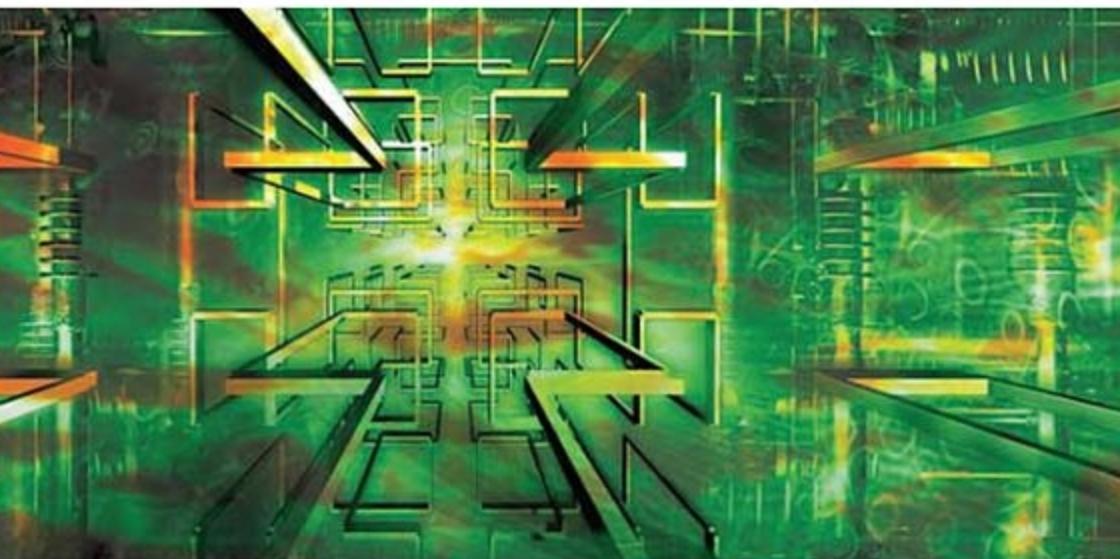


MPLS-ENABLED APPLICATIONS

Emerging Developments and New Technologies

THIRD EDITION



WILEY SERIES IN COMMUNICATIONS NETWORKING & DISTRIBUTED SYSTEMS

WILEY

INA MINEI
JULIAN LUCEK

'While MPLS is in itself simple, its apparent complexity lies in the proliferation of applications, which shows no signs of ceasing. To make things worse, catching up involves reading a large number of documents written by various authors at various times in various styles. Here at last is a single, all encompassing resource where the myriad applications sharpen into a comprehensible text that first explains the whys and whats of each application before going on to the technical detail of the hows.'

Kireeti Kompella, CTO Junos, Juniper Networks

'MPLS-Enabled Applications thoroughly covers the MPLS base technology and applications on MPLS-enabled IP networks. It guides you to a comprehensive understanding of standards, problems, and solutions in networking with MPLS. Before it had been necessary to go through material from many different sources, here we have everything in one place. All the MPLS protocols are covered, as are the applications of these protocols. This should be the textbook for MPLS courses, both for training of experienced networking professionals and for universities.'

Loa Andersson, Ericsson AB and IETF MPLS working group co-chair

'Although over ten years old, MPLS technology continues to evolve to meet the developing requirements of network operators and the advancing aspirations of network users. It is important that a book like this should continue to be updated in step with the changes to MPLS, and this new revision includes essential new material for those trying to understand the next steps in MPLS.'

Adrian Farrel, IETF Routing Area Director

'This book continues to be the industry and academic state-of-the-art on explaining the foundation and nuances of MPLS technology. It is extremely well written and tackles all of the most modern extensions of MPLS technology. If you are interested in how the internet works, it will be a well-worn read. It should be on every internet practitioner's bookshelf.'

Dave Ward, IETF WG chair: BFD, Softwires, ISIS, HIP

'This is the MPLS text that the industry has been waiting for. On one hand, the text presents MPLS technology clearly enough that the reader can absorb its content in a few easy sittings. On the other hand, the text provides a sufficiently in-depth treatment that even an MPLS expert can learn from it. The authors offer a clear and complete description of MPLS, its inner workings and its applications, in a manner that could only be achieved by persons who have been significant contributors to the MPLS development effort. Every network operator who has deployed or is considering the deployment of MPLS technology should read this book. It is appropriate reading for everyone from the CTO to the tier 1 NOC engineer.'

Ron Bonica, Juniper Networks, Co-director IETF Operations and Management Area

'MPLS-Enabled Applications provides excellent insight on how recently developed solutions can help address challenges for providing multicast in MPLS-based VPNs. The in-depth coverage of recent advances in MPLS technology that provide multicast support in L2 and L3 VPNs is essential to anyone needing to deploy both basic use cases and advanced scenarios as well.'

Thomas Morin, Network Architect at France Telecom Orange

'This is a highly recommended book for network design engineers who want to update themselves with the latest MPLS development, or those who want to learn this technology thoroughly. In addition to the impressive technology coverage and depth, the book is also a delightful reading!'

Lei Wang, Department manager Mobile IP Transport, Telenor

'MPLS-Enabled Applications is an excellent read for network engineers involved in the design of MPLS networks and services. It can serve as an introduction to MPLS networking or as a reference book for the advanced engineer. It discusses practical issues that must be considered in the design of MPLS networks and services, including MPLS-TE, MPLS-IPVPNs and MPLS L2VPNs. It also discusses current topics that are still evolving in the industry such as inter-AS/area MPLS-TE, point-to-multipoint LSPs and IPVPN multicast, providing a good overview of the issues being addressed and the current industry direction.'

Nabil N. Bitar, Principal member of Technical Staff and lead network architect, Verizon

'MPLS-Enabled Applications: Emerging Developments and New Technologies second edition, by Ina Minei and Julian Lucek, presents the current state-of-the-art in the specification, development, and application of MPLS and its related technologies. I believe, the readers will find the book to be a very valuable resource. I am pleased to see that the third edition of this book covers contemporary topics in the Internet industry such as MPLS-TP.'

Bijan Jabbari, PhD, Founder of Isocore, and Professor of Electrical Engineering, George Mason University

'This is the MPLS book that I reference the most and recommend to all my colleagues. It is written in an easy-to-follow approach that starts with basic concepts and then gradually ramps to advanced topics. It is timely in its coverage of new developments such as MPLS-TP and BGP/MPLS mVPNs, yet exhaustive by addressing all aspects of MPLS including the newer advances. I have personally used this book to architect designs such as broadcast video over IP/MPLS, hierarchical video-on-demand library distribution using BGP/MPLS mVPN, and a MPLS-based network supporting triple-play services over a BGP and PIM-free Core.'

Mazen Khaddam, Principal lead network architect, network architecture group, Cox communications

'This book is a wonderfully comprehensive overview of not just the underlying technology, but also the many use case applications of MPLS. It's a must have for networking professionals.'

Dorian Kim, Director of Network Development, NTT America

'MPLS-Enabled Applications takes a unique and creative approach in explaining MPLS concepts and how they are applied in practice to meet the needs of Enterprise and Service Provider networks. I consistently recommend this book to colleagues in the engineering, education and business community.'

Dave Cooper, Chief IP Technologist, Global Crossing Ltd.

'This book presents clear, comprehensive descriptions of the various scenarios in which the MPLS toolkit can be used to provide reliable and quality connectivity. It includes background information, detailed explanations on how to enable different services and applications, and precise technical and operational considerations. Business drivers for emerging technologies are discussed as well as practical and real deployment scenarios. Highlighting the hottest trends in the industry, this invaluable book describes how best to fit the pieces of the puzzle together to efficiently enable new applications and services.'

Nurit Sprecher, Senior specialist, Packet Transport Evolution, Nokia Siemens Networks

MPLS-Enabled Applications

WILEY SERIES IN COMMUNICATIONS NETWORKING & DISTRIBUTED SYSTEMS

Series Editors: David Hutchison, *Lancaster University, Lancaster, UK*
Serge Fdida, *Université Pierre et Marie Curie, Paris, France*
Joe Sventek, *University of Glasgow, Glasgow, UK*

The 'Wiley Series in Communications Networking & Distributed Systems' is a series of expert-level, technically detailed books covering cutting-edge research, and brand new developments as well as tutorial-style treatments in networking, middleware and software technologies for communications and distributed systems. The books will provide timely and reliable information about the state-of-the-art to researchers, advanced students and development engineers in the Telecommunications and the Computing sectors.

Other titles in the series:

- Wright: *Voice over Packet Networks* 0-471-49516-6 (February 2001)
Jepsen: *Java for Telecommunications* 0-471-49826-2 (July 2001)
Sutton: *Secure Communications* 0-471-49904-8 (December 2001)
Stajano: *Security for Ubiquitous Computing* 0-470-84493-0 (February 2002)
Martin-Flatin: *Web-Based Management of IP Networks and Systems* 0-471-48702-3 (September 2002)
Berman, Fox, Hey: *Grid Computing. Making the Global Infrastructure a Reality* 0-470-85319-0 (March 2003)
Turner, Magill, Marples: *Service Provision. Technologies for Next Generation Communications* 0-470-85066-3 (April 2004)
Welzl: *Network Congestion Control: Managing Internet Traffic* 0-470-02528-X (July 2005)
Raz, Juhola, Serrat-Fernandez, Galis: *Fast and Efficient Context-Aware Services* 0-470-01668-X (April 2006)
Heckmann: *The Competitive Internet Service Provider* 0-470-01293-5 (April 2006)
Dressler: *Self-Organization in Sensor and Actor Networks* 0-470-02820-3 (November 2007)
Berndt: *Towards 4G Technologies: Services with Initiative* 0-470-01031-2 (March 2008)
Jacquenet, Bourdon, Boucadair: *Service Automation and Dynamic Provisioning Techniques in IP/MPLS Environments* 0-470-01829-1 (March 2008)
Gurtov: *Host Identity Protocol (HIP): Towards the Secure Mobile Internet* 0-470-99790-7 (June 2008)
Boucadair: *Inter-Asterisk Exchange (IAX): Deployment Scenarios in SIP-enabled Networks* 0-470-77072-4 (January 2009)
Fitzek: *Mobile Peer to Peer (P2P): A Tutorial Guide* 0-470-69992-2 (June 2009)
Shelby: *6LoWPAN: The Wireless Embedded Internet* 0-470-74799-4 (November 2009)
Stavdas: *Core and Metro Networks* 0-470-51274-1 (February 2010)
Gómez Herrero, van der Ven, *Network Mergers and Migrations: Junos® Design and Implementation* 0-470-74237-2 (March 2010)
Jacobsson, Niemegeers, Heemstra de Groot, *Personal Networks: Wireless Networking for Personal Devices* 0-470-68173-X (June 2010)

MPLS-Enabled Applications

Emerging Developments
and New Technologies

Third Edition

Ina Minei
Juniper Networks

Julian Lucek
Juniper Networks

 **WILEY**
John Wiley & Sons, Ltd

This edition first published 2011
© 2011 John Wiley & Sons, Ltd.

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ,
United Kingdom

For details of our global editorial offices, for customer services and for information about
how to apply for permission to reuse the copyright material in this book please see our
website at www.wiley.com.

The right of the author to be identified as the author of this work has been asserted in
accordance with the Copyright, Designs and Patents Act 1988.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval
system, or transmitted, in any form or by any means, electronic, mechanical, photocopying,
recording or otherwise, except as permitted by the UK Copyright, Designs and Patents Act
1988, without the prior permission of the publisher.

Wiley also publishes its books in a variety of electronic formats. Some content that appears
in print may not be available in electronic books.

Designations used by companies to distinguish their products are often claimed as
trademarks. All brand names and product names used in this book are trade names, service
marks, trademarks or registered trademarks of their respective owners. The publisher is
not associated with any product or vendor mentioned in this book. This publication is
designed to provide accurate and authoritative information in regard to the subject matter
covered. It is sold on the understanding that the publisher is not engaged in rendering
professional services. If professional advice or other expert assistance is required, the
services of a competent professional should be sought.

Library of Congress Cataloging-in-Publication Data

Minei, Ina.

MPLS-enabled applications : emerging developments and new
technologies / Ina Minei, Julian Lucek.

p. cm.

Includes bibliographical references and index.

ISBN 978-0-470-66545-9 (pbk.)

1. MPLS standard. 2. Extranets (Computer networks) I. Lucek, Julian.

II. Title.

TK5105.573.M56 2010

621.382'16-dc22

2010029550

A catalogue record for this book is available from the British Library.

ISBN: 9780470665459 (P/B)

ePDF ISBN: 9780470976166

eBook ISBN: 9780470976173

ePub ISBN: 9780470976135

Set in 10/12pt Palatino by Aptara Inc., New Delhi, India.

Contents

About the Authors	xix
Foreword	xxi
Preface	xxv
Acknowledgements	xxxi
Part One	
1 Foundations	3
1.1 Historical perspective	3
1.2 Current trends	5
1.3 MPLS mechanisms	6
1.3.1 Forwarding plane mechanisms	7
1.3.2 Control plane mechanisms	11
1.3.3 Transport of IPv6 over an IPv4 MPLS core	32
1.4 Conclusion	35
1.5 References	35
1.6 Further reading	36
1.7 Study questions	36
2 Traffic Engineering with MPLS (MPLS-TE)	39
2.1 Introduction	39
2.2 The business drivers	39
2.3 Application scenarios	40
2.4 Setting up traffic-engineered paths using MPLS-TE	43
2.4.1 LSP priorities and preemption	43
2.4.2 Information distribution – IGP extensions	44
2.4.3 Path calculation – CSPF	46
2.4.4 Path setup – RSVP extensions and admission control	49

2.5	Using the traffic-engineered paths	51
2.6	Deployment considerations	54
2.6.1	Scalability	54
2.6.2	Reservation granularity	56
2.6.3	Routing challenges	57
2.7	Using traffic engineering to achieve resource optimization	57
2.7.1	Autobandwidth – dealing with unknown bandwidth requirements	58
2.7.2	Sharing links between RSVP and other traffic – dealing with unknown bandwidth availability	59
2.7.3	Other methods for optimization of transmission resources in MPLS networks	60
2.8	Offline path computation	61
2.9	Conclusion	64
2.10	References	65
2.11	Further reading	65
2.12	Study questions	65
3	Protection and Restoration in MPLS Networks	67
3.1	Introduction	67
3.2	The business drivers	68
3.3	Failure detection	69
3.4	End-to-end protection	70
3.4.1	Control over the traffic flow following a failure	71
3.4.2	Requirement for path diversity	71
3.4.3	Double-booking of resources	72
3.4.4	Unnecessary protection	72
3.4.5	Nondeterministic switchover delay	72
3.5	Local protection using fast reroute	73
3.5.1	Case (i): link protection, for the facility protection case	75
3.5.2	Case (ii): link protection, for the 1:1 protection case	77
3.5.3	Case (iii): node protection, for the facility protection case	78
3.5.4	Case (iv): node protection, for the 1:1 protection case	79
3.6	Link protection	81
3.6.1	What happens before the failure	82
3.6.2	What happens after the failure	87
3.7	Node protection	89
3.8	Additional constraints for the computation of the protection path	91
3.8.1	Fate sharing	91
3.8.2	Bandwidth protection	93
3.8.3	Bandwidth protection and DiffServ	96

3.9	Interaction of end-to-end protection and fast reroute	97
3.10	Deployment considerations for local protection mechanisms	98
3.10.1	Scalability considerations	98
3.10.2	Evaluating a local protection implementation	101
3.10.3	The cost of bandwidth protection	103
3.11	IP and LDP FRR	105
3.11.1	The tunnel-based approach	107
3.11.2	The alternate-path approach	108
3.12	Conclusion	110
3.13	References	111
3.14	Further reading	111
3.15	Study questions	111
4	MPLS DiffServ-TE	113
4.1	Introduction	113
4.2	The business drivers	114
4.3	Application scenarios	115
4.3.1	Limiting the proportion of traffic from a particular class on a link	115
4.3.2	Maintaining relative proportions of traffic on links	117
4.3.3	Providing guaranteed bandwidth services	117
4.4	The DiffServ-TE solution	117
4.4.1	Class types	117
4.4.2	Path computation	118
4.4.3	Path signaling	121
4.4.4	Bandwidth constraint models	122
4.4.5	Overbooking	127
4.4.6	The DiffServ in DiffServ-TE	129
4.4.7	Protection	130
4.4.8	Tools for keeping traffic within its reservation limits	131
4.4.9	Deploying the DiffServ-TE solution	132
4.5	Extending the DiffServ-TE solution with multiclass LSPs	133
4.6	Conclusion	134
4.7	References	134
4.8	Further reading	135
4.9	Study questions	135
5	Interdomain Traffic Engineering	137
5.1	Introduction	137
5.2	The business drivers	137
5.3	Setting up interdomain TE LSPs	139
5.3.1	Path setup	140
5.3.2	Path computation	144

5.3.3 Reoptimization	154
5.3.4 Protection and fast reroute	155
5.4 Interprovider challenges	157
5.5 Comparison of the LSP setup methods	158
5.6 Conclusion	159
5.7 References	160
5.8 Further reading	161
5.9 Study questions	161
6 MPLS Multicast	163
6.1 Introduction	163
6.2 The business drivers	164
6.3 P2MP LSP mechanisms	165
6.3.1 Forwarding plane mechanisms	165
6.3.2 Control plane mechanisms	167
6.4 LAN procedures for P2MP LSPs	176
6.4.1 Upstream label allocation	177
6.5 Coupling traffic into a P2MP LSP	178
6.5.1 Coupling Layer 2 traffic into a P2MP LSP	179
6.5.2 Coupling IP unicast traffic into a P2MP LSP	179
6.5.3 Coupling IP multicast traffic into a P2MP LSP	180
6.6 MPLS fast reroute	181
6.7 Ingress redundancy for P2MP LSPs	183
6.8 P2MP LSP hierarchy	184
6.8.1 P2MP LSP hierarchy forwarding plane operation	186
6.8.2 P2MP LSP hierarchy control plane operation	187
6.9 Applications of point-to-multipoint LSPs	187
6.9.1 Application of P2MP TE to broadcast TV distribution	188
6.9.2 Application of P2MP LSPs to L3VPN multicast	191
6.9.3 Application of P2MP LSPs to VPLS	193
6.10 Conclusion	193
6.11 References	193
6.12 Study questions	195

Part Two

7 Foundations of Layer 3 BGP/MPLS Virtual Private Networks	199
7.1 Introduction	199
7.2 The business drivers	200
7.3 The overlay VPN model	201
7.4 The peer VPN model	202
7.5 Building the BGP/MPLS VPN solution	205
7.5.1 VPN routing and forwarding tables (VRFs)	205
7.5.2 Constrained route distribution	207

7.5.3	VPN-IPv4 addresses and the route distinguisher (RD)	208
7.5.4	The route target (RT)	209
7.5.5	The solution so far – what is missing?	215
7.5.6	VPN label	216
7.6	Benefits of the BGP/MPLS VPN solution	221
7.7	References	222
7.8	Further reading	222
7.9	Study questions	223
8	Advanced Topics in Layer 3 BGP/MPLS Virtual Private Networks	225
8.1	Introduction	225
8.2	Routing between CE and PE	225
8.3	Differentiated VPN treatment in the core	230
8.4	Route reflectors and VPNs	231
8.5	Scalability discussion	235
8.5.1	Potential scaling bottlenecks	236
8.5.2	The cost of growing the VPN network	238
8.6	Convergence times in a VPN network	243
8.6.1	Convergence time for a customer route change	243
8.6.2	Convergence time for a failure in the provider's network	244
8.7	Security issues	244
8.7.1	Can traffic from one VPN ‘cross over’ into another VPN?	245
8.7.2	Can a security attack on one VPN affect another VPN?	245
8.7.3	Can a security attack against the service provider's infrastructure affect the VPN service?	246
8.8	QoS in a VPN scenario	246
8.9	IPv6 VPNs	248
8.10	Conclusion	251
8.11	References	251
8.12	Further reading	252
8.13	Study questions	252
9	Hierarchical and Inter-AS VPNs	255
9.1	Introduction	255
9.2	Carriers' carrier – service providers as VPN customers	256
9.2.1	ISP as a VPN customer	257
9.2.2	VPN service provider as a VPN customer – hierarchical VPN	262

9.3	Multi-AS backbones	266
9.3.1	Option A: VRF-to-VRF connections at the ASBR	266
9.3.2	Option B: EBGP redistribution of labeled VPN-IPv4 routes	268
9.3.3	Option C: multihop EBGP redistribution of labeled VPN-IPv4 routes between the source and destination AS, with EBGP redistribution of labeled IPv4 routes from one AS to the neighboring AS	269
9.4	Interprovider QoS	271
9.5	Conclusion	272
9.6	References	272
9.7	Further reading	273
9.8	Study questions	273
10	Multicast in a Layer 3 VPN	275
10.1	Introduction	275
10.2	The business drivers	276
10.3	mVPN – problem decomposition	278
10.4	The original multicast solution – PIM/GRE mVPN (draft-rosen)	279
10.4.1	PIM/GRE mVPN – routing information distribution using PIM C-instances	280
10.4.2	PIM/GRE mVPN – carrying multicast traffic across the core using multicast distribution trees	281
10.4.3	Properties of the PIM/GRE mVPN solution	283
10.5	NG multicast for L3VPN – BGP/MPLS mVPN (NG mVPN)	286
10.5.1	Requirements for support of PIM-SM SSM in an mVPN	286
10.5.2	BGP/MPLS mVPN – carrying multicast mVPN routing information using C-multicast routes	287
10.5.3	BGP/MPLS mVPN – carrying traffic across the provider network using inter-PE MPLS tunnels	292
10.5.4	BGP/MPLS mVPN – inter-PE tunnels – inclusive and selective tunnels	292
10.5.5	BGP/MPLS mVPN – carrying traffic from several mVPNs onto the same inter-PE tunnel	294
10.5.6	BGP/MPLS mVPN – creating inter-PE tunnels using BGP autodiscovery routes	295
10.5.7	Requirements for support of PIM ASM in an mVPN	299

10.5.8 BGP/MPLS mVPN – carrying mVPN active source information using BGP source active autodiscovery routes	300
10.6 Comparison of PIM/GRE and BGP/MPLS mVPNs	303
10.6.1 VPN model used	303
10.6.2 Protocol used in the control plane	304
10.6.3 Data-plane mechanisms	305
10.6.4 Service provider network as a 'LAN'	306
10.6.5 Deployment considerations	306
10.7 Conclusion	307
10.8 References	307
10.9 Further reading	308
10.10 Study questions	309
11 Advanced Topics in BGP/MPLS mVPNs	311
11.1 Introduction	311
11.2 BGP/MPLS mVPN – inter-AS operations	311
11.3 Support of PIM DM in BGP/MPLS mVPN	316
11.4 Discovering the RP – auto-RP and BSR support in BGP/MPLS mVPN	317
11.5 Implementing extranets in BGP/MPLS mVPN	319
11.6 Transition from draft-rosen to BGP/MPLS mVPNs	322
11.7 Scalability discussion	325
11.7.1 PIM/GRE mVPN control plane scaling	325
11.7.2 BGP/MPLS mVPN control plane scaling	326
11.8 Achieving multicast high availability with BGP/MPLS mVPN	328
11.8.1 Live-Standby multicast delivery using BGP/MPLS mVPN	329
11.8.2 Live-Live multicast delivery using BGP/MPLS mVPN	332
11.8.3 Comparison of the Live-Live and Live-Standby multicast high-availability schemes	335
11.9 Internet multicast service using the BGP/MPLS mVPN technology	335
11.10 Conclusion	337
11.11 References	338
11.12 Study questions	338
12 Layer 2 Transport over MPLS	341
12.1 Introduction	341
12.2 The business drivers	341
12.3 Comparison of layer 2 VPNs and layer 3 VPNs	344

12.4	Principles of layer 2 transport over MPLS	345
12.5	Forwarding plane	347
12.5.1	ATM cell	349
12.5.2	ATM AAL5	349
12.5.3	Frame relay	350
12.5.4	Ethernet	350
12.6	Control plane operation	351
12.6.1	Original LDP signaling scheme	351
12.6.2	BGP-based signaling and autodiscovery scheme	353
12.6.3	LDP signaling with BGP autodiscovery	357
12.6.4	Comparison of BGP and LDP approaches to Layer 2 transport over MPLS	358
12.7	Admission control of layer 2 connections into network	360
12.8	Failure notification mechanisms	361
12.9	Multi-homing	362
12.9.1	BGP case	362
12.9.2	LDP case	364
12.10	Layer 2 interworking	365
12.11	Circuit cross connect (CCC)	365
12.12	Point-to-multipoint Layer 2 transport	366
12.12.1	Point-to-multipoint CCC	367
12.12.2	Layer 2 Multicast VPNs	367
12.13	Other applications of Layer 2 transport	368
12.14	Conclusion	370
12.15	References	370
12.16	Study questions	371
13	Virtual Private LAN Service	373
13.1	Introduction	373
13.2	The business drivers	373
13.3	VPLS mechanism overview	375
13.4	Forwarding plane mechanisms	379
13.4.1	Forwarding of unicast frames	379
13.4.2	Broadcast and multicast frames	382
13.5	Control plane mechanisms	384
13.5.1	LDP-based signaling	384
13.5.2	BGP signaling and autodiscovery	389
13.5.3	Comparison of LDP and BGP for VPLS control plane implementation	396
13.5.4	IGMP and PIM snooping	399
13.5.5	Use of multicast trees in VPLS	401
13.6	LDP and BGP interworking for VPLS	406
13.7	Interprovider Option E for VPLS	413
13.7.1	Comparison of interprovider schemes for VPLS	415

13.8	Operational considerations for VPLS	416
13.8.1	Number of MAC addresses per customer	416
13.8.2	Limiting broadcast and multicast traffic	417
13.8.3	Policing of VPLS traffic	417
13.8.4	VPLS with Integrated Routing and Bridging (IRB)	417
13.8.5	Learning mode	417
13.9	Conclusion	418
13.10	References	419
13.11	Study questions	419

Part Three

14	Advanced Protection and Restoration: Protecting the Service	423
14.1	Introduction	423
14.2	The business drivers	423
14.3	Failure scenarios	425
14.4	Existing solutions	426
14.4.1	Single homed CE	426
14.4.2	Dual-homed CE	427
14.4.3	Analyzing existing dual-homing solutions	432
14.5	Protecting the egress – local protection solution	433
14.5.1	Protecting against an attachment circuit failure in a pseudowire scenario – edge protection virtual circuit	435
14.5.2	Protecting against an egress PE failure in an L3VPN scenario	437
14.6	Conclusion	440
14.7	References	440
14.8	Further reading	441
14.9	Study questions	441
15	MPLS Management	443
15.1	Introduction	443
15.2	Management – why and what	443
15.3	Detecting and troubleshooting failures	445
15.3.1	Reporting and handling nonsilent failures	445
15.3.2	Detecting silent failures – MPLS OAM	446
15.3.3	Troubleshooting failures	461
15.4	Configuration errors	467
15.4.1	Preventing configuration errors	467
15.4.2	Detecting and reporting misconfigurations	469
15.5	Visibility	473

15.6	Conclusion	474
15.7	References	475
15.8	Further reading	476
15.9	Study questions	476
16	MPLS in Access Networks and Seamless MPLS	479
16.1	Introduction	479
16.2	The business drivers	479
16.2.1	The transition from legacy access to Ethernet access	480
16.2.2	MPLS as the technology choice for the Ethernet access network	483
16.3	Models for MPLS deployment in access networks	486
16.4	Seamless MPLS Mechanisms	491
16.4.1	Extending MPLS to the Access Node	491
16.4.2	Seamless MPLS scaling	493
16.4.3	Scaling analysis of Seamless MPLS	497
16.4.4	Seamless MPLS for multicast	501
16.5	Conclusions	507
16.6	References	507
16.7	Study questions	508
17	MPLS Transport Profile (MPLS-TP)	509
17.1	Introduction	509
17.2	The business drivers	509
17.3	Requirements for a transport profile for MPLS	512
17.3.1	Characteristics of transport networks	513
17.3.2	Requirements and architectural goals of MPLS-TP	514
17.4	MPLS-TP functionality	516
17.4.1	MPLS-TP as a subset of MPLS	516
17.4.2	MPLS-TP resilience functions	517
17.4.3	MPLS-TP OAM functions	518
17.5	Deployment considerations	522
17.6	Misconceptions about MPLS-TP	526
17.7	Conclusion	527
17.8	References	527
17.9	Study questions	529
18	Conclusions	531
18.1	Introduction	531
18.2	Network convergence	533
18.3	Interaction with client edge equipment	536
18.4	Interprovider capability	538
18.5	MPLS in the data communications network (DCN)	539

18.6	MPLS in mobile networks	540
18.7	MPLS in the enterprise	542
18.8	MPLS in the transport	545
18.9	Final remarks	545
18.10	References	546
Appendix A: Selected Backhaul Scenarios in MPLS-Based Access Networks		547
Appendix B: MPLS Resources		559
Appendix C: Solutions to Selected Study Questions		561
Appendix D: Acronyms		575
Index		587

About the Authors

Ina Minei joined Juniper Networks in 2000 and is currently Director of IP and MPLS technologies. During this time she worked on the implementation of LDP and RSVP-TE, helped define new protocol extensions, and worked with numerous customers on network design. Her focus has been on next-generation network technologies, in particular MPLS protocols and applications. She previously worked at Cisco for two years in various software development projects for routers and switches. Ms Minei is an active participant in industry forums and conferences and holds several patents in the area of IP and MPLS. She earned a Master's degree in computer science from the Technion, Israel.

Julian Lucek joined Juniper Networks in 1999 and is currently a Distinguished Systems Engineer in the Europe, Middle East and Africa region, where he has been working with many service providers on the design and evolution of their networks. He previously worked at BT for several years, at first in the Photonics Research Department and later in the data transport and routing area. During this time, he gained a PhD in ultrahigh-speed data transmission and processing from Cambridge University. He is the holder of several patents in the area of communications technology. He has a Master's degree in Physics from Cambridge University and holds Juniper Networks Certified Internet Expert (JNCIE) #21.

Foreword

Yakov Rekhter, Juniper Fellow, Juniper Networks

Multi-Protocol Label Switching (MPLS) began in the mid-1990s with just two modest design objectives. The first was a better integration of ATM with IP, a goal that we hoped could be met by providing a single IP-based control plane that would span both ATM switches and IP routers. The second objective was to augment the IP control plane with some additional functionality, namely traffic engineering using constraint-based routing that was already present in the ATM control plane.

Not long after it started, MPLS usage was extended to applications such as Circuit Cross Connect (CCC), ATM and Frame Relay service over an IP/MPLS infrastructure (draft-martini), BGP/MPLS VPNs (2547 VPNs) and then Virtual Private LAN Services (VPLS). The original constraint-based routing functionality evolved beyond traffic engineering to applications such as fast reroute and Differentiated Services Traffic Engineering (DiffServ-TE).

The idea of a single control plane for both ATM switches and IP routers evolved into Generalized Multi-Protocol Label Switching (GMPLS), which provides a single control plane that could span not only routers and ATM switches but SONET/SDH and optical cross connects as well.

One of the recent MPLS developments deserving of mention here is the use of MPLS in the access network. Expanding MPLS into the access network brings with it scalability challenges. The third edition describes a solution, known as ‘Seamless MPLS’, that addresses these challenges.

Since the first edition of this book, considerable progress has been made in the area of MPLS multicast, IP multicast with BGP/MPLS VPNs, and IP multicast with VPLS. Advances in these areas were included in the second edition of this book. The third and current edition further expands upon these developments by covering such topics as supporting multicast extranets in BGP/MPLS VPNs and supporting Internet multicast over an MPLS infrastructure. This edition also presents in detail the scalability comparison between two schemes of supporting multicast in

BGP/MPLS VPNs – the first one based on the PIM/GRE solution (known informally as ‘draft-rosen’), and the second based on NG multicast for L3VPN (BGP/MPLS mVPN). As an ever-increasing array of services has been developed surrounding MPLS infrastructure, the importance of high service availability and its successful provision has come to light. Since the second edition of this book, significant progress has been made in the area of scalable fast protection based on the technique of local repair; these developments and their implications are included in this edition.

One important development since the publication of the second edition of this book is MPLS Transport Profile (MPLS-TP), a technology driven by the desire of the service providers to transition their transport infrastructure from circuit-switched based technologies (SONET/SDH) to packet-based switching technology based on MPLS. This edition provides an overview of MPLS-TP, and clarifies the motivations behind and requirements for its adoption.

It is important to keep in mind that in all of the applications mentioned above, MPLS is just one of the components of such applications, albeit a critical one. If we look back at the time when MPLS was created, and compare its design objectives with what MPLS is used for today, we notice several things. First of all, most of the applications of MPLS that we have today were not conceived of during the original design of MPLS, while some of the applications conceived of during the original design of MPLS are no longer relevant. For example, the original design goal of a better integration of ATM and IP routers by having a single control plane that spans both ATM switches and routers is a thing in the past. And while the ability to offer ATM service over an IP/MPLS infrastructure is still relevant, it becomes less and less important relative to the Ethernet service over an IP/MPLS infrastructure. While originally MPLS was conceived as a technology solely for the Service Providers, we see today how MPLS is gradually penetrating the enterprise environment. Additionally, over time the whole MPLS concept evolved from Multi-Protocol Label Switching to Multi-Purpose Label Switching.

A new technology quite often generates opposition, and MPLS was by no means an exception. You may all remember how MPLS was branded by its opponents in negative terms as ‘bad’, ‘evil’, ‘a social disease’ or ‘a nightmare of unprecedented proportions’. To put this in a proper perspective, we need to keep in mind that technologies exist not for their own sake but for the purpose of solving business problems. Therefore, talking about ‘good’ technologies versus ‘bad/evil’ technologies has little practical relevance; what is of great relevance is how well a particular technology meets business needs.

One might wonder how to judge how well a particular technology, like MPLS, meets business needs. To answer this question I would like to invoke the words of Cervantes’ Don Quixote: ‘the proof of the pudding

is in the eating', to which I would add: 'and not in the debate about the pudding'. That being said, the ultimate judge of how well a particular technology meets business needs is the marketplace. It is the judgment of the marketplace that determines whether a particular technology deserves to live or to die; and with respect to MPLS the market made its verdict loud and clear – MPLS is here to stay.

Preface

In the three years since we began the previous edition of this book, so many new MPLS developments have taken place that our publisher and many readers suggested that a third edition would be useful. Two particular note-worthy developments since the second edition are Seamless MPLS – an architecture to scale networks to 100,000+ MPLS nodes – and MPLS-TP, which provides the infrastructure for MPLS-based transport networks. The motivation for the book remains the same: MPLS is moving so fast that some of its new applications have already been deployed in production networks, yet are not described anywhere in book form. In many cases, the only available resources are the IETF drafts which list the extensions needed to produce interoperable implementations. These documents often assume familiarity with the problem at hand and do not discuss why a particular solution has been chosen or explain its pros and cons. The third edition of *MPLS-Enabled Applications* attempts to fill this gap and provide the reader with an understanding of both the problem and why the solution looks the way it does.

Therefore, when we describe the mechanisms underpinning an MPLS application, the emphasis is on giving an overview of the protocol machinery without delving into the bits and bytes of each protocol message. This allows us to convey the concepts without making it difficult to see the wood for the trees. Also, some of the mechanisms that we write about are currently being defined, so details of the protocol messages may change, but the concepts are less likely to. References at the end of each chapter point to the documents describing the message formats and processing rules. Because a lot of the content in this book deals with technologies that are still, literally, *works in progress*, several things may happen. Firstly, some proposals may be abandoned or fail to become widely adopted. Secondly, different vendors may introduce the technology at different times, and finally, the solution may evolve and change as implementation and deployment experience is gained. Therefore, the fact that we discuss a

particular technology in this book does not guarantee that it is available or deployed.

Although we both happen to work for the same router vendor, the book is not vendor-specific. Occasionally, we point out some vendor-specific quirks if they are relevant to the discussion, or aid in understanding a particular topic. Many of the topics discussed are still under debate in the IETF, and naturally our personal views on one topic or another may be stated more strongly than the opposing view.

WHO SHOULD READ THIS BOOK?

The intended audience of this book includes employees of network operators and network equipment vendors, customers of service providers who are interested in the mechanisms underpinning the services that they buy, network professionals who want to keep up to date with the latest advances in MPLS and students of network technology. To make this book more accessible to both the student and to the practitioner of MPLS, we have added study questions at the end of each chapter.

We assume that the reader has some degree of familiarity with network technology and routing protocols, in particular BGP and the link-state IGPs, but these are not a requirement to benefit from the book. Although our main aim is to cover the cutting-edge developments of MPLS, the Foundation chapter allows the reader unfamiliar with MPLS to get up to speed in order to benefit from the remainder of the book. Even when discussing basic topics such as traffic engineering or fast reroute, we also explore the more interesting and advanced aspects of the technology.

WHAT IS NEW IN THE THIRD EDITION?

In this third edition, we aim to capture the latest developments in the field. For this reason, we added three new chapters. Chapter 11 covers advanced topics in multicast in L3VPNs, focusing on new developments in the BGP/MPLS scheme, which has gained significant deployment over the last few years. Chapter 14 discusses advanced protection schemes for the LSP tail-end, thus enabling sub 50 ms end-to-end service restoration. Finally, Chapter 17 provides an overview of MPLS-TP, the transport profile for MPLS, which will form the foundation for packet-switched transport networks. Additional material was added and updated throughout the book. Chapter 16, covering MPLS in access networks, has new sections describing the Seamless MPLS architecture, including the solutions for both unicast and multicast. The book also has new material covering the

Live-live and Live-standby schemes for multicast resilience, point to multipoint pseudowires, pseudowire redundancy and VPLS Interprovider Option E. The study questions at the end of each chapter are intended to help readers test their understanding of the topics discussed and can serve to trigger debate on the pros and cons of a particular technology to a particular deployment.

HOW THIS BOOK IS ORGANIZED

The book is divided into three parts, each containing several chapters. Part One describes the MPLS infrastructure tools used as the foundation to build services, Part Two covers the MPLS-based services and Part Three explores advanced topics.

The structure of Part One

Chapter 1, the Foundations chapter, reviews the control plane and forwarding plane mechanisms associated with MPLS. In this chapter, we give an overview of the LDP and RSVP signaling protocols and compare the two.

Chapter 2 discusses MPLS Traffic Engineering, which gives service providers control over the path taken by traffic through their network and the ability to give bandwidth guarantees. In this context, we look at the impact of TE on network scalability, as well as at solutions for TE in LDP networks.

Chapter 3 explores the topic of Protection and Restoration in MPLS networks, essential to allowing MPLS networks to carry mission-critical traffic. We cover link and node protection, their respective scaling properties and the cost of bandwidth protection. We also explore more advanced topics such as fate sharing and the new developments for providing fast restoration in IP and LDP networks.

Chapter 4 presents Differentiated Services (DiffServ) Aware Traffic Engineering, which allows traffic engineering to be applied with per-class granularity, bringing QoS to the network.

Chapter 5 introduces Interdomain Traffic Engineering. Both the signaling and computation aspects are discussed, and path-computation elements are also reviewed.

Chapter 6 is devoted to MPLS multicast functionality. This chapter covers not just P2MP LSP setup with RSVP and LDP but also advanced topics such as upstream label allocation and hierarchies of P2MP LSPs. MPLS multicast is currently of great interest as it allows MPLS to be used

in broadcast TV and IPTV applications and because it is an essential part of the next-generation L3VPN multicast solutions discussed in Part Two.

The structure of Part Two

Chapters 7, 8, 9, 10 and 11 are devoted to Layer 3 VPNs – the most widespread application of MPLS to date. Chapters 7 through 9 focus on unicast traffic in VPNs. Chapter 7 provides a tutorial on L3VPN and explains the basic concepts, Chapter 8 discusses more advanced topics such as route target filtering and scalability analysis, and Chapter 9 covers hierarchical VPNs. Chapters 10 and 11 dive into the topic of multicast VPNs. Chapter 10 presents and compares the PIM/GRE and the BGP/MPLS solutions for multicast VPNs, while Chapter 11 focuses entirely on advanced topics such as extranet and inter-AS support in the BGP/MPLS solution, which has gained a lot of traction in the last few years.

Chapter 12 describes the rapidly growing area of Layer 2 transport over MPLS, including pseudowires and Layer 2 VPNs. These allow service providers to migrate ATM and Frame Relay services to an IP/MPLS network and to offer Ethernet-based alternatives to those services.

Chapter 13 describes the Virtual Private LAN Service (VPLS). This allows a service provider to offer a very simple-to-use service to enterprise customers, in which the customer's sites appear to be attached to the same LAN. Multicast support over VPLS, an area which has seen a lot of change in recent years, is also discussed.

The structure of Part Three

Chapter 14 describes advances in protection schemes aimed at providing 50 ms recovery times for end-to-end services. As we show in the chapter, a critical building block is providing protection of the LSP tail end.

Chapter 15 covers some aspects of the management and troubleshooting of MPLS networks. The subject of management of MPLS networks could fill an entire book by itself and a single chapter does not do it justice. However, we attempt to show some of the challenges (such as ICMP tunneling) and some of the available tools, such as LSPing.

Chapter 16 provides an overview of the emerging trend of using MPLS in the access network, explains why this technology is taking off and describes the various deployment models, as well as describing the new and increasingly popular Seamless MPLS architecture.

Chapter 17 discusses the much-debated topic of MPLS-TP, the transport profile for MPLS. MPLS-TP is currently the most active standardization area in MPLS. In order to track developments in this field, it is important

to understand both the technology itself and what drives the industry to rally behind it in this way.

The final chapter takes a look at the achievements of MPLS to date and how MPLS may in future extend to DCNs, mobile Radio Access Networks and enterprise networks.

Appendix A contains details of how xDSL architectures can be mapped onto an MPLS-based access network.

REFERENCES

At the end of each chapter, there is a list of references. In the body of the text, these references appear in brackets, like this [REF1]. Many of the references are IETF documents. As these documents progress in the IETF process, their revision number and document name may change. Therefore, when looking up a reference online, search by the author and title rather than by the document name.

In some chapters, we have included a section with further reading. These are documents that we thought would be useful for those wanting to broaden their knowledge on a particular topic.

Ina Minei, Sunnyvale, CA
Julian Lucek, Ipswich, UK

Acknowledgements

This book would not have existed if it were not for the following three people: Yakov Rekhter, Aviva Garrett and Patrick Ames, and to them we extend our most heartfelt thanks.

Yakov Rekhter encouraged us to pursue this project and provided valuable insight throughout the writing process, from the book proposal, in-depth discussions of many topics, in particular cutting edge topics such as multicast support in L3VPNs, seamless MPLS and advanced protection schemes and finally detailed technical reviews of numerous chapters. Most importantly, his faith in our ability to do this work was one of the main factors that determined us to go ahead with this project.

Aviva Garrett was the first person to hear about this idea, encouraged it and arranged all the required support within Juniper Networks, as well as providing detailed editorial review, sometimes on extremely short notice.

Patrick Ames guided us through the intricate process of bringing a book from proposal to the printing press and provided moral support and appropriate scolding as necessary. We would not have been able to pull this off without him. Patrick also did all the hard work of preparing the manuscript (and in particular the art manuscript) for editing.

All three editions of this book benefited from the contribution of many people. We would like to thank our following colleagues:

Pedro Marques, for his thorough review of almost all chapters, for many technical discussions and for contributing the analysis of VPN scaling and RR scaling.

Arthi Ayyangar, for her insight on all topics RSVP and TE-related, for many technical discussions throughout our years of working together and for the careful review of numerous chapters.

Steven Lin, for reading and commenting on the entire manuscript of the first edition, on a very tight schedule.

Der-Hwa Gan, for his mentoring role on TE, RSVP and MPLS, and for his very thorough technical review and comments.

Chaitanya Kodeboyina (CK), for very detailed reviews and discussions on several chapters.

Josef Buchsteiner, for always bringing up tough customer problems and for the timely reviews of selected chapters.

Serpil Bayraktar, for never leaving open ends and for very careful reading of the VPN and interdomain TE chapters.

Amir Tabdili, for always asking the hard questions and for reviewing selected chapters.

Quaizar Vohra, for his insight into Layer 2 circuits and the IGP, and for his technical review of these topics.

Margarida Correia, for always questioning proposed solutions to customer problems, and for technical review of selected chapters.

Hector Avalos, for valuable technical discussions and technical review of selected chapters.

Nischal Sheth, for being a mentor on LDP and for numerous discussions on all topics MPLS-related, as well as for a thorough review of the Advanced Protection chapter, where he provided valuable insights.

Kireeti Kompella, for many technical discussions, for his insight into all aspects of the MPLS technology and for his review of the MPLS in Access Networks chapter. The discussion of ‘Option 1’ and ‘Option 2’ deployment models for MPLS within that chapter is largely based on Kireeti’s analysis of this topic.

Hannes Gredler, for his review of the Protection and Restoration and the Advanced Protection chapters and numerous discussions on tailend protection.

Nitin Bahadur, for his review of the Management, Protection and Restoration, Multicast over MPLS and MPLS-TP chapters.

Amit Shukla, for a detailed review of the VPLS chapter and valuable insights in the LDP/BGP VPLS interworking scheme described there.

Derek Harkness, for his expertise on broadband DSL architectures and his review of the MPLS in Access Networks chapter.

Nils Swart, for his expertise on Metro Ethernet and his review of the MPLS in Access Networks chapter.

João Gomes, for his expertise on mobile networks and his review of the Conclusions chapter.

Pierre Bichon, for his expertise on mobile networks and his review of the Conclusions chapter.

Senad Palislamovic, for his thorough review of all the study questions and their answers for the second edition of the book.

Dave Ward, for his detailed review of the MPLS-TP chapter, and for his perspective on the technology, provided both through the presentations referenced in the chapter and through the review.

John Drake, for his thorough review of the MPLS-TP chapter, and his insight into the technology, which at the time of the writing was still very much work in progress.

Nurit Sprecher for her suggestions for additional topics to cover in MPLS-TP and comments on the MPLS-TP material.

Kurt Windisch, for his detailed and timely review of advanced topics in multicast in VPNs.

Kevin Wang, for his comments on the Advanced Protection chapter, for which he provided valuable implementation perspective.

Manish Gupta, for his comments on selected topics in VPLS and L2VPN.

As always, any errors and omissions are the responsibility of the authors.

We would also like to acknowledge the support and assistance of Juniper Networks for providing the necessary resources to work on the third edition of the book.

Last but not least we would like to thank the wonderful team at Wiley: Tiina Ruonamaa, Sarah Tilley, Anna Smart, and Jasmine Chang for their support and guidance.

Finally, the authors would like to express their personal thanks to family and friends:

Ina Minei – First of all, I would like to thank my husband Pedro Marques for being my strongest supporter and harshest critic, for having infinite patience throughout the entire project, not once, not twice, but three times around. I would not have been able to do this without his support. Second, I would like to thank my father for the long hours he spent with me in my high school years, teaching me English and writing.

Julian Lucek – I would like to thank my partner Rachel and our daughters Emma and Hannah for their great patience and support during the writing of both editions of this book. Also I would like to thank my parents for their encouragement and for looking after my daughters during my writing sessions.

Part One

1

Foundations

1.1 HISTORICAL PERSPECTIVE

In only a few years, Multi-Protocol Label Switching (MPLS) has evolved from an exotic technology to a mainstream tool used by service providers to create revenue-generating services. There is rapid deployment of MPLS-enabled services and active development of new mechanisms and applications for MPLS in the standards bodies. This book aims to describe the fundamental mechanisms used by MPLS and the main service types that MPLS enables, such as Virtual Private Networks (VPNs). We include descriptions of new applications of MPLS that are currently under development.

The history of MPLS and its precursors is described in [Davie Rekhter] and [Doyle Kolon]. The first Internet Engineering Task Force (IETF) MPLS Working Group Meeting took place in April 1997. That working group still exists, and MPLS has grown to the extent that it underpins much of the activity of several other working groups in the IETF, such as Layer 3 VPN (l3vpn), Layer 2 VPN (l2vpn), Pseudo Wire Emulation Edge-to-Edge (pwe3) and Common Control and Measurement Plane (ccamp). Part of the original MPLS problem statement [MPLS97] from the first MPLS working group meeting is shown below. It contains four items that the group aimed to address through the development of MPLS. It is interesting to examine these to see which items are still relevant today:

1. *Scalability of network layer routing.* Using labels as a means to aggregate forwarding information, while working in the presence of routing hierarchies.

Layer 3 VPNs have proved to be a good example of aggregation of forwarding information. As described in Chapter 7 of this book, edge routers need to contain routing information pertaining to each VPN that they service, but the core routers do not. Thus, assuming that any edge router services only a subset of the VPNs pertaining to the network, no router in the network needs to hold the entire set of routes present in the network.

2. *Greater flexibility in delivering routing services.* Using labels to identify particular traffic which are to receive special services, e.g. QoS. Using labels to provide forwarding along an explicit path different from the one constructed by destination-based forwarding.

MPLS has the ability to identify particular traffic flows which must receive special services such as Quality-of-Service (QoS). It also has traffic engineering properties that allow it to provide forwarding along a particular explicit path. These two properties are combined in DiffServ Aware Traffic Engineering, which is described in more detail in Chapter 4 of this book.

3. *Increased performance.* Using the label-swapping paradigm to optimize network performance.

Because modern routers perform packet forwarding in hardware, the forwarding rates for IP and MPLS packets are similar. However, 'optimizing network performance' implies a wider context than simply the performance of individual nodes. Certainly MPLS has helped in this wider context, e.g. through the use of traffic engineering to avoid congestion and the use of fast reroute to reduce the interruption to traffic when a link in the network fails.

4. *Simplify integration of routers with cell switching based technologies:* a) making cell switches behave as peers to routers (thus reducing the number of routing peers that a router has to maintain), b) by making information about physical topology available to Network Layer routing procedures, and c) by employing common addressing, routing, and management procedures.

When this item in the problem statement was written, many networks had a core of asynchronous transfer mode (ATM) switches surrounded by routers. The routers were typically fully meshed with ATM connections. This overlay model was proving difficult to scale because the number of routing adjacencies required grew as the square of the number of routers involved; hence there was a requirement to make the ATM switches act as peers to the routers. It is interesting to note that the situation has now been turned inside out: now many networks have an MPLS-based core, and service providers are migrating ATM services to this core network by

interconnecting ATM switches with Layer 2 connections over the MPLS core! This has the problem that the number of adjacencies between ATM switches grows as the square of the number of ATM switches involved. Hence, currently there is work on making ATM switches behave as peers to routers [MPLS ALLI]. This is to avoid having a full mesh of adjacencies between ATM switches rather than to avoid having a full mesh of adjacencies between routers, as stated in the problem statement. The concept expressed in the problem statement of using MPLS as a control plane for multiple technologies has manifested itself in Generalized MPLS (GMPLS). In GMPLS, a common control plane covers a wide range of network devices, such as routers, ATM switches, SONET/SDH equipment and optical cross-connects [RFC3945].

In summary, much of the original problem statement is still relevant today. Many of the mechanisms of MPLS described in Part 1 of this book were developed to address the items listed above, to the benefit of the MPLS applications discussed in Part 2 of this book.

1.2 CURRENT TRENDS

At the time of writing this book, the most widely deployed customer-visible MPLS service is the Layer 3 VPN (also known as an IP VPN or 2547bis VPN, after the IETF document describing them). MPLS is also used in some networks as an infrastructure tool to provide traffic engineering and fast-reroute capabilities. Another rapidly growing application is point-to-point Layer 2 transport, either as means of carrying a customer's Ethernet traffic across the wide area or as a component of ATM or Frame Relay Service emulation. Finally, Virtual Private LAN Service (VPLS) offerings, in which the service provider gives the impression to the customer that their sites are attached to the same Local Area Network (LAN), are also becoming available.

Many service providers are investigating the possibility of using an MPLS-based network to provide a common platform for a wide range of services that are currently typically delivered over multiple distinct networks. Such a multiservice network might carry Public Switched Telephone Network (PSTN) traffic, public Internet and private IP data services, Layer 2 ATM and Frame Relay services, Broadcast TV and TDM traffic. This offers capital and operational cost savings to the network operators by allowing them to operate a single network rather than a separate network for each service type. A key aim of this book is to show how MPLS can provide the necessary mechanisms for this network convergence, e.g. through the use of DiffServ Aware Traffic Engineering (TE), which allows the MPLS network to provide connection-orientated characteristics to particular traffic flows.

1.3 MPLS MECHANISMS

This section gives an overview of the mechanisms underpinning MPLS. Readers who are familiar with these may wish to skip this section.

A fundamental property of an MPLS network is that it can be used to tunnel multiple traffic types through the core of the network. Tunneling is a powerful tool because only the routers at the ingress and the egress of the tunnel need to understand the ‘context’ of the underlying traffic carried over the tunnel (e.g. the protocol that the traffic belongs to and the reachability information required to route and forward it in its native form). This detail is hidden from routers in the core of the network. As a consequence, core devices only need to carry sufficient state to enable them to switch MPLS-encapsulated packets without regard to their underlying content. Besides these aggregation properties, which apply to tunnels in general, MPLS tunnels have the following particular properties:

1. Traffic can be explicitly routed, depending on which signaling protocol is used.
2. Recursion is provided for; hence tunnels can exist within tunnels.
3. There is protection against data spoofing, as the only place where data can be injected into an MPLS tunnel is at the head end of that tunnel. In contrast, data can be injected into an IP tunnel from any source that has connectivity to the network that carries the tunnel.
4. The encapsulation overhead is relatively low (4 bytes per MPLS header).

An MPLS network consists of edge devices known as Label Edge Routers (LERs) or Provider Edge (PE) routers and core routers known as Label Switching Routers (LSRs) or Provider (P) routers. A mesh of unidirectional tunnels, known as Label Switched Paths (LSPs) is built between the LERs in order that a packet entering the network at the ingress LER can be transported to the appropriate egress LER. When packets enter a network, the ingress router determines which Forwarding Equivalence Class (FEC) the packets belong to. Packets that are to be forwarded to the same egress point in the network along the same path and with the same forwarding treatment along that path are said to belong to the same FEC. Packets belonging to the same FEC are forwarded with the same MPLS label. In a simple case, packets whose destination addresses correspond to the same Border Gateway Protocol (BGP) next-hop are regarded by the ingress router as belonging to the same FEC. In other cases, there may be a more granular assignment of packets to FECs. For example, in DiffServ Aware TE, each egress point in the network may have multiple FECs, each belonging to a different traffic class.

It is the role of the ingress LER to determine the appropriate egress LER and LSP to that egress LER associated with the FEC. MPLS has the property

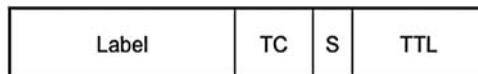


Figure 1.1 MPLS header structure

that multiple traffic types can be multiplexed on to a single LSP. Therefore, if desired by the network operator, a single LSP can be used to carry all the traffic (e.g. L3VPN, public IP and Layer 2) between a particular ingress LER and a particular egress LER. Transit routers along the path of the LSP make their forwarding decision on the basis of a fixed-format MPLS header, and hence do not need to store ‘routes’ (L3VPN routes, external IP routes, Layer 2 forwarding information) pertaining to the underlying tunneled packets. This is an important scaling property, as otherwise each of the core routers would have to carry routing information equivalent to the sum of the routing information carried by all the edge routers in the network.

The following sections describe the fundamental forwarding plane and control plane mechanisms underpinning MPLS.

1.3.1 Forwarding plane mechanisms

Data carried over an MPLS-capable network has one or more MPLS headers applied in order to transport it across the network. The MPLS header structure is shown in Figure 1.1. It contains the following fields:

1. *A 20-bit label value.* MPLS packets are forwarded on the basis of this field. This value is used as an index into the MPLS forwarding table.
2. *Traffic Class (TC) field (3 bits).* Previously known as the EXP bits,¹ convey the Class of Service to be applied to the packet. For example, LSRs and LERs can use these bits to determine the queue into which the packet should be placed. Note that in some cases, as described later in this chapter, the MPLS label value also determines the queuing behavior applied to the packet.
3. *Bottom of stack bit (S-bit).* As described later in this chapter, MPLS headers can be stacked. The S-bit is set on the header of the MPLS packet at the bottom of the stack.

¹Note that the bits in this field were known for many years as the ‘experimental bits’ or ‘EXP bits’ for short. At the time of writing of this edition of this book, they had only been renamed a few months previously, so most people still use the term ‘EXP bits’. For this reason, we will refer to these bits as the ‘EXP bits’ throughout this book.

4. *Time-to-live (TTL) field.* This is used to avoid forwarding loops and can also be used for path-tracing. The value is decremented at each hop and the packet is discarded should the value reach zero.

Packets arriving into the network have one or more MPLS headers applied by the ingress LER. The ingress LER identifies the egress LER to which the packet must be sent and the corresponding LSP. The label value used corresponds to the LSP on to which the packet is placed. The next router performs a lookup of that label and determines the output label that must be used for the next leg of the LSP. The lookup operation on a P router involves reading the incoming label; this yields a new label value to use and the output interface(s) on which the packet should be forwarded. In this way, through this label-swapping paradigm, the packet is conveyed along the LSP from the ingress to the egress LER.

In some simple cases, the use of a single MPLS label is sufficient, e.g. when transporting public IP traffic across a network. In this case, once the packet arrives at the egress LER, the LER performs a normal IP lookup in order to determine which egress link to use. Usually a scheme called Penultimate Hop Popping (PHP) is used. In this scheme, the LSR before the egress LER (i.e. the penultimate router along the LSP) pops the MPLS label and forwards it to the egress LER as an IP packet. This simplifies the processing required at the egress node, as otherwise it would be necessary to pop the label and perform an IP lookup at the egress node. It is not mandatory for the egress router to request PHP behavior, but is the default behavior of most implementations.

In other cases, a single MPLS header is insufficient. This is because the LERs in a particular network may be involved in multiple services – Layer 3 VPN, Layer 2 VPN, VPLS – rather than just the public IP. In this case, the egress LER needs to know which service and which instance of that service (i.e. which customer) the packet belongs to. This is achieved by having an additional MPLS header, which is applied by the ingress LER, corresponding to the service and service instance that the packet must be directed to by the egress LER once the packet has crossed the network. This is illustrated in Figure 1.2.

Let us see how an MPLS packet with two headers is transported between the ingress and egress LERs. The inner header with label Y denotes the

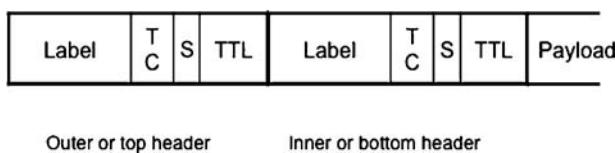


Figure 1.2 MPLS header stack

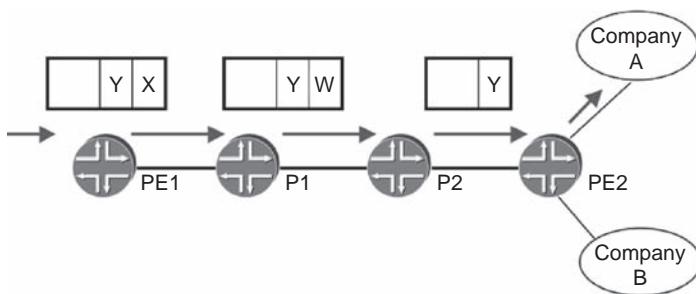


Figure 1.3 Forwarding a packet having two MPLS headers

service and service instance, and the outer header, often called the ‘transport’ header, is the one required to transport the packet from the ingress LER, PE1, to the correct egress LER, PE2. For example, a particular LER may be running several Layer 3 VPN, VPLS and Layer 2 VPN instances. Label Y tells the egress LER that the packet in question corresponds to the Layer 3 VPN service being provided to Company A, rather than any of the other Layer 3 VPN instances or the VPLS or Layer 2 VPN instances. The ability to stack headers in this way gives MPLS key multiplexing and hierarchical properties, allowing a single LSP between a particular ingress and egress point to carry all traffic between those points. As Figure 1.3 shows, the packet leaves the ingress LER, PE1, with an inner label value of Y and an outer label value of X. Routers P1 and P2 perform a lookup based on the outer transport label and do not need to read or take any action based on the inner label. P1 swaps outer label X with outer label W. If PHP is in use, which is typically the case, router P2 pops the outer header, and sends the remainder of the packet to PE2. Thus, when the packet arrives at PE2, the outermost (and only) label is the original inner label, Y, which PE2 uses to identify the packet as belonging to the Layer 3 VPN instance pertaining to Company A.

How does the ingress LER know the label value(s) to use? The transport label is learnt through either the RSVP or LDP signaling protocols, which are described in more detail later on in this chapter. The inner label in the case of most services is learnt via BGP (e.g. Layer 3 VPNs, BGP-signaled Layer 2 VPNs). However, there are also cases where LDP is used, e.g. LDP-signaled Layer 2 transport circuits.

1.3.1.1 MPLS support of DiffServ

DiffServ was developed as a solution to provide Quality-of-Service (QoS). It does so by dividing traffic into a small number of classes and allocating network resources on a per-class basis. To avoid the need for a signaling

protocol, the class is marked directly within the packet header. The DiffServ solution was targeted at IP networks so the marking is in the 6-bit DiffServ Code Point (DSCP) field in the IP header. The DSCP determines the QoS behavior of a packet at a particular node in the network. This is called the per-hop behavior (PHB) and is expressed in terms of the scheduling and drop preference that a packet experiences. From an implementation point of view, the PHB translates to the packet queue used for forwarding, the drop probability in case the queue exceeds a certain limit, the resources (buffers and bandwidth) allocated to each queue and the frequency at which a queue is serviced.

The first challenge with supporting DiffServ in an MPLS network is that LSRs make their forwarding decisions based on the MPLS header alone, so the per-hop behavior (PHB) needs to be inferred from it. The IETF solved this problem by assigning the three experimental (EXP) bits in the MPLS header to carry DiffServ information in MPLS.

This solution solves the initial problem of conveying the desired PHB in the MPLS header, while introducing a new one: how does one map DSCP values expressed in a 6-bit field that can encode up to 64 values into a 3-bit EXP field that can carry at most eight distinct values? There are two solutions to this problem, discussed separately below.

The first solution applies to networks that support less than eight PHBs. Here, the mapping is straightforward: a particular DSCP is equivalent to a particular EXP combination and maps to a particular PHB (scheduling and drop priority). During forwarding, the label determines where to forward the packet and the EXP bits determine the PHB. The EXP bits are not a property that is signaled when the label-switched path (LSP) is established; the mapping of EXP to PHB is configured on each node in the network. The EXP bits can be set according to the DSCP bits of the IP packets carried in the LSP, or they can be set by the network operator. LSPs for which the PHB is inferred from the EXP bits are called E-LSPs (where E stands for ‘EXP-inferred’). E-LSPs can carry packets with up to eight distinct per-hop behaviors in a single LSP.

The second solution applies to networks that support more than eight PHBs. Here, the EXP bits alone cannot carry all the necessary information to distinguish between PHBs. The only other field in the MPLS header that can be used for this purpose is the label itself. During forwarding, the label determines where to forward the packet and what scheduling behavior to grant it, and the EXP bits convey information regarding the drop priority assigned to a packet. Thus, the PHB is determined from both the label and the EXP bits. Because the label is implicitly tied to a per-hop behavior, this information needs to be conveyed when the LSP is signaled. LSPs that use the label to convey information about the desired PHB are called L-LSPs (where L stands for ‘label-inferred’). L-LSPs can carry packets from a single PHB or from several PHBs that have the same scheduling regimen

Table 1.1 Comparison of E-LSPs and L-LSPs

E-LSP	L-LSP
PHB is determined by the EXP bits	PHB is determined by the label or by the label and EXP bits together
Can carry traffic with up to 8 distinct PHBs in a single LSP	Can carry a single PHB per LSP or several PHBs with the same scheduling regimen and different drop priorities
User conservative label and maintains state, because the label is used only for conveying path information	Uses more labels and keeps more state, because the label conveys information about both the path and the scheduling behavior
No signaling is required to convey the PHB information	The PHB information needs to be signaled when the LSP is established
Up to 8 PHBs can be supported in the network when only E-LSPs are used E-LSPs can be used in conjunction with L-LSPs when more PHBs are required	Any number of PHBs can be supported in the network

but differ in their drop priorities (e.g. the set of classes AF_{xy} where x is constant are treated the same from the scheduling point of view but differ in their drop priority according to the value of y). Table 1.1 summarizes the differences between E-LSPs and L-LSPs.

1.3.2 Control plane mechanisms

So far we have seen how MPLS uses labels for forwarding, but how are the bindings between labels and FECs distributed throughout the network? Since manual configuration is not an option, there clearly is a need for a protocol to disseminate this information. From a practical point of view, there are two options: (a) invent a new protocol for distributing label bindings or (b) extend an existing protocol to carry labels in addition to routing information. The question of whether to invent a new protocol or extend an existing one is a popular one in the MPLS world, and we will discuss it in detail in later chapters. At this point, suffice it to say that when the question arises, the result is usually that both approaches are followed.

Regarding the distribution of label bindings, the engineering community invented a new protocol (LDP, or Label Distribution Protocol) and extended two existing protocols (RSVP, or Resource Reservation Protocol, and BGP, or Border Gateway Protocol). The packet formats and basic

operation of these protocols are explained in detail in many introductory texts [Doyle Kolon, Osborne Simha]. Instead of repeating this information here, let us instead examine the properties of the different protocols, and see the benefits and limitations of each of them.

1.3.2.1 LDP

LDP [RFC5036] is the result of the MPLS Working Group [MPLS WG] in the IETF. Unlike RSVP or BGP, which existed well before MPLS and were extended to do label distribution, LDP was specifically designed to distribute labels in the network. Since the goal of LDP is label distribution, LDP does not attempt to perform any routing functions and relies on an Interior Gateway Protocol (IGP) for all routing-related decisions. The original LDP specification was defined for setting up LSPs for FECs representing an IPv4 or IPv6 address. This is the functionality described in this section. The extensions of LDP used for pseudowire and VPLS signaling will be discussed in the appropriate chapters.

LDP was designed with extensibility in mind. All the information exchanged in LDP is encoded as TLVs (type-length-value triplets). The type and length are at the start of the encoding, and their length is known in advance. The type identifies which information is exchanged and determines how the rest of the encoding is to be understood. The value is the actual information exchanged and the length is the length of the value field. TLVs make it easy to: (a) add new capabilities by adding a new type and (b) skip unknown objects by ignoring the amount of data specified in the length field. Over the years, many new capabilities were added to the protocol thanks to this built-in extensibility.

LDP operation is driven by message exchanges between peers. Potential peers, also known as neighbors, that are directly connected to each other over a point-to-point or LAN interface are automatically discovered via hello messages multicast to a well-known UDP port. The protocol also allows for discovery of remote peers using targeted hello messages. In that case, unicast UDP hello messages are sent to the remote neighbor address and may travel through multiple hops to reach the peer.² Either way, once a potential peer is discovered, a TCP connection is established to it and an LDP session is set up. If a pair of peers is directly connected over more than one interface, although LDP hellos are exchanged on all those interfaces, there is only one LDP session between them. At session initialization time, the peers exchange information regarding the features and mode of operation they support. After session setup, the peers exchange information regarding the binding between labels and

² One case in which targeted hello messages are used is the case of LDP over RSVP tunneling, which is discussed in Section 1.3.2.3 of this chapter.

FECs over the TCP connection. The use of TCP ensures reliable delivery of the information and allows for incremental updates, rather than periodic refreshes. LDP uses the regular receipt of protocol messages to monitor the health of the session. In the absence of any new information that needs to be communicated between the peers, keepalive messages are sent.

The association between an FEC and a label is advertised via label messages: label mapping messages for advertising new labels, label withdraw messages for withdrawing previously advertised labels, etc. The fundamental LDP rule states that LSR A that receives a mapping for label L for FEC F from its LDP peer LSR B will use label L for forwarding if and only if B is on the IGP shortest path for destination F from A's point of view. This means that LSPs set up via LDP always follow the IGP shortest path and that LDP uses the IGP to avoid loops.

Relationship between LDP and the IGP

The fact that LDP relies on the IGP for the routing function has several implications:

1. LDP-established LSPs always follow the IGP shortest path. The LSP path shifts in the network when the IGP path changes, rather than being nailed down to a predefined path.
2. The scope of LDP-established LSPs is limited to the scope of the IGP. Thus, LDP LSPs cannot traverse autonomous system (AS) boundaries. The need for Inter-AS LSPs, as well as the solution proposed by the IETF for establishing them, is explained in the Interdomain Traffic Engineering chapter of this book (Chapter 5).
3. During reconvergence, traffic may be blackholed or looped. The existence of loops and the possibility of blackhole traffic is a fact of life for the IGPs during reconvergence. The same properties are inherited by LDP, by virtue of it relying on the IGP for routing decisions. We will discuss how such loops are created and what their impact is in the Protection and Restoration chapter of this book (Chapter 3).
4. In most deployments until recently, the IGP convergence time posed a lower bound on the LDP convergence time. Assuming that the IGP implements smart fast-convergence mechanisms the traffic loss would be of the order of several hundred milliseconds or so, at least an order of magnitude larger than RSVP's fast-reroute time. However, at the time of writing of this edition of this book, implementations had recently added fast-reroute capabilities to LDP. This is discussed in more detail in the Protection and Restoration chapter of this book (Chapter 3).
5. Loss of synchronization between the IGP and LDP can result in traffic loss. As always, for situations where two protocols must operate in tandem, there is a potential for race conditions.

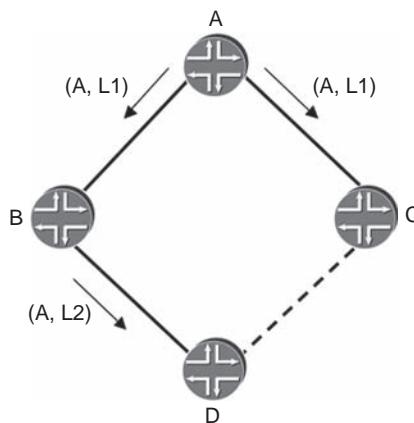


Figure 1.4 Race condition between the IGP and the LGP

Let us take a closer look at a race condition caused by the loss of synchronization between LDP and the IGP. In the diamond-shaped topology in Figure 1.4, LSR A is advertising a binding for its loopback FEC A. To start with, all links have the same metric, and the link C–D does not exist in the topology. From D's point of view, the LSP for FEC A follows the path D–B–A. At a later time the link C–D is added to the topology with a metric that is better than the metric of link B–D, causing the IGP shortest path from D's point of view to be D–C–A. Assume that the IGP reacts faster than LDP. As soon as D finds out about the routing change, it stops using the binding it received from B, thus breaking the LSP. The LSP stays down until a binding for FEC A is received on the LDP session C–D. This may take a while, depending on how fast the session establishment takes place. The situation described here is particularly unattractive, since an alternate path exists in the topology and could have been used until the LDP session comes up on the link C–D.

The above example shows a loss of synchronization caused by the fact that the LDP session on the new link comes up after the IGP session. This is not the only way in which loss of synchronization can occur: forgetting to enable LDP on the new interface, misconfiguring the LDP session authentication, setting up firewall filters that block LDP traffic, or any other event that would cause the IGP to take into account a link but would cause LDP not to use the link, has the same effect.

One solution to this problem is to tie (through configuration) the IGP metric for a particular link to the existence of an LDP session on the link [RFC5443]. When the LDP session is down, the IGP metric advertised for the link is deliberately a very high value. Therefore, if an alternate path is available, the LDP labels on that path can be used. This is discussed in more detail in the MPLS Management chapter of this book (Chapter 13).

Let us now suppose that the link between C and D is operational but undergoes a flap. That is to say, the link goes down and comes up again a few seconds later. Although the technique described in [RFC5443] prevents blackholing of traffic while the session between C and D re-establishes and labels are exchanged, the traffic could be following a suboptimal path through the network for several seconds during this time. An additional technique called ‘LDP Session Protection’ is supported by some LDP implementations to avoid this problem. This works as follows. While the link between C and D is up, they exchange regular hellos in the normal way. When LDP Session Protection is in use, in addition, C and D also exchange *targeted* hellos. Although there are two types of hello message being exchanged, there is only one LDP session between C and D. If the link between C and D fails, the regular hellos can no longer propagate, but as long as there is still IP connectivity between C and D (via A and B in the example), the targeted hellos can continue to travel between C and D so the LDP session stays up. This means that when the link between C and D subsequently comes up, the session does not need to be re-established or label bindings exchanged. Once regular LDP hello messages have been exchanged over the link, the link can be used for forwarding once more.

So far we have seen the implications of having LDP rely on the IGP for the routing function. Next, let us take a look at the choice of label distribution and retention modes made by common LDP implementations.

Label retention and label distribution modes

Label retention mode – which labels to keep? The LDP specification allows the use of both liberal and conservative label retention modes. Conservative retention means keeping only those labels which are used for forwarding, and discarding the rest. This policy makes sense for devices where the label space is a precious resource that must be carefully managed (such as ATM switches). The savings in the label usage come at a cost. Since the ‘uninteresting’ labels are discarded, they must be requested again if they become ‘interesting’ at a later point (e.g. due to a change in routing). Until the requested label arrives, traffic is lost. This undesirable property, coupled with the fact that label space is not a concern in modern routers means that most implementations today use liberal retention.

Label distribution mode – who assigns the labels? The key function of LDP is to distribute bindings between labels and FECs. The goal is to build a forwarding table containing a mapping between an incoming label and an outgoing label. Traffic arriving at the LSR labeled with the incoming label is forwarded labeled with the outgoing label. When building the forwarding table, the question is whether to use the locally picked label as the incoming or the outgoing label. The MPLS architecture [RFC3031] uses downstream label assignment, which means that the router expects to receive the traffic with the label that it picked locally. For example, if LSR

A receives label L1 for FEC F and advertises label L2 for it, then it expects traffic destined for FEC F to come labeled with label L2. When forwarding traffic for FEC F, LSR A labels the traffic with label L1. The traffic flows in the opposite direction from the distribution of labels. The method is called downstream because the label that is assigned to the traffic at point P in the network was actually picked by a router who is one hop further down in the direction of the traffic flow (downstream) from P.

The next question is: should labels be advertised only to those asking for them (on-demand label distribution) or to everyone (unsolicited label distribution)? We have already seen that on-demand label distribution has the undesirable property that traffic is blackholed until the request for the label is satisfied. For this reason, most implementations use the unsolicited label distribution mode. Since LDP uses downstream label allocation, the label distribution mode is usually referred to as downstream unsolicited.

Liberal retention, coupled with unsolicited label advertisements, ensures that labels received from peers are readily available. This is important for handling routing changes in a seamless fashion. To better understand this, let us look at LSR A, which receives two unsolicited label advertisements for FEC F: one with label L1 from peer B and one with label L2 from peer C. LSR A keeps both labels, since it is doing liberal retention. Assuming that the IGP route for FEC F points to peer B, LSR A installs label L1 in its forwarding table. If at some later point the IGP route changes and starts pointing at peer C, all that LSR A has to do is change its forwarding table to use label L2.

Control over the LSP setup

The sole purpose of distributing bindings between labels and FECs is to establish label-switched paths in the network. So far we have discussed a lot of interesting properties of LDP but have not yet answered two key questions: (a) which FEC to advertise a binding for and (b) when to advertise this binding.

The choice of FECs is derived from the LSPs that must be set up in the network. It is independent of the LDP protocol and therefore the LDP specification is silent on this topic. All vendors allow control over the choice of FECs through configuration, but the behavior in the absence of a user-defined configuration is different for different vendors. Some advertise a binding for every prefix in their routing table, while others only advertise a binding for the FEC corresponding to the LSR's loopback address. The outcome in terms of the numbers of LSPs that are set up and of the destinations reachable via these LSPs is quite different. There is no right or wrong decision here, as different implementations may have different constraints. However, from a network operations point of view,

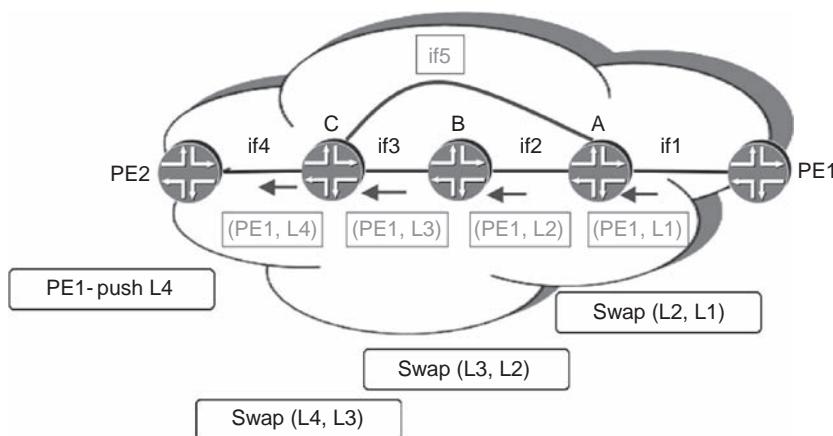


Figure 1.5 Different behavior for the ordered and independent label distribution modes

it is a bad idea to allow LDP to advertise bindings for FECs that will not be used for forwarding. The extra binding and LSP information uses up resources in the network and makes troubleshooting extremely difficult.

The choice of FEC determines which LSPs are set up. The decision when to advertise the label binding determines who has control over the LSP setup. The LDP specification allows two modes of operation: ordered control and independent control. Since not all vendors implement the same mode, let us take a closer look at the two options and their properties, by reference to Figure 1.5. For the purposes of this discussion, assume that link if5 does not exist. This link will be used for a later discussion in this section.

Ordered control. Under ordered control, egress LSR PE1 initiates the LSP setup by assigning label L1 to the FEC corresponding to its loopback address PE1 and advertising this mapping to its peer A. Upon receipt of the label mapping, A evaluates whether PE1 is on the IGP shortest path for that FEC. Since the check is successful, A assigns label L2 for FEC PE1, installs forwarding state swapping labels L2 and L1 and advertises a binding for label L2 and FEC PE1 to its peer B, who will do similar processing. If the check is not successful, A would not advertise the FEC any further. In this fashion, the LSP setup proceeds in an orderly way from egress to ingress. Each LSR consults the IGP for two decisions: (a) whether to advertise a mapping for an FEC and (b) whether to use a label for forwarding.

Independent control. With independent control, each LSR assigns a label for FEC PE1 and advertises this binding independently of the peers. Each LSR uses the locally assigned label as its incoming label in the forwarding table. The outgoing label in the forwarding table is filled

in when the LSR receives a label for PE1 from a peer lying directly on the IGP shortest path for prefix PE1. The LSRs use the IGP for just one decision: whether to use a label for forwarding or not. The success of the LSP establishment depends on all LSR advertising labels for the same set of FECs. If LSR A were configured not to advertise a label for FEC PE1, the LSP to PE1 would never be established.

At this point, it is probably already clear that the default behavior regarding the choice of FECs that are advertised, which we discussed earlier in this section, is not an arbitrary one. With ordered control, the router who is the egress of the LSP decides which FECs to initiate LSPs for. Thus, a reasonable default behavior for an implementation performing ordered control is to advertise a mapping for the loopback address of the egress. With independent control, all routers in the network must advertise the same set of FECs. Thus, the reasonable thing for an implementation performing independent control is to advertise a mapping for all prefixes in the routing table. Another point to note is that when changing the default behavior via configuration, with ordered control the change is applied to one router only (the egress), while with independent control the change must be uniformly applied throughout the network. The requirement for a uniformly applied change is due to the independent operation of the routers in the network: unless they agree on the same set of FECs to advertise, LSPs will not establish end-to-end throughout the network, causing traffic blackholing. This situation is made worse by the fact that the protocol has no built-in mechanisms for detecting such misconfigurations.

The different behavior with regards to the propagation of labels has important implications regarding the setup of LSPs. With ordered control, the bindings must propagate from the egress to the ingress before the LSP is established and traffic can be forwarded on to it. If an application (such as a Layer 3 VPN) relies on the existence of the LSP, then it cannot forward traffic. This behavior is not limited to the initial setup of LSPs. The same dynamics apply when routing changes. With ordered control labels must propagate to the routers in the new IGP path, while with independent control the labels are already available on these routers. This, however, is not as bad as it looks: when routing changes, the IGP messages themselves must propagate and new routes computed, so the propagation of LDP labels is no worse than the propagation of IGP messages.

A more interesting scenario is a failure case where LDP cannot follow the IGP. Let us go back to the example in Figure 1.5. Assume that the interface if5 does not yet exist in the network. The LSP for FEC PE1 (the loopback of router PE1) establishes along the routers PE2–C–B–A–PE1. At this point, the operator decides to add the interface if5 and includes it in the IGP, but forgets to enable LDP on it. As a result, the IGP best path from router C for FEC PE1 is C–A–PE1.

With ordered control, LSR C notices that the label advertisement that it received for FEC PE1 from LSR B does not match the IGP best path, withdraws its advertisement for FEC PE1 and removes its forwarding state. When LSR PE2 receives the withdrawal, it removes the forwarding state for FEC PE1. PE2 knows that the LSP is not operational and will not attempt to forward labeled traffic on it. With independent control, LSR C notices that the routing changed and that the outgoing label it installed in the forwarding table for FEC PE1 is no longer valid and removes the forwarding state for FEC PE1. PE2 does not change its forwarding state, since from its point of view the best path to PE1 is still through C. The net effect is that the LSP for PE1 is broken at point C, but PE2 is unaware of the failure. It will continue to send labeled traffic on this LSP and the traffic will be dropped at C. This type of silent failure is very problematic in a VPN environment, as we will see in later chapters. A solution to this issue is the scheme described in [RFC5443], in which the IGP metric for a link is given a high value if LDP is not fully operational over the link. As described earlier, this scheme is also a solution to race conditions between LDP and the IGP.

Implementations supporting each of the two modes of operation can be and are deployed together in the same network [RFC5037]. The key to interoperability is the fact that LSRs do not assume anything regarding the behavior of their peers, except consistent installation of the forwarding state following the IGP path.

Now that we have discussed the way LDP labels are distributed, let us look at an example of an LDP LSP. Figure 1.6 shows an LDP LSP whose egress point is router D. LDP forms a multipoint-to-point tree rooted at D, with each of the other routers as ingress points to the tree. In the same way, LDP also forms multipoint-to-point trees rooted at each of the other routers in the network, but these are not shown in the diagram for clarity. The numbers inside the boxes show the IGP metric on each link. The arrows show the direction of the data flow, and the number next to each arrow shows the LDP label used on that link for the LSP to D. It can be seen that the LSP path follows the best route as determined by the IGP. On any particular link, the label used to reach a particular destination router is the same, regardless of the origin of the packet. Thus, for example, on link F-C all packets whose destination is D have a label value of 27, regardless of whether they originated at G or A or F. Also, if per-platform label space is used, router C (for example) announces the same label value in order to reach D to all its neighbors, so all traffic passing via C to reach D has the same label value on all links into C. Hence traffic from B to D also has a label value of 27 on the B-C link. Note that in the example, D announces a label value of 3 to its neighbors. This label value of 3 is a special one called the 'Implicit NULL label' [RFC3032]. This triggers PHP on C and E. Because of the special meaning associated with a label value of 3, an

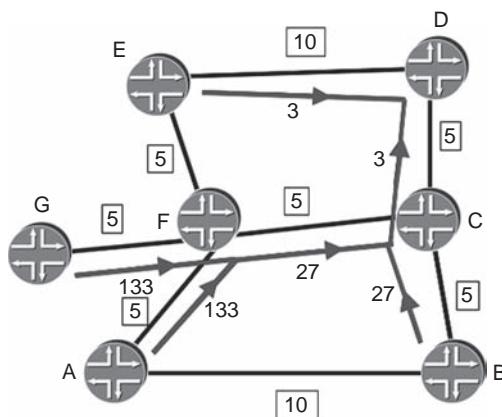


Figure 1.6 Inverted tree formed by LDP rooted at D

MPLS data packet could never have a header with a label value of 3. As already stated, the diagram only shows the tree rooted at D. In reality, there would be multiple overlapping trees, each rooted at a different router in the network. As a result, on any particular link various labels may be in use if multiple routers are reachable over that link.

As with the IGP, typically LDP implementations install multiple forwarding table entries in Equal Cost Multi-Path (ECMP) situations. For example, in Figure 1.6, if the metric between E and D were 5 rather than 10, there would be two equal cost paths from F to D, F-E-D and F-C-D. Hence F installs two forwarding entries for D, one corresponding to each path. Traffic arriving at F for D is load-balanced over the two paths.

LDP key properties

Here is a summary of the key properties of LDP:

- Automatic discovery of peers. LDP uses discovery messages to find peer LSRs. This yields two important benefits:
 - *Ease of configuration.* The operator does not need to configure each peer individually. Adding a new LSR in the network requires configuration of the new LSR, but not of any of the other LSRs in the network (in contrast to RSVP). The automatic discovery built into the LDP protocol is one of the most compelling reasons for picking LDP as the label distribution protocol in networks where traffic engineering is not required.
 - *Session maintenance.* The amount of session state an LSR must maintain is proportional to the number of neighbors. In the absence of targeted peers, this number is constant, regardless of the size of the network.

- *Reliable transport.* LDP uses TCP as the transport protocol for all except the discovery messages. Once advertised, information does not need to be refreshed. Keep-alive messages are sent periodically for session maintenance, but their number is proportional to the number of sessions, not to the amount of information that was exchanged over the session.
- *Extensible design.* LDP uses TLVs for passing information around. This has proven itself over and over as the protocol was extended over the years.
- *Reliance on the IGP.³* LDP relies on the IGP for the routing-related decisions. LDP-established LSPs follow the IGP shortest path and are influenced by changes in routing. During periods of network convergence, LDP LSPs are affected, and traffic may be looped or blackholed.
- *Liberal label retention and downstream unsolicited label distribution.* The labels are advertised to all peers and kept by the peers even if they are not actively used for forwarding. Thus LDP reacts quickly to changes in the IGP routing.

1.3.2.2 RSVP

Another scheme for distributing labels for transport LSPs is based on the Resource Reservation Protocol (RSVP). RSVP was invented before MPLS came into being, and was originally devised as a scheme to create bandwidth reservations for individual traffic flows in networks (e.g. a video telephony session between a particular pair of hosts) as part of the so-called ‘int-serv’ model. RSVP includes mechanisms for reserving bandwidth along each hop of a network for an end-to-end session. However, the original int-serv application of RSVP has fallen out of favor because of concerns about its scalability: the number of end-to-end host sessions passing across a service provider network would be extremely large, and it would not be desirable for the routers within the network to have to create, maintain and tear down state as sessions come and go.

In the context of MPLS, however, RSVP has been extended to allow it to be used for the creation and maintenance of LSPs and to create associated bandwidth reservations [RFC3209]. When used in this context, the number of RSVP sessions in the network is much smaller than in the case of the int-serv model because of the way in which traffic is aggregated into an LSP. A single LSP requires only one RSVP session, yet can carry all the traffic between a particular ingress and egress router pair, containing many end-to-end flows.

An RSVP-signaled LSP has the property that its path does not necessarily follow the path that would be dictated by the IGP. RSVP, in

³Recall that the discussion in this section is for FECs that are IP addresses.

its extended form, has explicit routing properties in that the ingress router can specify the entire end-to-end path that the LSP must follow, or can specify that the LSP must pass through particular transit nodes. Here are a few consequences of the explicit routing properties of RSVP:

1. *The path does not necessarily follow the IGP.* The path can be computed to comply with different constraints that may not be taken into account when the IGP paths are computed. As such, RSVP-signaled LSPs are a key component of MPLS-based traffic engineering, enabling the network administration to control the path taken by traffic between a particular pair of endpoints by placing the LSP accordingly.
2. *The path may be computed online by the router or offline using a path computation tool.* In the case of online computation, typically only the ingress router needs to be aware of any constraints to be applied to the LSP. Moreover, use of the explicit routes eliminates the need for all the routers along the path to have a consistent routing information database and a consistent route calculation algorithm.
3. *The path is not restricted to a single IGP instance.* As long as a path is specified in some way, RSVP is not restricted to a single IGP instance (so, for example, is not confined to one AS). In contrast, LDP is dependent on the IGP, so although LDP LSPs can cross from one IGP area or level to another, they cannot cross from one AS to another, since different ASs run separate IGPs.⁴
4. *An LSP can be signaled in such a way that its path can only be changed by the head end.* This is in contrast to LDP, where each LSR updates its forwarding state independently of all other LSRs as it tracks the IGP state. This property is very important in the context of traffic protection schemes such as fast reroute, discussed in detail in the Protection and Restoration chapter of this book (Chapter 3). Fast-reroute schemes involve each router along the path of an LSP computing a local repair path that bypasses a failure in the downstream link or downstream neighbor node. Traffic sent on the LSP is guaranteed to reach the router where the local repair path has been set up, since the routers do not change their forwarding state after a failure (this again is in contrast to the looping that may happen with LDP following a failure).

The creation of an RSVP-signaled LSP is initiated by the ingress LER. The ingress LER sends an RSVP Path message. The destination address of the Path message is the egress LER. However, the Path message has the

⁴ A workaround is to leak the addresses corresponding to the LDP FECs between the IGPs in the two ASs, but this is cumbersome and is only used in situations where the ASs involved belong to the same owner.

Router Alert option set so that transit routers can inspect the contents of the message and make any necessary modifications.

Here are some of the objects contained in a Path message:

1. *Label Request Object*. Requests an MPLS label for the path. As a consequence, the egress and transit routers allocate a label for their section of the LSP.
2. *Explicit Route Object (ERO)*. The ERO contains the addresses of nodes through which the LSP must pass. If required, the ERO can contain the entire path that the LSP must follow from the ingress to the egress.
3. *Record Route Object (RRO)*. RRO requests that the path followed by the Path message (and hence by the LSP itself once it is created) be recorded. Each router through which the Path message passes adds its address to the list within the RRO. A router can detect routing loops if it sees its own address in the RRO.
4. *Sender TSpec*. TSpec enables the ingress router to request a bandwidth reservation for the LSP in question.

In response to the Path message, the egress router sends an Resv message. Note that the egress router addresses the Resv message to the adjacent router upstream, rather than addressing it directly to the source. This triggers the upstream router to send a Resv message to its upstream neighbor and so on. As far as each router in the path is concerned, the upstream neighbor is the router from which it received the Path message. This scheme ensures that the Resv message follows the exact reverse path of the Path message. Figure 1.7 illustrates the Path and Resv message exchange along the path of an LSP.

Here are some of the objects contained in an Resv message:

1. *Label Object*. Contains the label to be used for that section of the LSP. For example, in Figure 1.7 when the Resv message is sent from the egress

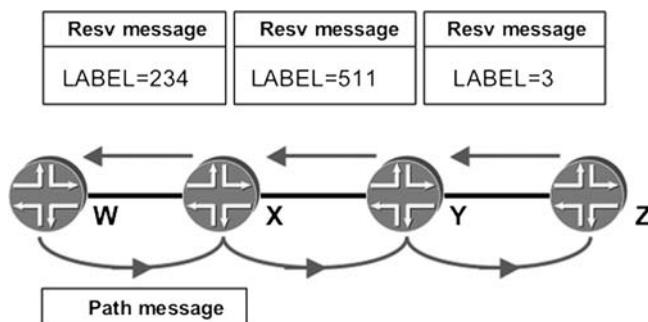


Figure 1.7 Illustration of the RSVP Path and Resv message exchange

router Z to the upstream neighbor Y, it contains the label value that Y must use when forwarding traffic on the LSP to Z. In turn, when Y sends the Resv message to X, it overwrites the Label Object with the label value that X must use when forwarding traffic on the LSP to Y. In this way, for the LSP in question, Y knows the label with which traffic arrives at Y and the label and outgoing interface that it must use to forward traffic to Z. It can therefore install a corresponding label swap entry in its forwarding table.

2. *Record Route Object*. Records the path taken by the Resv message, in a similar way to the RRO carried by the Path message. Again, a router can detect routing loops if it sees its own address in the Record Route Object.

As can be seen, RSVP Path and Resv messages need to travel hop-by-hop because they need to establish the state at each node they cross, e.g. bandwidth reservations and label setup.

As a consequence of the scheme described above, an RSVP-signaled LSP only requires configuration at the ingress router. In typical implementations, properties of the LSP and the underlying RSVP session, such as the ERO and RRO and requested bandwidth, can be viewed on any router along the path of the LSP since that information is known to all routers along the path.

RSVP requires a periodic exchange of messages once an LSP is established in order to maintain ('refresh') its state. This can be achieved by periodically sending Path and Resv messages for each active LSP. If a router does not receive a certain number of consecutive Path or Resv messages for a particular LSP, it regards the LSP as no longer required and removes all states (such as forwarding entries and bandwidth reservations) pertaining to that LSP. The processing overhead of such a scheme can become a scaling concern for a router maintaining a very large number of LSPs. In order to address this, the 'Refresh Reduction Extensions' to RSVP were devised to reduce this overhead. These include a Summary Refresh Extension that allows multiple RSVP sessions (and hence multiple LSPs) to have their state refreshed by a single message sent between RSVP neighbors for refresh interval [RFC2961].

RSVP has an optional node failure detection mechanism, in which hello messages are sent periodically between RSVP neighbors. Without this mechanism, a node might only become aware of the failure of a neighbor through the timeout of RSVP sessions, which can take a relatively long time.

Note that there is no concept of ECMP in RSVP as there is in LDP. A particular RSVP-signaled LSP follows a single path from ingress to egress. If, in performing the path computation, the ingress router finds that there are multiple potential paths for an LSP that have equal merit, it chooses one of those paths for the LSP and signals for its creation via

RSVP. Hence, once traffic has entered an RSVP-signaled LSP, there is no splitting and merging of traffic as sometimes occurs in the LDP case. On the other hand, if the ingress router has multiple RSVP-signaled LSPs to a particular egress router, it can load-balance the traffic across those LSPs. Some implementations allow the load-balancing to be weighted according to the bandwidth reservation of the LSPs.

In some cases, a network may only have a handful of RSVP-signaled LSPs, as a tactical way of controlling traffic flows around particular hot-spots in the network. In those situations, RSVP-signaled LSPs would be created between certain pairs of endpoints to achieve this aim. In other networks, the reason for deploying RSVP-signaled LSPs might be in order to make use of fast reroute, in which case the administrator may choose to fully mesh the PEs in the network with RSVP-signaled LSPs.

By way of summary, here are the key properties of RSVP:

- Explicit routing. The ingress LER has control over the path taken by the LSP, either by specifying the entire path or by specifying particular nodes that the LSP must pass through. As a consequence, RSVP lends itself to traffic engineering and traffic protection schemes that operate independently of, and faster than, the IGP.
- Periodic message exchange is required to renew the state of an LSP, although the RSVP Refresh Reductions reduce this overhead.
- The amount of session state on a node is proportional to the number of LSPs traversing the node. This tends to grow as the network grows (assuming a high degree of meshing of RSVP-signaled LSPs).

1.3.2.3 RSVP and LDP comparison

A frequently asked question is whether LDP or RSVP is the better protocol to use in a deployment. Let us compare the two protocols with regard to the factors that affect the choice of which to use:

1. Ease of configuration:
 - (a) *Initial configuration.* LDP has the advantage that it is easy to configure, only requiring one line of configuration in some implementations, to allow the protocol to run on a particular interface. RSVP, on the other hand, requires explicit configuration of the LSPs on the ingress router. Each router must know all other routers to which it must establish LSPs.
 - (b) *Incremental configuration when new edge devices are added.* For LDP, only the new device must be configured. For RSVP, adding a new router to the edge means configuring LSPs to it from all the existing routers, potentially requiring configuration changes on all other edge routers in the network.

There are currently moves to reduce the configuration effort when using RSVP. One scheme is an automatic meshing capability, where each edge router in the network automatically creates an RSVP-signaled LSP to the other edge routers in the network. Another is an autobandwidth capability, where the bandwidth reservation for an LSP changes in accordance with the volume of traffic using that LSP. Used in combination, the configuration effort would not be very different to that associated with LDP. Such schemes may not help in all cases, however, e.g. when each LSP has particular constraints associated with it or requires a fixed bandwidth reservation rather than one that dynamically varies.

2. Scalability:

- (a) *Control plane sessions.* For LDP, each router must maintain a number of sessions equal to the number of LDP neighbors. For RSVP, the number of sessions is equal to the total number of LSPs that the router is involved with (whether in the role of ingress, transit or egress router). For a fully meshed topology, the total number of LSPs in the network is of order N -squared in the RSVP case, where N is the number of edge routers, but is proportional to N in the LDP case, because each edge router is the egress for an LDP multipoint-to-point tree having every other edge router as an ingress point.
- (b) *State maintenance.* LDP sends periodic keepalive and hello messages, but only for a limited and constant number of neighbors/ sessions. RSVP must refresh all sessions for the LSPs traversing a router, a number over which it has no control. RSVP refresh reduction reduces the number of RSVP messages that have to be created and sent in order to refresh the sessions; however, the router still needs to track the state of each session.
- (c) *Forwarding state.* LDP maintains the forwarding state for all FECs in the network. By nature of the protocol each FEC is reachable from every point in the network. The ability of LDP to support ECMP means that often more than one path is maintained. RSVP, on the other hand, only keeps the state for the LSPs traversing it, and potentially their protection paths.

For practical purposes, the above considerations may not be of practical importance unless one has a very large number of routers that need to be fully meshed with RSVP-signaled LSPs, resulting in an unsustainably large number of LSPs to be maintained by routers in the core of the network. In those cases, either the LDP over RSVP or the LSP hierarchy schemes described later in this section can be used.

3. Features supported.

Currently, only RSVP supports traffic engineering.

From the above analysis it should come as no surprise that if the traffic engineering properties offered by RSVP are not required, LDP is almost always chosen. Let us take a closer look at the choice of protocol in the context of the application for which the MPLS connectivity is required:

1. *L3VPN.* These services often do not have stringent SLAs in terms of outage time in the event of a link failure and although they may offer several Diff-Serv traffic classes, none of the traffic classes have associated bandwidth reservations through the core. The main considerations in this case are ease of management and provisioning. There-fore, to date, LDP has received wider deployment than RSVP in such networks.
2. *Migration of Layer 2 services to MPLS networks.* Emulation of services such as ATM and Frame Relay over MPLS networks often requires tangible bandwidth guarantees. For example, if a service provider offers a particular access rate at a particular class of service between two access points in the network, it is necessary to ensure that the bandwidth between those points is reserved and uncontended. Due to its traffic engineering capabilities (and in particular DiffServ Aware Traffic Engineering), RSVP is better suited than LDP in such deployments.
3. *Services requiring fast restoration, such as voice services.* In some cases, there may be no TE requirement, because link utilization is low and bandwidth plentiful. However, fast-reroute capabilities may still be required, due to the nature of the service (e.g. voice). In the past, RSVP was the only protocol that supported fast restoration so such services often used RSVP-signaled LSPs as the transport. However, more recently fast restoration support has been added to LDP-signaled LSPs so these can be used as an alternative to RSVP-signaled LSPs in certain topologies. The approaches to fast restoration for RSVP-signaled LSPs and LDP-signaled LSPs are discussed in detail in Chapter 3.

In many deployments, each Point-of-Presence (PoP) consists of several access routers and one or two core facing routers. The SP may wish to use RSVP for its traffic engineering properties in the core, but has no need for traffic engineering within the PoP. Similarly, there may be a need for fast reroute in the core but not within the PoP infrastructure, on the premise that intra-PoP link failure is relatively rare.

In these cases, the SP can use LDP within the PoPs and RSVP-signaled LSPs in the core. This is illustrated in Figure 1.8.

In Figure 1.8, each PoP has five LERs and two core-facing LSRs. Each core-facing LSR has an RSVP-signaled LSP to each of the core-facing LSR in the other PoPs. In the figure, we show only the RSVP-signaled LSPs from PoP C to PoP A for clarity. Targeted LDP sessions are created between

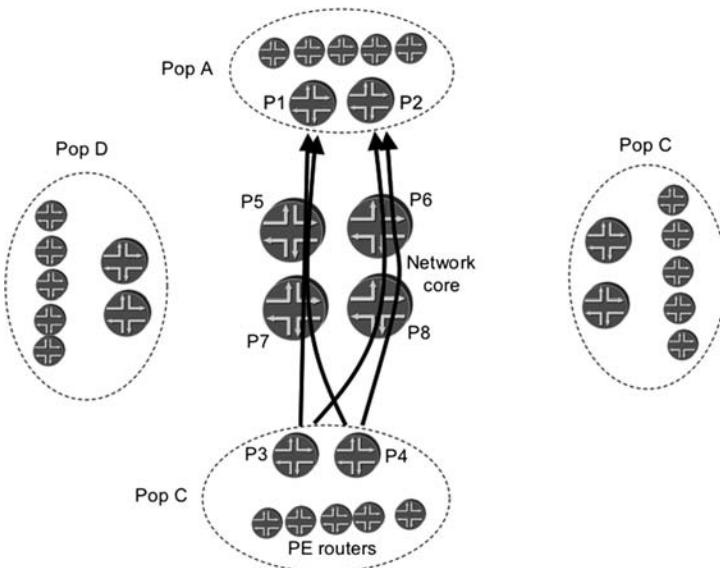


Figure 1.8 Creating an RSVP mesh between core-facing P routers in each PoP

the ingress and egress routers of each RSVP-signaled LSP. A targeted LDP session allows LDP labels to be exchanged between routers even if they are not directly connected to each other so that LDP labels are exchanged without involving the transit routers of the RSVP-signaled LSPs. For example, there would be a targeted LDP session between P3 and P1, and the routers in the core of the network (P5, P6, P7 and P8) would not be involved in this session. Let us look at the impact that the LDP over RSVP scheme has on the total number of RSVP-signaled LSPs in the network. If the number of core-facing routers in the network is X and the number of edge routers in the network is Y, then the number of RSVP-signaled LSPs is reduced from $Y(Y - 1)$ to $X(X - 1)$. This could be a large reduction if the ratio Y to X is large. For example, consider a network that has 30 PoPs, each containing two core-facing routers and five edge routers. In the case where the edge routers are fully meshed with RSVP-signaled LSPs, there would be 22 350 (i.e. 150×149) RSVP-signaled LSPs in the network. In the case where only the two core-facing routers in each PoP are fully meshed, there would be a total of 3480 (i.e. 60×58) RSVP-signaled LSPs in the network.⁵ This is almost an order of magnitude smaller than the full mesh case. The smaller number of LSPs means a lighter load on the

⁵ This calculation assumes that the core-facing router in each PoP does not need an RSVP-signaled LSP to the other core-facing router in the same PoP.

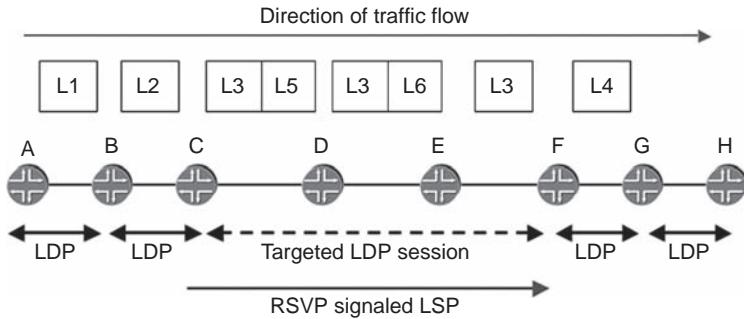


Figure 1.9 LDP over RSVP forwarding

protocols and the routers. This, in itself, is only of practical consequence if the load in the fully meshed edge router case is unsustainably high. More importantly, fewer LSPs means easier provisioning and management from the operator's point of view.

The LDP over RSVP process is illustrated in more detail in Figure 1.9, which shows a cross-section through the edge and core of a network. Routers A, B and C are within the same PoP. Routers F, G and H are within another PoP. D and E are core routers. LDP is used within the PoPs. In the network, the core-facing routers in the PoPs are fully meshed with RSVP-signaled LSPs. Hence there is a pair of RSVP-signaled LSPs between C and F (one in each direction). Also, there are targeted LDP sessions between the core-facing routers in each PoP, i.e. between C and F in the diagram. The targeted LDP session allows C and F to directly exchange labels for the FECs associated with the edge routers in their respective PoPs even though C and F are not directly connected. For example, C learns the label from F to use when forwarding traffic to H. Routers D and E are not involved in the LDP signaling process and do not store LDP labels.

Let us consider the transport of packets arriving into the network at router A and leaving the network at router H. The forwarding plane operation is as follows: ingress router A pushes a label which is learnt via LDP. In the example, the label value is L1, and is the label associated with H, the egress point of the packet. Router B swaps the label for one having the value L2. Router C is the ingress router for the RSVP-signaled LSP across the core. C swaps the existing label L2 for a label value L3 that it learnt via the targeted LDP session with F. Also, it pushes on to the packet a label of value L5 learnt via RSVP. Hence, at this point, the label stack consists of an outer label of value L5 and an inner label of value L3. The core routers D and E are only aware of the RSVP-signaled LSP and hence only carry out operations on the outer label. D swaps the outermost label of value L5 for a label having value L6. Note that the underlying label having value L3 is left untouched. If PHP is in use, router E pops the

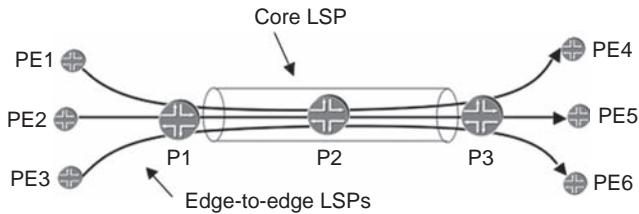


Figure 1.10 LSP hierarchy

label learnt via RSVP, thus exposing the label, L3, learnt via LDP. Router F swaps the LDP label for one having value L4. If PHP is in use, router G pops the label, exposing the header of the underlying packet. This could be an IP header or could be another MPLS header, e.g. a VPN label.

In cases where the properties brought by RSVP are required from edge-to-edge, the above LDP over RSVP scheme is not suitable. However, in the case of very large networks, it may not be feasible either to fully mesh all the edge routers with RSVP-signaled LSPs because of the resulting amount of the RSVP state in the core of the network. The concept of LSP hierarchy [RFC4206] was introduced to solve this problem. In this scheme, a layer of routers is fully meshed with RSVP-signaled LSPs. The layer is chosen such that the number of routers involved in the mesh is less than the number of edge routers. For example, as with the LDP over RSVP scheme discussed earlier, the routers chosen might be the core-facing routers within each PoP. The edge routers are also fully meshed with RSVP-signaled LSPs which are nested within the LSPs between the core-facing routers.⁶ The LSPs in the core of the network are called forwarding adjacency (FA) LSPs. Referring again to Figure 1.8, in the context of LSP hierarchy, the LSPs between P1, P2 and P3 and P4 are the FA LSPs. Each LER would have an RSVP-signaled LSP to each other LER in the network, which would be tunneled in one of the FA-LSPs in order to cross the core. In this way, routers in the heart of the network (P5, P6, P7 and P8 in the figure) only have to deal with the session state corresponding to the core LSPs and are unaware of the fact that LSPs from LER to LER are nested within them.

The LSP hierarchy concept is illustrated in more detail in Figure 1.10. The diagram shows six LERs, three in each of two PoPs. P1 is a core-facing router in one PoP and P3 is a core-facing router in the other PoP. The diagram shows an RSVP-signaled LSP between P1 and P3. Using LSP hierarchy, edge-to-edge LSPs between the LERs in the two PoPs can be nested within the core LSP between P1 and P3. For example, there is an LSP between PE1 and PE4, another between PE2 and PE5 and so on.

⁶Note that, as a consequence, the use of the LSP hierarchy does not solve the issue of the overhead of configuring a full mesh of RSVP-signaled LSPs.

However, P2 in the core of the network is unaware of the existence of these LSPs and is only involved in the maintenance of the core LSP. This is because the RSVP messages associated with the edge-to-edge LSPs pass directly between P1 and P3 without being processed by the control plane of P2. Note that in the data plane, the label operations are analogous to those in the LDP over RSVP case that we showed in Figure 1.9. The ingress router of the FA-LSP pushes a label corresponding to the FA-LSP onto the existing label stack. This label is swapped at each hop of the FA-LSP, leaving the labels underneath untouched and is then typically popped at the penultimate router of the FA-LSP.

1.3.2.4 BGP label distribution

The third type of label distribution also relies on a preexisting protocol, BGP. BGP has support for multiple address families, which make it straightforward to define and carry new types of reachability information and associated attributes. Thus, by adding a new address family to BGP, it is possible to advertise not just a prefix but also one or more labels associated with the prefix. In the Hierarchical and Inter-AS VPNs chapter of this book (Chapter 9), we will see that this capability is essential in the context of inter-AS MPLS/VPNs. The chapter describes several solutions in which BGP is used to:

- (a) distribute the ‘inner’ labels (VPN labels) required by the egress LER to identify the service and service instance that the packet belongs to and/or
- (b) distribute the outer label required to transport a packet to the appropriate egress LER.

The reasons for picking BGP as the protocol for the solution are discussed in detail in the Hierarchical and Inter-As VPNs chapter (Chapter 9). At this point, let us see some of added benefits of using BGP for label distribution:

- The ability to establish LSPs that cross AS boundaries. An example of where this is required is an MPLS-based VPN service having attachment points within multiple providers. In this case, it is necessary to distribute labels pertaining to LER reachability, so that the transport label required to reach a LER in another AS is known. BGP is a protocol that is used today to convey reachability information across AS boundaries; therefore it can easily convey label information across AS boundaries.
- As an aid to scalability in very large MPLS networks – this is discussed in more detail in the context of Seamless MPLS in Chapter 16.

- Reduction in the number of different protocols running in the network. Rather than deploying an entirely new protocol, reuse one of the existing protocols to provide one more function.
- Reuse of existing protocol capabilities. BGP supports a rich set of attributes that allow it to filter routing information, control the selection of exit points, prevent loops, etc. All these capabilities are readily available when label information is distributed along with a prefix.

BGP label distribution is also used in the context of the 6PE scheme to enable transport of IPv6 over an IPv4 MPLS core. This is discussed in the next section.

1.3.3 Transport of IPv6 over an IPv4 MPLS core

Increasingly, service providers are seeing the need to carry IPv6 traffic as well as IPv4 traffic across their networks. As for IPv4 traffic, the IPv6 traffic can be divided into two main categories:

- (i) *Public IPv6 traffic (or 'IPv6 Internet' traffic)*. In this case, the requirement for the service provider is to transport IPv6 packets between IPv6 users across the public Internet infrastructure. In some cases, packets might be transported between users attached to the same service provider's network, but more typically the task of the service provider is to transport IPv6 packets between a service provider customer and an IPv6-enabled peering exchange, for hand-off to another service provider.
- (ii) *Private IPv6 traffic*. In this case, the requirement is to provide a VPN service, to enable IPv6 traffic to be transported between a customer's sites while maintaining separation and privacy from other customers.

In this section, we will examine case (i), the public IPv6 case, in more detail. Case (ii), private IPv6 traffic, will be discussed in the Advanced Topics in Layer 3 BGP/MPLS VPNs chapter.

The service provider has the following choices in terms of how to carry the IPv6 traffic across the network core:

1. Turn on IPv6 forwarding and an IPv6-enabled IGP on all the routers in the network and send the packets in native IPv6 form.
2. Create a mesh of tunnels (such as GRE tunnels) between the PE routers in the network. Thus, the IPv6 packets can be encapsulated in IPv4, avoiding the need to turn on IPv6 in the core of the network.
3. Use MPLS LSPs between the PE routers in the network to carry the IPv6 packets.

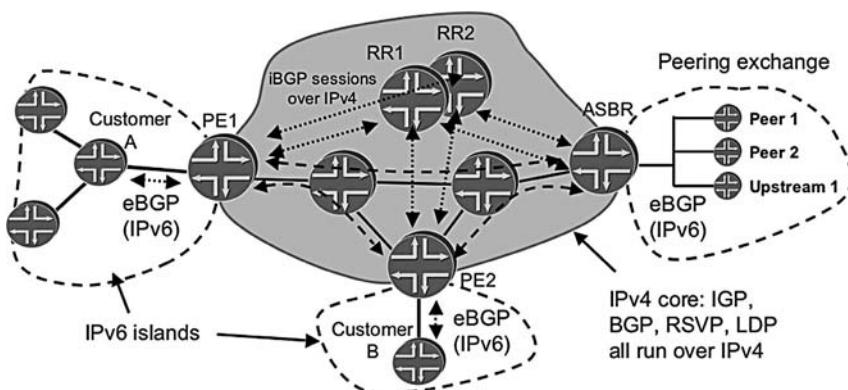


Figure 1.11 Carrying IPv6 traffic across the core using the 6PE scheme. Dotted lines denote BGP sessions. Dashed lines denote MPLS LSPs

Option 1 is of interest to service providers that already carry IPv4 internet traffic across the core in native IPv4 form, because the IPv6 traffic is carried in an analogous way. However, in some cases, a service provider may have core routers that are not capable of running IPv6 or there may be a reluctance to turn on IPv6 on the core routers.

In contrast, Option 2 avoids the need to turn on IPv6 on the core routers, because the IPv6 packets are encapsulated inside IPv4 packets. The issue with this scheme, however, is that typically it involves manual configuration of the tunnels and so has a high operational overhead.

Option 3 is attractive to service providers who already use MPLS LSPs to carry their IPv4 internet traffic, as it allows the same LSPs to be used to carry the IPv6 internet traffic too. The configurational overhead is much less than for Option 2.

Let us examine Option 3 in more detail. A scheme called ‘6PE’ [RFC 4798] has been devised to cater for this scenario. The premise behind the scheme is that the core routers in the network do not support IPv6, so only the LERs need to support IPv6 forwarding and an IPv6 protocol. The LSPs used to transport the packets are signaled using IPv4 and can be the same LSPs that are used to transport IPv4 traffic and other traffic such as Layer 2 traffic.⁷ Figure 1.11 illustrates the infrastructure required for the 6PE scheme.

The service provider’s LERs routers each have an eBGP session running over IPv6 to the attached CE routers. Similarly, the service provider’s peering router has eBGP sessions running over IPv6 with peers and

⁷ Although the IETF has defined schemes for signaling LSPs using IPv6, these are not supported by most implementations today.

upstream providers. Within the core of the network, shaded in grey, the addressing and the IGP are IPv4 based. The LERs in the network are meshed with LSPs that are signaled using IPv4. The iBGP sessions for exchanging routes between the LERs also run over IPv4.

In order to discuss the label operations associated with the 6PE scheme, let us re-examine the LSP in Figure 1.7. The LSP in the figure happens to have been signaled using RSVP, but the same holds if it had been signaled using LDP. Imagine if one simply encapsulated the IPv6 packet into the LSP shown in the figure. Between X and Y, the packet would have a label value of 511. At Y, the label would be popped, since PHP is in operation. Although the use of PHP is not mandatory, in practice it is used in the majority of MPLS deployments. This would give rise to the issue that a bare IPv6 packet would be exposed on router Y. However, the premise behind the 6PE scheme is that the P routers do not support IPv6. If this is the case for router Y, then Y would not know how to set the appropriate protocol type in the Layer 2 header before forwarding the packet on the link to Z. For example, if an Ethernet link is being used between Y and Z, router Y would need to set the Ethertype on the Ethernet frame to the value assigned for IPv6 payloads. In order to overcome this problem, the 6PE solution makes use of an additional label to ensure that the IPv6 packet is not exposed to the penultimate router. This is illustrated in Figure 1.12.

There is an MPLS LSP from PE1 to PE2, signaled by LDP or RSVP. On PE1, an MPLS header having label value Y is pushed onto the IPv6 packet. On top of that, another MPLS header having label value X is pushed onto the packet. The label value X is the one signaled by LDP or RSVP. At P1, the outer label value is swapped for a label having value W. At P2, the outer label is popped, exposing the inner label having value A. The packet is forwarded with this label to PE2. But how does PE1 know what label value is required for the inner label? The answer is to use Multi-Protocol (MP) BGP. In this way, when PE2 advertises its IPv6 prefixes in BGP, it also advertises the label value associated with them. The Address Family

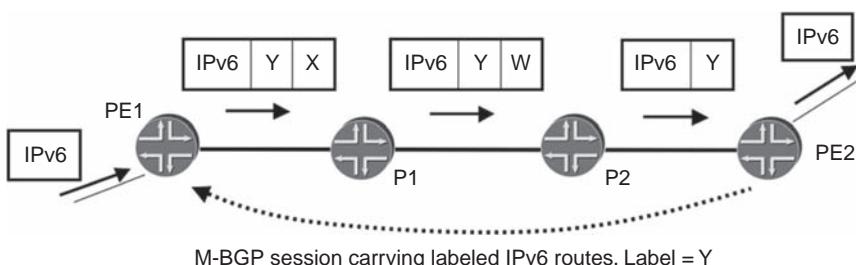


Figure 1.12 Using an extra label in the 6PE scheme

Indicator (AFI) used for this advertisement has a value of 2, signifying IPv6. The Subsequent Address Family Indicator (SAFI) used has a value of 4, signifying a labeled route. The same BGP session can also be used to advertise IPv4 prefixes, without a label. The same LSP can be used to carry IPv4 traffic as is used to carry IPv6 traffic between PE1 and PE2. At each hop, the label stack for an IPv6 packet would have one extra label compared to the label stack for an IPv4 packet.

Note that 6PE is not a VPN scheme. If the requirement is to provide a VPN service capable of transporting a customer's IPv6 packets, the scheme discussed in the Advanced Topics in Layer 3 BGP/MPLS Virtual Private Networks chapter should be used.

1.4 CONCLUSION

We have started this chapter by looking at the original goals of the MPLS Working Group back in 1997. As is often the case for successful technologies, MPLS has become a key component in the development of new applications that were not envisioned at the time MPLS started out. The following chapters take a closer look at many of the innovations made possible by MPLS.

1.5 REFERENCES

- | | |
|-----------------|---|
| [Davie Rekhter] | B. Davie and Y. Rekhter, <i>MPLS: Technology and Applications</i> , Morgan Kaufmann, 2000 |
| [Doyle Kolon] | J. Doyle and M. Kolon (eds), <i>Juniper Networks Routers: The Complete Reference</i> , McGraw-Hill, 2002 |
| [MPLS97] | Original problem statement for the IETF-MPLS Working Group, http://www.ietf.org/proceedings/97apr/97apr-final/xrtftr90.htm |
| [MPLS ALLII] | T. Walsh and R. Cherukuri, <i>Two Reference Models for MPLS Control Plane Interworking</i> , MPLS/FR Alliance Technical Committee document mpls2005.050.00, March 2005 |
| [MPLS WG] | IETF MPLS Working Group, http://ietf.org/html.charters/mpls-charter.html |
| [Osborne Simha] | E. Osborne and A. Simha, <i>Traffic Engineering with MPLS</i> , Cisco Press, 2002 |
| [RFC2961] | L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasiand and S. Molendini, <i>RSVP Refresh Overhead Reduction Extensions</i> , RFC 2961, April 2001 |

- [RFC3031] E. Rosen, A. Viswanathan and R. Callon, *Multi-protocol Label Switching Architecture*, RFC 3031, January 2001
- [RFC3032] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li and A. Conta, *MPLS Label Stack Encoding*, RFC 3032, January 2001
- [RFC3209] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan and G. Swallow, *RSVP-TE: Extensions to RSVP for LSP Tunnels*, RFC 3209, December 2001
- [RFC3945] E. Mannie, *Generalized Multi-protocol Label Switching (GMPLS) Architecture*, RFC 3945, October 2004
- [RFC4206] K. Kompella and Y. Rekhter, *LSP Hierarchy with Generalized MPLS TE*, RFC 4206, October 2005
- [RFC4798] J. De Clercq, D. Ooms, S. Prevost, F. Le Faucheur, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*, RFC 4798, February 2007
- [RFC5036] L. Andersson, I. Minei and B. Thomas (eds), *LDP Specification*, RFC 5036, October 2007
- [RFC5037] L. Andersson, I. Minei and B. Thomas, *Experience with the Label Distribution Protocol (LDP)*, RFC 5037, October 2007
- [RFC5443] M. Jork, A. Atlas and L. Fang, *LDP IGP Synchronization*, RFC 5443, March 2009

1.6 FURTHER READING

- [RFC3478] M. Leelanivas, Y. Rekhter and R. Aggrawal, *Graceful Restart Mechanism for Label Distribution Protocol*, RFC 3478, February 2003
- [RFC3988] B. Black and K. Kompella, *Maximum Transmission Unit Signaling Extensions for the Label Distribution Protocol*, RFC 3988, January 2005

1.7 STUDY QUESTIONS

1. List the fields in an MPLS header and describe their function.
2. Describe the two different schemes by which the Diff-Serv Per-Hop Behavior (PHB) can be inferred for an LSP.
3. Describe the differences between ordered and independent control modes for LDP LSP creation.

4. List some of the differences between LDP and RSVP.
5. A network has 100 LERs. How many LSPs are there in the network in total if it is required to fully mesh the LERs with RSVP-signaled LSPs?
6. A service provider wishes to carry IPv6 Internet traffic. The edge routers in the network support IPv6, but the core routers do not. List the methods by which the service provider can carry the IPv6 traffic across the network.
7. Describe the protocol machinery required for the 6PE scheme.

2

Traffic Engineering with MPLS (MPLS-TE)

2.1 INTRODUCTION

Controlling the path taken by traffic through a network is called traffic engineering (TE). There are many reasons why network operators want to influence the path traffic is taking in their networks. The most popular reason is improving utilization of network resources. The goal is simple: avoid a situation where parts of the network are congested while others are underutilized. Other reasons for using traffic engineering include ensuring that the path has certain characteristics (e.g. it does not use high-latency links), ensuring that transmission resources are available along a particular path, and determining which traffic gets priority at a time of resource crunch (e.g. following a link cut).

This chapter describes why MPLS is a useful technology for implementing traffic engineering and how it accomplishes the goal of steering traffic around the network.

2.2 THE BUSINESS DRIVERS

Influencing the path that traffic takes in the network can increase revenues in two ways:

1. Offering new services with extra guarantees.
2. Lowering the investment in new network resources (primarily bandwidth) by improving the utilization of existing resources.

‘Offering new services’ means any guarantee that the operator can charge extra money for. One example is the ‘guaranteed bandwidth service’, which simply means that a certain amount of bandwidth is available for a particular customer’s traffic, both in the steady state and under failure conditions.

‘Improving resource utilization’ means avoiding a situation where part of the network is congested while other parts are underutilized. For example, if some part of the traffic is routed around a congested link on to a path where enough bandwidth is available, the upgrade of the congested link can be delayed. Avoiding congestion also means better quality for the customer traffic: less loss, less delay and better throughput.

Another important cost-saving measure achieved through traffic engineering is increasing the maximum percentage of link utilization. Operators constantly monitor link utilization to determine at which point it is necessary to schedule a link upgrade. Each network has its own upgrade rules, expressed in terms of the percentage of link utilization that triggers an upgrade. A typical rule is to upgrade at 50% utilization, to be able to accommodate traffic shifting from a failed link. The benefit of traffic engineering in this context is that it can allow a higher utilization of the links, because there is more control over the path that traffic takes in the network, both under normal operation and in the failure case. By increasing the average percentage of link utilization, the upgrade of links can be delayed.

From this discussion, it should be clear that traffic engineering is not always required. If bandwidth resources are plentiful or utilization is low, there will be no congestion, not even following a link failure. If there are no high-latency links in the network, there is no need to worry about traffic crossing high-latency links. Indeed, as seen in the Foundations chapter of this book (Chapter 1), not all MPLS deployments are used for traffic engineering, and depending on the label distribution protocol used, not all MPLS networks can indeed provide traffic engineering. Thus, although traffic engineering is often equated with MPLS, an MPLS network does not necessarily mean a traffic engineered network.

An important thing to bear in mind when discussing the benefits of traffic engineering is that the means through which traffic engineering is achieved must be simple enough to deploy and maintain. In financial terms, the added cost of operating a more complex network must be justified by the new revenue brought in by traffic engineering. MPLS provides the required operational simplicity, along with the flexibility for implementing complex traffic engineering policies.

2.3 APPLICATION SCENARIOS

[RFC2702] lays out the requirements for traffic engineering with MPLS by listing the desirable properties of a TE solution. Rather than discussing the

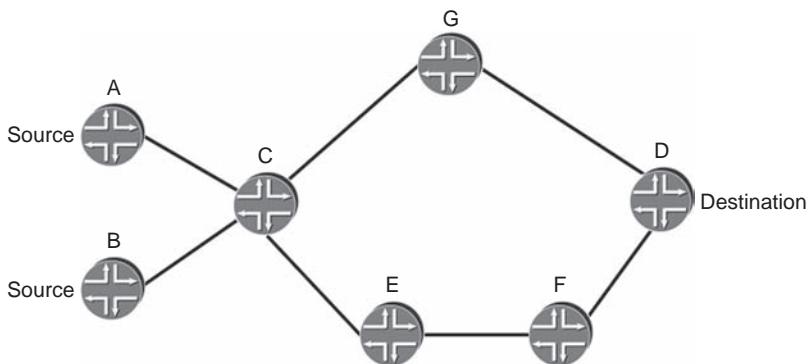


Figure 2.1 A network with two sources, A and B, and two unequal cost paths to destination D

requirements in abstract terms, the following section illustrates them by looking at three application scenarios. At the end of this section we discuss why MPLS is a powerful tool for satisfying these requirements.

The network topology used in all scenarios is shown in Figure 2.1. Two sources, A and B, send traffic to destination D through node C.

The cost of a path is measured in terms of the link metrics. When all metrics are equal, the cost simply translates to the number of hops in the path. In the sample network, two unequal cost paths exist between C and D. All links are of the same capacity and speed, unless specified otherwise.

The first application scenario highlights the need to forward traffic along a predefined path. In the example network in Figure 2.1, assume that A and B are two customers and that customer A buys a service that guarantees low latency. Assume that the link G–D is a satellite link. From the operator's point of view, the requirement is that traffic originating at A should avoid the high-latency link G–D. Thus, a path must be found from A to D that avoids the link G–D. In the simple network from Figure 2.1, the path from A to D can be easily identified as A–C–E–F–D. Regardless of how this path is computed, the problem becomes one of how to forward traffic originating at A along the path A–C–E–F–D.

It is difficult to satisfy this requirement in an IP network. The challenge with IP lies in the fact that forwarding is done independently at every hop, based on the destination of the traffic. Therefore, enforcing a forwarding policy that takes into account the source of the traffic is not always possible and when possible it is not always straightforward. In this example, it is required not only that both paths to D be of equal cost but also that node C must know that it handles a packet from A in order to forward the traffic towards E rather than towards G. In this simple network, the source can be identified based on the incoming interface, but in more complex topologies this may not be the case (e.g. in a similar network where A

and B are connected to router C1, which is directly connected to C). The requirement for traffic engineering is the ability to forward traffic along a path specified by the source – in other words, explicit routing.

The second application scenario shows the requirement for improving resource utilization. In the network shown in Figure 2.1, assume that the capacity of all links is 150 Mbps and that source A sends 120 Mbps and source B sends 40 Mbps towards destination D. If traffic follows the shortest path, 160 Mbps of traffic cross path C–G–D, exceeding link capacity and causing congestion and loss. By splitting the traffic between both the shortest path (C–G–D) and the longer path (C–E–F–D), congestion is avoided and 80 Mbps of traffic traverse both links. This can be achieved, for example, by manipulating the costs of the links to make them behave as equal cost. Once two equal paths exist, traffic can be split (load balanced) between them.

However, this approach is not foolproof. Imagine now that under the same conditions, the capacity of all links is 150 Mbps, except for link E–F, which has a capacity of only 50 Mbps. In this case, the link E–F will be congested under the 80 Mbps load provided by the previous solution. The requirement is to specify the bandwidth requirements between each source/destination pair, find a path that satisfies these requirements and forward the traffic along this path.

The final scenario shows the need for control over the resources at a time of resource contention. The same network as the previous example is used and all links are 150 Mbps except E–F, which is 50 Mbps. Assume A sends 100 Mbps of traffic and B 40 Mbps. Also assume that customer B buys a service with strict service guarantees, while A does not. Under normal conditions, the 140 Mbps can be placed on the shortest path. However, when the link G–D fails, there are not enough resources on the alternate path to carry both A and B's traffic, so congestion and loss occur. To protect B's traffic, one could take a DiffServ-based approach and map B's traffic to a more preferred class. However, such an approach is not always feasible, for two reasons:

1. Under normal conditions, it may well be the case that both A and B's traffic should receive the same treatment.
2. Operators typically strive to minimize the number of behavior aggregates they support in the network and implementing a priority scheme between traffic originated by two different sources increases the number of DiffServ code points.

In this case, the requirements are to find a path from the source to the destination that complies with the bandwidth constraints and to enforce the priority of the path sourced at B over the path sourced at A. Thus, after the link G–D fails, only the path sourced at B can set up on the alternate

links. The path sourced at A will not find enough resources and so traffic from A will not interfere with traffic from B.

The three application scenarios described in this section boil down to two requirements: computing a path between source and destination that complies with a set of constraints and forwarding traffic along this path. As discussed in the Foundations chapter (Chapter 1), MPLS can easily forward traffic along an arbitrary path. The explicit routing capabilities of MPLS, implemented in RSVP [RFC3209] with the Explicit Route Object (ERO), allow the originator of the LSP to establish the MPLS forwarding state along a path defined at the source. Once a packet is mapped on to an LSP, forwarding is done based on the label, and none of the intermediate hops makes any independent forwarding decisions based on the packet's IP destination. In the following sections, we will see how the constrained-path computation is accomplished.

2.4 SETTING UP TRAFFIC-ENGINEERED PATHS USING MPLS-TE

As seen in the application scenarios in the previous section, traffic engineering is accomplished in two steps: computing a path that satisfies a set of constraints and forwarding traffic along this path. These steps are discussed in detail in the following sections. However, it is first necessary to introduce the concept of LSP priorities.

2.4.1 LSP priorities and preemption

MPLS-TE uses LSP priorities to mark some LSPs as more important than others and to allow them to confiscate resources from less important LSPs (preempt the less important LSPs). Doing this guarantees that:

1. In the absence of important LSPs, resources can be reserved by less important LSPs.
2. An important LSP is always established along the most optimal (shortest) path that fits the constraints, regardless of existing reservations.
3. When LSPs need to reroute (e.g. after a link failure), important LSPs have a better chance of finding an alternate path.

MPLS-TE defines eight priority levels, with 0 as the best and 7 as the worst priority. An LSP has two priorities associated with it: a setup priority and a hold priority. The setup priority controls access to the resources when the LSP is established and the hold priority controls access to the resources for an LSP that is already established. When an LSP is set

up, if not enough resources are available, the setup priority of the new LSP is compared to the hold priority of the LSPs using the resources in order to determine whether the new LSP can preempt any of the existing LSPs and take over their resources. If so, the other LSP(s) are torn down. By using different LSP priorities in the third application scenario from Section 2.2, the requirement to give traffic from B priority after a failure can be easily satisfied by simply giving the LSP B–D better priority than the LSP A–D.

So far so good, but is it ever necessary to assign distinct setup and hold priorities to an LSP? The answer is ‘yes’, and doing so is the default for many implementations. Assigning an important hold priority (say 0) and a less important setup priority (say 7) to an LSP creates a stable network environment. Using these priorities, a new LSP can never preempt an existing LSP and in turn can never be preempted. Conversely, assigning an unimportant hold priority (say 7) and an important setup priority (say 0) is a recipe for disaster, because it guarantees constant churn if two LSPs compete for the same resource. Imagine that LSP1 has been established over a particular path and that LSP2 wants to use the same links. LSP2’s setup priority is better than LSP1’s hold priority; thus LSP2 can preempt LSP1. When LSP1 attempts to reestablish, it notices that it can preempt LSP2, and so the cycle of preemption continues indefinitely. For this reason, most implementations disallow the configuration of a hold priority that is worse than the setup priority.

Priorities determine the treatment of an LSP in cases of resource contention in the network. They are essential for ensuring that ‘important’ traffic obtains the necessary resources at a time of shortage (e.g. after a link failure). However, this is not their only application. In a network where large LSPs and small LSPs exist, large LSPs are usually given better priorities to prevent setup failures. The reasoning is that smaller LSPs have a better chance of finding the necessary resources over an alternate path.

Having introduced the concept of priorities, we are now ready to start the discussion of path computation.

2.4.2 Information distribution – IGP extensions

As seen in the example scenarios, the requirement is to find a path in the network that meets a series of constraints. Therefore, the constraints must be taken into account when calculating feasible paths to a destination. Some of the constraints are:

1. The bandwidth requested for a particular LSP (such as 10 Mbps from source x to destination y).

2. The administrative attributes ('colors') of the links that the traffic is allowed to cross. An example of a constraint expressed in terms of link colors is to avoid high-latency links, where these links are marked with a particular administrative attribute. Link coloring is discussed in more detail in Section 2.4.3.
3. The metric that is assigned to a link for the purpose of traffic engineering.
4. The number of hops that the traffic is allowed to transit.
5. The setup priority of the LSP.

Other constraints are also possible, such as the inclusion or exclusion of a particular hop in the path or the requirement to place two related LSPs on different links, to ensure that failure of a single link does not affect both LSPs. Note that the constraints fall into two categories:

- (a) link properties such as available bandwidth, link color and traffic engineering metric; and
- (b) LSP properties such as number of hops or priority.

Calculating a path that satisfies a set of constraints requires that the information about whether the constraints can be met is available for each link and that this information is distributed to all the nodes that perform path computation. Therefore, the relevant link properties have to be advertised throughout the network. This is achieved by adding TE-specific extensions to the link-state protocols IS-IS (Intermediate System-to-Intermediate System) and OSPF (Open Shortest Path First) [RFC3784, RFC3630], which allow them to advertise not just the state (up/down) of the links but also the link's administrative attributes and the bandwidth that is available for use by LSPs at each of the eight priority levels. In this way, each node has knowledge of the current properties of all the links in the network. This information is stored in the traffic engineering database (TED) on each router and used in the path computation.

The question, however, is not just what to distribute but also when to distribute it. Link-state advertisements are sent periodically at typically large intervals (on the order of 30 minutes). New advertisements must be sent whenever the link information changes (e.g. when the available bandwidth changes) and they must be propagated throughout the network. To protect the network from being overwhelmed by link-state advertisements, new advertisements are not sent on every change, only on changes that are deemed significant (e.g. a change in the available bandwidth by more than a certain percentage). This necessary throttling creates a tradeoff between the accuracy of the information stored in the TED and the number of link advertisements that the network elements must process.

To summarize, the IGP extensions for traffic engineering ensure that the TE-related link attributes are available at all the nodes in the network. Next we will see how they are used in the computation of the constrained path.

2.4.3 Path calculation – CSPF

Like conventional SPF, constrained SPF (CSPF) computes a shortest path with regard to some administrative metric. CSPF takes into account only paths that satisfy one or more user-defined constraints (such as available bandwidth) by pruning out of the network topology links that do not satisfy the constraints. For example, if the constraint is bandwidth, CSPF prunes from the topology links that do not have enough bandwidth. In the second application scenario in Section 2.2, once the LSP A–D is set up for 120 Mbps, only 30 Mbps are available along the path A–C–G–D. Thus, when computing the path for LSP B–D, with a requirement of 40 Mbps, the links C–G and G–D are removed from the topology and CSPF picks the alternate path as the best available.

Another frequently used constraint is link coloring (also called administrative attributes). The concept of link colors is very intuitive. Links are marked with different colors through configuration and a link can be marked with multiple colors if desired, or no colors at all. Up to 32 different colors are available.¹ Figure 2.2 shows an example network where links E–F and F–D are colored ‘red’, link C–D is colored ‘blue’, link C–G is not colored at all while link C–E is colored both ‘red’ and ‘green’. There is no restriction on how link colors are assigned, but they typically correspond to link properties such as latency, loss, operational cost or geographic location.

The colors are used to express the desire to include or exclude a link or set of links from a particular path. For example, if the operator marks all high latency links with the color ‘blue’, he or she can then compute a path that does not cross high-latency links by excluding links marked with the color ‘blue’ from the path. For example, in Figure 2.2, assume link C–D is a high-latency link. LSP1 is set up between C and D with the constraint ‘exclude blue links’. This means that none of the links in the path can be marked ‘blue’. Thus, although the shortest path is through link C–D, this link is excluded from the computation due to its coloring and the LSP must establish the best path in a topology that does not include link C–D, yielding path C–G–D. Similarly, LSP2 is set up between C and D with a constraint of ‘include red links’. Thus, all links in the path must be marked red. Note that for this purpose link C–E, which is marked with two colors (red and green), is acceptable. Although the

¹ The limitation is because of the way link colors are encoded in the IGP advertisements.

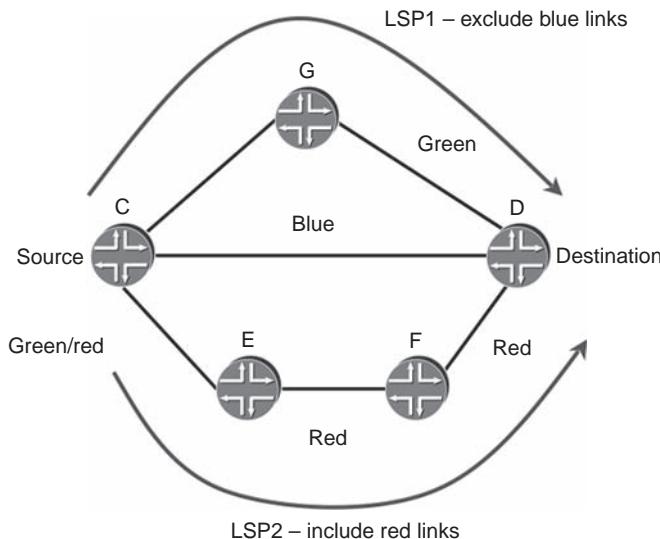


Figure 2.2 Using link coloring

example shown includes and excludes constraints separately, they can be used together for the same LSP. In effect, link coloring creates several TE topologies in the network, where some of the links belong to several topologies.

The reason administrative attributes (colors) are sometimes perceived as intimidating is because of how this feature is implemented by some vendors. Administrative attributes are encoded in a bit field in the IGP link advertisement. From an implementation point of view, the inclusion or exclusion of a link from the computation is accomplished by encoding the user-defined constraints in a similar bit-field format and then performing the necessary bit arithmetic between the link bit field and the constraint bit field. Some implementations force the user to express the constraints in bit format, something non-intuitive for most users. Other implementations offer friendlier interfaces, where instead of bits the user deals with attribute names expressed as strings (words). In both cases, however, the concept is equally simple: tag a link to be able to reference it in the computation. From the CSPF point of view, the links whose colors do not match the constraints are pruned from the topology.

CSPF bases its computation on the information stored in the TED, which is built from information disseminated by the IGP. Therefore:

1. The computation is restricted to a single IGP area. The solutions for extending LSP across area and AS boundaries are explored in the chapter discussing Interdomain TE (Chapter 5).

2. The result of the computation is only as good as the data on which it is based. Recall from Section 2.4.2 that the generation of link advertisements on link attributes changes is throttled. Therefore, the TED is not 100% up to date and the result of the computation is not always accurate. In Section 2.4.4 a scenario is presented where a path is found when in fact there are no available resources, leading to a failure during the path setup.

The LSP path may not have been optimal when it was computed, or may have been optimal at the time but became non-optimal later on, due to changes in the network. Recall that link-state advertisements are sent both periodically and whenever significant events occur. The temptation is to recompute the LSPs based on the updated information and move the traffic to the new paths if they are deemed more optimal. This process is called reoptimization. Reoptimization ensures optimality of the paths, at the cost of stability, shifting traffic patterns in the network and possibly preempting LSPs in networks where multiple priority levels are enforced between the different LSPs. Stable and known traffic patterns are desirable both for debugging and for capacity planning. For this reason, reoptimization is off by default in most vendor implementations and can be turned on with different levels of granularity (periodic, event-driven or manual). The requirements for signaling paths following a reoptimization are discussed in more detail in Section 2.4.4.

Just like SPF, the result of CSPF is a single path. Even if several equally good paths are available, only one is chosen. The tie-breaking rule is one of the following: random, least-fill (causing links to be least full) or most-fill (causing links to be most full). Without a global view of all reservations, both present and future, it is always possible to find a case where any of the algorithms is not optimal.

Let us see an example for least-fill. In Figure 2.1, assume all links are 150 Mbps and the metrics are such that the two paths C–G–D and C–E–F–D are considered of equal cost. The following three LSPs must be set up between C and D: LSP1 and LSP2, with 75 Mbps each, and LSP3, with 150 Mbps. There is enough bandwidth in the network to accommodate all three LSPs. However, depending on the order of the computation and on the tie-breaking algorithm used, the LSPs may not all be created. Under the least-fill algorithm, LSP1 is placed on C–G–D, LSP2 is placed on C–E–F–D and LSP3 cannot be placed. What is needed to make the correct decision in this case is knowledge of all the reservations ahead of time. This is possible when doing offline (rather than dynamic online) path computation. Offline path computation is discussed further in Section 2.8.

Regardless of how the path is actually computed, the label-switching forwarding state must be set up along the path in order to ensure that

traffic does not stray from the desired course. The mechanisms for setting up the label-switching path are described in the next section.

2.4.4 Path setup – RSVP extensions and admission control

After a path has been successfully calculated, it is set up using RSVP-TE.² As discussed in previous sections, the path is specified at the LSP head end in the Explicit Route Object (ERO). However, the ERO is not the only TE-related information that must be carried in the RSVP messages. RSVP must also carry:

- the TE information that intermediate nodes must keep track of, such as the bandwidth requested by the LSP, and
- the information that is relevant in the path setup, such as the setup and hold priorities of the LSP.

As the RESV messages travel from the LSP tail end towards the LSP head end, admission control is performed at each node. Admission control during RSVP signaling is required for the following reasons:

1. The LSP may not have necessarily been computed with CSPF.
2. Even if it was computed with CSPF, the state of the available resources between the time the computation was performed and the path was signaled may have changed (e.g. because another LSP was set up, sourced at a different node).
3. The result of CSPF is only as accurate as the information in the TED (which may not always be up to date because of link advertisement throttling).

If enough resources are available at a particular node, admission control is successful, the path is set up through the node and the available resources are updated. This information is fed back into the IGP so that other nodes in the network become aware of the new state of the available resources. The information may not be immediately distributed, owing to the throttling of link-state advertisements, discussed in Section 2.4.2.

It is important to understand that the bandwidth reservations are in the control plane only and that there is no enforcement of the reservations in the data plane. This means that the data plane usage may be higher than the control plane reservation. When it is important to keep the two equal,

² Although CR-LDP [RFC3212] also supports explicit routing, it never gained much traction. In the context of MPLS, RSVP has become synonymous with TE. The IETF decided in [RFC3468] to abandon new development for CR-LDP.

policing must to be enforced at the ingress of the LSP to ensure that traffic stays within the bounds of the reservation.³

If not enough resources are available, it may be necessary to preempt other LSPs passing through the node. This is where the setup and hold priorities of the LSPs come into play, as explained in Section 2.4.1. If preemption cannot solve the resource problem, the reservation fails and an error message is sent to the head end.

On receipt of the admission control error message, the head end of the LSP recomputes the path. However, if the TED at the head end was not updated in the meantime, it is very likely that the same path is recomputed and the path setup fails again. The IETF standards do not specify a method to avoid this problem. In practice, two things can be done:

1. Exclude the link where the admission control failure was encountered from the CSPF computation for a period of time. Thus, the new path is guaranteed not to use the problematic link. The advantage of this approach is that it is localized to the head end and does not require any extra actions on the node where the failure occurs. The drawback is that the TED database does not get updated, so LSPs sourced by other nodes will encounter a similar failure.
2. Force the generation of a link-state advertisement on an admission control failure, regardless of the throttling mechanism. This ensures that the TED is up to date and the new path does not use the problematic link. The advantage of this approach is that the link information is updated in the TED on all the nodes in the network. The drawbacks are: (a) it requires the computation to happen after a delay, to make sure that the TED was updated, (b) it relies on help from a downstream node which may not implement the same behavior because it is not standardized in any document and (c) it generates extra link-state advertisements that need to be flooded through the network and processed by all the nodes in the network.

Note that the two approaches described above are not mutually exclusive. In fact, they complement each other and are often implemented together.

Another interesting admission control problem arises in the context of reoptimization. Recall from Section 2.4.3 that reoptimization finds a more optimal path in the network, based on new information in the TED. Switching the traffic from the old path to the new must happen without any traffic loss. Therefore, the new path must be set up before the old one is torn down. This method is known as make-before-break. After the new

³ It is not always required to keep the control plane and data plane usage equal. For example, overbooking can be implemented by reporting higher available resources in the control plane than in the data plane.

path is set up, traffic is switched to it and the old path is torn down. This means that for a short period of time, the forwarding state is maintained for both the old path and the new path throughout the network, causing the LSP to consume twice the forwarding resources it would normally use.

Another challenge with make-before-break arises because the new path may use some of the same links as the old path. To avoid double-counting of the resources used by the LSP, which can lead to admission control failures, it is necessary for the old path and the new path to share the bandwidth resources that they reserve. To accomplish this, two pieces of information must be conveyed to all nodes along the path: (a) the desire for reservation sharing and (b) the fact that the two paths belong to the same reservation. The shared explicit (SE) reservation style in RSVP provides support for this behavior.

Once an LSP is set up, traffic can be forwarded along it from the source to the destination. But how does traffic actually get mapped to the LSP?

2.5 USING THE TRAFFIC-ENGINEERED PATHS

The simplest, most basic way to map traffic to LSPs is through static routing. The LSR can be configured to send traffic to a destination by sending it over the LSP. However, the fact that the route must be manually configured to use the LSP is both restrictive and unscalable from an operational point of view, thus limiting widespread use.

To reap the benefits of the traffic-engineered paths, it is necessary for the routing protocols to become aware of the LSPs. From the routing protocol's point of view, an LSP is treated as an interface (a tunnel) and has a metric associated with it. The metric can be the same as that of the underlying IP path or it can be configured to a different value to influence the routing decision. Different routing protocols have different properties and therefore their use of the LSP is different.

The rule for LSP usage in BGP is that when an LSP is available to the BGP next-hop of a route, the LSP can be used to forward traffic to that destination. This property is crucial for the implementation of Layer 3 BGP/MPLS VPNs, as will be seen in the chapter discussing the basics of VPNs (Chapter 7). In a plain IP/MPLS network (non-VPN), this means that if an LSP is set up between the AS border routers (ASBRs), all traffic transiting the AS uses the LSP, with the following consequences:

1. Forwarding for transit traffic is done based on MPLS labels. Thus, none of the routers except the ASBRs need to have knowledge of the destinations outside the AS, and the routers in the core of the network are not required to run BGP. By using an LSP to carry traffic inside the domain it is thus possible to achieve a 'BGP-free core'.

2. The use of an LSP allows tight control over the path that transit traffic takes inside the domain. For example, it is possible to ensure that transit traffic is forwarded over dedicated links, making it easier to enforce service-level agreements (SLAs) between providers.

The use of LSPs by the IGP makes it possible to mix paths determined by constraint-based routing with paths determined by IP routing. Therefore, even when traffic engineering is applied to only a portion of the network, label-switched paths are taken into account when computing paths across the entire network. This is a very important property from a scalability point of view, as will be seen in Section 2.6.1.

In the context of IGPs, there are two distinct behaviors:

1. Allow the IGP on the LSP head end to use the LSP in the SPF computation.
2. Advertise the LSP in the link-state advertisements so that other routers can also take it into account in their SPF (shortest path first).

There is often a lot of confusion about why two different behaviors are needed and how they differ. This confusion is not helped by the fact that the two behaviors are individually configurable and that vendors use nonintuitive names for the two features. To illustrate the difference between the two, refer to Figure 2.3, which shows a simple network topology, with a single LSP set up between E and D, along the path E–F–D, with a metric of 15. Note that the LSP metric in this case is smaller and therefore better than the IGP metric of the path E–F–D, which is 50.

Traffic is forwarded towards destination W from two sources, E and A. The goal is to forward the traffic along the shortest path. For source E, this means taking the LSP E–D and then the link D–W, yielding a metric of 25 (15 + 10). When the SPF algorithm runs at node E, in order to find this path E has to be able to take the LSP E–D into account in the SPF computation. This is the first behavior described above, called autoroute

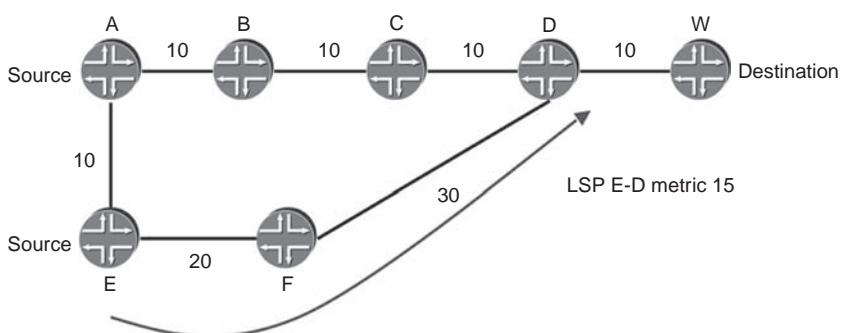


Figure 2.3 IGP use of LSPs

or traffic engineering shortcuts in different vendors' implementations. The concept, however, is very simple: use LSPs originating at a particular node in its SPF computation.

When source A sends traffic to destination W, the path with the smallest metric is through E and the LSP E-D, with a metric of 35 ($10 + 15 + 10$). However, A is oblivious of the existence of the LSP E-D, because the LSP originates at node E. For A to be able to take the LSP into account when computing its SPF, it is necessary for node E to advertise the LSP as a link in the link-state advertisements. This is the second behavior described above, called forwarding adjacency or advertise LSP in different vendors' implementations. The concept is simple: distribute the knowledge about the existence of the LSP to other nodes in the network so they can use it in their SPF computation.

Relying on LSP information distributed by other nodes can sometimes cause surprising behavior. This is because the routing decision is made based on a different router's judgment on what the shortest path should be. Let us continue the example above with a slight modification: the metric of the link E-F is 10 instead of 20, as illustrated in Figure 2.4. Because E advertises the LSP in its link-state advertisements, the node F also receives this advertisement. Consequently, F concludes that the shortest path to destination W is through E along the path F-E-LSP-D-W with a metric of 35 ($10 + 15 + 10$), rather than through the path F-D-W, with a metric of 40. What happens is that the traffic from F is forwarded to E and then right back to F, only to follow the same links as the pure IGP path. This happens because F has no insight into the LSP's path and relies on E's advertisement that traffic to W should be forwarded through it.

Regardless of whether the protocol used is BGP or one of the IGPs, when several LSPs are available to the same destination, most vendors allow the user the flexibility to pick one out of several LSPs for forwarding, based on various local policies. One such policy can use the class-of-service

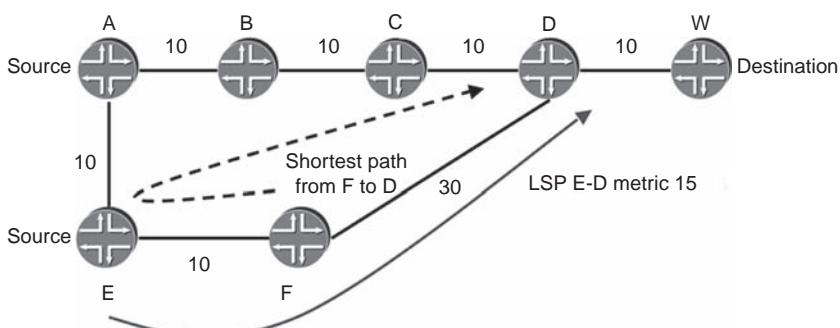


Figure 2.4 Surprising behavior when using LSPs in the shortest path computation

classification of the incoming IP traffic for picking the LSP. For example, best-effort traffic is mapped to one LSP, while expedited forwarding traffic is mapped to another. By manipulating the properties of these LSPs, the operator can provide more guarantees to the more important traffic. Mapping traffic to different LSPs in this way is particularly useful in the context of MPLS DiffServ-TE, as will be seen in the DiffServ-TE chapter (Chapter 4).

To summarize, the ability of the routing protocols to make use of the traffic-engineered paths set up in the network enables control over the path that transit traffic takes in a domain and allows deployment of MPLS-TE in just parts of the network. After seeing how traffic-engineered paths are computed and used, the next thing to look at are some of the considerations for deploying a traffic engineering solution.

2.6 DEPLOYMENT CONSIDERATIONS

2.6.1 Scalability

The number of LSPs that are used to implement the traffic engineering requirements is one of the major scalability concerns for MPLS-TE deployments. Two major factors impact the number of LSPs in the network:

1. *The extent of the deployment.* For any-to-any connectivity, a solution based on RSVP-TE requires a full mesh of LSPs. Assuming N devices meshed, this yields $O(N^2)$ LSPs. For this reason, many MPLS-TE deployments are limited to the core of the network, as explained in the Foundations chapter (Chapter 1). Solutions to this problem using LSP hierarchy are also discussed there.
2. *The size of the reservations.* If the size of the traffic trunk between two points exceeds the link capacity, one solution is to set up several LSPs, each with a bandwidth reservation that can be accommodated by the link capacity. Traffic can then be load-balanced between these LSPs. Although logically a single LSP is necessary, the limitation on the maximum size of the reservation causes several LSPs to be set up in this case. An increasingly popular solution to this problem is to use aggregated interfaces, where several physical interfaces are ‘bundled’ together and treated as a single logical interface from the routing protocol’s point of view. Thus, a single LSP can be set up in this case. The downside is that failure of any interface in the bundle causes the LSP not to be able to establish.

In the context of MPLS-TE deployments, vendors typically express scaling numbers by listing the maximum number of LSPs supported. Most

equipment vendors distinguish between the head end, mid-point and tail end when reporting these numbers. Typical scaling numbers are in the range of several tens of thousand LSPs. It is not uncommon that different numbers are supported for head end and transit (mid point). This is mainly because of two factors: (a) the amount of state that must be maintained is different at different points in the path and (b) the fact that the head end must perform path computation for the LSPs.

When evaluating an MPLS-TE design, an important question is whether the equipment can support the number of LSPs called for by the design. It is usually pretty straightforward to evaluate the number of LSPs for which a box is ingress or egress because this information is derived directly from the tunnel endpoints.

It is less obvious how to determine the number of LSPs that may transit a particular box. The temptation is to assume that all LSPs may cross a single node in the network. This is true for networks where ‘choke points’ exist. Such choke points could be, for example, the PoPs connecting major regions (such as the US and Europe areas of a network). However, in most designs it is safe to assume that transit LSPs are distributed among the routers in the core. Either way, the analysis must be performed not just for the steady state but also for the failure scenarios, when LSPs reroute in the network. Finally, one factor often overlooked when computing the total number the LSPs on a box is the extra LSPs that are created due to features that are turned on in the network. One example is the bypass LSPs used for fast reroute and another example is the extra LSPs created with make-before-break on reoptimization.

The number of LSPs in the network is a concern not only because of the scaling limits of the equipment used but also because of the operational overhead of provisioning, monitoring and troubleshooting a large number of LSPs. In particular, configuring a full mesh of LSPs between N devices can be very labor intensive to set up and maintain. When a new LSR is added to the mesh, LSPs must be established from it to all the other N LSRs in the mesh. However, because LSPs are unidirectional, LSPs must also be set up from all the existing LSRs to the new LSR in the mesh. This is a problem, because the configurations must be changed on N different devices.

The RSVP automesh capability discussed in the Foundations chapter (Chapter 1) automates the process of LSP initiation. The IGP are extended to carry information regarding the membership of an LSR to a particular LSP mesh. When a new LSR is added to the mesh, all the existing LSRs find out about it automatically and can initiate LSPs back to it.⁴

⁴This solution assumes that the properties of the LSPs are fairly uniform for all LSPs originating at a particular LSR or that mechanisms such as autobandwidth (discussed later in this chapter) are used to handle properties that are not uniform.

While RSVP automesh can help alleviate the provisioning challenge of dealing with a large number of LSPs, the burden of monitoring and troubleshooting the LSPs still remains. Operators use automated tools for monitoring, e.g. sending test traffic at regular intervals, gathering statistics and doing Simple Network Management Protocol (SNMP) queries. When the number of LSPs is very large, these operations may take a large amount of resources on the router and of bandwidth in the network. Thus, a tradeoff must be made between the polling frequency and the number of LSPs supported.

The number of LSPs in the network is perhaps the most important deployment consideration for an MPLS-TE network. We have seen that the reservation size impacts the number of LSPs that must be created. However, the reservation size has other effects as well, discussed in the next section.

2.6.2 Reservation granularity

The size of the individual reservations affects not just the number of LSPs that are set up in the network but also the ability to find a path, especially following a failure and the overall utilization of the links. Link capacity is the gating factor on the size of the reservation. In the previous section we saw how this impacts the number of LSPs when the reservation requests exceed the link capacity. In a network that uses links of different capacities, using the minimum link capacity as a gating factor ensures that paths can be established across any of the links. This is especially important when rerouting following a failure. The downside is that using a smaller reservation size creates more LSPs in the network and introduces the challenge of efficiently load balancing the traffic over the LSPs.

The granularity of the reservation can also affect the efficiency of the link utilization. In a network with links of equal capacity, if all the reservations are close to the maximum available bandwidth on each link, there will necessarily be unutilized bandwidth on all links that cannot be used by any of the reservations. In this case it might have been preferable to set up several reservations of sizes such that all the available bandwidth could be used. For example, if all links are of capacity 100 Mbps and all LSPs require 60 Mbps, better utilization can be achieved if instead of a single 60 Mbps reservation, several 20 Mbps reservations are made.

The approach of setting up several LSPs rather than one is not always applicable. The LSPs may not have the same delay and jitter properties, and for this reason balancing the traffic between them may not always be possible. For example, if all 60 Mbps in the previous example are used for a large data transfer by a single application, sending the packets over the different LSPs may cause them to arrive out of order at the destination. However, if the LSP is set up between two BGP peers and is used to

carry traffic to destinations advertised using BGP, then packets can be load-balanced across the LSPs based on the destination address.

The preemption regime employed in the network can also impact the MPLS-TE performance. A common rule of thumb is to preempt smaller LSPs rather than large ones. This is done not only to keep the large LSPs stable but also because small LSPs have a better chance of finding an alternate path after being preempted. This method is also useful in avoiding bandwidth fragmentation (having unutilized bandwidth on the links). This is because smaller LSPs are more likely to be able to establish with the ‘leftover’ bandwidth remaining after the setup of larger LSPs.

2.6.3 Routing challenges

From a routing point of view, the challenges created by having LSPs in the network stem more from different implementation decisions by different vendors’ software than from the technology itself. One aspect in which implementations differ is the default behavior with regards to LSP usage by the BGP protocol. BGP route advertisements include a next-hop address, which is the address of the next BGP router in the path to the destination. Thus, to forward traffic to that destination, a path to the next-hop address must be found. This process is called resolving the route. By default, all vendors use LSP next-hops to resolve VPN-IP routes. As explained in the L3VPN introductory chapter (Chapter 7), this is required because VPN-IP routes are used for forwarding labeled traffic. However, the default resolution regime of non-VPN-IP routes differs from one vendor to another. The issue here is not one of correct versus incorrect behavior, especially because implementations typically allow the user to control the behavior through the configuration, but rather it is an issue of being aware that such differences may exist and accounting for them properly.

So far, we have seen some of things that must be taken into account when deploying a traffic engineering solution. Next, we will take a look at one of the most popular applications for traffic engineering, namely the optimization of transmission resources.

2.7 USING TRAFFIC ENGINEERING TO ACHIEVE RESOURCE OPTIMIZATION

One of the most popular applications of traffic engineering is for the optimization of transmission resources. In this context, traffic engineering is deployed in one of two ways:

1. *Selectively deployed only in parts of the network.* The goal in this case is to route traffic away from a congested link. This can be thought

of as a tactical application, aimed at solving an immediate resource problem.

2. *Deployed throughout the entire network.* The goal is to improve the overall bandwidth utilization and by doing so, delay costly link upgrades. This can be thought of as a strategic application of the technology, aimed at achieving a long-term benefit.

Both applications solve valid problems and the terms ‘tactical’ and ‘strategic’ should not be assigned any negative or positive connotations.

The classic example of a tactical MPLS-TE deployment is the problem of a scheduled link upgrade that gets delayed. What is needed is a temporary solution to move some of the traffic away from the link until the upgrade actually takes place. Another example is the requirement to optimize a particularly expensive resource, such as an intercontinental link.

The classic example of a strategic MPLS-TE deployment is traffic engineering the core of the network (the WAN links). Another example is a network spanning several geographic locations, where traffic engineering is required in only some of the regions. For example, a network with a presence in both the US and Asia may run traffic engineering only in the Asia region, where traffic rates are high and links run at high utilization.

Regardless of the type of deployment, when optimizing resource utilization using RSVP-TE, the assumption is that the following information is available:

1. The bandwidth requirement for the LSP at the head end.
2. The available bandwidth at each node in the network.

In real deployments, however, this necessary information may not always be readily accessible. The following sections discuss how to deal with missing information.

2.7.1 Autobandwidth – dealing with unknown bandwidth requirements

For both tactical and strategic deployments, the first requirement for setting up a traffic-engineered LSP is to know how much bandwidth to request for it. Estimating this information can be done by looking at traffic statistics such as interface or per-destination traffic statistics or by setting up an LSP with no bandwidth reservation and tracking the traffic statistics for this LSP. Once the traffic patterns are known, an LSP can be set up for the maximum expected demand.

The problem with this approach is that typically the bandwidth demands change according to the time of day or day of the week. By always reserving bandwidth for the worst-case scenario, one ends up wasting bandwidth rather than optimizing its utilization. A more flexible solution is to allow the LSP to change its bandwidth reservation automatically, according to the current traffic demand.

This solution is called auto-bandwidth. The ingress router of an LSP configured for auto-bandwidth monitors the traffic statistics and periodically adjusts its bandwidth requirements according to the current utilization. A new path is computed to satisfy the new bandwidth requirements, in a make-before-break fashion. Once the path is set up, traffic is switched to it seamlessly, without any loss. Auto-bandwidth is not defined in the IETF standards, but rather it is a feature that vendors have implemented to address the problem of traffic engineering when the bandwidth constraints are not known.

2.7.2 Sharing links between RSVP and other traffic – dealing with unknown bandwidth availability

The bandwidth reservation model works under the assumption that the reservations on a link accurately reflect the traffic that is crossing the link. This assumption can break in two cases:

1. Traffic is not kept within the limits of the reservation. The implications of not keeping the traffic within the reservation limits and the use of policers for doing so are discussed in more detail in Section 4.4.7 of the DiffServ-TE chapter (Chapter 4).
2. Not all traffic traversing the link is accounted for. This can be the case when there is a mix of IP and MPLS traffic or a mix of LDP and RSVP traffic on the links, which is usually the case for a tactical MPLS-TE deployment.

The problem with having a mix of RSVP and non-RSVP traffic on a link is that bandwidth accounting breaks. A common misconception is that RSVP traffic is somehow special because it was set up with resource reservations. This is not true. The RSVP reservation exists in the control plane only, and no forwarding resources are actually set aside for it. This fact is often overlooked in network designs, especially ones for converged networks, where some of the traffic must receive better QoS than others. The result is a solution that relies on RSVP with resource reservations to carry the QoS-sensitive traffic and uses LDP for the best-effort traffic. The problem with such a solution is that both the RSVP and the LDP

traffic cross the same links. Currently, routers take into account only RSVP reservations when reporting available resources and when doing admission control. Because the bandwidth utilized by LDP is not accounted for, the bandwidth accounting is not accurate and there is no guarantee that the RSVP reservations will actually get the required bandwidth in the data plane.

One solution to this problem is to rely on DiffServ and map RSVP traffic to a dedicated scheduler queue (or more than one queue). In this model, the bandwidth that is available for RSVP reservation is the bandwidth pool allocated to the scheduler queue (or the sum of the pools if more than one queue). Another solution is to estimate the amount of bandwidth used by ‘other’ (IP/LDP) traffic and reduce (through configuration) the link bandwidth that is available for RSVP reservations. This approach works as long as the non-RSVP traffic does not exceed the bandwidth set aside for it. Statistics monitoring can be used to estimate the traffic demand in the steady state, but no mechanism is available to react to changes in the non-RSVP traffic dynamically (e.g. following a link break somewhere else in the network). Offline tools can help evaluate better the amount of bandwidth set aside for non-RSVP traffic by simulating failure events in the network and how they impact the requirement for bandwidth for ‘other’ traffic on the links.

The important thing to remember is that bandwidth reservations are not a magic bullet. Unless the bandwidth consumption is correctly evaluated, bandwidth reservations do not give any of the guarantees that MPLS-TE strives to achieve. This is particularly important in networks where RSVP is used locally to route traffic around points of congestion.

2.7.3 Other methods for optimization of transmission resources in MPLS networks

The only solution presented so far for doing resource optimization in an MPLS network is traffic engineering with RSVP-TE. However, most MPLS deployments use LDP for label distribution. In an LDP network, the proposition of adding a second MPLS protocol for the sole purpose of achieving resource optimization may not be an attractive one.

An alternative way of doing resource optimization in LDP networks is based on the observation that LDP label-switched paths follow the IGP. Thus, by traffic engineering the IGP paths, the LDP LSPs are implicitly traffic-engineered. A real-world example of a traffic-engineered LDP deployment was presented at the Nanog33 conference [LDP-TE]. The goal was to achieve better resource usage by allowing a higher percentage of

the link to be utilized before triggering an upgrade. Traffic engineering the IGP paths was accomplished by manipulating the link metrics.

There are two main challenges when doing traffic engineering through IGP metric manipulation:

1. Changing the IGP metric on one link in one part of the network may impact routing in a different part of the network.
2. To allow higher link utilization safely, it is necessary to prove that traffic on any given link does not exceed 100% under any kind of failure.

Being able to analyze both these factors requires the ability to simulate the network behavior with different link metrics and under different failure scenarios. Thus, an offline tool is required both for planning and for validation of the design. This means that the metrics are computed offline, based on the current traffic information and after simulating different types of failures in the network. Once the metrics are set in the network, the link utilization is monitored to detect when it becomes necessary to re-optimize the computation. It is not the intention to modify the IGP metrics on a failure, because this approach would not be feasible from an operations point of view. Instead, the IGP metrics are chosen in such a way that even under failure conditions no link gets over-loaded.

Given all these constraints, the question is how good is the result obtained through metric manipulation when compared to explicit routes computed with constrained routing and signaled with RSVP-TE. The answer is that it does not matter, as long as doing traffic engineering improves the existing situation by an amount that justifies the extra work involved. When choosing a metric-based approach over explicit routing, the operator is making a conscious decision to trade off some of the benefits of explicit routing, such as unequal-cost load sharing or fine-grained traffic engineering, for the sake of a simpler network design. Test results from one vendor [IGP-TE] [LDP-TE] show, not surprisingly, worse results using a metric-based approach than explicit routing, but much better results than no traffic engineering at all.

2.8 OFFLINE PATH COMPUTATION

Traffic engineering with RSVP-TE relies on explicit paths. Most of the discussion so far focused on a model where the paths are computed dynamically by the routers. As seen in previous sections, the results of this computation may not be the most optimal. Offline computation tools are used to provide better results. This model is particularly familiar to operators from an ATM PVC (permanent virtual channel) background.

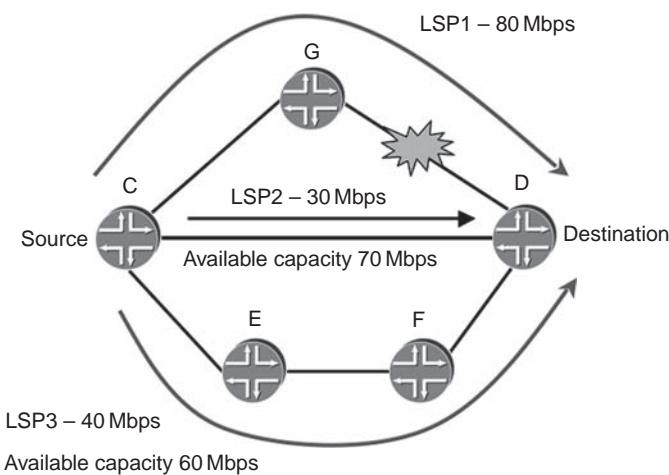


Figure 2.5 LSP placement must take into account failure scenarios

Offline computation tools provide the following advantages in the context of traffic engineering:

1. *Exact control of where the paths are placed.* The operator knows where the traffic is going to flow. There are no surprises from dynamic computation.
2. *Global view of the reservations and of the bandwidth availability.* As seen in Section 2.4.3, this global knowledge enables optimal placement of the LSPs.
3. *Ability to cross area and AS boundaries.* The computation is not based solely on the information in the TED; therefore, the restriction to a single IGP area does not apply.⁵
4. *Computation can take into account both the normal and the failure cases.* One of the biggest strengths of offline tools is that they can take into account the impact of one or more link failures when computing the optimal placement of LSPs. Doing so can ensure that LSPs will always be able to reroute following a failure. Figure 2.5 shows an example of such a scenario. Assume all links are 100 Mbps and three LSPs are set up: LSP1 on C-G-D with 80 Mbps, LSP2 on C-D with 30 Mbps and LSP3 on C-E-F-D with 40 Mbps. Under this setup, a failure of the link G-D will cause LSP1 not to be able to reestablish, because none of the alternate paths has the capacity to accommodate it. If instead

⁵ This assumes that the TE information for the other area/AS is available through some other means.

LSP2 and LSP3 were taking the same links, LSP1 could have rerouted under failure.

5. *Optimality of the solution.* The computation is done offline and can take a long time to complete. More sophisticated algorithms than CSPF can be employed to look for the most optimal solution. The solution can be optimized for different factors: minimize the maximum bandwidth utilization on all links, minimize the number of changes to existing LSPs, achieve protection in case of a single failure and so on. Perhaps the biggest advantage of an offline computation tool is that it can perform optimizations taking into account all the LSPs in the network, while CSPF can take into account only the LSPs originated at the node performing the computation.

The benefits of offline computation come at a cost. Here are a few of the challenges of offline tools:

1. *Input to the computation.* The result of the computation is only as good as the data that the computation was based on. The traffic matrix, the demands of the LSPs and the available bandwidth must be correctly estimated and modeled.
2. *Global versus incremental optimizations.* As network conditions change, the computation must be repeated. The result of the new computation may require changes to a large number of LSPs and configuration of a *large number of routers*. To perform configuration changes, routers are typically taken offline for maintenance. For practical reasons it may not be desirable to use the result of a computation that calls for a lot of changes in the network. Instead, an incremental optimization may be more appealing: one that strives to take into account the new network conditions while leaving as many of the LSPs in place as possible. The result of an incremental optimization is necessarily worse than that of global optimization, but the tradeoff is that fewer routers need to be reconfigured.
3. *Order of the upgrade.* Following a re-computation, it is not enough to know the paths of the new LSPs, but also in which order these LSPs must be set up. This is because the reconfiguration of the routers does not happen simultaneously, so an old reservation setup from router A that is due to move may still be active and take up the bandwidth on links that should be used by a new reservation from router B.
4. *Limitations of the computation.* The result of the computation assumes certain network conditions (such as a single failure in the network). To respond to changing conditions in a network, such as a link cut, the computation must be redone. However, the computation is fairly slow and applying its result requires router configuration, which is not

always possible within a short time window. Therefore, reacting to a temporary network condition may not be practical. By the time the new computation has been performed and the changes have been applied in the network, the failure might already be fixed.

Offline computation tools can ensure optimality of the paths at the cost of the effort required to keep the simulation results and the network state synchronized. Operators have the choice of using (a) offline computation for the primary and the secondary (backup) paths, (b) offline computation of the primary and dynamic computation of the secondary paths or (c) dynamic computation for both primary and secondary paths (secondary paths are discussed in detail in the Protection and Restoration chapter, Chapter 3). Offline tools are available from several vendors, including Wandl (www.wandl.com), Carden (www.carden.com) and Opnet (www.opnet.com). Some operators develop their own simulation and computation tools in-house, tailored to their own network requirements.

2.9 CONCLUSION

We have seen how MPLS-TE can be used to build paths with band-width guarantees and how paths can be made to avoid certain links by marking such links with the appropriate administrative value and excluding them from the path computation. Using the traffic-engineered path, it is possible to achieve efficient bandwidth utilization, guarantees regarding resource allocation in times of resource crunch and control over the path that the traffic is taking in the network.

However, the traffic engineering solution presented so far has three limitations:

1. It operates at the aggregate level across all the DiffServ classes of service and cannot give bandwidth guarantees on a per-DiffServ-class basis.
2. It is limited to a single IGP area and to a single AS.
3. It provides no guarantees for the traffic during failures.

In Chapters 4 and 5 we will see how the traffic engineering solution is extended to overcome the first two limitations, using MPLS DiffServ Aware TE and interdomain traffic engineering. In Chapter 3 we will look at mechanisms available for protection and restoration, which overcome the third limitation listed above.

2.10 REFERENCES

- [IGP-TE] A. Maghbouleh, *Metric-Based Traffic Engineering: Panacea or Snake Oil? A Real-World Study*, presentation at Nanog 27, <http://www.nanog.org/mtg-0302/arman.html>
- [LDP-TE] M. Horneffer, *IGP Tuning in an MPLS Network*, presentation at Nanog 33, <http://nanog.org/mtg-0501/horneffer.html>
- [RFC2702] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, *Requirements for Traffic Engineering over MPLS*, RFC2702, September 1999
- [RFC3209] D. Awduche et al., *RSVP-TE: Extensions to RSVP for LSP Tunnels*, RFC3209, September 2001
- [RFC3212] B. Jamoussi, L. Andersson, R. Callon, R. Danter, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, and A. Malis, *Constraint-Based LSP Setup Using LDP*, RFC3212, January 2002
- [RFC3468] L. Andersoon and G. Swallow, *The Multiprotocol Label Switching (MPLS)*, Working Group Decision on MPLS Signaling Protocols, RFC3468, February 2003
- [RFC3630] D. Katz, K. Komppella and D. Yeung, *Traffic Engineering Extensions to OSPF*, RFC3630, September 2003
- [RFC3784] H. Smit and T. Li, *IS-IS Extensions for Traffic Engineering*, RFC3784, June 2004

2.11 FURTHER READING

- [Awduche Jabbari] D. Awduche and B. Jabbari, *Internet Traffic Engineering Using Multiprotocol Label Switching (MPLS)*, *Journal of Computer Networks* (Elsevier-Science), **40**(1), September 2002

2.12 STUDY QUESTIONS

1. Referring to RFC 3209, what is the protocol encoding decision that restricts the number of LSPs that an LSR can be head end for?

2. Compare the following approaches for keeping traffic within one geographical area: (a) IGP metric manipulation and (b) link coloring.
3. Preemption is applied at LSP setup time, when RSVP signaling is applied. The specifications do not define what the behavior should be when an LSP is preempted. One option is to immediately confiscate the resources and tear down the forwarding state of the preempted LSP and inform the head end that the LSP was preempted. This approach is sometimes referred to as 'hard preemption'. The other option is to inform the head end that the LSP is preempted and needs to be moved, but only tear down the forwarding state at the node where the LSP was preempted after a delay (unless it is removed by the head end before). This approach is sometimes referred to as 'soft preemption'. What would be the rationale for each approach?
4. Setting up LSPs with bandwidth reservations in the control plane does not ensure that resources are reserved in the data plane. Two approaches exist for making sure the data path is not 'overrun': (a) admission control of services into the LSPs and (b) policing. Compare the two approaches.
5. When advertising the LSP into the IGP, its metric can be set in one of the following two ways: (a) it can inherit the metric of the underlying IGP path or (b) it can have a fixed metric. What are some of the advantages/disadvantages of each approach?
6. As discussed in Section 2.6.1, an important question when evaluating a network design is to see whether the equipment can support the number of LSPs that is likely to cross it. Assume a network with 10 POPs, each with a single WAN router, where full-mesh connectivity is required. Compute the maximum number of transit LSPs in two cases: (a) Simple full-mesh topology and (b) five POPs in Europe and five in the US, with trans-continental links running between the DC and London POPs.

3

Protection and Restoration in MPLS Networks

3.1 INTRODUCTION

In the Traffic Engineering chapter (Chapter 2) we have seen how MPLS traffic engineering allows operators to carry traffic with stringent QoS guarantees such as voice and video. However, these applications require high-quality service, not just when the network is in a normal operating condition but also following a failure. Voice and video are referred to as ‘fragile traffic’ because they are real-time in nature and therefore cannot recover from traffic loss using retransmissions. Therefore, protection and restoration mechanisms are necessary to handle the failure case quickly. The ability to provide such fast protection is essential for converging voice, video and data on to a single MPLS network infrastructure.

This chapter deals with protection and restoration in MPLS networks. We will start by discussing the use of bidirectional forwarding detection (BFD) for fast-failure detection. Then we will take a look at path protection and at fast reroute using local protection and will see why MPLS-TE has become a synonym for fast reroute in MPLS networks. Finally, we will look at schemes for protecting MPLS traffic that is not forwarded along a TE path, such as LDP traffic. This chapter assumes familiarity with RSVP and with basic TE concepts.

3.2 THE BUSINESS DRIVERS

Traditionally, providers used IP/MPLS backbones to carry traffic with loose service level agreements (SLAs) and TDM networks for traffic with tight SLAs. Converging all services on to the same core is attractive because it eliminates the need to build and maintain separate physical networks for each service offering and because the flexibility of IP enables new services such as video-telephony integration. However, traffic with tight SLAs such as voice, video or ATM CBR has stringent requirements for availability and traffic loss. Thus, fast recovery following a failure is an essential functionality for multiservice networks.

One way to provide fast recovery following a link failure is to provide protection at Layer 1. This is the solution provided by SONET APS (Automatic Protection Switching). The idea is simple. Maintain a standby link that is ready to take over the traffic from the protected one in case of failure and switch traffic to it as soon as the failure is detected. Because the decision to move to the standby link is a local one, the switchover can happen within 50 ms, making any disruption virtually unnoticeable at the application layer. The quick recovery comes at the cost of maintaining the idle bandwidth and the additional hardware required for the switchover.

The goal of MPLS fast reroute (FRR) is to provide similar guarantees for MPLS tunnels. The advantage of fast reroute over SONET APS is that (a) it is not limited by the link type, (b) it offers protection for node failures and (c) it does not require extra hardware. For a provider contemplating the deployment of a network requiring subsecond recovery (such as voice-over IP) the first question to ask is whether MPLS FRR is the only option.

Exactly how much loss can be tolerated by a particular application is an important consideration when choosing a protection method. Many non real-time applications do not really need 50 ms protection and can tolerate higher loss.¹ Given the more lax requirements of such applications, some service providers may decide to deploy pure IP networks and rely on subsecond IGP convergence (which is now available from many vendors) for the protection. The main differentiator for MPLS FRR in this context is that it can consistently provide a small recovery time because it is a local decision, while IGP convergence, which is the result of a distributed computation, may be affected by factors such as when the last SPF was run, churn in a different part of the network or CPU (central processing unit) load caused by other unrelated operations. Hence, although the average IGP convergence time might be low, the upper bound on the recovery time may be relatively high.

¹ A loss of 300 ms or more will be noticed in the phone conversation; a loss of more than 2 seconds will affect the control traffic and may cause the call to be dropped.

The amount of time during which traffic is lost depends on how fast the failure is detected and how fast the traffic is switched over to an alternate path. Most of this chapter deals with the mechanisms for quickly moving the traffic to an alternate path around the point of failure. However, no matter how efficient these mechanisms are, they are useless if the failure is not detected in a timely manner. Thus, fast failure detection, though not directly related to MPLS, is an important component of MPLS protection and is assumed throughout this chapter. In the next section we will take a look at some of the challenges with fast detection.

3.3 FAILURE DETECTION

The ability to detect that a failure has happened is the first step towards providing recovery and therefore is an essential building block for providing traffic protection. Some transmission media provide hardware indications of connectivity loss. One example is packet-over-SONET/SDH (synchronous digital hierarchy), which is widely used in the network backbones and where a break in the transmission path is detected within milliseconds at the physical layer.

When failure detection is not provided in the hardware, this task can be accomplished by an entity at a higher layer in the network. Let us take a look at the disadvantages of doing so, using IGP hellos as an example. The IGPs send periodic hello packets to ensure connectivity to their neighbors. When the packets stop arriving, a failure is assumed. There are two reasons why hello-based failure detection using IGP hellos cannot provide fast detection times:

1. The architectural limits of IGP hello-based failure detection are 3 seconds for OSPF and 1 second for ISIS. In common configurations, the detection times range from 5 to 40 seconds.
2. Handling IGP hellos is relatively complex, so raising the frequency of the hellos places a considerable burden on the CPU.

The heart of the matter is the lack of a hello protocol to detect the failure at a lower layer. Based on this realization, the BFD protocol was developed. Having rapidly gained acceptance, the BFD protocol has its own working group (with the same name) in the IETF [BFD]. So what exactly is BFD?

BFD is a simple hello protocol designed to do rapid failure detection. Its goal is to provide a low-overhead mechanism that can quickly detect faults in the bidirectional path between two forwarding engines, whether they are due to problems with the physical interfaces, with the forwarding engines themselves or with any other component. The natural question is just how quickly BFD can detect such a fault. The answer is that it depends

on the platform and on how the protocol is implemented. Available early implementations allowed detection times of about 100 ms, newer implementations can provide detections in the range of 10s of milliseconds. While 100 ms detection time is not perfect if recovery times of 50 ms are sought, it is a huge improvement over detection times on the order of seconds and still falls within the requirements of many applications. BFD started out as a simple mechanism intended to be used between adjacent routers,² but has since found numerous other applications. We will see one such application in the context of LSP failure detection in the chapter discussing management of MPLS networks (Chapter 15).

It is beyond the scope of this book to describe the details of the BFD protocol, its packet formats and processing rules, which are explained in detail in the relevant IETF drafts [BFD-BASE] [BFD-MHOP]. From the point of view of MPLS protection and restoration techniques, BFD is simply a tool for solving the fast detection problem. With the knowledge that this tool exists, the problem of fast detection can be considered to be solved for all media types. Therefore, in the rest of the chapter, fast failure detection is assumed.

Let us now turn our attention to the mechanisms available for actually protecting the traffic: end-to-end (path) protection and hop-by-hop (local) protection.

3.4 END-TO-END PROTECTION

The first type of protection discussed is end-to-end protection, also known as path protection. Although not as popular as local protection using fast reroute, it is important to examine it because it highlights some of the issues solved by local protection.

A common practice in network deployments is the use of a primary/backup approach for providing resiliency. Following this model, LSP protection is achieved using two LSPs: the primary, used under normal operation, and the secondary, used if there is a failure on the primary. For example, in Figure 3.1 LSP2 (S–R4–D) provides path protection for LSP1 (S–R1–D). For fastest recovery times, the secondary is presignaled and ready to take over the traffic, in effect being in hot standby mode. When a failure (such as an interface down event) is detected on the primary LSP, an RSVP error is propagated to the LSP head end. Upon receipt of this error message, the head end switches the traffic to the secondary. The problem is that until the error reaches the head end, traffic continues to be sent over

² An alternative approach in the case of Ethernet links is Ethernet OAM. At the time of writing of this book, Ethernet OAM had been deployed less than BFD because it was relatively new, although its prevalence was increasing.

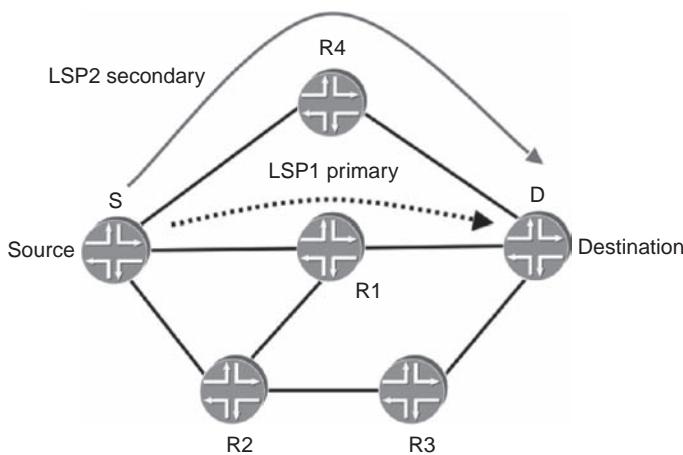


Figure 3.1 Path protection

the primary. If the secondary is not presignedaled, the extra time required to set it up further increases the switchover delay.

From the example, several properties of path protection become apparent.

3.4.1 Control over the traffic flow following a failure

The use of a presignedaled secondary path is very powerful because it provides exact knowledge of where the traffic will flow following the failure. This is important not just for capacity planning but also for ensuring properties such as low delay. Note that the same control can be achieved even if the secondary is not in standby mode, if its path is explicitly configured.

3.4.2 Requirement for path diversity

For the secondary to provide meaningful protection in case of a failure on the primary, it is necessary that a single failure must not affect both the primary and the secondary. Clearly, if both LSPs use a common link in their path, then they will both fail when the link breaks. To avoid this, the primary and the secondary must take different paths through the network. Path diversity is relatively easy to achieve when the LSPs are contained within a single IGP area and many implementations attempt to provide this functionality by default. However, in the chapter discussing Interdomain

TE (Chapter 5), we will see that it is not trivial to ensure for LSPs that cross domain boundaries.³

3.4.3 Double-booking of resources

The secondary LSP is usually set up with the same resource reservations as the primary to ensure the same quality of service when the traffic moves from the primary to the secondary. The net result is that twice as many resources are used throughout the network if the secondary is set up before the failure. This problem could be avoided if the secondary were not presignaled, at the expense of a longer switchover time. Assuming that the secondary is presignaled and therefore reserves resources, an interesting situation can arise when there is a resource shortage in the network: secondary LSPs that effectively carry no traffic may reserve bandwidth while other primary LSPs may fail to establish. To prevent this situation, some providers choose to use LSP priorities and assign better values to all the primary LSPs in the network, to ensure they can always establish.

3.4.4 Unnecessary protection

End-to-end protection protects the entire path. Thus, even if most links in the primary paths are protected using other mechanisms (such as APS), it is not possible to apply protection selectively for just those links that need it.

3.4.5 Nondeterministic switchover delay

The delay in the switchover between the primary and the standby is dictated by the time it takes for the RSVP error message to propagate to the LSP head end. This is a control plane operation and therefore the time it takes is not deterministic. For example, if the CPU is busy processing BGP updates at the time of the failure, there may be a delay in the propagation of the RSVP error. Moreover, unless the secondary is set up in the standby mode, further delay is incurred by RSVP signaling of the secondary path. The discussion above assumes that the error is detected by BFD at the link level and propagated to the head end by RSVP. In the MPLS management chapter (Chapter 15), we will see a different approach to trigger switchover to a secondary LSP. This approach relies on running BFD at the LSP level and avoids the failure propagation delay.

³ Unfortunately, path diversity alone does not guarantee that the primary and secondary will not share the same fate when a resource fails. Fate sharing is discussed in detail later in this chapter.

The main advantage of end-to-end path protection is the control it gives the operator over the fate of the traffic after the failure. Its main disadvantages are double-booking of resources, unnecessary protection for links that do not require it and nondeterministic switchover times. They arise from the fact that the protection is provided by the head end for the entire path. Local protection attempts to fix these problems by providing the protection locally rather than at the head end and by protecting a single resource at a time.

3.5 LOCAL PROTECTION USING FAST REROUTE

The goal of protection is to minimize the time during which traffic is lost. Thus, it makes sense to apply protection as close to the point of failure as possible. The idea of local protection is simple. Instead of providing protection at the head end for the entire path, the traffic around the point of failure is rerouted. This is very similar to what happens when the highway between two cities closes somewhere between exits A and B. Rather than redirecting all the traffic away from the highway altogether, vehicles are directed on to a detour path at exit A and rejoin the highway at exit B or at some other exit down the road from B.

The use of a detour is a very intuitive concept, easily applicable to TE LSPs, as shown in Figure 3.2. An alternate path, called the detour or bypass, is created by R1 (the ‘Point of Local Repair’ or PLR for short) in

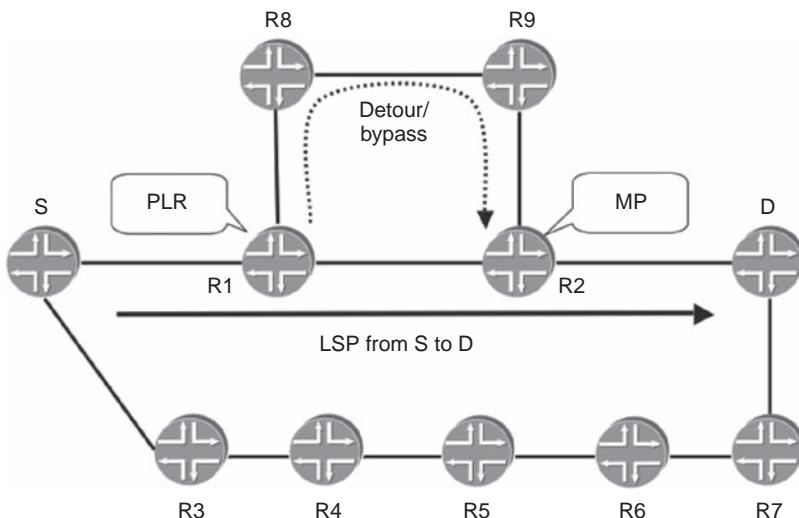


Figure 3.2 Fast reroute using local protection

order to avoid the link R1–R2. In case of a failure, traffic is shuttled around the failed link using this path and rejoins the LSP at R2 (the ‘Merge Point’ or MP for short). Thus, the traffic is quickly rerouted around the point of failure and for this reason this mechanism is called fast reroute. The idea is not to keep the traffic on the detour until the link recovers, but rather to keep it long enough for the LSP head end to move the LSP to a new path that does not use the failed link. There are several attractive properties to fast reroute:

1. A single resource is protected and therefore it is possible to pick and choose which resource to protect.
2. Protection can be applied quickly because it is enforced close to the point of failure.
3. Traffic is forwarded around the failure over the detour/bypass, on a path that has been computed and signaled before the failure happens. This means that as soon as the node immediately upstream of the point of failure detects that a failure has occurred, it can immediately move the traffic onto the protection path without having to signal to other nodes.

If fast reroute is so intuitive and brings so many advantages, why is it not available in IP networks? The answer is because it relies heavily on source routing, where the path is determined at the source and no independent forwarding decisions are made by the individual nodes in the path. Let us see how. For local protection to work, traffic must continue to reach the beginning of the protection path after the failure has occurred. When traffic is forwarded as IP, the forwarding decision is made independently at every hop based on the destination address. In Figure 3.2, link R1–R2 is protected by a protection path along R1–R8–R9–R2. All the link metrics are equal to 1, except link R8–R9, which has a metric of 10. If after the failure node S computes its shortest path as S–R3–R4–R5–R6–R7–D and redirects the traffic towards R3, the packets will not reach the protection path. Furthermore, until router R3 has also performed its path computation, its best path for destination D points back towards S (because the path R3–S–R1–R2–D is shorter than the path R3–R4–R5–R6–R7–D).

Source routing is one of the most powerful properties of TE LSPs. The LSP is set up along a path determined at the head end. Once traffic is placed into the LSP, it is guaranteed to be forwarded all the way to the tail end, regardless of the routing changes that happen in the network. Thus, traffic is always guaranteed to reach the beginning of the protection path. Once it rejoins the LSP at R2, it is guaranteed to reach the tail end.

The mechanisms for providing fast reroute in MPLS networks were developed in the MPLS Working Group in the IETF and are documented in [RFC4090]. Local protection mechanisms are qualified based on two criteria:

1. The type of resource that is protected, either a link or a node. Thus, local protection is either link protection or node protection. As we will see in later sections, this influences the placement of the backup path. Regardless of the protected resource, local protection mechanisms are collectively referred to as local protection or fast reroute (FRR).
2. The number of LSPs protected by the protection tunnel, either 1:1 or N:1. These are called one-to-one protection and facility protection, respectively, and because the protection paths are sometimes referred to as ‘backups’, the terms one-to-one backup and facility backup will also be used in this chapter. The ability to share the protection paths is not an issue of scalability alone. As we will see in later sections, it also determines how traffic is forwarded over the protection path.

From the above, we can see that there are in total four variants of local protection. To our knowledge, at the time of writing, no current implementation supports all four variants. Because one-to-one backup and facility backup have their respective advantages, implementations exist for at least one variant of both these methods.

In order to examine the differences between the four variants, in terms of the number of protection paths created and their placement, let us examine Figure 3.3.

The figure shows three LSPs, all of which pass through R2 and R3. LSP X follows the path R1, R2, R3, R4, R5. LSP Y follows the path R1, R2, R3, R4, R6. LSP Z follows the path R1, R2, R3, R11, R12, R13. For each variant of local protection, we will show the protection path(s) created to protect against failure scenarios in which R2 is the PLR.

3.5.1 Case (i): link protection, for the facility protection case

In the facility protection case, the protection path created by a PLR is known as a bypass tunnel. Figure 3.4 shows the bypass tunnel that is used should the link between R2 and R3 fail.

The bypass tunnel shown in the figure is shared by all three LSPs that use the link between R2 and R3. R2 must set up the bypass tunnel such that the MP is R3, the router immediately downstream of the link failure, that is to say the ‘next-hop’ node of the LSPs normally passing through R2

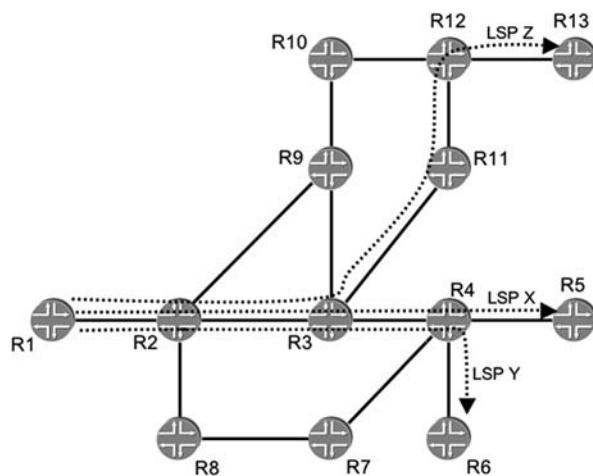


Figure 3.3 Network topology for local protection discussion, before failure event

towards R3. This is because the bypass tunnel is shared between multiple LSPs, and R3 is the only router in the topology that is guaranteed to lie along the main path of all the LSPs being protected. Note that this means that for the period of time that the bypass tunnel is in use, the overall path taken by some of the traffic from the PLR to the egress node of the LSP may not be optimal. For example, traffic using LSP Z follows the path R1,

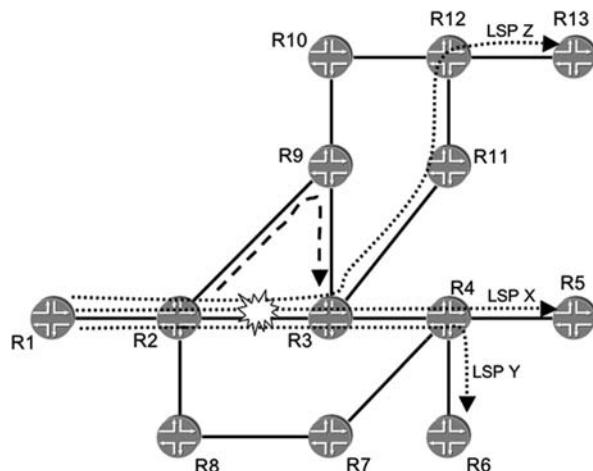


Figure 3.4 Bypasses for link protection, facility protection case

R2, R9, R3, R11, R12, R13 because the nature of this variant of protection means the traffic must pass through R3.

3.5.2 Case (ii): link protection, for the 1:1 protection case

In the 1:1 protection case, the protection paths created by a PLR are called detours. Figure 3.5 shows the detour paths created for the LSPs by R2 for use should the link between R2 and R3 fail.

A separate detour path is created for each LSP that uses the link between R2 and R3. Because each detour path is dedicated to one LSP, it simply needs to follow the shortest path to the egress point of the LSP being protected. There is no need for the detour path to rejoin the main LSP at R3 if that would not give the optimum path to the egress node from the PLR. If the shortest path to the egress node intersects the path of the main LSP, the detour path merges with the main LSP at that point. This can be seen in the figure: the detour for LSP Z follows the path R2–R9–R10–R12 and merges with the main LSP at R12. As can also be seen from the figure, the detour LSPs for LSP X and LSP Y both follow the path R2–R9–R3 and merge with their respective LSPs at R3. Note that although the detour LSPs for LSP X and LSP Y follow exactly the same path, separate detours are nevertheless created.

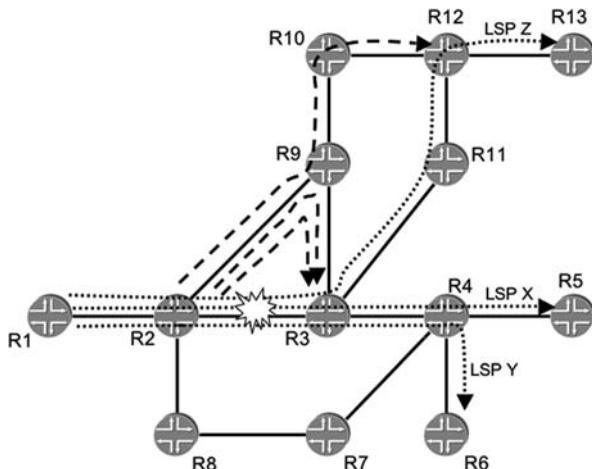


Figure 3.5 Detours for link protection, 1:1 protection case

Figures 3.4 and 3.5 illustrate that case (i) results in fewer protection paths being required than case (ii), at the expense of a potentially suboptimal path being followed in case (i).

3.5.3 Case (iii): node protection, for the facility protection case

Figure 3.6 shows the bypass tunnels created for the LSPs by R2 for use should node R3 fail.

R2 needs to identify all the ‘next next-hop’ nodes, that is the routers two hops away along the LSPs passing through R2, and create a bypass tunnel to each next next-hop node. In general, not all the LSPs have the same next next-hop node in common, but LSPs that do have the same next next-hop node share the same bypass tunnel. In the figure, the next-hop-hop node for both LSP X and LSP Y is R4. Therefore, traffic from both LSPs share the same bypass tunnel, R2–R8–R7–R4, and merge with their respective LSPs at R4. LSP Z has a different next next-hop node, R11, so R2 creates another bypass tunnel that follows the path R2–R9–R10–R12–R11. Note that as with case (i), the overall path followed by traffic using LSP Z from the PLR to the egress node following a failure event is not optimal, as the traffic ‘doubles-back’ at R11.

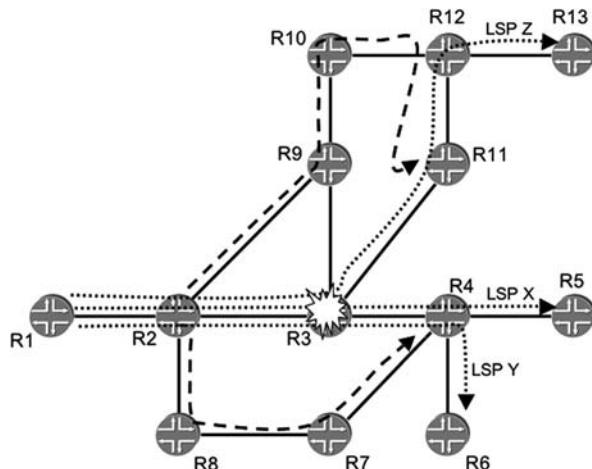


Figure 3.6 Bypasses for node protection, facility protection case

3.5.4 Case (iv): node protection, for the 1:1 protection case

Figure 3.7 shows the detour paths created for the LSPs by R2 for use should node R3 fail.

As in case (ii), a separate detour path is created for each LSP that uses the link between R2 and R3. Because each detour path is dedicated to one LSP, it simply needs to follow the shortest path to the egress point of the LSP being protected. Thus for LSP Z, the detour follows the path R2–R9–R10–R12 and merges with the main LSP at R12. The detour paths for LSP X and LSP Y both follow the path R2, R8, R7, R4 and merge with their respective LSPs at R4. Although the detour LSPs for LSP X and LSP Y follow exactly the same path, separate detours are nevertheless created.

Comparing cases (iii) and (iv), in general for case (iii), each PLR needs to create several bypass tunnels, one for each next next-hop node. In case (iv), a separate detour is required for each LSP being protected – in most cases this will result in more protection paths being created than in case (iii), but with the benefit of having more optimal paths for the protected traffic.

Note that for the 1:1 protection cases, detour LSPs created by different PLRs can merge if they protect the same LSP. This means that for many topologies, the overall number of detour LSPs using each link of the network may not be as high as might be supposed (although typically will still be higher than for the facility protection case, if more than a small

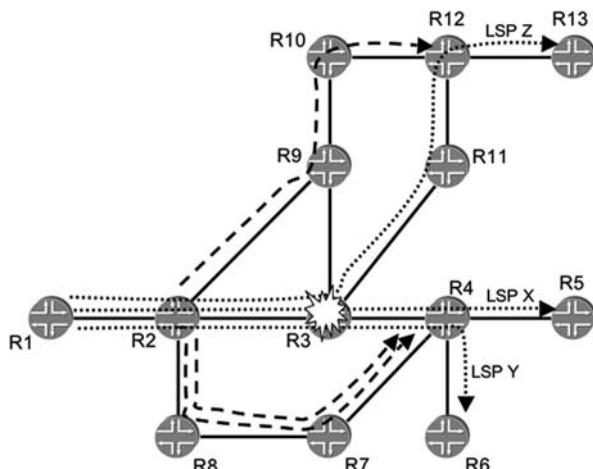


Figure 3.7 Detours for node protection, 1:1 protection case

number of LSPs are present). For example, let us consider the 1:1 node protection case for LSP Z. As discussed before, R2 computes a detour path R2–R9–R10–R12 for LSP Z to be used should node R3 fail. R3 computes a detour path R3–R2–R9–R10–R12 for LSP Z should node R11 fail. Hence, the two detours can merge at R2, so only one detour LSP to protect LSP Z is present along the path R2–R9–R10–R12.

A question often asked by people studying MPLS fast-reroute schemes is whether the bypass or detour path is allowed to pass through the ingress node. The answer is yes – this situation arises in ring topologies. It is interesting to compare the protection paths for the 1:1 and the facility protection cases in a ring topology. In Figure 3.8, suppose we have an LSP that follows the path A, B, C, D, E and the link between B and C breaks.

If the link-protection variant of facility protection is in use, then the bypass tunnel must go from B, the PLR, to the MP which is the next-hop node downstream from the PLR, namely C. Hence, the resulting path followed by the traffic while the bypass is in use is A, B, A, H, G, F, E, D, C, D, E as can be seen in Figure 3.8. In contrast, Figure 3.9 shows the path followed by the traffic for the 1:1 protection case.

Whether the link or node protection variant of 1:1 protection is in use, the resulting path followed by the traffic when the detour is in use is A, B, A, H, G, F, E. This is shorter than the facility protection case because the detour path ends at the egress node of the main LSP. Note that this difference in the path length may not be an issue to all operators as typically the protection path is only used for a short period of time. Once the ingress router, A, is aware of the failure, it computes and signals a new path for the main LSP, which would be A, H, G, F, E.

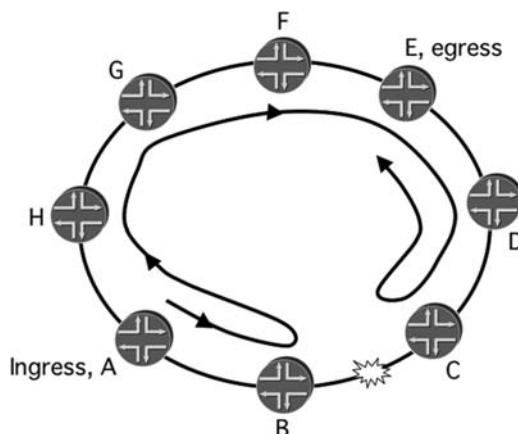


Figure 3.8 Local protection in a ring topology: link protection, facility protection case

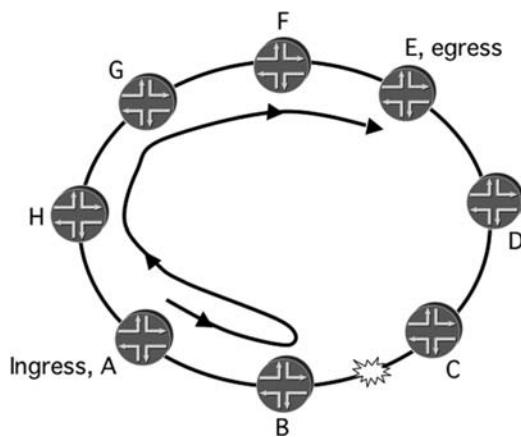


Figure 3.9 Local protection in a ring topology: link or node protection, 1:1 protection case

The following sections examine the mechanisms of fast reroute in more detail, in terms of how the protection paths are computed and signaled and the label operations that occur along the forwarding path.

3.6 LINK PROTECTION

As discussed in the previous section, link protection refers to the ability to protect traffic being forwarded on an LSP when a link along the LSP fails. To protect against the failure of a link, a backup tunnel is set up around the link. This backup is called a detour in the case of one-to-one protection and bypass in the case of many-to-one protection.

Figure 3.10 shows one LSP, LSP_{xy} from X to Y, along the path X–A–B–Z–Y. Link A–B is protected by a backup LSP taking the path A–C–D–B. When link A–B fails, traffic from LSP_{xy} (the protected path) is forwarded on this backup around the broken link at A and delivered to B, from where it continues on its normal path to destination Y. Node A, where traffic is spliced from the protected path on to the backup, is the PLR, and node B, where traffic merges from the backup into the protected path again, is the MP.⁴ Throughout this chapter, we will use the terms ‘protected path’ and ‘main path’ interchangeably to mean the LSP receiving protection.

⁴ As noted earlier in this chapter, in the case of one-to-one protection, the MP does not necessarily need to be the router immediately downstream of the protected link. However, for the discussion in this section, we have chosen a topology such that the MP is the same for both the one-to-one protection and facility protection cases, so that the label operations can be compared more easily for the two cases.

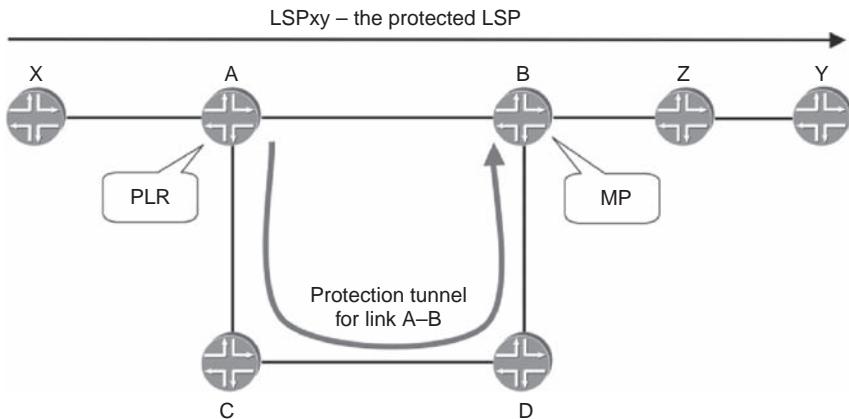


Figure 3.10 Topology for detailed link protection discussion

Let us take a look at the different actions that need to happen before and after the failure.

3.6.1 What happens before the failure

To ensure fast protection, the backup must be ready to forward traffic when the failure happens. This means that:

- The backup path must be computed and signaled before the failure happens and the forwarding state must be set up for it, at the PLR, MP and all the transit nodes.
- The forwarding state must be in place at the head end of the backup tunnel (the PLR) and at its tail end (the MP) so that traffic can be forwarded into the backup at the PLR and back on to the main LSP at the MP.

Let us examine each of these separately below.

3.6.1.1 Path computation

What triggers the computation and setup of the backup path? To answer this question, let us first examine what information will cause LSR A to set up a backup path. First, LSR A needs to know that it is required to protect link A-B. Second, it must know that it is required to protect traffic flowing on LSP_{xy}. Remember that one of the advantages of local protection is that the operator can pick and choose which resources and which LSPs to protect. For example, the operator may protect an LSP carrying voice

traffic but not one carrying data. Similarly, the operator may decide not to protect a link that is already protected using APS. Thus, the operator must specify in the configuration which LSPs and which links to protect. For the link, the configuration is on the router containing the link, router A. For the LSP, the configuration is at the head end X, and therefore the information must be propagated in the RSVP Path messages for the LSP. This is done using either the 'local protection desired' flag in the Session Attribute Object or the Fast Reroute Object.⁵

Once this information is available, node A computes a protection path for link A–B by running a CSPF computation to destination B, with the obvious constraint to avoid the link A–B. The head end can signal other constraints to be applied to the backup path computation, such as the maximum number of hops that the backup is allowed to cross, the required bandwidth for the backup or its setup and hold priorities. The purpose of these constraints is to ensure that even when using the protection path, traffic continues to receive certain guarantees. These constraints are signaled from the head end to the PLR in the Fast Reroute Object. In addition, for one-to-one backup, where the backup path protects a single LSP, some of the properties of the backup path, such as bandwidth and link colors, are inherited from the protected LSP and do not require explicit signaling.

Once the backup path is computed, it is set up using RSVP. How is traffic forwarded on to it? To answer this question, let us take a look at the forwarding state that is installed.

3.6.1.2 *Forwarding state installation*

The goal of the backup is to carry the traffic from the protected (main) path around the failed link and merge it back into the mainpath at the MP located at the other end of the failed link. Two different techniques exist for directing traffic from the backup into the mainpath, which differ in the label with which the traffic arrives at the MP. This in turn influences the number of LSPs that can be protected by a single backup tunnel, yielding either $N:1$ (facility backup) or $1:1$ (one-to-one backup).

Facility backup

Traffic arrives over the backup tunnel with the same label as it would if it arrived over the failed link.⁶ The only difference from the point

⁵ However, it is recommended that the bit (desired) in the Session Attribute should always be set if local protection is desired. Additionally, of course, the Fast Reroute Object can also be signaled.

⁶ The use of a different label at the MP is not precluded in the specification. In practice, this scheme is not implemented.

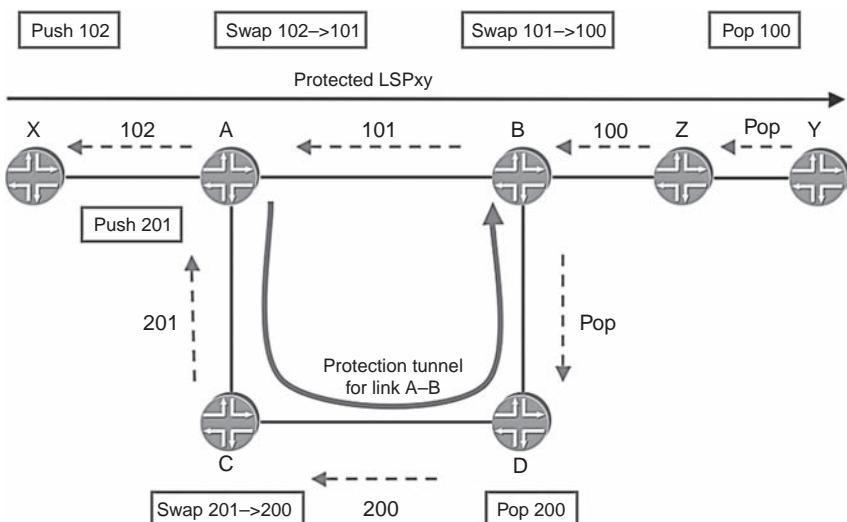


Figure 3.11 Setting up a bypass in the link protection, facility protection case

of view of forwarding is that traffic arrives at the MP over different interfaces when arriving over the protected path and over the backup. To ensure that traffic arrives at the MP with the correct label, all that needs to be done is to tunnel it into the backup by pushing the backup tunnel label on top of the protected LSP label at the PLR (label stacking) and do penultimate hop-popping for the backup tunnel label before the MP (label stacking and penultimate hop-popping are explained in the Foundations chapter, Chapter 1). Note that using this scheme, the depth of the label stack increases when traffic is forwarded over the backup tunnel.

Figure 3.11 shows the setup of the backup tunnel before the failure and the forwarding state that is installed at every hop in the path.

Figure 3.12 shows traffic forwarding after a failure. In the figure, the payload happens to be an IP packet, but this is just by way of example; the packet could be anything that can be carried in an LSP. Let us take a look at some of the key properties of facility backup:

1. No new forwarding state is installed at the MP. At the PLR, the forwarding state must be set in place to push the label of the backup path (label 201 in the example) on to the labeled traffic from the protected LSP in the event of a failure.
2. Any number of LSPs crossing link A–B can be protected by the backup shown in the figure. There is no extra forwarding state for each

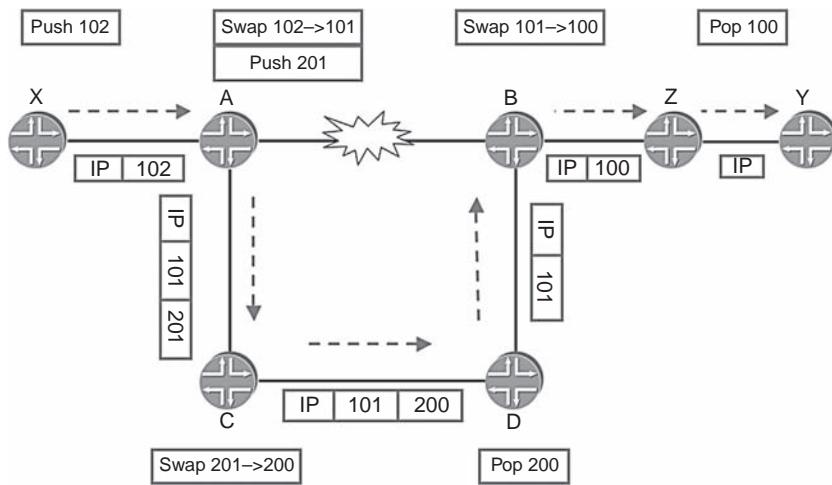


Figure 3.12 Forwarding traffic using the facility backup in the link protection case

LSP protected either at the MP or at any of the routers in the path (e.g. routers C and D) and the action taken by the PLR is always the same: push the backup tunnel label on to the label stack of the main LSP. The ability for several LSPs to share the same protection path is an important scaling property of facility backup.

3. The label that is advertised by the MP is an implicit null label and therefore penultimate hop popping is performed for the backup tunnel. Thus, traffic arrives at the MP with the same label with which it would have arrived over the main LSP.

One-to-one backup

Traffic arrives at the MP with a different label than the one used by the main (protected) LSP. Figure 3.13 shows the setup of a one-to-one backup for the LSP from the previous example and Figure 3.14 shows forwarding over the backup following a failure. Traffic arrives at the MP with label 300, the backup tunnel label and is forwarded using label 100, the protected LSP label. Thus, the MP must maintain the forwarding state that associates the backup tunnel label with the correct label of the protected LSP. If a second LSP were to be protected in this figure, a separate backup tunnel would be required for it, and a separate forwarding state would be installed at the MP.

Similar to facility backup, the forwarding state must be set up to map traffic from the protected LSP into the backup. For example, traffic arriving with label 102, the label of the protected LSP, is forwarded over the backup

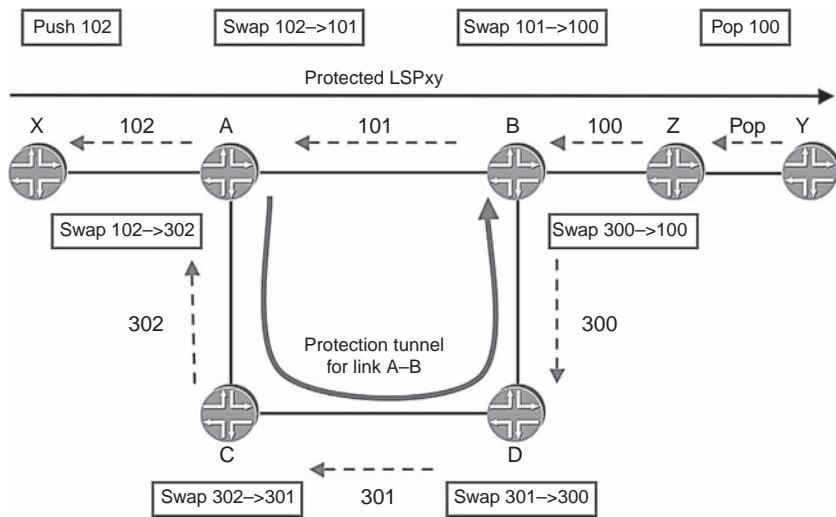


Figure 3.13 Setting up a detour for one-to-one backup

using label 302, the backup tunnel label. Note that, using this approach, the depth of the label stack does not increase when packets are forwarded over the backup path, because the top label is simply swapped to the backup tunnel label.

To summarize, the use of one-to-one backup requires installing new forwarding state at both the MP and the PLR. Because the backup protects a single main LSP, the amount of state that the MP, the PLR and all the nodes in the backup path must maintain increases proportionally to the

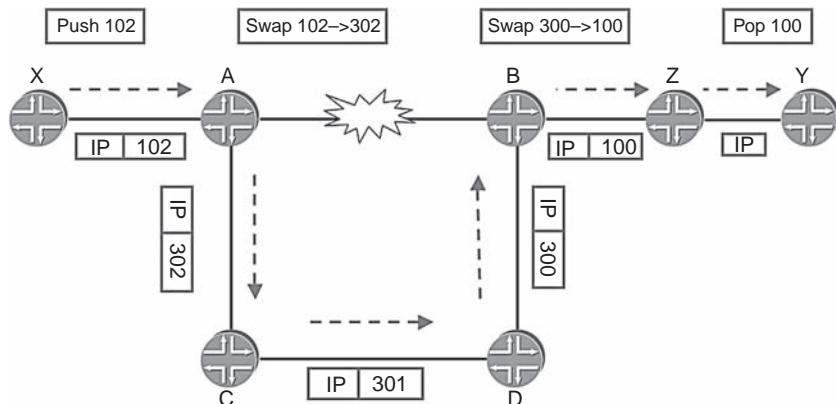


Figure 3.14 Forwarding traffic over a one-to-one backup

number of LSPs protected. Using separate backup tunnels means that the depth of the label stack need not increase when traffic is forwarded from the main LSP to the backup.⁷ Furthermore, the fact that each backup services a single LSP allows tighter control over the backup tunnel and its properties. This is difficult with facility backup where multiple LSPs share the same backup.

Regardless of the type of protection used, the time it takes to update the state in the forwarding engine following the failure detection adds to the total protection time. This is an important, though often overlooked, component in the total protection time. To achieve fast switchover, many implementations install the forwarding state for the protection path ahead of time.

With the backup tunnel computed and signaled and the forwarding state in place to shift traffic from the main tunnel to the backup, let us now take a look at what happens after the failure of the protected link.

3.6.2 What happens after the failure

Once the failure of the protected link is detected, traffic is switched from the main LSP to the backup path. Assuming that the forwarding state was preinstalled, this operation can be done in the forwarding engine, without the intervention of the router's control plane and therefore the traffic flow is quickly restored. Would this be the end of the discussion? The answer is not quite. Following the failure, more action needs to be taken in the control plane.

3.6.2.1 Suppression of LSP teardown

Even if the protected LSP head end or tail end receives IGP notifications about the failure of the link, it must suppress any error generation that would lead to the teardown of the LSP when local protection is available. Otherwise, the purpose of local protection is defeated.

3.6.2.2 Notification of the LSP head end

Remember that the purpose of the backup is to protect traffic while the LSP head end looks for an alternate path for the LSP, avoiding the failed link. For this to happen, the head end must first find out about the failure. The PLR takes care of this by notifying the head using an RSVP Path Error message with a 'Notify' error code and 'Tunnel locally repaired' subcode.

⁷ Increasing the depth of the label stack was a problem in early implementations of MPLS.

In addition to this, a new flag indicating that the path is locally repaired is turned on in the Record Route Object. However, why bother with any of this when the head end will find out about the failure through the IGP anyway? Because relying on a different protocol for the failure notification will not always work. For example, when the LSP spans several IGP areas or ASs, the IGP notifications will not reach the head end.

3.6.2.3 *New path computation and signaling*

When the head end finds out about the switch to the backup path, it recomputes the LSP, avoiding the failed link, and sets it up in make-before-break fashion (make-before-break was discussed in the Traffic Engineering chapter, Chapter 2). This means that LSPs for which local protection is desired are always signaled as ‘shared explicit’, allowing the new path to share resources with the old path. It is possible for the new path to establish over the very same links used during protection. This is the case, for example, in Figure 3.11, after failure of the link A–B. Although traffic will not move to a different path, the new LSP will still be set up along this path. When the last LSP using the protection path has moved, the bypass is torn down.

RSVP message processing

The receipt of a ‘Tunnel locally repaired’ notification informs the head end that traffic is forwarded over a potentially suboptimal protection path. As a result, the head end attempts to re-reroute the LSP. What happens if the head end cannot find an alternate path? This can happen, for example, if the LSP is configured for an explicit path that does not allow it to move away from the failed link. The decision whether to tear down the LSP or let it stay on the protection path is one of local policy/implementation at the head end. Assuming that the policy decision allows the LSP to continue to use the protection path, the next question becomes: can the LSP stay on the protection path forever? In principle, yes, but in practice more details must be taken care of. Remember that RSVP requires periodic refreshes of its state, using Path and Resv messages. Unless these messages continue to be correctly generated and processed, the LSP will time-out. Thus, it is required to forward these messages over the backup tunnel after link failure.

To summarize, local protection is achieved through a combination of actions both before and after the failure. Before the failure, the protection tunnel must be set up and the forwarding state must be installed to switch the traffic from the main tunnel over the protection tunnel around the failed link. After the failure, actions must be taken to prevent the teardown of the

main tunnel until it is rerouted. The basic mechanisms of local protection were described in the context of link protection, but they apply equally to node protection, with a few modifications, as discussed in the next section.

3.7 NODE PROTECTION

Link failures are the most common type of failure in a network. A link failure may happen because of a problem with the link itself or it may be caused by a failure of the node at the other end of the link. In the latter case, the link protection mechanisms described in the previous section will not work if they rely on the adjacent node to act as the MP. Node protection covers this case by setting up the backup tunnel around the protected node to the next next-hop in the path, in the case of facility protection, or towards the egress point of the main LSP in the case of 1:1 protection. Note that in the case of 1:1 protection, the procedure to set up state and the resultant label operations are virtually the same for the node protection case as the link protection case discussed in the previous section. For this reason, the emphasis in this section is on the procedures and label operations for the facility protection case.

Figure 3.15 shows LSP_{xy} from X to Y, along the path X-A-B-Z-Y. LSP_{xy} is protected against node B's failure by a backup tunnel taking the path A-C-D-Z that merges back into LSP_{xy} at node Z downstream from

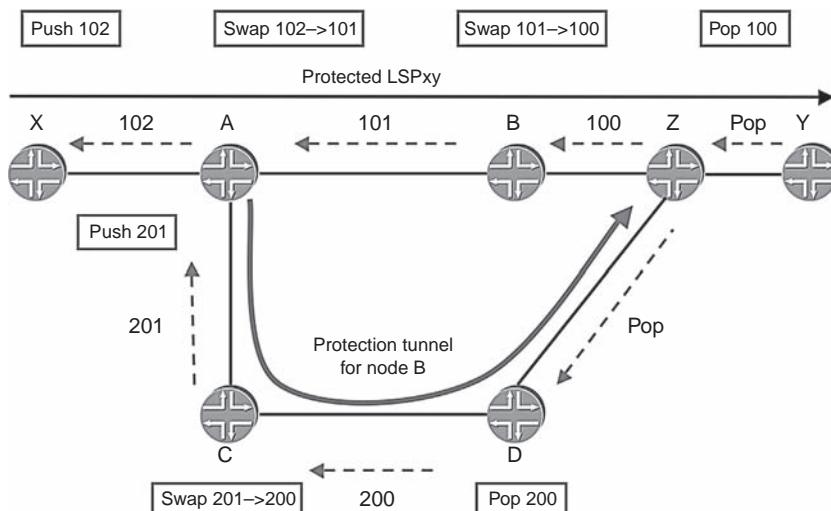


Figure 3.15 Setting up a protection tunnel around node B

node B. When node B fails, traffic from LSP_{xy} (the protected path) is placed on this backup at A and delivered to Z, where it continues on its normal path to destination Y. The figure shows the setting up of the state for the facility protection case.

Looking at this description and at the figure, it becomes clear that A must obtain two pieces of information to set up the backup tunnel:

1. The address of node Z, the tail end of the backup tunnel. This information can be obtained from the Record Route Object (RRO). This address is used as a loose hop for reaching the MP. It can be a router ID or an interface address belonging to the MP.
2. In the case of facility backup, the label used by the main LSP at node Z. Recall that when using the facility backup, traffic arrives to the MP with the same label as that used by the main LSP. Thus, A must be able to swap the incoming label 102 to the label 100, expected by node Z rather than the label 101, which is the one used in normal forwarding along the main LSP. How can A obtain this information? The answer is to use a similar approach as for the discovery of the downstream node and rely on the information in the RRO. However, the label is normally not recorded in the RRO. To solve this problem, the new flag 'label recording desired' is defined for use in the Session Attribute Object. Setting this flag indicates that the label information should be included in the RRO. As a result, labels are recorded in the RRO and becomes available to the PLR.

Given this information, the backup tunnel can be established. Figure 3.16 shows forwarding of traffic over the backup tunnel, assuming facility backup. Note that at node A traffic is already labeled with the label expected by Z before the tunnel label is pushed on to it. The rest of the mechanisms for providing node protection are very similar to link protection and are not repeated here. Instead, let us sum up the differences between node protection and link protection.

1. Node protection protects against both link and node failures.
2. The MP for the backup path is a router downstream from the protected node. The MP is the next next-hop node in the facility protection case.
3. When protecting a node using facility protection, label recording is required because the PLR must know the label that the MP expects the traffic to arrive with.

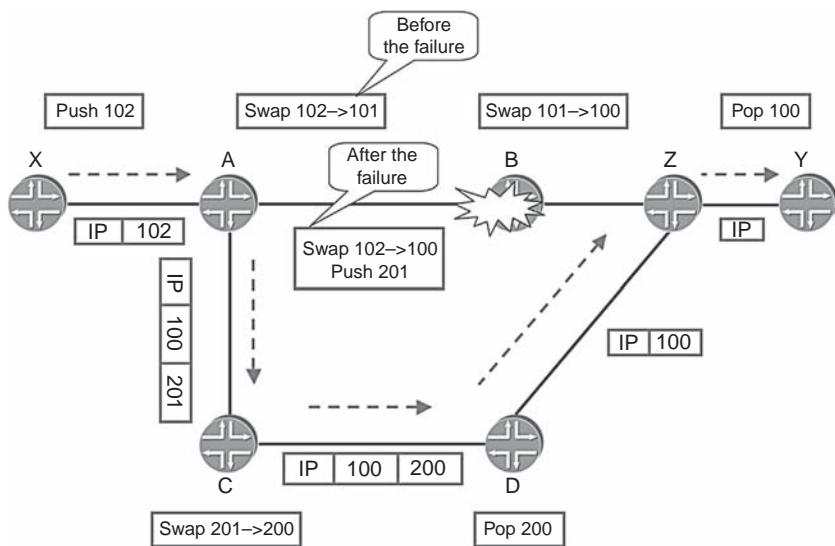


Figure 3.16 Node protection when forwarding traffic, in the facility protection case

3.8 ADDITIONAL CONSTRAINTS FOR THE COMPUTATION OF THE PROTECTION PATH

Until now, when discussing computation of the protection path we have focused on a single constraint: avoid using the protected resource in the protection path. However, is this enough to ensure adequate protection? The answer may be ‘no’, on a case-by-case basis, as we will see in this section.

3.8.1 Fate sharing

Using different links or nodes in the protected and backup paths seems like a good way to ensure resiliency in the case of a failure. However, underlying resources used in the two paths may be affected by a single event. For example, if the optical fiber for two links follows the same physical path, a single event (such as a bulldozer cutting through the fiber) affects them both. In this case, the two links are said to be in the same Shared Risk Link Group (SRLG) or fate-sharing group. Therefore, the protection path must avoid all links that are in the same SRLG with links in the main path (the protected path). This is applicable for both

end-to-end path protection (discussed at the beginning of this chapter) and local protection. For example, for the path protection scenario in Figure 3.1, if links S–R1 and S–R4 belonged to the same SRLG, the secondary path would have been computed along the (less optimal) path S–R2–R3–D.

From the point of view of the CSPF computation, SRLG membership is just another constraint and it is conceptually very similar to link colors (discussed in the Traffic Engineering chapter, Chapter 2). There are two main differences between SRLG and link colors:

1. SRLG is a dynamic constraint, while link colors are static. Although SRLG membership is static, the SRLG is a dynamic constraint that changes according to the links used by the main LSP (the protected one). If the main path uses links from group x, the constraint is to avoid links from group x, but as soon as the main path reroutes and starts using links from group y, the constraint becomes to avoid links from group y. This is in contrast to link colors, where the constraint is a static one, easily expressed as a rule such as ‘exclude red links’, independently of the path taken by the main LSP.
2. Links from the same SRLG need not be completely excluded from the computation. CSPF does not need to exclude links that share a fate from the computation. Doing so might cause the alternate path not to establish at all if the topology is not rich enough to provide an alternate path. Instead, the cost of these links may be raised, making them less preferred in the computation. However, if no other path exists, the links can still be used, providing protection for those failures that do not affect both links. For example, in the link protection scenario in Figure 3.11 assume that links A–B and C–D are in the same risk group. Ideally, the protection path for link A–B should not cross link C–D. However, no other path exists in the given topology. In this case, it is preferable to set up the protection path through C–D anyway than not to set up any protection path at all.

So far we have seen how the SRLG information is used in the CSPF computation. How does a router find out about the SRLG membership? From an implementation point of view, there are two options on how this knowledge can be provided to the routers in the network. In both cases, the assumption is that the network operator knows which resources share what risks and can build a database of SRLG membership. The first option for distributing this information to the routers is to configure the SRLG information on the routers. The second option is to use the IGP to distribute SRLG information similarly to how other TE information is carried in the IGP. (Such extensions were defined for OSPF and ISIS in the context of Generalized Multi-Protocol Label Switching, or GMPLS, extensions for

these protocols [RFC4203] and [RFC5307].) Because this information is fairly static, carrying it in the IGP does not place a large burden on the protocol. Note that regardless of whether the SRLG information is statically configured or dynamically exchanged, the router must be configured with some measure of SRLG information: in the first case, regarding all the links in the network, in the second case, regarding membership of its own links in different SRLG groups. Both approaches are valid and the choice depends on the amount of information that must be configured and on the capabilities of the equipment used.

To summarize, taking into account fate-sharing information ensures that a single event has the least chance of impacting both the protected LSP and its protection tunnel. Fate sharing applies to both path protection and link/node protection and is implemented by computing the protection tunnel with additional constraints. The constraints remove from the computation resources that are considered to be in the same ‘risk group’ with those used in the protected path. Thus, the protection path is guaranteed to be able to take over traffic following a failure in the protected path. However, what if there is not enough available bandwidth on the protection path?

3.8.2 Bandwidth protection

Bandwidth protection refers to the ability to guarantee that enough bandwidth is available on the protection path. For path protection, this is achieved by setting up the secondary with the same bandwidth requirements as the primary. The consequences of doing bandwidth protection end-to-end for the entire path were discussed in Section 3.4 and are not repeated here. Instead, this section focuses on bandwidth protection in the case of local protection.

The very fact that bandwidth protection is discussed separately from the local protection mechanisms may seem counterintuitive. Without bandwidth protection, can one really ensure that no traffic is lost when switching to the backup path? And why go through all the effort of setting up a backup and switching traffic to it if packets will be lost anyway? Should bandwidth protection be an integral part of the local protection mechanisms? The answer is ‘not necessarily’, as we will see below.

Local protection is a short-term solution for avoiding loss by shuttling the traffic around the failed resource until the LSP reroutes to a new path. Remember that traffic is expected to stay on the protection path for a few seconds only, until the reroute happens at the head end. Even if there is not enough bandwidth on the protection path, some of the traffic will still make it through and loss will happen only for a short amount of time. This

might be acceptable for some types of traffic and is an improvement when compared to total traffic loss for the same amount of time. Furthermore, links are typically not used to full capacity and some bandwidth is always available on the protection path. For example, many networks use a 50% upgrade rule, meaning that links are upgraded to higher capacity as soon as they are half full, to account for the possibility of a failure shifting traffic from a different link. For these reasons, bandwidth protection is not mandated in the local protection specification.

Bandwidth protection is an optional functionality that the head end can request for the LSP. It is signaled in the same way as link protection is signaled, by using a new flag in the Session Attribute and Fast Reroute Objects, the 'bandwidth protection' flag. The flag informs the PLR that bandwidth protection is desired. The ability of the PLR to provide the requested bandwidth protection is signaled in the same way as its ability to provide local protection, by using a flag in the Record Route Object, the 'bandwidth protection available' flag. Based on the 'local protection in-use' and the 'bandwidth protection available' flags in the Record Route Object, the head end can determine whether the LSP receives the required guarantees when it switches to the protection path. If these guarantees are not met, the head end can take action to move traffic away from this LSP at the routing level, e.g. by increasing the metric of the LSP and thus making it less desirable for use in routing.

How much bandwidth is required to protect the LSP and how should the PLR act based on this information? For one-to-one backup, the amount of bandwidth required for the backup is the same as for the protected LSP.⁸ Because of the 1:1 mapping between backup paths and protected LSPs, the PLR can set up the backup path with the desired bandwidth as soon as it realizes bandwidth protection is requested.

If the same approach were used for facility backup, then separate backup paths would be built for each of the LSPs requiring protection and there would be no sharing of the backup path between several LSPs, defeating the nice scaling properties of facility backup. Many of the existing implementations deal with this problem by reversing the trigger for the setup of the backup LSP. Rather than setting up the backup based on the bandwidth of the protected LSPs, the backup is set up with a predefined bandwidth and admission control is performed for the protected LSPs into the backup based on their bandwidth requirements. For example, assume that the backup path for link A–B is set up with a bandwidth reservation of 50 Mbps and that two LSPs, LSP1 and LSP2, cross this link. If they each require bandwidth protection for 30 Mbps, bandwidth protection is provided to only one of them.

⁸The exact bandwidth is actually signaled and can be different from that of the protected LSP.

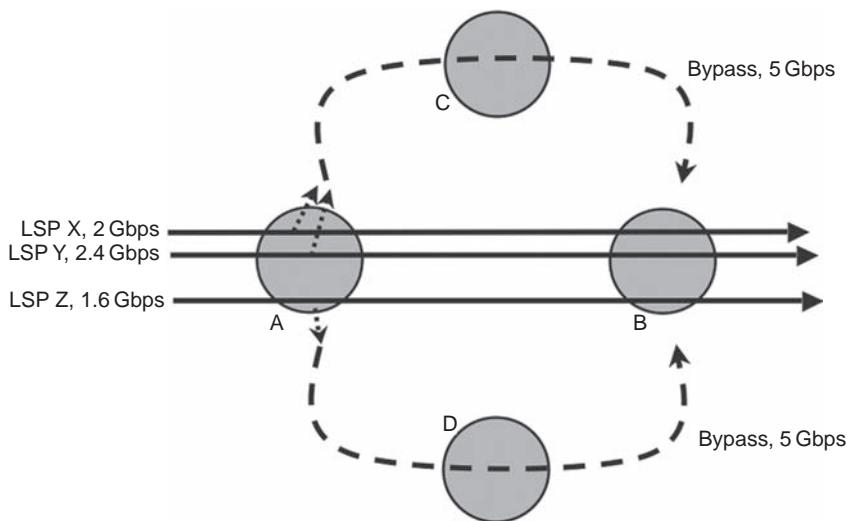


Figure 3.17 Facility protection using multiple bypass tunnels to spread the traffic load

The approach of setting up the backup LSP with a bandwidth reservation and performing admission control into it is very attractive for two reasons:

1. It is easy to estimate the bandwidth that must be reserved for the backup path. For example, a safe guess for a link-protection case is to use the same value as the bandwidth of the protected link⁹ because the total bandwidth of all LSPs crossing the link cannot exceed this value.
2. It is possible to achieve bandwidth protection by setting up several backup paths. When none of the paths computed satisfies the bandwidth requirements, several distinct protection paths can be set up and traffic from different protected LSPs can be spread among these paths. Traffic from a single protected LSP cannot be split or spread over different protection paths because doing so might cause packets to arrive out of order at the LSP tail end. This is illustrated in Figure 3.17.

Let us suppose the link between A and B has a bandwidth of 10 Gbps. Let us suppose that the maximum acceptable bandwidth reservation of a bypass tunnel is 5 Gbps, and there are three LSPs, X, Y and Z having bandwidth reservations of 2, 2.4 and 1.6 Gbps, respectively. Router A could set up one bypass tunnel via router C to accommodate LSP X and Y and another bypass tunnel via D to accommodate LSP Z. Some vendors offer

⁹ Assuming that all of the bandwidth is available for RSVP reservations.

implementations that automate the entire process by (a) creating sufficient bypass tunnels to accommodate the LSPs for which bandwidth protection is required, (b) optimizing the ‘packing’ of LSPs onto the bypass tunnels and (c) removing excess bypass tunnels when no longer required.

To summarize, bandwidth protection guarantees that enough resources are available to the traffic when it switches to the protection path, ensuring that QoS is maintained following a link or node failure. An interesting question arises with regards to the amount of bandwidth that must be reserved for the protection path, especially for facility backup, where several LSPs share the same backup. Regardless of how the bandwidth for the backup path is determined, the cost of bandwidth protection is the idle resources that are used only in case of failure.¹⁰ Just how much bandwidth is kept idle depends on the optimality of the computation and will be discussed further in Section 3.10.3. Although bandwidth protection is expensive, not having it can impact the traffic, not just of the protected LSP but also of other LSPs as well, as we will see in the following section.

3.8.3 Bandwidth protection and DiffServ

Bandwidth protection is expensive. However, not using bandwidth protection can be very destructive in certain environments. Recall from the Traffic Engineering chapter (Chapter 2) that bandwidth reservations are done in the control plane only and no bandwidth is ‘set aside’ in the forwarding plane. If more traffic is forwarded than the reservation, there will be traffic loss.

In Figure 3.18, all links are 100 Mbps. Two LSPs are set up, each with a bandwidth reservation of 60 Mbps: LSP1 from A to E along the path A–B–E and LSP2 from C to B along the path C–D–B. Link protection for link A–B is provided by the backup path A–C–D–B. LSP1 did not request bandwidth protection because it can tolerate some loss during the failure. Assume that both LSP1 and LSP2 carry 60 Mbps of traffic each. When traffic from LSP1 switches to the protection path, 120 Mbps of traffic are sent over link C–D, causing congestion and loss. The traffic that is dropped may belong to either LSP1 or LSP2, causing the very undesirable situation where LSP2 is affected by a failure in LSP1. What is needed is a way to mark the packets of LSP1 as more desirable for dropping.

One way to accomplish this is to give a different DiffServ marking to the traffic as it switches to the backup. This can be easily accomplished by manipulating the EXP bits on the label that is used by A when switching traffic to the protection path. The value of the EXP bits would be such that

¹⁰ Unless the idle resources are used by best-effort traffic at the forwarding time.

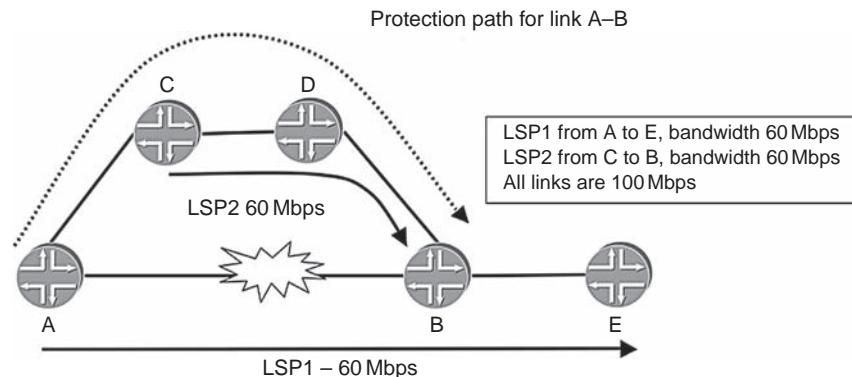


Figure 3.18 Not using bandwidth protection may be destructive

packets have a higher drop preference and during congestion only LSP1's packets are dropped.

3.9 INTERACTION OF END-TO-END PROTECTION AND FAST REROUTE

So far this chapter has focused on local protection mechanisms using fast reroute. Fast reroute has attractive properties in terms of the speed of recovery, deterministic switchover delay and the ability to protect selected resources in the network. In contrast, using path protection to provide recovery after failure cannot offer similar guarantees, as discussed in Section 3.4.

It would seem therefore that fast reroute makes end-to-end path protection unnecessary. Why compute secondary paths when fast reroute can provide the desired level of protection? If so, are path protection and local protection mutually exclusive? The answer is 'no'; they are complementary.

Path protection allows the operator exact control over the path of the traffic after the failure. Fast reroute has the ability to limit the loss to a few milliseconds. The two can be combined by configuring LSPs with both secondary paths and local protection. Because local protection forwards the traffic around the point of failure, the main LSP can switch over to the secondary path slowly. Furthermore, there is no need to presignal the secondary path and reserve resources for it ahead of time, because there is time to set it up after the failure has happened. On the other hand, the use of a secondary path allows tight control over the traffic patterns

following a failure. The secondary can be computed offline taking into account different failure scenarios.

Having seen the different mechanisms for providing protection, let us take a look at some of the deployment considerations.

3.10 DEPLOYMENT CONSIDERATIONS FOR LOCAL PROTECTION MECHANISMS

Service providers are attracted to MPLS FRR by the promise of fast and predictable recovery times. However, the challenges of deploying it, its cost and the guarantees it can deliver in a real deployment are not always well understood. This makes it difficult to make the correct tradeoff between costs and benefits. In the following sections we discuss some of the deployment issues for MPLS FRR.

3.10.1 Scalability considerations

Local protection comes at the cost of setting up extra LSPs in the network. In the Traffic Engineering chapter (Chapter 2) we saw that the number of LSPs is a scalability concern, both because the routers themselves have limits on the number of LSPs they support and because a large number of LSPs in the network is cumbersome to manage. In the context of protection, another scaling dimension is the amount of extra forwarding state created along the protection path. Let us take a look at several of the scalability aspects of local protection.

3.10.1.1 *Extra configuration work*

One of the prevailing misconceptions is that local protection is operationally very difficult to deploy and manage. This belief has its origins in some of the early implementations of local protection which required listing the protection paths in the configuration. Apart from being labor intensive, this approach also required that the computation of the path be done offline, either manually or by using a specialized tool. Furthermore, the use of preconfigured protection paths can work well only if the protected LSPs were also preconfigured to ensure that they actually do use the protected resources. (This is especially true in the case of node protection, where the MP is determined based on the current path of the LSP.) For networks relying on dynamic computation for the primary paths, the requirement to specify the protection in the configuration was therefore unacceptable.

For this reason, many of the implementations available today can compute and set up protection paths dynamically. From a network operations point of view, this can be an important factor when choosing one implementation over another. Ultimately, the complexity of deploying such a solution impacts the number of resources that can be protected in the network.

3.10.1.2 Number of LSPs created

Regardless of whether the protection paths are preconfigured or dynamically computed, they do contribute to the overall number of LSPs that are created and maintained in the network. Just how many extra LSPs are created depends on the type of protection implemented and can be an important consideration when choosing a protection method.

The easiest to analyze is the 1:1 protection. Clearly, for each LSP traversing a protected resource, a new LSP is created and a state is maintained for it. The total number of new LSPs is a function of the number of existing LSPs and the average number of protected resources for each LSP. In principle, the protected resources should be a function of the number of hops crossed by the LSP. However, remember that one of the advantages of local protection is that it can be applied selectively, so fewer resources may be protected. Deployments of 1:1 protection show an increase of a factor of 1.5 or 2 in the number of LSPs after deploying local protection. The relatively low numbers (in the context of 1:1 protection) may be attributed both to selective application of the protection and to the fact that LSPs do not cross many hops.

What about N:1 protection? The whole idea of N:1 protection is to allow sharing of the protection path. It is very tempting to think that when sharing is allowed the amount of new state created becomes solely a function of the number of resources protected. This is true for link protection, where the MP of the protection LSP is unambiguously identified as the endpoint of the link. However, for node protection, the MP depends on the path of the LSP crossing the node. Several LSPs crossing the same node may ‘fan out’ past the protected node and require separate protection paths, as shown later in Figure 3.21, where LSP1 and LSP2 fan out past node B. Therefore, the backup paths for LSP1 and LSP2 for node protection of node B have different MPs. Thus, for node protection, both the network topology and the LSP placement influence the number of protection paths.

The discussion so far made the assumption that protection paths are implemented as a single LSP. This is not always the case. Recall from Section 3.8.2 that when bandwidth protection is required, several LSPs may need to be set up to satisfy the bandwidth requirements. This may yield a higher number of LSPs than the analysis above implies.

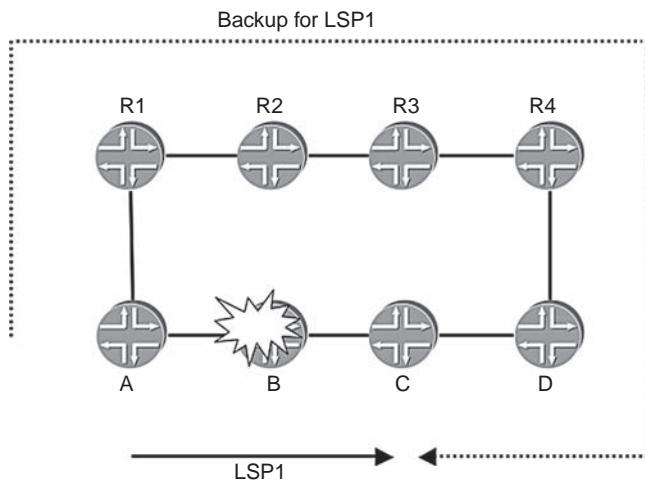


Figure 3.19 Some network topologies yield very long protection paths

3.10.1.3 Increase in the forwarding state

An important although often overlooked scalability consideration is the amount of extra forwarding state that must be maintained following the deployment of local protection. No equipment has unlimited resources therefore understanding the impact of local protection on the forwarding resources used is essential.¹¹ If analysis shows that the limits are reached, then alternative solutions such as adding extra routers, protecting less resources or lowering the number of LSPs can be evaluated.

There are several factors impacting the amount of forwarding state:

- (a) *Network topology*. The protection requires forwarding resources on all the routers in its path. Some network topologies (such as string-of-pearls or dual plane) yield extremely long protection paths, as shown in Figure 3.19. Therefore, the extra forwarding state consumes resources on a large number of routers.
- (b) *Protection type*. Intuitively it is easy to understand that the type of protection (one-to-one or facility) increases the amount of forwarding state in the routers in the protection path because of the different number of paths created.
- (c) *Make-before-break*. Remember that the new LSP is created using make-before-break. Thus, while the new LSP is established, twice as many forwarding resources are being used (one for the old path and one for

¹¹ This is especially true when forwarding is done in hardware.

the new path). If the new path triggers the creation of new protection paths, the forwarding state may triple (one for the old path, one for the new path and one for the protection path). It is true that the old path is removed very shortly afterwards, but make-before-break does temporarily increase the utilization of forwarding resources. What this means is that there must be enough free forwarding resources to accommodate the new paths. Doing so may mean that fewer LSPs are allowed to cross a particular router under steady state, which in turn may impact the network design (different LSP placements or the addition of an extra router). This is particularly important because many times at the design stage the only question that is asked is how many LSPs will cross a particular router, and no thought is given to either the protection paths or the make-before-break scenario.

To summarize, the benefits of local protection come at a cost. It is important to understand how the deployment of local protection affects the resource consumption on the routers, to determine whether the equipment can support the proposed deployment, not just in the steady state but also following a failure.

3.10.2 Evaluating a local protection implementation

The expectation is that local protection will provide very fast recovery following a failure of the protected resource. Just how fast ‘very fast’ is depends on the time it takes to detect the failure and the time it takes to update the forwarding state. Let us take a look at some of the ways to evaluate an implementation with regards to local protection.

3.10.2.1 Detection time

When fast detection is not available in hardware, support of BFD and the speed at which it operates directly impacts on the total recovery time. When deploying BFD, it is important to bear in mind that because BFD is a hello protocol, it must run on the routers at both ends of the link for which fast detection is required.

3.10.2.2 Switchover time for a single LSP

Vendors often express their local protection performance in terms of the time it takes to switch over one LSP. This time translates to the amount of traffic that is lost following a failure. It is typically measured by sending traffic at a known rate over an LSP set up with local protection and

measuring the traffic loss following the failure of a protected resource. However, it is seldom the case that a single LSP needs to be protected.

3.10.2.3 Number of LSPs switched over in a certain amount of time

The second scaling number that vendors quote is the number of LSPs that can be successfully protected within a certain amount of time. Assuming that N LSPs cross the failed resource, the question is how much time it takes to switch over all of them, which boils down to how many forwarding entries must be changed for moving them all to the protection path. For 1:1 protection, the answer is unambiguously N , because each LSP has a different label on the protection path. What if all N LSPs were to share the same protection path? In this case, it might be possible to push the same label on to all LSPs by making a single change to the forwarding table. In any case, the maximum number of updates is bounded by the maximum number of LSPs traversing the protected resource. From a practical point of view, the number of LSPs traversing any link or node is not more than a few thousand.

3.10.2.4 Forwarding state update for IP routes resolving over a protected LSP

Even when testing the recovery of a single LSP, the location of the failure (head end, mid-point or tail end) may play a role in the protection time. This is because the use of the LSP and the forwarding state that is installed (and thus needs to be updated) is different at the head end and mid-point. Imagine a network where BGP traffic traverses the network encapsulated in MPLS and all BGP routes resolve over a single LSP. The LSP head end has a forwarding state for each and every one of the BGP destinations. The routers in the middle of the network maintain only a single forwarding entry (the one created for the LSP). When the failure happens at the LSP mid-point, protecting the traffic to any of the BGP destinations requires only fixing the underlying LSP, basically switching over a single LSP to the protection path.

However, when the failure happens at the head end, the forwarding state for each of the BGP destinations must be updated. The heart of the matter is whether these updates need to be implemented as separate forwarding state updates or not. It is easy to understand that if route 10.0.0.1 is forwarded over (L1, if1) and the protection path is through (L2, if2), then the forwarding entry for route 10.0.0.1 has to be updated to point to (L2, if2). Assume that 100 000 other routes share the same forwarding state (because they are forwarded along the same LSP).

Clearly, there need to be 100 000 distinct forwarding entries for the 100 000 distinct destinations in the forwarding table. How are they

represented? One option is to create 100 000 distinct forwarding states, one for each IP prefix. When the LSP switches to the protection path, all 100 000 entries must be updated. A more efficient option is to share the actual forwarding state between all IP prefixes by introducing a level of indirection. For example, rather than maintaining the exact label and interface, maintain an abstract entity representing the LSP. Even if the LSP switches to the protection path, from the BGP routes point of view forwarding has not changed and the BGP forwarding entries need not be modified. By using indirection, fewer forwarding state updates must be made.

The issue of updating forwarding state for IP routes falls at the boundary between local-protection and routing protocol implementation. Understanding the design options provides an insight into what kind of problems to look for when testing the performance of a vendor's software.

3.10.3 The cost of bandwidth protection

Bandwidth protection requires reserving bandwidth on the protection path. The cost for doing so is quite high: idle bandwidth that is used only under failure conditions. Therefore, the goal is to minimize the overall amount of bandwidth that is reserved.

Intuitively, it is easy to understand that shorter is better in this context: the longer the path, the more resources are kept idle. Reserving bandwidth along the long paths shown in Figure 3.19 is clearly not appealing. However, the picture is not that simple. The placement of the LSPs themselves can influence the protection path, especially in setups where some links are protected using other mechanisms and do not require protection using fast reroute. Figure 3.20 shows a network where all links except the link A–B are protected using other mechanisms, such as APS. An LSP is set up between nodes A and E, with a bandwidth requirement of 100 Mbps and bandwidth protection. Let us take a look at the total bandwidth reservation in this network, counted as the sum of all reservations on all links. When the LSP is set up along the shortest path A–B–E, 300 Mbps are reserved for the protected path and another 300 Mbps for the protection path for the link A–B, a total of 600 Mbps. If, instead, the LSP was set up over the longer path A–C–D–B–E, no protection would be necessary and only 400 Mbps would be reserved in the network.

Another interesting scenario arises with regards to the computation of the protection paths for different links used by the same LSP. Figure 3.21 shows two LSPs, LSP1 from node A to G along the path A–B–G and LSP2 from node A to E along the path A–B–E. The capacity of all links is 100 Mbps. The requirement is to provide bandwidth protection in case of a failure of node B. Ideally a single backup would be built around B and shared by both LSPs. However, because the two LSPs diverge after node B,

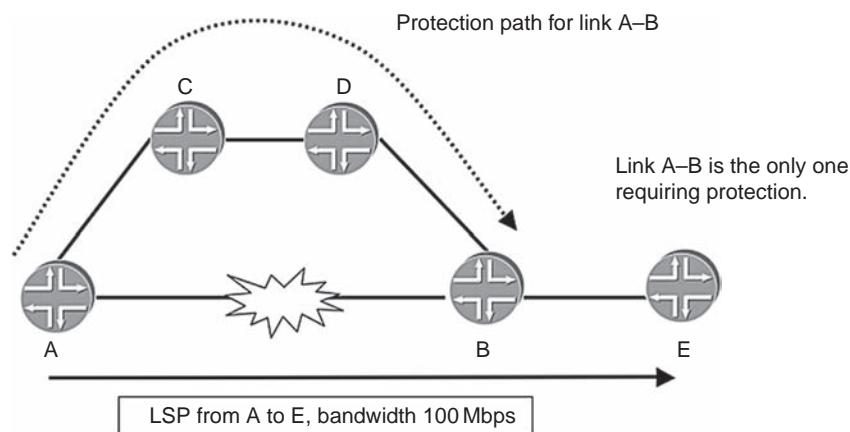


Figure 3.20 Setting up the LSP along the shortest path may not yield the best bandwidth utilization

they require different MPs and therefore different backup paths, as shown by the dashed lines in the figure.

The next question is what should the bandwidth be for each backup. The first backup protects all LSPs crossing the path A–B–G, up to 100 Mbps. The second protects all LSPs crossing the path A–B–E, which can also add up to 100 Mbps. If the two backups are set up with 100 Mbps each, a total

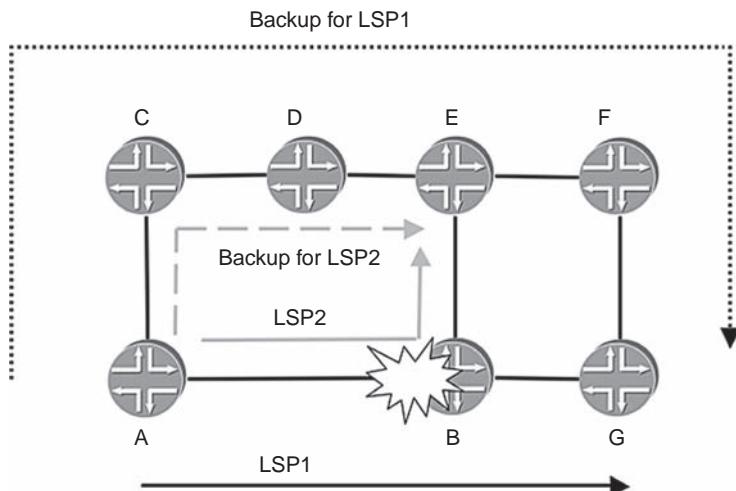


Figure 3.21 Using CSPF to compute the backup paths may not yield optimal bandwidth utilization

of 200 Mbps are needed for links A–C, C–D and D–E, and the setup of one of the backup paths with fail. However, the total band-width of all LSPs crossing the paths A–B–E and A–B–G cannot exceed 100 Mbps, because of the shared link A–B. This knowledge is lost when computing the backup paths and more bandwidth is required for them than necessary.

What the two examples discussed above illustrate is that to achieve optimal bandwidth utilization, more information is needed than what is available to CSPF. Therefore, the use of an offline tool may make sense, especially in situations where bandwidth is at a premium and strict guarantees are required during failure. This could be the case for delivery of large amounts of video traffic (e.g. satellite TV channels), where loss is not acceptable and traffic volumes are very high.

To summarize, bandwidth protection effectively means paying for bandwidth that is used only when a failure happens. To minimize the cost of bandwidth protection, it is necessary to have a global view of all LSPs and protection paths and to employ more specialized algorithms than CSPF. Offline tools can provide this functionality, at the cost of increased operational complexity, as explained in the Traffic Engineering chapter (Chapter 2).

3.11 IP AND LDP FRR

Fast failure recovery is a fundamental requirement for carrying sensitive traffic such as voice or video and is an important building block for providing QoS in MPLS networks. The problem is that the local protection schemes described so far only work in conjunction with RSVP, but many MPLS deployments use LDP as the label distribution protocol.

If an LDP network is to carry voice or video traffic, it must ensure fast failure recovery. Let us see the options available:

- Move away from LDP and switch to RSVP. This is an unacceptable proposition for most providers because it requires a massive reengineering of the network.
- Use one-hop RSVP LSPs with link protection to protect selected links, and continue to run targeted LDP sessions over these RSVP tunnels, as shown in Figure 3.22. Note that both under normal conditions and following a failure, LDP traffic is tunneled inside RSVP. When the link fails, traffic on the one-hop RSVP tunnels is protected, so the LDP traffic is also protected. This approach is attractive because it allows the operator to continue to use LDP for the setup of LSPs and does not require changes to the edge routers. However, protection for only link failures can be achieved.

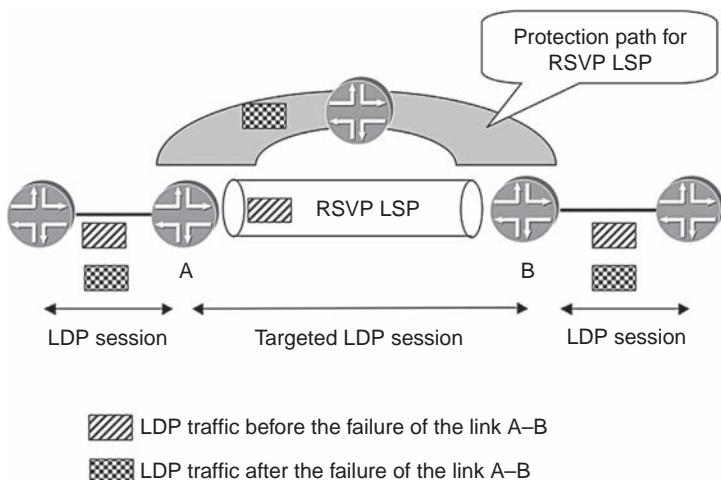


Figure 3.22 Protecting LDP traffic by using one-hop RSVP tunnels with link protection

- Find a mechanism that will provide fast-reroute behavior for LDP. This is the most attractive proposition for the network operators, and for this reason such schemes have been studied extensively over the past few years in the IETF and implementations are becoming available from the vendors.

The fundamental difference between LDP and RSVP in the context of fast reroute is that LDP LSPs follow the IGP while RSVP LSPs are set up along a predefined path and are immune to changes in the IGP. Thus, LDP is influenced by changes in IP routing while RSVP is not. Based on this observation, it can be argued that if the problem of IP fast reroute is solved, the same protection can be automatically extended to LDP. Under this assumption, this chapter discusses fast reroute for IP and LDP interchangeably.

One special case in which it is relatively straight forward to implement fast-reroute style behavior for LDP is for certain ECMP cases. Let us consider the topology in Figure 3.19, for a situation where LDP is being used (rather than RSVP as shown in the figure). Let us suppose A is sending LDP (or indeed plain IP) traffic to R4 (this traffic could be entering the network at A or A could be a transit router for this traffic). The sum of the metrics along the path A, B, C, D, R4 is the same as the sum of the metrics along the path A, R1, R2, R3, R4. Therefore, A has two ECMP paths for the traffic to R4. Some implementations install forwarding entries in the router forwarding engine in such a way that normally the traffic uses both paths. If, say, the link between A and B goes down, the traffic using that link can

be immediately snapped onto the other path via R1, without waiting for the control plane to reconverge from the existing ECMP paths to the single path. This behavior is valid because it is already known that this path is the shortest path to R4, so the traffic can be safely moved onto this path before IGP convergence has taken place. In this way, for this special ECMP case, convergence times similar to those in RSVP fast-reroute can be achieved. Note that this behavior is an implementation-specific improvement and does not require any change in the protocol behavior. When the control plane eventually reconverges, it will do so to the exact same path as the one already being used for forwarding.

There are several proposals for providing IP fast reroute in a wider set of scenarios than just the ECMP case discussed above. These are currently under discussion by the Routing Area Working Group [RTGWG] in the IETF. The proposals fall broadly into two categories: tunnel based or alternate-path based. Let us discuss them separately below.

3.11.1 The tunnel-based approach

The tunnel-based approach is conceptually very similar to RSVP FRR. A bypass tunnel is set up ahead of time around the protected resource and is used to shuttle the traffic around the point of failure. In this case, the tunnel is set up with RSVP and the same mechanisms as described above for RSVP can be used. Note that this approach is different from protection using one-hop RSVP LSPs discussed at the beginning of the section, because in this case LDP traffic is forwarded over the RSVP tunnel only following a failure, whereas in the one-hop RSVP LSP case, LDP traffic is always tunneled in RSVP.

The tunnel-based approach can provide protection for both link and node failures. However, to provide node protection, the label used at the MP must be known to the PLR, as discussed in Section 3.7. For RSVP FRR, this label was learned from the Record Route Object. For LDP, new procedures must be set in place to advertise this label from the MP and the PLR. Different methods to do so were discussed by the MPLS Working Group in the IETF, but no solution has yet been adopted.

The tunnel-based approach is very attractive because it relies on the same elements that have been deployed and are proved to be working for RSVP FRR. However, is it really all this simple? For it to work, traffic must reach the PLR. For RSVP LSPs, this is guaranteed, because the traffic is source routed and follows a preestablished path that is not influenced by routing changes. Unfortunately with LDP this is not the case, as LDP paths track IP routing.

Changes in IP routing as the network reconverges following the link failure may cause the traffic never to reach the beginning of the bypass

tunnel, which is similar to the situation described in Section 3.5 for Figure 3.2. To recap, before the failure the route from node S to node D is through node R1 and the route from node R3 to D is through node S. In the event of a failure of the link R1–R2, the route from S to D is through node R3. If S starts forwarding traffic to R3, packets will not arrive at R1 and at the protection path. This can be a problem if the IGP on R3 has not yet performed its SPF and still points to S as its best path (the route from before the failure). In this case, traffic will loop between S and R3. This situation is referred to as microloops and there are several proposals for avoiding it. The most intuitive is to slow down the convergence of the routers so as to avoid the situation where node S reconverged faster than node R3. The solution to the microloop problem is currently being worked on by the Routing Area Working Group in the IETF.

Apart from the issue of microloop prevention, a tunnel-based approach is attractive for the following reasons:

- (i) Most elements of the solution have been deployed in networks today in the context of MPLS FRR.
- (ii) The computations are well bounded (one computation per link for link protection, or one per next next-hop for node protection).
- (iii) The forwarding paradigm during failure is simply MPLS.
- (iv) The path of the packets during failure is known and there is control over it.

3.11.2 The alternate-path approach

The alternate-path approach relies on maintaining an alternate path towards the destination and taking that path when the primary fails.

This is best shown in an example. In Figure 3.23 there are two paths from source A to destination D, A–B–D and A–C–D. From the IGP's point of view, path A–B–D is shorter, with a metric of 9, and normally the forwarding state for D at A would simply point to if1. When using the alternate-path approach both paths are maintained. This means two things: both paths must be computed and the forwarding state must be kept for both paths. This must be done for each of the destinations reachable from A.

When link A–B fails, traffic is sent on the alternate path over link if2 as soon as A detects the failure. In the simple topology in the figure, this approach works, traffic arrives to C over interface if2 and is forwarded along the link C–D to D, using the primary route at C. The solution described above is the basic approach for IP fast reroute as documented in [RFC5286], also known as loop-free alternates.

At this point, it should already be clear that this solution depends on the network topology. If the metric of link C–D was 30 instead of 3, then

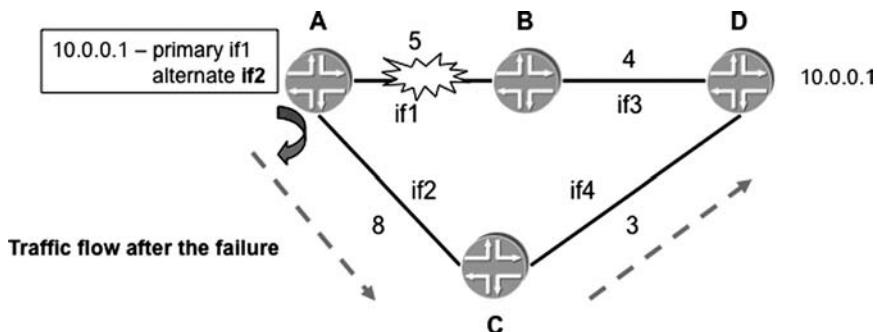


Figure 3.23 Protection using loop-free alternates

the primary route for 10.0.0.1 at C would point back to A (through path C–A–B–D). Traffic arriving at C from A would then do a ‘U-turn’ and loop between A and C until the IGP on C converges. To solve this problem, the notion of U-turn alternates is introduced. If C could detect that it is causing the traffic to take a U-turn, then C could use its alternate route to forward the traffic and packets would arrive safely at D, as shown in Figure 3.24. This solution is also known as U-turn alternates and it is currently work in progress in the IETF [U-TURN].

Let us take a look at some of the properties of this solution, in contrast to a tunnel-based approach:

1. *Partial coverage.* The solution does not work in arbitrary topologies. Neither loop-free alternates nor U-turn alternates can provide coverage for all failure cases. In some cases, the network operator is happy with

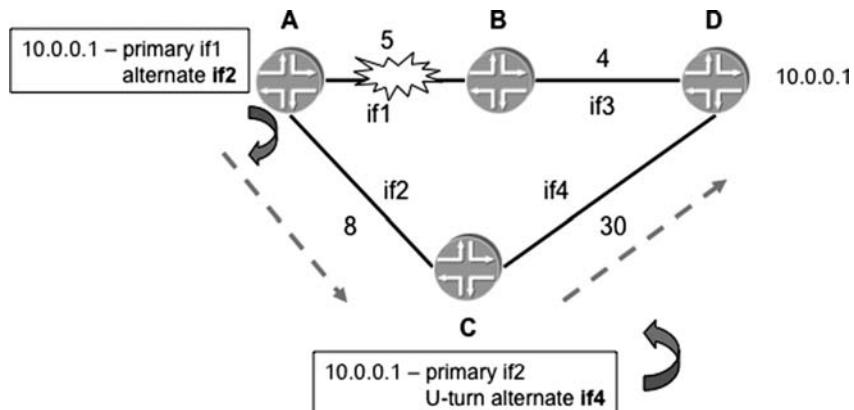


Figure 3.24 Protection using U-turn alternates

partial coverage because at least the convergence time is improved for the covered failure cases and is not made worse for the non-covered cases. In cases where the neighboring nodes cannot provide coverage for a destination, some implementations allow coverage to be extended by using an RSVP signaled LSP to tunnel protected traffic to a node further downstream that can provide coverage. Often tunneling to a next-next-hop node is sufficient to achieve coverage in many topologies.

2. *Change in the forwarding paradigm.* Forwarding behaves differently in steady state and following a failure, making it difficult to debug failure scenarios.
3. *Computational complexity.* Each router must compute two paths for every destination. The number of computations required to set up alternate paths is proportional to the number of destinations in the network, rather than being proportional to the number of neighbors or next next-hop neighbors. For U-turn alternates, more complex computations need to be performed to detect the presence of U-turns.
4. *Growth in the forwarding state.* The forwarding state doubles at each and every router in the network, because a separate alternate must be maintained for each destination.
5. There is no control over the path traffic will take after a failure.

To summarize, the success of MPLS fast reroute and the move towards converged networks has prompted an interest in fast-reroute solutions for both IP and LDP. The available solutions are based on either maintaining alternate paths for use during failure or on providing backup tunnels similarly to MPLS FRR.

3.12 CONCLUSION

MPLS fast reroute provides protection for the traffic following link or node failures, within times that are virtually undetectable at the application layer. This is a fundamental requirement for carrying sensitive traffic such as voice or video and is an important building block for converging all services on to a common MPLS core.

Used together with traffic engineering, fast reroute can ensure adherence to strict QoS guarantees, not just in the normal case but also following a failure, thus completing the TE solution as described in Chapter 2. In the next chapter, we will explore DiffServ Aware TE, which refines the TE solution by allowing bandwidth reservations to be carried out on a per-DiffServ class basis.

3.13 REFERENCES

- [BFD] BFD Working Group, <http://ietf.org/html.charters/bfd-charter.html>
- [BFD-BASE] D. Katz and D. Ward, *Bidirectional Forwarding Detection*, draft-ietf-bfd-base-11.txt (work in progress)
- [BFD-MHOP] D. Katz and D. Ward, *BFD for Multihop Paths*, draft-ietf-bfd-multipath-09.txt (work in progress)
- [RFC4090] P. Pan, G. Swallow and A. Atlas, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*, RFC 4090, May 2005
- [RFC4203] K. Kompella and Y. Rekhter, *OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*, RFC 4203, October 2005
- [RFC5286] A. Atlas and A. Zinin, *Basic Specification for IP Fast-reroute: Loop-free Alternates*, RFC 5286, September 2008.
- [RFC5307] K. Kompella and Y. Rekhter, *IS-IS Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)*, RFC 5307, October 2008
- [RTGWG] Routing Area Working Group <http://ietf.org/html.charters/rtgwg-charter.html>
- [U-TURN] A. Atlas, *U-turn Alternates for IP/LDP Fast-reroute*, draft-atlas-ip-local-protect-uturn-03.txt (work in progress)

3.14 FURTHER READING

- [BMWG] Benchmarking Methodology Working Group <http://ietf.org/html.charters/bmwg-charter.html>
- [BMFRR] S. Poretsky et al., *Benchmarking Methodology for MPLS Protection Mechanisms*, draft-poretsky-mpls-protection-meth-05.txt (work in progress)

3.15 STUDY QUESTIONS

1. One of the disadvantages of path protection is the double booking of resources, which is a problem in a situation of resource crunch, such as after one or more failures in the network. One approach to alleviate the double booking problem is to set up the secondary paths without any bandwidth reservations. (a) What is the rationale behind this approach and what mechanisms does it rely on? Another approach is to set up the secondary paths with lower priority than the primary, as explained in Section 3.4. (b) What is the disadvantage in this approach?

2. Look at the protocol extensions described in RFC4090. Pick five of these extensions (new objects or flags in existing objects) and explain the rationale for the extension.
3. Describe why facility protection can lead to longer protection paths than 1:1 protection.
4. Figure 3.8 shows a ring topology where traffic is forced to 'double-back' when taking the protection path. Such scenarios are not limited to ring topologies alone. Can you give another example of a topology where the same might happen?
5. A particular network is made up of OC48/STM16 and OC192/STM64 links and has two classes of LSPs: high-priority and low-priority ones (in this context, priority does not necessarily refer to the setup/hold priority). High-priority LSPs are signaled over OC192/STM64 links only, while low-priority LSPs are signaled over OC48/STM16 links. The operator would like to implement facility-based link protection, while at the same time requiring protection paths for low-priority LSPs to only use OC48/STM16 links. Discuss the ways in which this problem could be solved.
6. We have seen in Section 3.10.3 that when bandwidth protection is not provided, traffic can be impacted. We have also seen that this impact can be constrained to the protected traffic only, through the use of DiffServ markings for the packets entering the protection path. However, an operator may wish to avoid attracting traffic onto LSPs that are on the protection path. What are some of the techniques he can use to achieve this goal?
7. When using path protection, what are the pros and cons of having a presignaled secondary path?
8. Describe why the traffic loss for path protection tends to be longer than that for local protection.

4

MPLS DiffServ-TE

4.1 INTRODUCTION

In the MPLS Traffic Engineering chapter (Chapter 2), we saw how MPLS traffic engineering (TE) allows the user to create end-to-end paths across the network with bandwidth reservations. This guarantees that the resources are available to carry traffic of volume less than or equal to the bandwidth reservation. A disadvantage of the basic MPLS-TE model is that it is not aware of the different DiffServ classes, operating at an aggregate level across all of them.

This chapter introduces the concept of DiffServ Aware MPLS-TE, which refines the MPLS-TE model by allowing bandwidth reservations to be carried out on a per-class basis. The result is the ability to give strict QoS guarantees while optimizing use of network resources. The QoS delivered by MPLS DiffServ-TE allows network operators to provide services that require strict performance guarantees, such as voice, and to consolidate IP and ATM/FR (Frame Relay) networks into a common core.

This chapter explores MPLS DiffServ-TE and its extensions. It assumes familiarity with DiffServ in general and MPLS DiffServ in particular, discussed in the Foundations chapter (Chapter 1), as well as with MPLS Traffic Engineering, discussed in the Traffic Engineering chapter (Chapter 2).

4.2 THE BUSINESS DRIVERS

Traditionally, IP/MPLS-based networks were used only for services with relatively relaxed requirements in terms of delay, jitter or bandwidth guarantees. Increasingly, providers have started carrying a wider range of services, such as PSTN-quality voice or providing ATM/FR or Ethernet over the MPLS core. The driver for offering these services is the cost savings achieved by eliminating the need to have several separate physical networks. Indeed, one of the most attractive promises of MPLS is the ability to converge all services on to a common core. The challenge lies in the fact that most of these services often require stricter service-level agreements (SLAs) than the previous norm on IP/MPLS networks.

The SLAs define the service quality experienced by traffic transiting the network and are expressed in terms of latency, jitter, bandwidth guarantees, resilience in the face of failure, and down time. The SLA requirements translate to two conditions: (a) different scheduling, queuing and drop behavior based on the application type and (b) bandwidth guarantees on a per-application basis.

To date, service providers have rolled out revenue-generating services in their networks using DiffServ alone. By assigning applications to different classes of service and marking the traffic appropriately, condition (a) was met. However, this approach assumes that there are enough resources to service the traffic according to the marking. If the traffic follows a congested path, traffic may be dropped, or it may experience different delay and jitter characteristics than required by the SLAs. In principle, service providers could solve this problem by using overprovisioning to avoid congestion altogether. Besides being wasteful with regards to resource utilization, this approach of ‘throwing bandwidth at the problem’ cannot provide any guarantees when congestion is caused by link and/or node failures.

In the Traffic Engineering chapter (Chapter 2) we have seen how MPLS traffic engineering sets up label-switched paths (LSPs) along links with available resources, thus ensuring that bandwidth is always available for a particular flow and avoiding congestion both in the steady state and in failure scenarios. Because LSPs are established only where resources are available, overprovisioning is not necessary. Further optimization of transmission resources is achieved by allowing LSPs not to follow the shortest path, if the available resources along the shortest path are not sufficient. An added benefit of MPLS is that built-in mechanisms such as link protection and fast reroute (discussed in the Protection and Restoration chapter, Chapter 3) provide resilience in the face of failure. The catch is that MPLS-TE is oblivious of the class-of-service (CoS) classification, operating only on the available bandwidth at an aggregate level across all classes.

MPLS DiffServ-TE makes MPLS-TE aware of CoS, allowing resource reservation with CoS granularity and providing the fault-tolerance properties of MPLS at a per-CoS level. By combining the functionalities of both DiffServ and TE, MPLS DiffServ-TE delivers the QoS guarantees to meet strict SLAs such as the ones required for voice, ATM and Frame Relay, thus meeting condition (b).

Note that even if resources are reserved on a per-CoS basis and that even if traffic is properly marked to conform to the CoS appropriate for the application, the SLAs still cannot be guaranteed unless further mechanisms, such as policing and admission control, are set in place to ensure that the traffic stays within the limits assumed when the resource reservation was made, as will be seen in Section 4.4.8.

4.3 APPLICATION SCENARIOS

The DiffServ-TE solution is the product of the TEWG Working Group in the IETF.¹ In [RFC3564], the Working Group documented a few application scenarios that cannot be solved using DiffServ or TE alone. These scenarios form the basis for the requirements that led to the development of the DiffServ-TE solution and are presented in this section. The scenarios show why per-traffic-type behavior is necessary.

4.3.1 Limiting the proportion of traffic from a particular class on a link

The first scenario involves a network with two types of traffic: voice and data. The goal is to maintain good quality for the voice traffic, which in practical terms means low jitter, delay and loss, while at the same time servicing the data traffic. The DiffServ solution for this scenario is to map the voice traffic to a per-hop behavior (PHB) that guarantees low delay and loss, such as the expedited-forwarding (EF) PHB.

The problem is that DiffServ alone cannot give the required guarantees for the following reason. The delay encountered by the voice traffic is the sum of the propagation delay experienced by the packet as it traverses the network and of the queuing and transmission delays incurred at each hop. The propagation and transmission delays are effectively constant; therefore, in order to enforce a small jitter on the overall delay, the queuing delay must be minimized. A short queuing delay requires a short queue,

¹ The TEWG finished all its work items and has been closed.

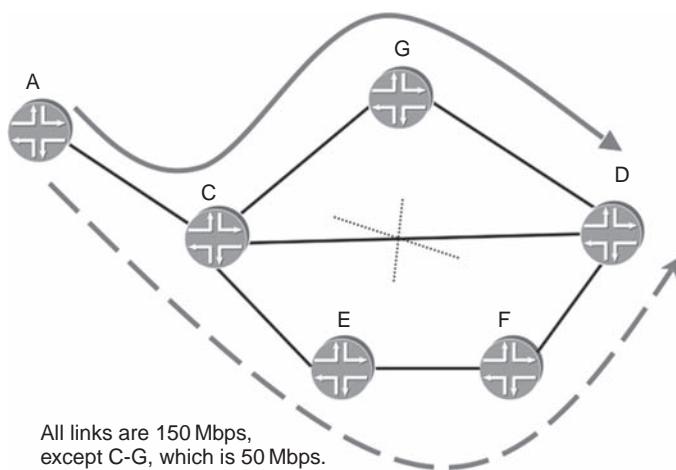


Figure 4.1 Overprovisioning cannot provide guarantees in a failure scenario

which from a practical point of view means that only a limited proportion of the queue buffers can be used for voice traffic.

Thus, the requirement becomes 'limit the proportion of voice traffic on each link'. In the past, service providers used overprovisioning to achieve this goal, making sure that more bandwidth was available than would ever be necessary. However, overprovisioning has its own costs and, while it may work well in the normal case, it can give no guarantees in the failure scenario. Figure 4.1 shows a network operating under such a regimen. Under normal conditions, the voice traffic takes the path A-C-D, which is the shortest path. The link capacity is large, so the percentage of the voice traffic on each link is acceptable. When the link C-D fails, the traffic reroutes on the next best path, A-C-G-D. The link C-G is low-capacity and the percentage of voice traffic becomes too large. Instead, the traffic should have rerouted on the path A-C-E-F-D.

Taking the solution a step further, limiting the proportion of voice traffic on all links can be achieved by artificially limiting the available bandwidth on a link to the proportion suitable to satisfy the voice traffic requirements alone and using TE to ensure that traffic (voice and data) is mapped in such a way as to honor these artificially lower available resources. This solution provides the requested functionality but wastes resources because bandwidth that could be allocated to delay-insensitive data traffic is now idle and unavailable. The root of the problem is that TE cannot distinguish between the two types of traffic and cannot enforce allocations at a per-traffic-type granularity.

4.3.2 Maintaining relative proportions of traffic on links

The second scenario extends the previous example to a network that supports three traffic types that map to three ‘classes of service’. The proportion of the different traffic types depends on the source and destination of the traffic. The challenge for the service provider is to configure the queue sizes and queue scheduling policies on each link to ensure that the correct PHB is given to each class.² It is impractical to configure these parameters based on the link load at a given time: changes in routing, link or node failures and preemption between LSPs make the link load a very dynamic property. Instead, from an operational and maintainability point of view, it would be ideal to fix the relative proportions of each traffic type on the links, allocate the queue sizes and scheduling policies accordingly, and use TE to make the traffic comply with the available resources. This solution requires TE to enforce different bandwidth constraints for different classes of traffic.

4.3.3 Providing guaranteed bandwidth services

In this application, which is very similar to the example in Section 4.3.1, there are two types of traffic: best effort and ‘guaranteed bandwidth’. The guaranteed bandwidth traffic must comply with a given SLA. The goal is to provide the required service level to the guaranteed traffic and also to be able to traffic-engineer the best-effort traffic. As in the first example, in order to enforce strict SLAs, the guaranteed bandwidth traffic must be engineered not to overflow the allotted bandwidth of the link, and TE must be employed to ensure this requirement. In addition, the best-effort traffic must also be traffic-engineered, to increase link utilization. Here again, TE must have knowledge of the type of traffic.

4.4 THE DiffServ-TE SOLUTION

This section examines how per-traffic-type behavior is enforced, both when setting up an LSP and when forwarding traffic.

4.4.1 Class types

The basic DiffServ-TE requirement is to be able to make separate bandwidth reservations for different classes of traffic. This implies keeping

²The example uses three classes rather than the two (voice and data) from the previous scenario. This is because when just voice and data are used, it can be argued that the queue size must be set uniformly for the voice traffic.

track of how much bandwidth is available for each type of traffic at any given time on all routers throughout the network. [RFC3564] spells out the requirements for support of DiffServ Aware MPLS-TE and defines the fundamental concepts of the technology.

For the purpose of keeping track of the available bandwidth for each type of traffic, [RFC3564] introduces the concept of a class type (CT). [RFC3564] does not mandate a particular mapping of traffic to CTs, leaving this decision to the individual vendors. One possible implementation is to map traffic that shares the same scheduling behavior to the same CT. In such a model one can think of a CT in terms of a queue and its associated resources. Because the PHB is defined by both the queue and the drop priority, a CT might carry traffic from more than one DiffServ class of service, assuming that they all map to the same scheduler queue.

The IETF standards require support of up to eight CTs referred to as CT0 through CT7. LSPs that are traffic-engineered to guarantee bandwidth from a particular CT are referred to as DiffServ-TE LSPs. In the current IETF model, a DiffServ-TE LSP can only carry traffic from one CT. LSPs that transport traffic from the same CT can use the same or different preemption priorities. By convention, the best-effort traffic is mapped to CT0. Because all pre-DiffServ-TE LSPs are considered to be best effort, they are mapped to CT0.³

Let us revisit the application scenario from Section 4.3.1 and discuss it in terms of CTs. The voice and data network in this example supports two DiffServ PHBs, EF and BE (for voice and data traffic respectively). The goal is to provide service guarantees to the EF traffic. Two scheduler queues are configured on each link, one for BE and one for EF. CT0 is mapped to the BE queue and CT1 is mapped to the EF queue. The bandwidth available for CT1 (the voice traffic) is limited to the percentage of the link required to ensure small queuing delays for the voice traffic. Separate TE LSPs are established with bandwidth requirements from CT0 and from CT1.

In the following sections, we look at how LSPs are established with per-CT bandwidth requirements.

4.4.2 Path computation

In the Traffic Engineering chapter (Chapter 2), we discussed how CSPF computes paths that comply with user-defined constraints such as bandwidth and link attributes. DiffServ-TE adds the available bandwidth for each of the eight CTs as a constraint that can be applied to a path. Therefore, CSPF is enhanced to take into account a CT-specific bandwidth at a given

³ Pre-DiffServ-TE LSPs and DiffServ-TE LSPs from CT0 are signaled in exactly the same way.

priority as a constraint when computing a path. For example, the user might request an LSP of CT1 at priority 3 with a bandwidth of 30 Mbps. CSPF computes a path that meets these criteria. For the computation to succeed, the available bandwidth per CT at all priority levels must be known for each link.

This means that the link-state IGP must advertise the available bandwidth per CT at each priority level on every link. Recall that there are eight CTs and eight priority levels, giving a total of 64 values that need to be carried by the link-state protocols. In an ideal world, all 64 values would be advertised and stored for each link. However, the IETF decided to limit the advertisements to eight values out of the possible 64 [RFC4124].⁴

How are these eight values picked? TE classes are defined for this purpose as a combination of CT and priority. The IGP advertises the available bandwidth for each of the TE classes defined. DiffServ-TE supports a maximum of eight TE classes, TE0 through TE7, which can be selected from the 64 possible CT-priority combinations through configuration. At one extreme, there is a single CT with eight priority levels, very much like the existing TE implementation. At the other extreme, there are eight distinct CTs, with a single priority level. The combinations chosen depend on the classes and priorities that the network must support. Figure 4.2 shows the 64 combinations of class type and priority, and a choice of eight TE classes, called a TE class matrix. Note that both the setup and the hold priorities used by LSPs must be used in the TE class matrix. This is because available bandwidth is reported per TE class and this information is required for both priority levels. Because TE classes are used in the IGP advertisements, all routers must have a consistent configuration of the TE class matrix. Otherwise, the advertisements will be incorrectly attributed to the wrong CT-priority combination.

⁴ One of the most heated debates in the Working Group was around the question of whether to advertise all 64 available bandwidth values. The opponents argued that doing so would yield a very small gain, since 64 different combinations are more than anyone would deploy in a real network, and that advertising all the 64 values would place a large burden on the IGP. This would happen because (a) the information would not fit in one link-state advertisement and would require sending several of them, greatly increasing the number of IGP advertisements in the network, and (b) new advertisements would need to be sent every time any of the values changed, creating churn. Although the concerns may seem valid, they are not entirely justified. The 64 different combinations are useful for allowing flexible configuration of CT and priorities without the need to coordinate TE class matrices across the entire network. The concern regarding the need for several link-state advertisements is also not founded given the current MTU sizes. Even without any smart packing of the data, the information would still not require more than one advertisement. If space were a concern, efficient packing could have solved the problem. Finally, the question of churn when the values change can be solved by dampening the advertisements, in the same way it is done for regular TE.

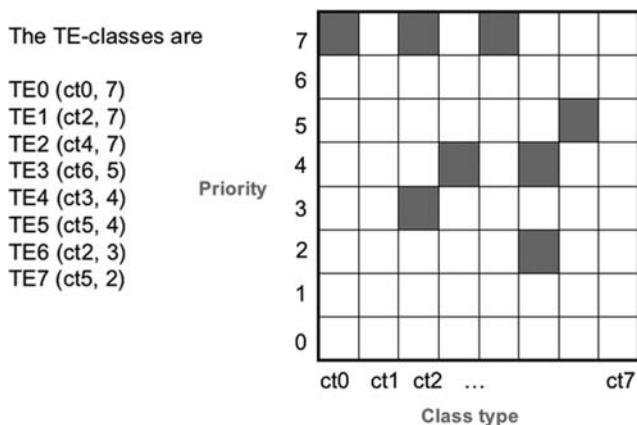


Figure 4.2 Picking eight TE classes out of the 64 possible combinations

The link-state IGP s advertise the available bandwidth for each TE class. [RFC4124] mandates that this advertisement be made using the existing Unreserved Bandwidth TLV, which was previously used to disseminate unreserved bandwidth for TE [RFC3784] [RFC3630]. Therefore, the information that is available to CSPF through the IGP s is relevant only for the CT and priority combinations that form valid TE classes. Thus, in order for CSPF to perform a meaningful calculation, the CT and priority levels chosen for an LSP must correspond to one of the configured TE classes.

To summarize, two crucial design decisions were taken with respect to the advertisement of per-class available bandwidth:

1. Advertising information for only eight CT-priority combinations rather than for all 64 of them.
2. Overriding the semantics of an existing TLV in the IGP s to carry the available bandwidth information for the TE classes chosen.

We have already seen that decision 1 required the introduction of the not very intuitive concept of a TE class and limits the characteristics of the LSPs in the network to the classes and priorities supported by the TE class matrix. In the section discussing deployment of the DiffServ-TE solution (Section 4.4.9) we will see how decision 2 constrains the deployment of DiffServ-TE in networks that already use TE.

Despite these constraints, implementations exist today supporting this model and have been demonstrated to interoperate. As long as the classes and priorities are consistently configured, the solution is backwards-compatible with routers not supporting the functionality.

To summarize, to compute a path with per-CT bandwidth constraints, CSPF is enhanced to handle per-CT reservation requirements and the IGPs are enhanced to carry per-CT available bandwidth at different priority levels.

4.4.3 Path signaling

After the path is calculated, it is signaled, and admission control and bandwidth accounting are performed at each hop. [RFC4124] defines the necessary extensions to RSVP-TE that allow it to establish paths with per-CT bandwidth reservations.⁵

The CT information for an LSP is carried in the new Class Type Object (CT object) in the RSVP path message, and specifies the CT from which the bandwidth reservation is requested. Two rules ensure that it is possible to deploy DiffServ-TE incrementally in the network:

1. The CT object is present only for LSPs from CT1 through CT7 (if the CT object is missing, CT0 is assumed).
2. A node that does not understand the DiffServ-TE extensions and that receives a path message with the CT object rejects the path establishment.

These two rules ensure that establishment of LSPs with per-CT reservation is possible only through DiffServ-TE-aware nodes, while pre-DiffServ-TE LSPs, which are considered to belong to CT0, can cross both old and new nodes. In a mixed network, where some of the routers support DiffServ-TE and others do not, DiffServ-TE LSPs can establish through the routers that have the support.

The CT information carried in the path message specifies the CT over which admission control is performed at each node along the path. If a node along the path determines that enough resources are available and the new LSP is accepted, the node performs bandwidth accounting and calculates the new available bandwidth per-CT and priority level. This information is then passed back into the IGPs.

To summarize, for each LSP, the CT is implicitly signaled for CT0 and explicitly signaled for all other CTs. The CT is necessary to perform the calculation of the available resources. How is this calculation performed?

⁵Note that although CR-LDP also supports explicit routing, no extensions are defined for it because the IETF decided in [RFC3468] to abandon new development for CR-LDP.

4.4.4 Bandwidth constraint models

One of the most important aspects of the available bandwidth calculation is the allocation of bandwidth among the different CTs. The percentage of the link's bandwidth that a CT (or a group of CTs) can take up is called a bandwidth constraint (BC). [RFC3564] defines the term 'bandwidth constraint model' to denote the relationship between CTs and BCs. Several bandwidth constraint models exist; the most popular are the maximum allocation model (MAM) and the Russian dolls model (RDM).

4.4.4.1 The maximum allocation model (MAM)

The most intuitive bandwidth constraint model maps one BC to one CT. This model is called the maximum allocation model (MAM) and is defined in [RFC4125]. From a practical point of view, the link bandwidth is simply divided among the different CTs, as illustrated in Figure 4.3.

The benefit of MAM is that it completely isolates different CTs. Therefore, priorities do not matter between LSPs carrying traffic from different CTs. In the network shown in Figure 4.4, all links are of capacity 10 Mbps and are partitioned to 9 Mbps for data (CT0) and 1 Mbps for voice (CT1). The operator sets up two data LSPs: LSP1 with 9 Mbps and LSP2 with 1 Mbps. LSP1 is set up along the shortest path A–B–C. As a result, the available bandwidth for CT0 along this path becomes 0 and LSP2 is forced to establish along the longer path A–D–E–C. This is despite the fact that 1 Mbps is free along the path A–B–C. When the operator wants to set up a voice LSP, the resources are guaranteed to be available for class CT1 on the shortest path and no preemption of data LSPs (CT0) is necessary, or indeed possible.

The problem with MAM is that because it is not possible to share unused bandwidth between CTs, bandwidth may be wasted instead of being used for carrying other CTs. Consider the network shown in Figure 4.4. In the absence of voice LSPs, bandwidth is available on all the links on the shortest path for data traffic, but this bandwidth cannot be used for setting up another data LSP. The second data LSP is forced to follow a nonoptimal

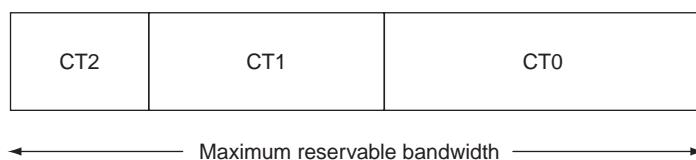


Figure 4.3 The allocation of bandwidth to CTs in the MAM model (for simplicity, only three CTs are shown)

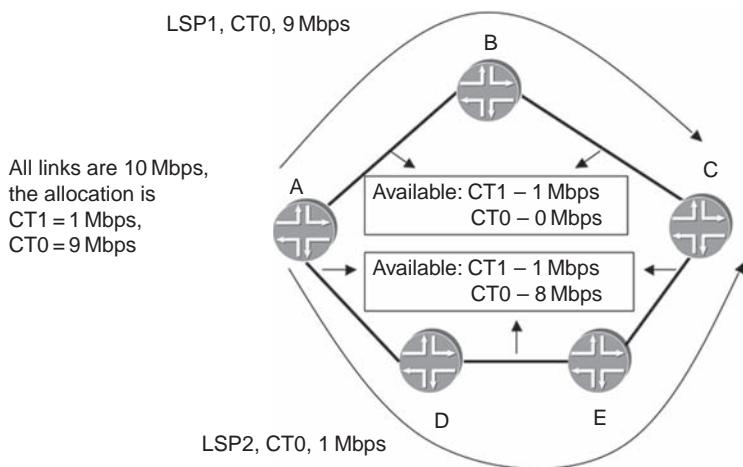


Figure 4.4 Even if no CT1 LSPs are established, the bandwidth allocated for CT1 cannot be used for CT0

path, even though bandwidth is available on the shortest path. On the other hand, after both data LSPs have been set up, if a voice LSP needs to be established, bandwidth is available for it on the shortest path.

The available bandwidth for the MAM model is accounted in a similar way as for TE, except that it is done on a per-CT basis. To calculate the bandwidth available for CT_n at priority p, subtract from the bandwidth allocated to CT_n the sum of the bandwidths allocated for LSPs of CT_n at all priority levels that are better or equal to p.

4.4.4.2 The Russian dolls model (RDM)

The Russian dolls bandwidth allocation model (RDM), defined in [RFC4127], improves bandwidth efficiency over the MAM model by allowing CTs to share bandwidth. In this model, CT7 is the traffic with the strictest QoS requirements and CT0 is the best-effort traffic. The degree of bandwidth sharing varies between two extremes. At one end of the spectrum, BC7 is a fixed percentage of the link bandwidth that is reserved for traffic from CT7 only. At the other end of the spectrum, BC0 represents the entire link bandwidth and is shared among all CTs. Between these two extremes are various degrees of sharing: BC6 accommodates traffic from CT7 and CT6, BC5 from CT7, CT6 and CT5, and so on. This model is very much like the Russian doll toy, where one big doll (BC0) contains a smaller doll (BC1) that contains a yet smaller doll (BC2), and so on, as shown in Figure 4.5.

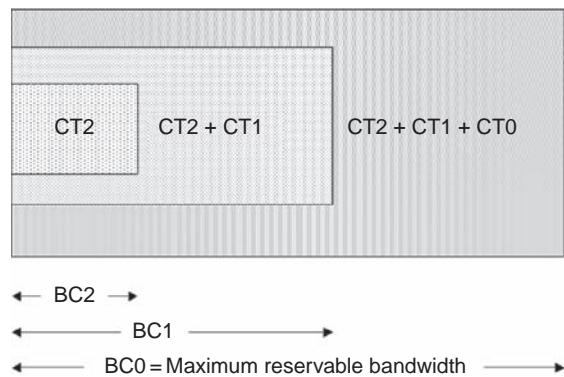


Figure 4.5 Russian dolls bandwidth allocation (for simplicity, only three CTs are shown)

Figure 4.6 shows how bandwidth accounting works for the RDM model. The figure shows a network very similar to the one in Figure 4.4, which carries two classes of traffic, data (CT0) and voice (CT1). The total bandwidth available on each link is 10 Mbps; 1 Mbps is allocated to BC1 and 10 Mbps are allocated to BC0. This means that each link can carry between 0 and 1 Mbps of voice traffic and use the rest for data. A data LSP, LSP1 from CT0, is already established over the shortest path A-B-C, with a reservation of 9 Mbps. Therefore, 1 Mbps remains available on this path, for use by either CT0 or CT1 traffic. Therefore, the available bandwidth for each of these classes is reported as 1 Mbps.

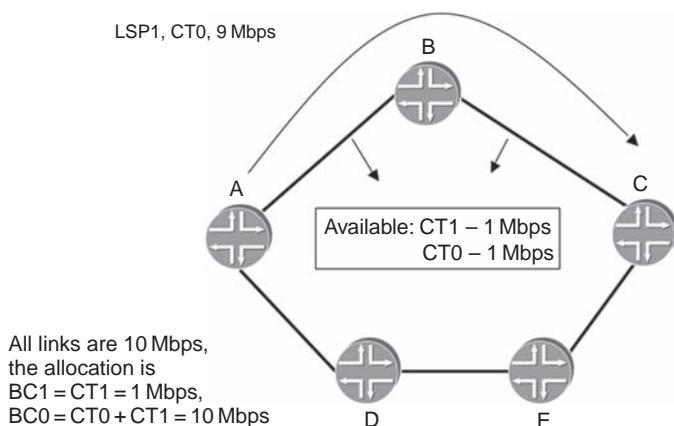


Figure 4.6 Bandwidth accounting for the RDM model

4.4.4.3 Comparison of the RDM and MAM models

The advantage of RDM relative to MAM is that it provides efficient bandwidth usage through sharing. In Figure 4.6, a second data LSP, LSP2, with a reservation of 1 Mbps can also be established on the shortest path to take advantage of the unused bandwidth. Another useful property that is achieved through sharing is cheap overprovisioning for real-time traffic. Because the extra bandwidth can be used by other types of traffic, allocating it to the real-time class does not affect the overall throughput of the network.

The disadvantage of RDM relative to MAM is that there is no isolation between the different CTs and preemption must be used to ensure that each CT is guaranteed its share of bandwidth no matter the level of contention by other CTs. This is shown in Figure 4.7. After establishing the second data LSP, LSP2, if the operator wants to establish a voice LSP, no resources are available for the voice traffic on the shortest path. Thus, one of the data LSPs must be preempted: otherwise, bandwidth is not guaranteed for the voice traffic. This means that voice and data LSPs must be given different priorities, because they share bandwidth resources.

Figure 4.8 shows the same network, with voice LSPs at priority 0 and data LSPs at priority 1. (Recall that the best priority is priority 0 and the worst priority is priority 7.) When the voice LSP, LSP3, is established, it preempts one of the data LSPs (LSP2) and establishes over the shortest

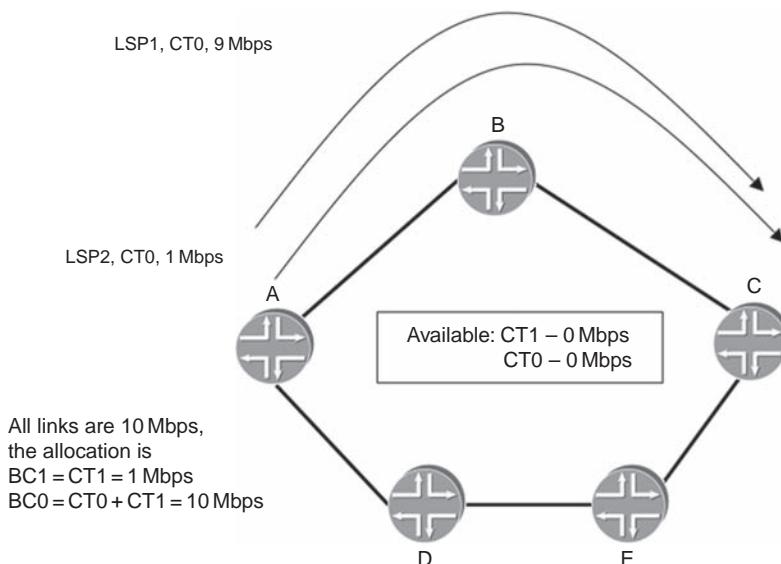


Figure 4.7 Why preemption is necessary when using RDM

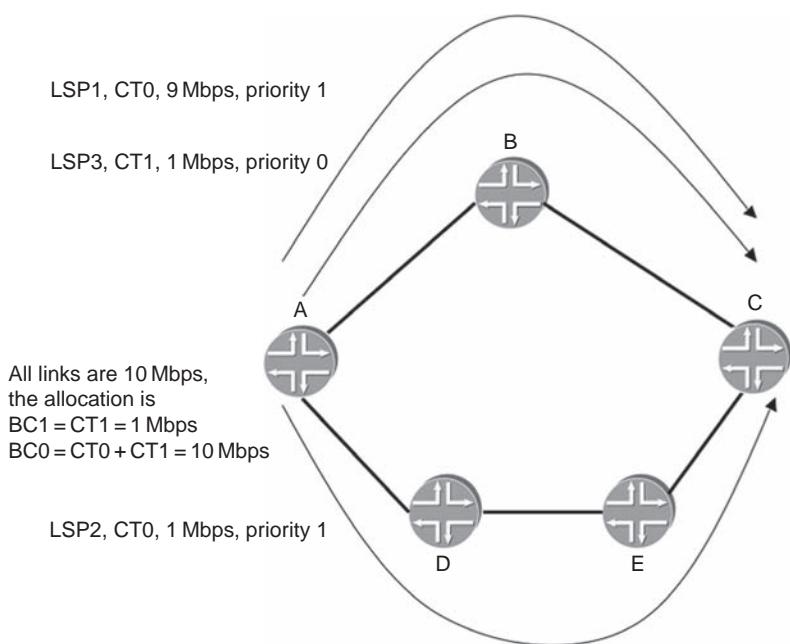


Figure 4.8 When using the RDM model, priorities are necessary to guarantee bandwidth to different CTs

path. LSP2 then reestablishes over the longer path A–D–E–C. Note that a voice LSP can preempt the data LSP only if the voice LSP bandwidth requirement is such that the CT1 allocation on the link is not exceeded. For example, if LSP3 had a requirement of 9 Mbps from CT1, it would not preempt LSP1. This is because the maximum bandwidth that class CT1 is allowed to reserve on any link is 1 Mbps (from the definitions of the BCs). In that case, LSP3 would simply not establish. What the example in Figure 4.8 shows is that the bandwidth-sharing capabilities of RDM come at the cost of extra planning and configuration: LSPs from different classes must be assigned different priorities, to ensure that ultimately each class gets its share of the bandwidth on a link.

The calculation of available bandwidth for the RDM model is a bit more complicated, because it must take into account LSPs at several priority levels and for all the CTs that share the particular BC. For example, the available bandwidth for an LSP from CT0 at priority p is equal to BC0 minus the allocations for all LSPs from all CTs at priorities better or equal to p . Table 4.1 summarizes the differences between MAM and RDM.

It is clear that the BC model plays a crucial role in determining the bandwidth that is available for each one of the TE classes on a link. The

Table 4.1 Comparison of MAM and RDM

MAM	RDM
Maps one BC to one CT, easy to understand and manage	Maps one BC to one or more CTs, harder to manage
Achieves isolation between CTs and guaranteed bandwidth to CTs without the need for preemption	No isolation between CTs requires preemption to guarantee bandwidth to CTs other than the premium
Bandwidth may be wasted	Efficient use of bandwidth
Useful in networks where preemption is precluded	Not recommended in networks where preemption is precluded

BC model and the bandwidth allocation for each BC are advertised by the IGP in the BC sub-TLV. The IETF does not mandate usage of the same BC model on all links in the network. However, it is easier to configure, maintain and operate a network where the same BC model is used, and some implementations require consistent configuration of the bandwidth model on all links.

To summarize, the BC model determines the available bandwidth for each CT at each priority level. MAM and RDM are two possible BC models. They differ in the degree of sharing between the different CTs and the degree of reliance on preemption priorities necessary to achieve bandwidth guarantees for a particular CT. The IGP advertises the BC model and the unreserved bandwidth for the CT-priority combinations corresponding to valid TE classes.

4.4.5 Overbooking

In the discussion so far, LSPs are established with bandwidth reservations for the maximum amount of traffic that is bound to traverse the LSP. However, not all LSPs are carrying the maximum amount of traffic at all times. Thus, even if a link is full from the point of view of existing reservations, there is idle bandwidth on the link. This bandwidth could be used by allowing other LSPs to establish over the link, in effect overbooking it. Several methods exist for implementing overbooking:

1. *LSP size overbooking.* The overbooking is achieved by reserving a lower bandwidth value than the maximum traffic that will be mapped to the LSP.
2. *Link size overbooking.* The overbooking is achieved by artificially raising the maximum reservable bandwidth on a link and using these artificially higher values when doing bandwidth accounting. Note that

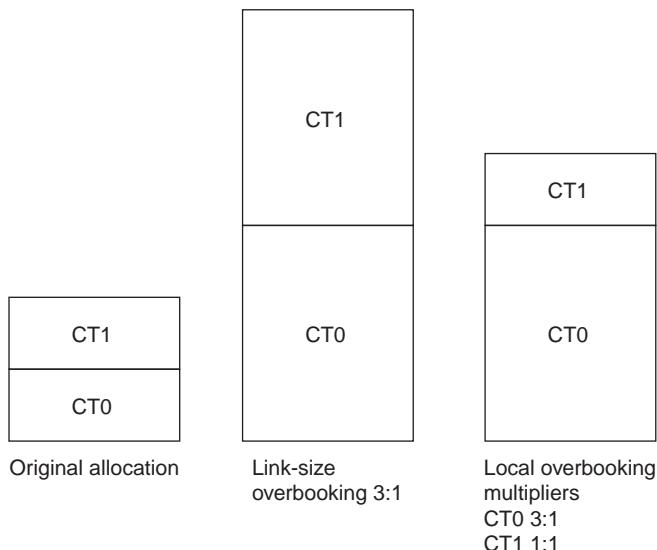


Figure 4.9 Comparison of link size overbooking and local overbooking multipliers

with this approach the overbooking ratio is uniform across all CTs, as shown in Figure 4.9.

3. *Local overbooking multipliers (LOM)*. This refines the link size overbooking method by allowing different overbooking values for different CTs. Rather than ‘inflating’ the bandwidth for all CTs by the same factor, different factors can be used for each CT, e.g. 3:1 overbooking for CT0 but 1:1 overbooking for CT1, as shown in Figure 4.9. The per-CT LOM is factored in all local bandwidth accounting for the purpose of admission control and IGP advertisement of unreserved bandwidths. LOM is tightly coupled to the bandwidth model used, because the effect of overbooking across CTs must be accounted for very accurately (recall that, for example, in RDM bandwidth can be shared across classes). The details of the computation are described in [MAM-LOM] and [RDM-LOM].
4. *Manual configuration of the BC*. This method allows the user to specify the bandwidth constraints and by doing so to overbook a particular class. The drawback of this approach is that it is less intuitive to configure, because it does not translate easily into an overbooking percentage for a particular class.

Overbooking is useful in a multiservice network that will carry a number of different traffic classes where the statistical likelihood of congestion

for each of the traffic classes varies greatly. A typical scenario is a network providing voice and data services. In this case, it is likely there will be high overbooking for the data traffic and no overbooking for the voice traffic.

4.4.6 The DiffServ in DiffServ-TE

In the previous sections, we have seen how network resources are partitioned among different types of traffic and how paths with per-traffic-type resource reservations are set up. In the solution we have presented, the traffic type equates to a desired scheduling behavior, and the available resources for a traffic type are the available resources for a particular scheduler queue. The assumption is that traffic automatically receives the correct scheduling behavior at each hop. This is achieved through DiffServ [RFC2475].

Recall from the Foundations chapter (Chapter 1) that the DiffServ CoS determines the packet's PHB and in particular the scheduling behavior at each hop. In practice, there are two ways to ensure that traffic mapped to a particular DiffServ-TE LSP maps to the correct scheduler queue, as explained in [RFC3270]:

1. *Set the EXP bits appropriately at the LSP ingress (E-LSPs).* Recall from the Foundations chapter that using E-LSPs at most eight PHBs are supported, so this method is good for networks in which less than eight PHBs are required. An important thing to keep in mind is that once the packet is marked with a particular value, its QoS treatment is defined at each hop it crosses. Thus, to ensure consistent QoS behavior, it is imperative to maintain consistent EXP-to-PHB mappings.
2. *Encode the scheduling behavior in the forwarding state (label) installed for the LSP and use the EXP bits to convey the drop preference for the traffic (L-LSP).* The scheduling behavior associated with a forwarding entry is signaled at the LSP setup time. Any number of PHBs can be supported in this way.

A combination of both E-LSPs and L-LSPs can be used in a network, assuming that they can be identified (e.g. through configuration).

Thus, once the traffic is mapped to the correct LSP, it will receive the correct DiffServ treatment. The remaining challenge is to ensure that the mapping is done appropriately. Most vendors today provide mechanisms for picking the LSP based on flexible policies. One of the most intuitive policies is one where the IP DSCP (for IP traffic) or the EXP bits (for MPLS traffic) is used to map the packet to the correct LSP. Other policies may employ BGP communities attached to the route advertisements to pick the LSP. In that case, destinations are tagged with BGP communities, e.g. one

community for a destination that requires EF treatment (such as a game server) and a different community for destinations for which traffic can be treated as best-effort (such as Internet routes). In both cases, traffic must be forwarded to the BGP next-hop, which corresponds to the address of the peer who sent this advertisement. Thus, several LSPs are set up to this destination, one per traffic class. When sending traffic to different destinations, the community is used to pick the LSP. In this way, traffic to the game server can be mapped to an LSP that gives the correct guarantees for EF.

In summary, DiffServ provides the correct scheduling behavior to each type of traffic. Vendors provide flexible policies for picking an LSP that was set up with bandwidth reservations from the correct class type. The combination of DiffServ and per-CT traffic engineering ensures compliance to strict SLAs.

4.4.7 Protection

No discussion on SLAs is complete without looking at the mechanisms available for protecting traffic following a failure. As mentioned at the beginning of this section, the same mechanisms used to protect TE LSPs, discussed in the Protection and Restoration chapter (Chapter 3), can be used for DiffServ Aware LSPs. However, an interesting issue arises in the context of bandwidth protection.

When bandwidth protection is provided, the backup path must reserve bandwidth from the same class type as the protected path. The solution is straightforward for one-to-one backup, because separate protection paths are set up for each LSP. In case of facility backup, there are two options:

- *Single backup.* Use a single backup for the LSPs from all classes and treat all traffic on the backup as best-effort (this implies the backup is set up from CT0). Note that this approach is likely to cause performance degradation.
- *Separate backup per-CT.* Instead of a single backup, there is one backup for each class type and admission control of LSPs into the appropriate backup is performed based on both the class type and the requested bandwidth.

Because different CTs are tied to different guarantees, the operator might choose to reserve backup bandwidth for some classes but not for others or to protect LSPs from some classes but not from others. The ability to provide protection for DiffServ-TE LSPs ensures that SLAs can be guaranteed both under normal conditions and following a failure. However, is all this enough?

4.4.8 Tools for keeping traffic within its reservation limits

The carefully crafted solution presented in the previous sections would all go to waste if more traffic were forwarded through the LSP than the resources that were allocated for it. In such an event, congestion would occur, queues would be overrun and traffic dropped, with disastrous QoS consequences, not just on the misbehaving LSP but on all other LSPs from the same CT crossing the congested link.

One solution is to police the traffic entering the network at the interface between the user and the provider. Another solution is to use LSP policers to prevent such scenarios. LSP policers operate at per-CT granularity at the LSP head end and ensure that traffic forwarded through an LSP stays within the LSP's bounds. Out-of-profile traffic can be either dropped or marked, affecting the QoS of the misbehaving LSP but shielding well-behaved LSPs that cross the same links from QoS degradation, as shown in Figure 4.10. LSP policers make it easy to identify the traffic that needs to be policed, regardless of where traffic is coming from (e.g. different incoming interfaces) or going to (e.g. different destinations beyond the LSP egress point). If the traffic is mapped to the LSP, it will be policed.

LSP policing provides a tool for policing traffic that is forwarded through an LSP. But how can one prevent mapping more traffic to an LSP than the LSP can carry? The answer is admission control. For example, some implementations provide admission control for Layer 2 circuits. A circuit does not establish unless the underlying RSVP LSP has enough available resources, thus avoiding oversubscription. For example, if a new

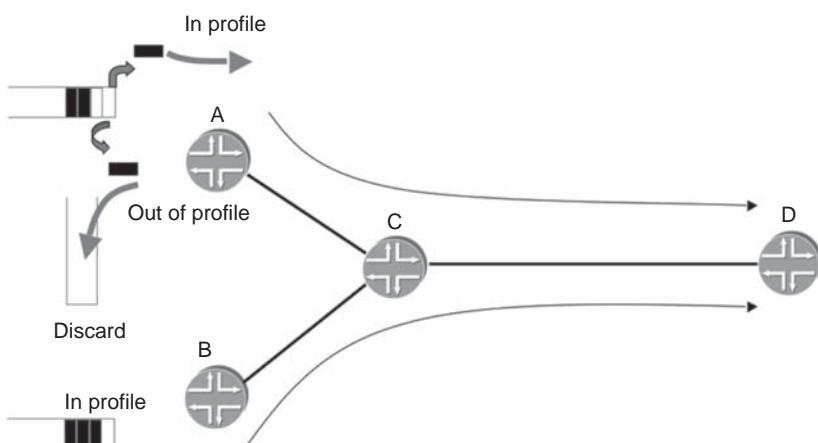


Figure 4.10 Misbehaving source A has its traffic policed and experiences QoS degradation. The well-behaved traffic from B is not affected

Layer 2 circuit requires a 20 Mbps bandwidth, the ingress router identifies an LSP that goes to the required destination that has sufficient bandwidth, and decrements 20 Mbps from the bandwidth available for other potential Layer 2 circuits that may need to use the LSP in the future.

To summarize, LSP policing is a simple tool that ensures that traffic stays within the bounds requested for an LSP. Admission control into the LSPs prevents mapping more traffic to an LSP than the resources allocated for it. By combining admission control with policing, traffic is kept within its reservation limits and QoS can be enforced.

4.4.9 Deploying the DiffServ-TE solution

To summarize the previous sections, the following steps are required to deploy a DiffServ-TE solution:

1. Decide on a BC model and the bandwidth associated with each BC on each link.
2. Configure the buffer and bandwidth allocations on each link to be consistent with step 1 (assuming a model where CTs map to scheduler queues).
3. Decide which CTs and priorities are required.
4. Choose an IGP.
5. Configure LSPs with the desired bandwidth reservation, CT and priority.
6. Configure policers if required.
7. Decide whether the DiffServ treatment will be determined from the EXP bits or the label. If the DiffServ treatment is based on the EXP bits, configure the EXP-to-PHB mappings consistently throughout the DiffServ domain and make sure the traffic is marked correctly. If the DiffServ treatment is based on the label, make sure that all routers have a consistent view of what the PHBs are, so that when the DiffServ treatment is advertised at LSP setup time, it results in uniform behavior on all routers.

Let us briefly look at the migration of a traffic-engineered network to the DiffServ-TE model. As a first step, the network operator must decide which combinations of CTs and priorities are required in the network. Recall from Section 4.4.1 that LSPs with no per-CT requirements are mapped to CT0. Therefore, in a migration scenario, the combinations of CT0 and of the priorities used for TE LSPs that already exist in the network must be selected as valid combinations. The second step is to map the CT-priority combinations selected in the first step to TE classes. Recall from

Section 4.4.2 that the Unreserved Bandwidth TLV is overwritten with the per-TE class information. Network migrations are typically done in stages, so there will be both old and new nodes advertising the Unreserved Bandwidth TLV to each other, but with different semantics. Old nodes will fill in field i of the Unreserved Bandwidth TLV the available bandwidth for (CT_0, i) . New nodes will fill the available bandwidth for TE_i . To provide a consistent picture of the available resources to both old and new nodes, (CT_0, i) must map to TE_i . Such a definition ensures smooth interoperation between nodes that support the DiffServ-TE extensions and nodes that do not.

4.5 EXTENDING THE DiffServ-TE SOLUTION WITH MULTICLASS LSPS

So far we have seen that LSPs set up according to [RFC4124] carry traffic from a single DiffServ class and are set up along a path that satisfies the bandwidth constraints specified for that class. However, sometimes traffic with different DiffServ behaviors must be mapped to the same LSP and the LSP must satisfy the bandwidth constraints for each one of these classes. We will call these multiclass DiffServ Aware LSPs.

An example scenario for multiclass LSPs arises in the context of ATM trunk emulation using MPLS LSPs. To effectively emulate an ATM trunk, all the traffic classes should follow the same path in the network and should exhibit the same behavior in case of failure. If the EF class fails, so should the BE class. If traffic switches to a protection path, it should use the same path for all classes. In principle, one could argue that this behavior can be achieved by setting up a separate LSP for each class and then adding the necessary control-plane intelligence to keep them synchronized. Apart from being cumbersome to implement, such a solution also has drawbacks with regards to the number of LSPs that must be created and maintained.

This brings us to another application of multiclass DiffServ-TE LSPs: reducing the number of LSPs in a network by setting up reservations for several classes in one LSP rather than one reservation per class. When LSPs are set up with bandwidth reservations from a single class, the total number of LSPs in the network is equal to the number of classes times the number of LSPs in the mesh. With multiclass LSPs, the total number of LSPs is equal to the size of the LSP mesh. The reduction in the number of LSPs is important both from a scaling and manageability point of view, as seen in the Traffic Engineering chapter (Chapter 2). From a protection point of view, it also makes sense to protect multiclass LSPs with multiclass protection paths to ensure the SLAs for the traffic following a failure in the network.

Without a solution from the IETF, vendors developed proprietary extensions to the DiffServ-TE solution, in order to support multiclass LSPs. One such solution is documented in [MULTI-CLASS]. In this case, multiple-class types are configured per LSP and the LSP is established only if there is a path that fulfils the bandwidth requirements of each configured class-type.

4.6 CONCLUSION

Differentiated Services (DiffServ) provides QoS by dividing traffic into a small number of classes and allocating network resources on a per-class basis. MPLS-TE enables resource reservation and optimization of transmission resources. MPLS DiffServ-TE combines the advantages of both DiffServ and TE, while at the same time benefiting from the fast-reroute mechanisms available for MPLS.

The result is the ability to set up traffic-engineered LSPs with per-traffic-class granularity and to guarantee resources for each particular type of traffic. Equipment vendors offer mechanisms to map traffic to the appropriate LSPs based on flexible policies, as well as tools for ensuring that traffic stays within the limits of the resources that were reserved for it. Thus, strict QoS guarantees are achieved both for the steady state and the failure cases. Based on the service guarantees that are achieved, service providers can offer services with high SLA requirements, such as voice or migration of ATM/FR on to an MPLS core.

However, as discussed so far, both TE and DiffServ-TE are limited in their scope to a single IGP area and a single AS. In the next chapter, we will see how this limitation can be overcome by Interdomain TE.

4.7 REFERENCES

[MAM-LOM]	draft-ietf-tewg-diff-te-mam-00.txt, older version of the MAM draft, which includes discussion of LOM
[MULTI-CLASS]	I. Minei et al., <i>Extensions for Differentiated Services-aware Traffic Engineered LSPs</i> , draft-minei-diffserv-te-multi-class-01.txt (expired draft)
[RDM-LOM]	draft-ietf-tewg-diff-te-russian-01.txt, older version of the RDM draft, which includes discussion of LOM
[RFC2475]	S. Blake et al., <i>An Architecture for Differentiated Services</i> , RFC2475, December 1998

[RFC3270]	F. Le Faucheur et al., <i>MPLS Support of Diff-Serv</i> , RFC3270, May 2002
[RFC3468]	L. Andersson and G. Swallow, <i>The Multiprotocol Label Switching (MPLS) Working Group Decision on MPLS Signaling Protocols</i> , RFC3468
[RFC3564]	F. Le Faucheur et al., <i>Requirements for Support of Differentiated Services-Aware MPLS Traffic Engineering</i> , RFC3564, July 2003
[RFC3630]	D. Katz, K. Komella and D. Yeung, <i>Traffic Engineering Extensions to OSPF</i> , RFC 3630, September 2003
[RFC3784]	H. Smit and T. Li, <i>IS-IS Extensions for Traffic Engineering</i> , RFC3784, June 2004
[RFC4124]	F. Le Faucheur et al., <i>Protocol Extensions for Support of Differentiated-Service-Aware MPLS Traffic Engineering</i> , RFC4124, June 2005
[RFC4125]	F. Le Faucheur and K. Lai, <i>Maximum Allocation Bandwidth Constraints Model for Diff-Serv-aware MPLS Traffic Engineering</i> , RFC4125, category experimental, June 2005
[RFC4127]	F. Le Faucheur et al., <i>Russian Dolls Band-width Constraints Model for Diff-Serv-Aware MPLS Traffic Engineering</i> , RFC4127, category experimental, June 2005

4.8 FURTHER READING

[RFC2702]	D. Awdanche et al., <i>Requirements for Traffic Engineering over MPLS</i> , RFC2702, September 1999
[Awdanche Jabbari]	D. Awdanche and B. Jabbari, Internet traffic engineering using multiprotocol label switching, <i>Journal of Computer Networks</i> (Elsevier Science); 40 (1), September 2002

4.9 STUDY QUESTIONS

1. Discuss the restrictions on the TE-class mappings on a deployment transitioning from TE to DiffServ-TE, when the LSPs in the TE deployment have setup and hold priorities of 2 and 3.
2. A customer is contemplating a DiffServ-TE deployment where eight CTs are used. Which bandwidth model is better suited for such a deployment?

3. One of the challenges of incrementally deploying DiffServ-TE in a network where TE is already deployed is the risk that the path computation will yield paths traversing nodes that do not yet support the new CTs, thus causing the path signaling to fail. Explore some of the techniques that can be used to avoid this problem.
4. What could be some of the reasons why some implementations require consistent configuration of the bandwidth constraint model on all links?
5. A network is deploying voice and data services. Overbooking by a factor of 3 is desired for data traffic, but no overbooking is desired for voice traffic. Describe how this could be used using (a) LSP-size overbooking or (b) local overbooking multipliers and compare the two approaches.
6. LSP policing is a powerful tool not just for keeping traffic within its reservation bounds, but also for accounting purposes. Give an example where LSP policing can provide per-destination billing, while simple accounting on either the incoming or outgoing interface couldn't.

5

Interdomain Traffic Engineering

5.1 INTRODUCTION

In the Traffic Engineering chapter (Chapter 2), we have seen how to compute and signal traffic-engineered paths that comply with a set of user-defined constraints. A key step in this process is acquiring the information regarding the constraints for all the links in the network. This information is distributed by a link-state IGP and is therefore confined within the same boundaries as the link-state advertisements. Because the visibility of the topology and of the constraints is limited to a single IGP area, TE LSPs dynamically computed by the head end are also limited in the same way. This becomes a problem in large networks that deploy several IGP areas for scalability or in the case of services spanning across several service providers.

In this chapter we will see how RSVP-signaled TE LSPs can extend across IGP areas and across AS boundaries. These solutions are known as interarea TE and inter-AS TE respectively and are referred to collectively as interdomain TE. They apply equally to classic TE and to Diff-Serv Aware TE (described in the DiffServ-TE chapter, Chapter 4). In this chapter the term ‘domain’ is used to denote either an IGP area or an AS.

5.2 THE BUSINESS DRIVERS

The benefits of traffic engineering were discussed in the Traffic Engineering chapter (Chapter 2). Providers use traffic-engineered paths for

optimization of network resources, support of services with QoS guarantees, fast reroute and the measurement of the aggregated traffic flow between two points in the network. To achieve these functions in large networks with multiple IGP areas, the LSPs used for traffic engineering need to cross area boundaries (interarea LSPs).

Interdomain LSPs¹ are not limited to traffic engineering; they are also pivotal to the deployment of services spanning across different geographical locations. These can be services requiring assured bandwidth, such as connection of voice gateways, or they may be applications that rely on the existence of an MPLS transport tunnel, such as pseudowires or BGP/MPLS Layer 3 VPNs. When the service spans several IGP areas, the LSP is interarea; when it spans different ASs, the LSP is inter-AS.

Inter-AS LSPs exist both within the same provider and across different providers. Multiple ASs can be present within a single service provider's network, e.g. following the acquisition of another provider's network in a different geographical location. The separate ASs are maintained for reasons ranging from administrative authority to the desire to maintain routing protocol isolation between geographical domains and prevent meltdown of the entire network in the event of a local IGP meltdown.

A useful application of LSP establishment across provider boundaries is the interprovider option C of BGP/MPLS Layer 3 VPNs (discussed in detail in the Hierarchical and Recursive L3VPNs chapter, Chapter 9). The inter-AS RSVP LSP brings two benefits: (a) the ability to traffic-engineer the path between the remote PEs and (b) the ability to simplify the configuration by not having to rely on BGP to 'glue' the LSP segments for setting up LSP between the remote PEs.

Figure 5.1 shows an interprovider VPN setup, where two customers, VPNa and VPNb, have sites attached to PE3 and PE4 (in different ASs). The loopback addresses of PE3 and PE4 are advertised as VPN routes to ensure connectivity between the PEs. Once the addresses of PE3 and PE4 are known, an External Border Gateway Protocol (EBGP) session can be set up between the two PEs and the VPN routes of the two customers, VPNa and VPNb, are exchanged over this session. Forwarding traffic to these addresses requires an LSP between PE3 and PE4. When discussing interprovider VPNs, we will see how this is done using BGP. If an end-to-end RSVP LSP was available, it could be used instead.

So far we have seen why interdomain LSPs are important. Next, we will look at how they can be set up.

¹ Recall that 'domain' is used in this chapter to denote either an area or an AS.

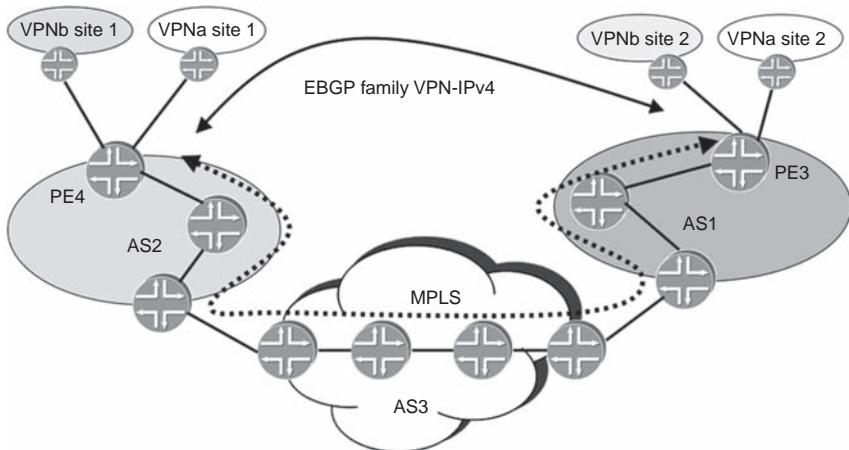


Figure 5.1 An inter-AS LSP can be used in the setup of an interprovider VPN

5.3 SETTING UP INTERDOMAIN TE LSPS

As discussed earlier in this chapter, the limitation of a TE LSP to a single IGP area is caused by the limited visibility into the topology at the LSP head end. However, once a path is specified, there is no problem signaling it across IGP areas or across ASes. Therefore, the setup of interdomain LSPs is possible without any extensions, e.g. by computing the path offline and specifying the hops at the head end. The problem with this approach is that it forces a service provider to move to an operations model relying on offline computation for both the primary and secondary paths, with the implications discussed in the section on offline computation in the TE chapter.² In addition, the issue of fast reroute is not addressed in this model unless the bypass tunnels protecting the interdomain links are also computed offline.

As TE and MPLS-based applications started gaining traction in the industry, more scalable and optimal solutions than the simple setup of an LSP across domain boundaries (whether area or AS) became necessary. The requirements for interdomain traffic engineering [RFC4105] [RFC4216], were developed in the TEWG³ in the IETF. The solutions are currently being developed in the CCAMP and the PCE Working Groups

²In the case of a multiprovider environment the offline tool would also need to know the TE information of all the links of all the providers involved. That may require, among other things, that one provider discloses its internal topology to another provider, a not very attractive prospect in many cases.

³The TEWG has in the meantime completed its work and has been closed.

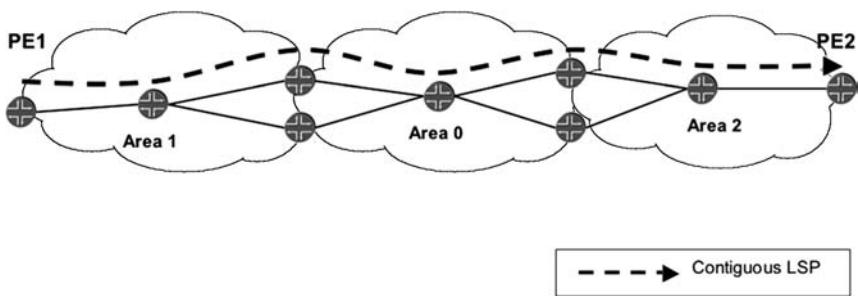


Figure 5.2 Setting up an interarea LSP using the contiguous LSP method

[CCAMPWG] [PCEWG] and at the time of this writing are still works in progress.

Setting up RSVP-signaled TE LSPs across IGP domains is done in three steps: (a) discovering reachability and TE information, (b) computing the path and (c) signaling the LSP. However, only the latter two are modified for interdomain operation. The propagation of reachability and TE information cannot be changed to cross IGP boundaries, because this would severely impact the scalability of the IGPs.⁴ For this reason, information distribution is not discussed further. Instead, the following sections focus on path computation and path signaling. For clarity of the discussion, the setup methods for interdomain LSP setup are discussed first.

5.3.1 Path setup

It may seem like a strange thing to start the discussion on interdomain TE from the setup instead of the path computation. The reason for doing so is because examining the different setup methods makes it easier to understand the choices that must be made with regards to path computation. There are three methods for setting up LSPs across domain boundaries.

5.3.1.1 *Contiguous LSP*

In this case, an end-to-end LSP between PE1 and PE2 is built across domain boundaries, using hop-by-hop signaling between adjacent neighbors. This method is the most intuitive, because it resembles exactly the setup of a TE LSP within one domain. Figure 5.2 shows the setup of an interarea contiguous LSP.

⁴ It may also not be feasible for the reason of privacy discussed before (as each provider may not want to disclose its internal topology to other providers).

5.3.1.2 LSP stitching

The end-to-end LSP between PE1 and PE2 in Figure 5.3 is built from several smaller LSPs (called TE LSP segments) that are set up in the different domains and ‘stitched’ together at locations called ‘stitching points’ [RFC5150]. This patching together of segments is accomplished by installing the forwarding state that takes traffic reaching the endpoint of one LSP segment and maps it into the next LSP segment. A 1:1 mapping is enforced between the segments in the different domains, meaning that traffic that is mapped into any LSP segment is guaranteed to be coming from a single LSP segment. Railway cars are a useful analogy for LSP segments. They can be stitched together to allow traffic (people) to pass from one car to another and there is a 1:1 mapping between the segments because the stitching point connects exactly two cars.

Figure 5.3 shows a stitched LSP crossing three IGP areas. Separate TE LSP segments exist in each area (in this case spanning between the area border routers) and are stitched together at the ABRs to form one LSP. If in this example a second LSP were to be set up between the same endpoints, new TE LSP segments would have to be created in each domain, because a segment can participate only in a single end-to-end LSP. Thus, the amount of state created and maintained in a transit domain grows proportionally with the number of LSPs crossing it.

There are several important things to note about TE LSP segments that influence the properties of an end-to-end LSP set up using the stitching method:

1. *Scope.* By definition, TE LSP segments span a single domain. This means that the computation of their path is limited to the domain and that functions such as reoptimization and fast reroute are also confined in the same way. The ability to perform these operations locally is a useful property of the stitching solution, as will be seen in later sections.

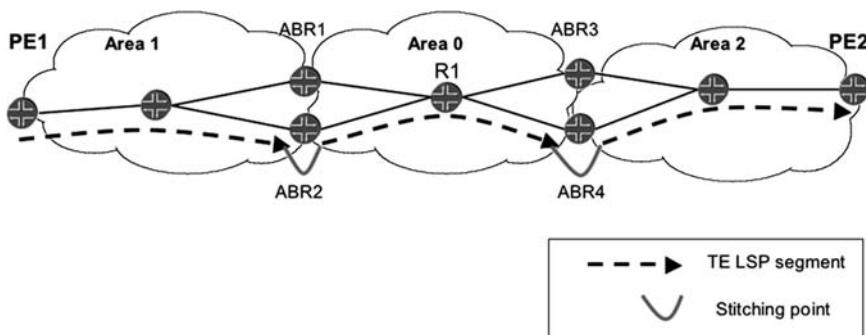


Figure 5.3 Setting up an interarea LSP using the LSP stitching method

2. *Endpoints.* TE LSP segments have a head end and a tail end, just like normal LSPs. These are usually the border routers at the entry into and exit from the domain, but can be other routers as well, depending on the topology and the provisioning used. In the section discussing LSP computation we will see why it is convenient to set up the TE LSP segments between border routers.
3. *Setup trigger.* TE LSP segments may be preconfigured or their setup may be triggered by the arrival of an LSP setup message from a neighboring domain.

Thus, LSP stitching creates an interdomain LSP from several segments with per-domain scope. However, because any segment can be part of only a single LSP, the state created in transit domains increases with each transit LSP. LSP nesting solves this scalability limitation.

5.3.1.3 LSP nesting

An end-to-end LSP between PE1 and PE2 is tunneled inside an LSP with per-domain scope as it crosses the domain, creating a hierarchy of LSPs and hiding the details of the end-to-end LSP from the routers in the transit domain [RFC4206] [RFC4726]. This process is called ‘nesting’, because one LSP is placed into another one. The LSP that acts as the ‘nest’ or container for other LSPs is called the Forwarding Adjacency (FA) LSP. LSP1 and LSP2 in Figure 5.4 are both end-to-end LSPs crossing three IGP areas. In the middle area, LSP1 and LSP2 are both nested into an FA LSP that spans between the area boundaries.

Nesting is accomplished by using label stacking. At the head end of the FA LSP, the label of the FA LSP is pushed on top of the label stack

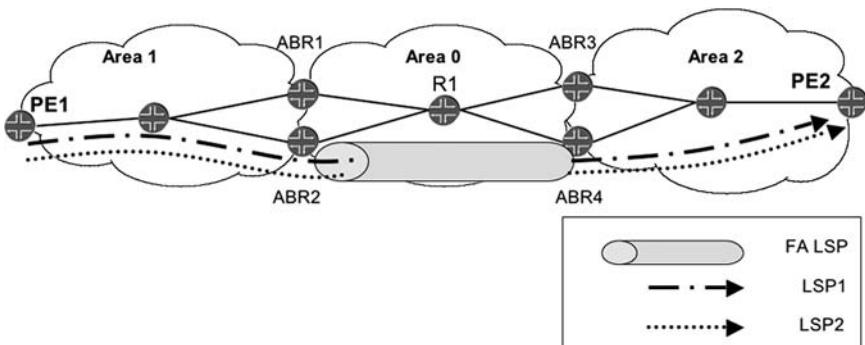


Figure 5.4 Setting up an interdomain LSP using the LSP nesting method. Several LSPs can be nested onto the same FA LSP

of the nested LSP, in a process similar to the one described for bypass tunnels in the protection and restoration chapter. Forwarding proceeds based on the top label only and routers within the FA LSP's domain are not required to maintain any state for the transit LSPs. In the example setup from Figure 5.4, router R1 is not aware of the existence of LSP1 or LSP2.

More than one transit LSP can be nested into the same FA LSP. Figure 5.4 shows two LSPs, originating and terminating on different PE routers, that share the same FA LSP in the transit area. Thus, nesting uses a 1:N mapping between the FA LSP and the transit LSPs.

LSP nesting brings the scaling advantages of LSP hierarchy to interdomain TE: no state needs to be maintained for the interdomain LSPs inside the transit domain and the setup and teardown requests for the nested LSPs do not place any additional burden on the routers inside the transit domain. FA LSPs also allow for easier enforcement of policies for LSPs traversing the domain. One example is the ability to control the links used by transit LSPs by limiting the links used by the FA LSP. Another example is the ability to perform admission control for LSPs traversing the domain by simply looking at the available resources on the FA LSP rather than checking each and every link that the traffic would traverse.

FA LSPs share the same properties as TE LSP segments from the point of view of scope, endpoints and triggers for their setup. Thus, the useful properties of domain-local path computation, reoptimization and repair also apply to FA LSPs and end-to-end LSPs set up with stitching and nesting share similar properties.

The main difference between nesting and stitching is in the amount of state that is created in the transit domain. Stitching requires individual segments for each transit LSP while nesting can share a single FA LSP, yielding a more scalable solution. The natural question is why bother with stitched LSPs at all? To answer this, recall that interdomain LSPs are often used for traffic engineering. Let us take a look at an end-to-end LSP with a certain bandwidth requirement. When the LSP is set up using stitching, the bandwidth requirement can be easily satisfied by ensuring that all the TE LSP segments are set up with the correct bandwidth allocation. In contrast, when the LSP is set up using nesting, the same approach does not automatically work, because any number of LSPs may be mapped into the same FA LSP. To ensure adequate resources for all transit LSPs there is a need to perform admission control into the FA LSP. In addition to the admission control, one may also need to perform traffic policing at the entrance to the FA LSP, especially if such an entrance is on the administrative domain boundary between two providers.

Having seen the different LSP setup methods, the natural question is how the setup method is picked at the domain boundary. The answer is that it is chosen based on administrative policies locally configured at

the border router.⁵ This implies that a single end-to-end LSP may be set up using different methods in different domains: for example, it may use nesting in one domain and stitching in another. This should not come as a surprise, especially when one thinks of the interprovider case. How the LSP is set up within each domain should be a local decision.

Regardless of the setup method used, the path of the LSP must be computed. The following section discusses the challenges of computing the path for an interdomain LSP and the different methods for performing the computation.

5.3.2 Path computation

The main limiting factor for an interdomain⁶ path computation is the visibility that the computing node has into the topology. This influences both the scope of the computation (per-domain or interdomain) and the ownership of the computation (which element is performing the computation).

1. *Scope of the computation.* The scope of the computation is limited by the visibility that the computing entity has into the topology. This is true irrespective of the owner of the computation. Therefore, it is either confined to a single domain (per-domain path computation) or it spans multiple domains (interdomain path computation, also referred to as end to end).
2. *Ownership of the computation.* The entity performing the computation can be an offline tool, the LSR at the head end, a domain-boundary node or another element (such as the path computation element, which is discussed in more detail in the following sections). The visibility that the computing entity has into the topology affects its ability to perform the computation.

At first glance it may seem that the LSP setup method dictates the scope of the computation and therefore also implicitly determines which element has enough data to perform the computation. Wouldn't the setup of a contiguous LSP require that the path computation span its entire path? And if an interdomain computation is indeed required, wouldn't it have to be performed by an entity with global visibility across all domains? The answer is 'no', as will be seen in the following example discussing the setup of an interdomain LSP using the contiguous signaling method.

⁵ When the head end requires the setup of a contiguous LSP, it can explicitly signal this desire using a flag in the Session Attribute Object. In all other cases, the signaling is based on local administrative policies.

⁶ The IETF documents discussing path computation use the term 'domain' to denote either an area or an AS. For this reason, the same terminology is used here.

It is very intuitive to think of a contiguous LSP setup where all the hops in the path are precomputed and then signaled with the Explicit Route Object (ERO). In this case, the path computation must have interdomain scope and therefore must be performed by an entity that has interdomain visibility, such as an offline tool.

A less intuitive, but perfectly valid, way of setting up the same end-to-end LSP is to perform the path computation separately within each domain. Assuming that the exit points out of the domains are known⁷ or can be determined by some means, a path can be computed up to the border router at the domain exit. Thus, the path is not known in its entirety at the LSP head end. Instead, as the LSP is signaled, at the entrance to each domain the path to the next border router is computed and added to the ERO (this process is called ERO expansion). Using this approach, the path computation is always limited in scope to a single domain and the path is computed piece by piece as it traverses the different domains. The computation may be performed by the domain boundary nodes or it may be obtained through other means, as we will see in the following sections.

The above example illustrates a fundamental point regarding interdomain TE, namely that the path computation can be performed either interdomain or per-domain, regardless of the signaling method used for the LSP setup.

The discussion so far focused on finding a path for the LSP across the different domains. However, remember from the introduction that one of the main requirements for the interdomain solution was support for TE. It is important to understand that regardless of whether the path is computed per-domain or interdomain, the assumption is that the traffic engineering characteristics of the LSP are uniformly maintained across all domains. This implies a common understanding of the LSP's constraints across all domains. The problem is that different domains may be under different administrations and therefore their local definition of DiffServ-TE class types, as discussed in the DiffServ-TE chapter (Chapter 4), or link properties may not be compatible. For example, the class type (CT) suitable for voice traffic may be CT1 in one AS and CT3 in another, or the link color for high-latency links may be X in one domain and Y in the neighboring one. For this reason, when the path computation crosses from one domain to the next, the constraints must be translated appropriately, e.g. through a mapping. Note that this implies that the administrations of the two domains must cooperate by exchanging the relevant information and agreeing on such a mapping. This is particularly true in the inter-provider case, where it is very likely that different constraints are used.

⁷ The exit points out of the domain (border routers) can be configured as loose hops in the ERO or they may be discovered based on the IP reachability information for the LSP's destination address.

Thus, when talking about ‘visibility’ into neighboring domains, both the topology information and the TE characteristics of the topology (or the appropriate mapping) must be known.

Given a common understanding of the constraints, the interdomain computation assumes visibility into all the domains in the path, but does not introduce any new requirements. However, the per-domain computation raises interesting challenges.

5.3.2.1 *Per-domain path computation*

Per-domain path computation is performed when there is no visibility across all domains at any one single central point, irrespective of the owner of the computation. For this reason, the computation is performed separately within each domain, from one border router to the next, each such computation bringing the path one domain closer to the final destination. The assumption is that the address of a border router on the path to the LSP destination is known. The border router is either configured as a loose hop in the path or it is discovered dynamically based on the IP reachability for the LSP destination address. The result of the computation is a path to the border router. How this path is used depends on the LSP setup method. For contiguous signaling, it can be used during ERO expansion; for stitching and nesting, it can be used to set up or select the relevant TE LSP segment or FA LSP.

Thus, when using per-domain computation, the path is traffic engineered separately within each domain rather than being traffic engineered end to end. However, the fact that each piece in the path is optimal does not necessarily mean that the entire path is optimal.

Figure 5.5 shows an example of how the complete path can be nonoptimal. The goal is to set up a shortest-path inter-AS LSP from A to B, with a bandwidth reservation of 100 Mbps. There are two inter-AS links and both exit points are equally good from a routing point of view. All links are of the same capacity, 100 Mbps. However, link ASBR3-B in AS2 has no available bandwidth because LSP2 is set up over it, with a 100 Mbps bandwidth requirement. In this case, the optimal path is A–ASBR2–ASBR4–B. However, from the viewpoint of AS1, both ASBR1 and ASBR2 appear to be valid, optimal options. If A chooses ASBR1 as its exit point, then A–ASBR1–ASBR3–ASBR4–B is the most optimal path that can be found (it is, in fact, the only feasible path, so it is the ‘best’ one that meets the constraints). Although the computation is optimal within each domain, the end-to-end path is not optimal.

This example also raises an interesting question with regards to the information distribution within a single domain. Imagine that congestion occurs on the inter-AS link ASBR1–ASBR3 rather than on ASBR3–B. The

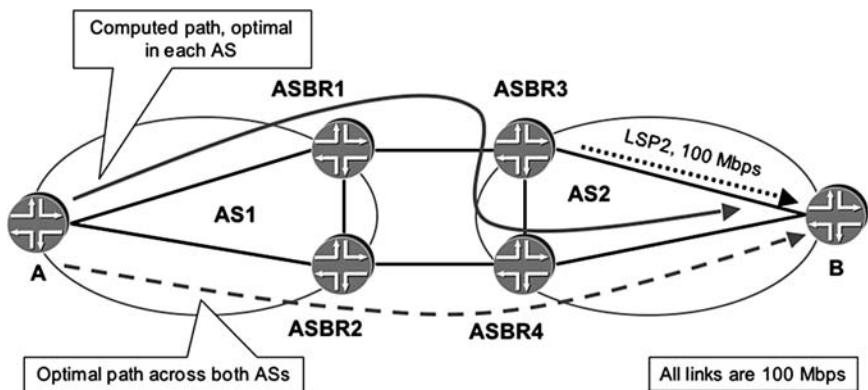


Figure 5.5 Nonoptimal resource utilization when using per-domain path computation

problem is that the inter-AS link is not in the TED, so the congestion on the link is not detected until the path setup request is received and admission control fails. One way to alleviate this problem is to inject the TE information for links on domain boundaries (such as the ASBR–ASBR link) into the IGP TED, to improve the accuracy of the computation and minimize failures at the LSP setup time.

However, this approach cannot guarantee that admission control will succeed when the LSP is actually signaled. Of course this is no different from any other links in the TED and true for any computation method and any signaling method. The question, therefore, is how are LSP setup failures handled in the case of interdomain LSPs?

5.3.2.2 Crankback

The previous example showed an LSP setup failure caused by a computation based on inaccurate TE information. However, even if the computation is perfectly accurate, the LSP setup can still fail if, between the time the path was computed and the time that the path was signaled, one of the resources becomes unavailable (e.g. due to the setup of another LSP). The Traffic Engineering chapter (Chapter 2) describes how this situation is handled when TE is confined to a single domain. In this case, the node where admission control fails sends a path error message to the head end, indicating the location of the failure. Based on this information, the head end can compute a new path that avoids the problematic resource. In addition, the updated resource information may be advertised by the IGP to

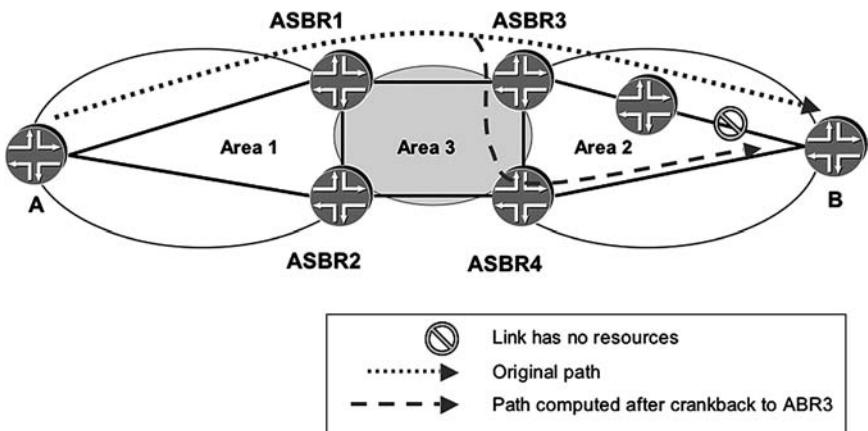


Figure 5.6 Using crankback to deal with path setup failures

ensure that inaccurate computations are not performed by other nodes in the network.

This solution assumes that the LSP head end can use the failure information in a meaningful way when computing the new path. This may not be the case when the path crosses several domains into which the head end does not have visibility. Figure 5.6 shows a network with three IGP areas (which are labeled areas 1, 2 and 3 for the sake of clarity). An LSP must be established between node A in area 1 and node B in area 3. Imagine that the path computation yields the path A–ABR1–ABR3–R1–B and the setup fails because of unavailable resources on link R1–B (we will call link R1–B the blocked resource). In this case, propagating the failure to the LSP head end is not very useful, as there is not much that it can do with the information identifying the blocked resource as link R1–B. Instead, it makes more sense to forward the failure information to the first node that can put it to good use, in this case border router ABR3, which can look for an alternate path within area 3 that avoids the offending link (e.g. ABR3–ABR4–B).

However, what if border router ABR3 cannot find such a path? In the example, this can happen if the link ABR3–ABR4 along the path ABR3–ABR4–B does not have enough resources. In this case, ABR3 is treated as the blocked resource and an error is forwarded to the first router that can use this information in a meaningful way, border router ABR1. What is effectively happening is that the computation is cranked back one computation step at a time, away from the failure point. This process is called crankback and is a popular technique in TDM-based networks.

Crankback is a natural fit for LSPs made up of nested or stitched segments. When there is a setup failure in one domain, rather than recomputing the entire LSP the computation can first be redone in the failed

domain. If the computation fails, the error is reported to the domain upstream and a path to an alternate border router or alternate domain can be evaluated. This local repair of the path computation shields the LSP head end from recomputation requests caused by failures in domains over which it has no control.

The desire to shield upstream domains from unnecessary computations is one of the main goals of crankback. However, containing the computation within a particular domain is not enough. In the previous example, imagine that there is no feasible path within area 3 and that the computation has been cranked back to border router ABR1 in area 2, as shown in Figure 5.7. At this point, any setup request from ABR1 will fail. What is to stop border ABR1 from continuously toggling between the two blocked resources and trying to set up paths through ABR3 and ABR4 alternatively? What is needed to avoid such a situation is a way to inform ABR1 that there is no point in trying to continue the search and that it should crankback the computation. Thus, two pieces of information must be carried in the failure notification: the location of the failure and whether to continue the search or crankback. In addition to this mechanism, routers can maintain a history of failed computation attempts to improve the accuracy of computations, and a hard limit can be set for the recomputation attempts of any path.

Note that crankback does not provide any guarantees regarding the time it takes to find a path. Furthermore, because of its per-domain nature, it cannot ensure optimality of the path either. In fact, because of the limit imposed on the recomputation attempts, crankback cannot even ensure that any path will be found. Having said all this, it may seem that crankback is

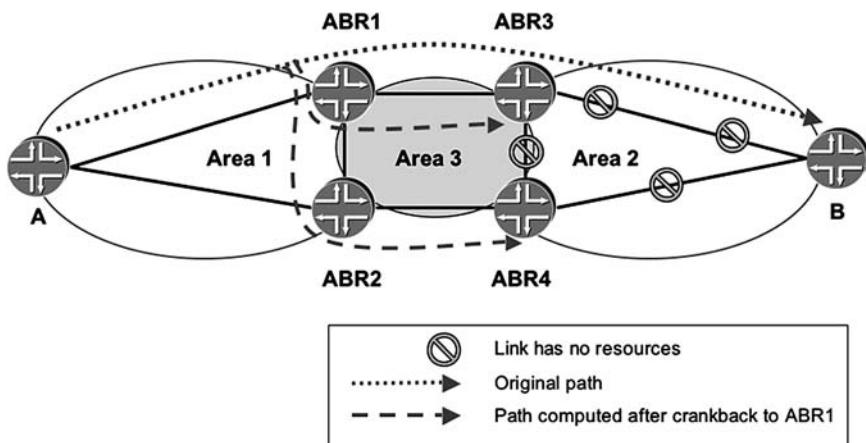


Figure 5.7 Crankback when no feasible paths exist in a downstream domain

not a very good idea at all. However, remember that crankback does provide an efficient solution in nonpathological situations. In an engineering world, an imperfect solution may sometimes be better than no solution at all. As with anything else, the decision whether to use crankback becomes a question of whether the benefits outweigh the costs for a particular deployment. For example, think of an LSP crossing 10 ASs across the globe, when there is some congestion in the destination AS. Without an attempt to local repair, the errors would need to be propagated back all the way to the head end LSR, 10 ASs away.

As a protocol, RSVP lends itself readily to crankback signaling because it already includes the failure notification. The IETF defined further extensions to RSVP for crankback support [RFC4920]. In particular, the ability to do crankback and the node that should perform the recomputation (head end, border router or other router in the path) can be signaled using flags in the Session Attribute Object in Path messages. New objects are added to carry more detailed failure information in the Path Error message issued when the LSP setup fails.

Although by no means perfect, crankback is an important tool when the path is computed separately within each domain. From the discussion so far, it may seem that the entity performing the computation is always one of the routers in the network. However, in an interdomain setup, the entity performing the computation may need more information than is available to a router in the network. For this reason, a path computation element may be used.

5.3.2.3 *Path computation element (PCE)*

Recall from earlier discussions that path setup across domain boundaries is not a problem when the path is specified in its entirety at the head end. For this reason, it is tempting to want to offload the path computation to an all-knowing, all-seeing entity that can deliver the necessary information on demand. Path computation elements (PCEs) first started out as such entities, and thus in many people's minds became inextricably tied to the interdomain solution. From the discussion so far it should already be clear that this is not the case. In the previous sections we have talked about setting up LSPs using per-domain computation without the aid of PCEs, but the same computations performed by the border routers could have been performed by PCEs in each domain.

Let us take a look at a few of the challenges of doing path computation in an interdomain setup, as this will shed some light on the benefits of PCEs:

1. *Constraint communication.* Recall from the Traffic Engineering chapter (Chapter 2) that not all the constraints are signaled when the LSP is set

up. The assumption is that some of the constraints (e.g. link colors) need to be known only at the head end, which is the node doing the path computation. This assumption breaks in an interdomain setup where the computation is performed per-domain and the constraints must be communicated to all nodes participating in the path computation.

2. *Need for extra visibility.* Intuitively, it is easy to think of an interdomain LSP as one whose head end and tail end are in different domains. However, this may not always be the case. For example, for optimization or policy reasons it may be necessary to set up an LSP whose head end and tail end reside in the same domain but crosses into a different domain at some point along its path. To compute the path of such an LSP, more information is required than is available in a single domain.
3. *Constraint translation.* Earlier in this chapter we discussed the issue of translating constraints such as class types or priorities at domain boundaries. Although mappings may solve the problem, a different approach is to have a different entity do the translation.
4. *Optimality of the solution.* For an optimal solution, it may be necessary to run more specialized algorithms than CSPF or to have more information than just the resource availability, as discussed in the Traffic Engineering chapter.

From looking at these requirements, it looks like an offline path computation tool could almost fit the bill. However, thinking of the way the path is used during LSP setup, it is clear that the computing entity should be able to service requests dynamically in real time. The notion of a clever entity that can compute paths based on more information than what is available to a router in the network, using more sophisticated algorithms than simple CSPF, was first introduced in the early interdomain drafts. The entity was called the path computation element (PCE). Its operation and protocols are being defined in the PCE Working Group [PCEWG] in the IETF and will be briefly presented in the remainder of this section.

Let us start by looking at how the PCE is envisaged to operate in the interdomain setup. The PCE can either be a module on the router or a separate entity that the router can communicate with. Its role is to service path computation requests. The resulting path may be confined to a single domain or may cross several domains. For example, when setting up an interdomain path using the ERO expansion technique, the border router can query a PCE for the path to the next loose hop instead of simply running CSPF on the router itself. To compute such a path, the PCE must have at least the same information as is available in the TED, but to provide added value it may store additional information not available to the routers (such as global knowledge of all LSPs set up in the domain). Other ways in which the PCE can improve the path computation are by running more

sophisticated algorithms than CSPF and by collaborating with other PCEs to find the most optimal path across several domains.

Based on the description of the PCE operation and requirements, let us list some of the PCE-related functionality that is currently being standardized in the PCE Working Group:

1. *PCE communication protocol.* The PCE services path computation requests from its path computation clients (PCCs). When the PCE and PCC are not collocated, a communication protocol is needed to pass the computation requests and replies back and forth between the two entities. The PCE communication protocol (PCEP) [RFC5440] handles this interaction. PCEP is a TCP-based protocol using RSVP-like objects to carry information required for the path computation, such as bandwidth, and the result of the path computation in an ERO-like format. In addition to PCE-PCC communication, the protocol also supports PCE-PCE communication. Inter-PCE collaboration may be required for tasks such as computing an end-to-end path or for minimizing the risk of having to run crankback when computing per-domain paths. In such cases, several PCEs collaborate in the computation of an end-to-end path.
2. *PCE autodiscovery.* The question of how a PCC finds out which PCE to query seems a simple one: why not simply configure the address of the PCE? A statically configured PCE becomes cumbersome to maintain in a large network and does not solve the problem of switching to a backup PCE when the primary fails or is too busy servicing other requests. Furthermore, in a network where multiple PCEs are available, the PCC may need additional information such as the capabilities, scope or availability of the PCE to help in the selection of one of the PCE from the possible candidates. Similar to the RSVP automesh solution discussed in the Foundations chapter (Chapter 1), the automatic PCE discovery is done using extensions to IS-IS and OSPF defined in [RFC5089] and [RFC5088], respectively.
3. *Acquiring the TED database.* The TED is the minimum information that the PCE must have in order to provide meaningful computation. For the computation to be as accurate as possible, the TED on the PCE must be at least as accurate as the one on the router. This is not a problem when the PCE is part of the router. For PCEs that are separate entities, the TED can either be built dynamically by ‘sniffing’ the IGP advertisements or it can be requested from the routers. Sniffing IGP advertisements implies that the PCE is part of the network (which in turn means that the operator must qualify the equipment appropriately). Synchronizing large TED databases in an efficient and reliable way requires an appropriate database exchange protocol built into both the router and the PCE, at the time of this writing, such a mechanism had not been standardized.

4. *Stateful versus stateless PCE.* Clearly if the PCE has a global view of all the current reservations, its computation can be much more accurate. Maintaining extra information also allows the PCE to perform more interesting functions such as computing disjoint paths for the primary and secondary or avoiding links that failed in a previous computation. For these reasons, a stateful PCE is attractive. The price for doing so is not just maintaining large amounts of information on the PCE but also synchronizing this information between several PCEs in the network (either between the primary and the secondary or between collaborating PCEs) and possibly maintaining this state across failures.
5. *Computation algorithms.* One of the earliest and least controversial decisions of the PCE Working Group was that the path computation algorithms used in the PCE are not an area of standardization. Instead, they are left to the individual implementations as a differentiator between vendors. Although the algorithms themselves are not standardized, the objective functions are. The objective functions correspond to the optimization criteria used in the computation, for example number of hops vs. bandwidth used is desired. A core of six fundamental objective functions are defined as part of the generic PCEP requirements spelled out in [RFC4657].⁸ Both the speed of the computation and the quality of the result are important when evaluating PCE performance and extensions to the PCEP protocol were made to monitor these in [PCE_MON].

The notion of PCE is not foreign either to vendors or to large network operators. Some large operators have home-grown tools that provide a lot of the functionality required by the PCE (such as gathering TE information from the network or computing paths) and that are used as offline computation tools. Therefore, the standardization work is driven equally by service providers and vendors.

The PCE is a tool that can be used to improve and ease path computation, both within a single domain and across domains. Although PCEs are often equated with interdomain solutions,⁹ they are not a requirement, regardless of whether the computation is done per-domain or interdomain.

So far we have described the different path computation methods. It is important to understand that the path computation methods can be used with the different path setup methods and are not tied to the LSP setup method.

⁸In fact, at the time of this writing, new functions were being standardized in the working group.

⁹It is important to keep in mind that PCEs are not specific to interdomain solutions. In Chapter 6 we will discuss the use of PCEs for computing P2MP LSPs.

5.3.3 Reoptimization

Reoptimization refers to the process of finding a more optimal path for an LSP and moving to it in a seamless fashion. The trigger for doing so may be an operator request, the expiration of a timer or the availability of updated information for the path computation. The point of reoptimization is to move the LSP to a better path if such a path can be found. In the Traffic Engineering chapter (Chapter 2) we saw how this can be done in a make-before-break fashion by setting up the LSP along the new path before tearing down the old path.

The important thing to understand with regards to reoptimization is that it is done in two steps: path computation and path signaling. Within a single domain, reoptimization is driven by the LSP head end and requires recomputation and resignaling of the entire path. For interdomain LSPs the situation is different: both the path computation method (per-domain or interdomain) and the signaling method (contiguous, stitching or nesting) influence how reoptimization happens.

When per-domain computation is used, it is possible to compute a new path in just one domain without disturbing segments in other domains. If, in addition, the LSP is set up using stitching or nesting, it is also possible to signal the new path within the domain without disturbing the head end or other domains. Thus, the entire reoptimization process is contained within a single domain. This is important for two reasons:

1. *Locality.* Remember that the reasons for reoptimization are usually local ones: new TE information or a decision on the part of the operator. A local decision in one domain should not impact the neighboring domain. This is especially true for interprovider situations, where the administrative decision to perform optimization of a segment in one domain should not create control plane operations (and thus load on the routers' CPUs) in the neighboring domain.
2. *Scalability.* Containing the optimization work to the domain where it was triggered is important for scalability. The head end does not need to be aware of the path changes happening in domains downstream from it and does not have to be involved in carrying out the reoptimization process (make-before-break) for every event in a domain several AS hops away. This approach also shields intermediate domains from the extra activity that would be triggered were the head end to initiate the reoptimization. Therefore, the ability to contain reoptimizations to a single domain is important for scalability.

However, is per-domain reoptimization always used? The answer is 'no'. In some cases, per-domain reoptimization is not desirable. For example, if an LSP is set up with tight constraints, allowing local reoptimization

can cause violation of the constraints. This is similar to the situation in Figure 5.5, where locally optimal paths yield a nonoptimal end-to-end result. In other cases, per-domain reoptimization is not possible; e.g. if the LSP is set up as a contiguous LSP. In such cases, the head end must be involved in the reoptimization process. There are two questions to be answered: is it possible for the head end to initiate reoptimization requests for nodes downstream and is it desirable to allow it to do so?

The answer to the first question is straightforward, because it involves only the mechanics of signaling. RSVP can be extended to provide the necessary signaling capabilities [RFC4736]. The head end can signal a reoptimization request to the nodes that perform the per-domain computation, using a bit in the path message and, conversely, these nodes can inform the head end that a better path is available using a path error message.

The answer to the question of whether such a mode of operation is desirable is not as straightforward. In an interarea setup it might be acceptable to hand over control to the head end LSR, but in an inter-provider scenario (as seen earlier) it might not be desirable to do so.

To summarize, the reoptimization of a contiguous LSP requires head end intervention, while for stitched/nested LSPs the process can be restricted to the routers in the domain where the path is optimized. Thus, the LSP setup method impacts the scaling properties of the reoptimization process, and must therefore be taken into account when choosing whether to set up a contiguous LSP or a stitched/nested one.

So far, we have discussed how to compute, set up and reoptimize interdomain TE LSPs. The last important part of TE is the protection and fast reroute aspects, discussed in the next section.

5.3.4 Protection and fast reroute

As seen in the Protection and Restoration chapter (Chapter 3), the first type of protection is end-to-end protection. This is accomplished by setting up an alternate (also called secondary) path that can be used in case of failure of the primary path. For this approach to provide protection, the failure of a link or node in the primary path should not affect the secondary path. Simply put, the primary and secondary paths must be routed differently in the network. In an interdomain setup, when the computation is done per-domain, finding diversely routed paths is not trivial. Even if the domain exit points chosen for the primary and secondary paths are different, this does not necessarily ensure diversely routed paths. For example, an LSP from A to B is set up in Figure 5.5. Imagine that the primary path enters AS2 through ASBR3 but, because of unavailable resources on link ASBR3–B, it establishes through ASBR3–ASBR4–B. Choosing a different entry point

(ASBR4) for the secondary path does not ensure the path diversity that was desired.

The second type of protection is local protection. This is accomplished by setting up protection paths around the failed link or node, as explained in the Protection and Restoration chapter. Within each domain, link/node protection operates in the same way for interdomain LSPs as for single-domain LSPs: a backup tunnel is built around the protected resource between the point of local repair (PLR) and the merge point (MP), and traffic is forwarded through it when the protected resource fails. When the LSP is set up using stitching, the protection path is applied to the TE LSP segment. When nesting is used, protection is applied to the FA LSP. Doing so implicitly protects the traffic of all LSPs nested on to it. No special actions need to be taken to protect the nested LSPs, because no control-plane state is maintained for them. To summarize, local protection within a domain operates in the same way for inter-domain LSPs and for intradomain LSPs. For this reason it will not be discussed further.

The interesting situation for local protection of interdomain LSPs is when the PLR and the MP fall in different domains. Regardless of whether the protected resource is a link or a node, there are two challenges in this case: how to identify the MP and how to compute the path to it. These challenges are not limited to any particular LSP setup method and they apply equally to LSPs set up as contiguous, stitched or nested. Let us take a look at a link protection scenario, where the failure of a link at the domain boundary requires a backup tunnel between the two border nodes, around the interdomain link.

In Figure 5.8, the link between ASBR1 and ASBR3 is protected by the tunnel ASBR1–ASBR2–ASBR4–ASBR3. How does ASBR1 identify the MP? Recall from the Protection and Restoration chapter (Chapter 3) that typically the MP address is taken from the RRO and that the FRR specifications recommend using interface addresses in the RRO. In an interdomain

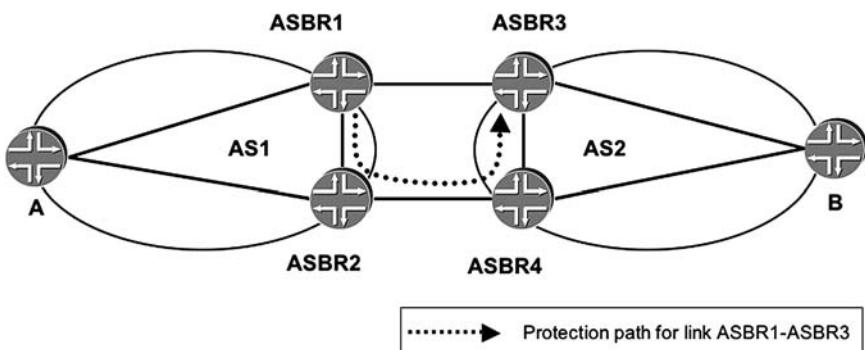


Figure 5.8 Link protection for inter-AS link

setup, interface addresses are not advertised in the IGP so the MP cannot be identified. To solve this problem, the FRR specification was extended to advertise node ids along with interface information [RFC4561]. Node ids are usually loopback addresses. Once the MP is identified, the challenge is to find a backup path to it that does not cross the protected link. Because the PLR does not have visibility into the neighboring domain, it must rely on the same path computation methods described earlier for interdomain LSP computation.

Identifying the MP is not as easy as described in the previous example. An interesting challenge arises in the node protection scenario. The failure of a node requires computing a backup path to a node on the LSP path that lies beyond the failed node. When the LSP is set up as a contiguous LSP, the MP can be any node in the LSP path. However, when the LSP is set up with stitching or nesting, the MP can only be the endpoint of the TE LSP segment or of the FA LSP.

In Figure 5.4, protecting node ABR2 can be accomplished by setting up the bypass tunnel to any node beyond ABR2. When the LSP is set up as a contiguous LSP, R1 is a good MP candidate. However, when the LSP is set up using stitching/nesting, the MP can only be node ABR4, resulting in a much longer protection path. The drawback of a long protection path is that when bandwidth protection is ensured by reserving bandwidth on the protection path, more resources are reserved in the network. In this case, instead of reserving extra bandwidth for the protection path on the links up to R1, the reservation is made on all the links all the way to ABR4. Thus, the LSP setup method affects choice of the MP and thus the properties of the protection path.

To summarize, the same protection mechanisms available for intradomain TE LSPs can be extended to apply in the interdomain case. This is an important property because protection has come to be equated with MPLS-TE and is a requirement for most QoS guarantees.

5.4 INTERPROVIDER CHALLENGES

So far we have focused on the technical details of setting up interdomain LSPs. However, when LSPs span several administrative boundaries, additional concerns arise, in particular over security and compensation agreements, which translate to additional requirements from the interdomain TE solution.

Let us start by looking at the security concerns. Any kind of interprovider interaction requires a level of trust. However, operators seldom rely on trust alone to prevent accidental or malicious impact on their networks because of interprovider relations. Interprovider LSPs are no exception.

The use of RSVP for path signaling creates an interesting problem in interprovider interactions. The path of the LSP is recorded in the Record Route Object (RRO) that is propagated all the way to the head end.

This means that the addresses of the links/nodes in one domain become visible in the neighboring domain. Providers are wary of exposing the internal addressing outside their networks, because by doing so their routers become vulnerable to attacks (the reasoning is that if the router address is not known, the router cannot be attacked). Therefore, the ability to conceal the hops in the path at the exit from a domain, by either filtering them out or modifying the addresses used in the RRO, becomes a requirement for interprovider LSPs. A similar requirement exists for PCEs collaborating in an interdomain path computation that exchange information regarding path segments.

Another security concern is the fact that excessive LSP setup or reoptimization requests can be sent by the upstream domain, with the same effect on the router control plane as a denial-of-service attack. Therefore, the ability to rate-limit such requests at the domain boundary becomes a requirement for interprovider LSP implementation. Furthermore, because an upstream domain can create both control and forwarding state in the network, it is necessary to ensure that LSP setup requests come from an authentic peer and cannot be faked by an attacker. This can be done by using authentication of the protocol messages and by negotiating ahead of time the type of requests accepted at a domain boundary (e.g. accept LSP setup requests only from a certain range of sender addresses).

Negotiation is necessary not just for security purposes but also for compensation agreements between the two administrative domains. As part of such agreements, the exact profile of the interprovider interaction is defined. For example, the two providers negotiate ahead of time how many LSP setup requests can be sent per unit of time, what LSP priorities are acceptable and whether FRR requests are honored. This implies that equipment vendors can provide tools to enforce the terms negotiated in such an agreement (e.g. the ability to reject setup requests based on configured policies).

To summarize, deployments spanning several providers' networks place additional requirements on the interdomain TE solution discussed so far. The extensions are necessary for providing the additional security guarantees needed in such setups and for enforcing compliance with the negotiated interaction profiles between providers.

5.5 COMPARISON OF THE LSP SETUP METHODS

The LSP setup method is one of the important decisions in an interdomain deployment and like any other design choice it involves tradeoffs. For

Table 5.1 Comparison of the different LSP setup methods

	Contiguous	Stitching	Nesting
Number of LSPs in the transit domain, assuming N LSPs in the head end domain	N	N	Smaller than N , depends on the number of FA LSPs
Support of per-domain path computation	Yes	Yes	Yes
Requires protocol extensions	Yes	Yes	Yes
Reoptimization in the transit domain affects other domains	Yes	No	No
Control over reoptimization	Head end	Local (head end if desired)	Local (head end if desired)
MP when protecting a boundary entry node	Any node in the path	TE LSP segment endpoint	FA LSP endpoint

example, contiguous LSPs are more intuitive but they have less desirable scaling properties when compared to nested or stitched ones. The question is not which LSP setup method is better, but rather which one is better for a particular deployment. For example, the fact that a stitched LSP can be reoptimized locally is not an advantage in a setup where reoptimization will never be run. Table 5.1 presents a summary comparison of the different setup methods.

5.6 CONCLUSION

Interdomain TE enables setting up TE LSPs across different areas and different ASs both within a single provider's network and across providers, with the same TE properties and features as intradomain TE. Along with DiffServ Aware TE, interdomain TE completes the traffic engineering solution presented in the Traffic Engineering chapter (Chapter 2). Interdomain TE tunnels are important, not just for interprovider deployments but also for enabling MPLS applications such as Layer 3 VPNs in large networks encompassing several IGP areas when the transport tunnel is RSVP-signaled.

Before we can start exploring the different MPLS applications there is one more piece of functionality that is useful for some of the advanced applications. This is point-to-multipoint LSPs, discussed in the next chapter.

5.7 REFERENCES

- [CCAMPWG] CCAMP working group in the IETF
- [PCE_MON] A set of monitoring tools for Path Computation Element based Architecture, *draft-ietf-pce-monitoring-09.txt* (currently in the RFC editor queue)
- [PCEWG] PCE working group in the IETF, <http://ietf.org/html.charters/pce-charter.html>
- [RFC4105] J. Le Roux, J.P. Vasseur, J. Boyle et al., *Requirements for Inter-Area MPLS Traffic Engineering*, RFC4105, June 2005
- [RFC4206] K. Kompella and Y. Rekhter, *LSP hierarchy with generalized MPLS TE*, RFC 4206, October 2005
- [RFC4216] R. Zhang and J.P. Vasseur, MPLS inter-AS traffic engineering requirements, RFC4216, November 2005
- [RFC4561] J.P. Vasseur, Z. Ali and S. Sivabalan, *Definition of an RRO node-id subobject*, RFC4561, June 2006
- [RFC4657] J. Ash and J.L. LeRoux, Path Computation Element (PCE) Communication Protocol Generic Requirements. RFC4657, September 2006
- [RFC4726] A. Farrel, J.P. Vasseur and A. Ayyangar, *A framework for inter-domain MPLS traffic engineering*, RFC4726, November 2006
- [RFC4736] J.P. Vasseur, Y. Ikejiri and R. Zhang, *Reoptimization of multiprotocol label switching (MPLS) traffic engineering (TE) loosely routed label switch path (LSP)*, RFC4736, November 2006
- [RFC4920] A. Farrel et al., *Crankback signaling extensions for MPLS and GMPLS signaling*, RFC4920, July 2007
- [RFC5088] J.L. Le-Roux, J.P. Vasseur, Y. Ikejiri and R. Zhang, *OSPF Protocol Extensions for Path Computation Element (PCE) Discovery*, RFC5088, January 2008
- [RFC5089] J.L. Le-Roux, J.P. Vasseur, Y. Ikejiri and R. Zhang, *IS-IS Protocol Extensions for Path Computation Element (PCE) Discovery*, RFC5089, January 2008
- [RFC5150] A. Ayyangar, K. Kompella, J.P. Vasseur and A. Farrel, *Label Switched Path Stitching with Generalized Multiprotocol Label Switching Traffic Engineering (GMPLS TE)*, RFC5150, February 2008
- [RFC5440] J.P. Vasseur and J.L. Le-Roux, *Path Computation Element (PCE) communication Protocol (PCEP)*, RFC5440, March 2009

5.8 FURTHER READING

[INTER-DOMAIN-PC]	J.P. Vasseur and A. Ayyangar, <i>Inter-domain Traffic Engineering LSP Path Computation Methods</i> , draft-vasseur-ccamp-inter-domain-path-comp-00.txt (expired draft)
[RFC5151]	A. Farrell, A. Ayyangar and J.P. Vasseur, <i>Inter Domain MPLS and GMPLS Traffic Engineering–Resource Reservation Protocol–Traffic Engineering (RSVP-TE) Extensions</i> , RFC5151, February 2008
[RFC5152]	J.P. Vasseur, A. Ayyangar and R. Zhang, <i>A Per-Domain Path Computation Method for Establishing Inter-Domain Traffic Engineering (TE) Label Switched Paths (LSPs)</i> , RFC5152, February 2008
[RFC2702]	D. Awduche et al., <i>Requirements for Traffic Engineering over MPLS</i> , RFC2702, September 1999
[RFC4655]	A. Farrel, J. Vasseur and J. Ash, <i>Path Computation Element (PCE) Architecture</i> , RFC4655, August 2006
[RFC4657]	G. Ash and J. Le-Roux, <i>PCE Communication Protocol Generic Requirements</i> , RFC4657, September 2006

5.9 STUDY QUESTIONS

1. Referring back to Figures 5.2, 5.3 and 5.4, how does reoptimization of the LSP path in area 1 affect the amount of state in areas 0 and 2 when the LSP is signaled using (a) the contiguous method, (b) stitching and nesting?
2. Referring back to Figures 5.2, 5.3 and 5.4, how is protection against a failure of link ABR2-R1 accomplished when the LSP is signaled using the contiguous method, (b) stitching and (c) nesting?
3. Referring back to Figures 5.2, 5.3 and 5.4, assume the discussion is about three ASs, each having a different EXP bit mapping for voice traffic. How could a voice LSP be set up across the three ASs, such that traffic is forwarded using the correct EXP bits in each AS, when the LSP setup method is (a) end-to-end, (b) stitching and (c) nesting?

4. What are the assumptions required for successfully using the ERO expansion method of path computation if the addresses of the border routers are manually configured at the head end?
5. What are some of the advantages and disadvantages of using a path computation element to perform path computation?
6. In what ways do an offline path computation tool and a path computation element differ?
7. What are some of the challenges of setting up interdomain LSPs in an interprovider deployment?

6

MPLS Multicast

6.1 INTRODUCTION

In the Foundation chapter of this book (Chapter 1), we discussed how MPLS is used to establish LSPs in the network and how the form of the LSP depends on the signaling protocol used. We saw that when RSVP is the signaling protocol, each LSP is point to point in nature, carrying traffic from one ingress point to one egress point. In contrast, when LDP is the signaling protocol, each LSP is multipoint to point in nature, carrying traffic from several ingress points to a single egress point.

In this chapter we will see how RSVP or LDP can be used to create point-to-multipoint (P2MP) LSPs which carry traffic from one ingress point to several egress points, thus enabling multicast forwarding in an MPLS domain. Using P2MP LSPs, traffic is multicast from one source to multiple destinations in a bandwidth-efficient manner, without the ingress having to send separate copies to each receiver.

The use of RSVP-based P2MP traffic engineering gives the ingress router control over the path taken by the traffic and allows bandwidth guarantees to be made. As described later in this chapter, this unification of traffic engineering and multicast enables applications that were previously difficult to support on an IP or MPLS network, such as the distribution of broadcast-quality television.

In later chapters, we discuss the use of P2MP LSPs in the context of L3VPN and VPLS, for the transport of customers' IP multicast traffic. The P2MP LSPs are set up in the service provider's core using either RSVP or

LDP, depending on the needs of the service provider and its customers. This chapter assumes an understanding of RSVP, LDP and TE and some basic knowledge of multicast.

6.2 THE BUSINESS DRIVERS

Without P2MP LSPs, many networks use MPLS for unicast traffic and IP multicast for multicast traffic. Therefore, separate control and forwarding planes for unicast and multicast traffic operate concurrently and independently in the network, without knowledge of each other. This is sometimes referred to as a ‘ships-in-the-night’ situation. When using P2MP LSPs for multicast distribution, the control plane for all traffic within the core of the network is based on RSVP or LDP and the forwarding plane for all traffic is based on MPLS encapsulation. This reduction in the number of protocols used in the core of the network, and the reduction in the number of encapsulations in the data plane, results in simplified network operations.

IP multicast enables the distribution of traffic to multiple receivers without the need to send separate copies to each one of them, but it allows no control over the path the traffic takes and provides no guarantees about the bandwidth availability on the path so it cannot make any QoS guarantees. However, some applications require multicast distribution in conjunction with QoS guarantees such as reserved bandwidth and low loss. The most notable example is professional real-time video transport, which is discussed in more detail in Section 6.7.1. Other applications include core distribution infrastructure for IPTV services and large database downloads to multiple remote sites.

It is useful to compare [P2MPWC] some of the properties of IP multicast to those of P2MP TE. As described later in this chapter, hybrid schemes are possible in which IP multicast operates in conjunction with P2MP TE. The list below does not consider such schemes, and instead compares IP multicast in its native form to P2MP TE:

1. *Failover mechanisms.* For IP multicast traffic, the failover mechanisms are relatively slow (on the order of seconds), the timescale being partly dependent on IGP convergence times. This makes IP multicast unsuitable for real-time video distribution applications in which an interruption of this timescale would be unacceptable. In contrast, as described in Chapter 3, RSVP-TE fast-reroute mechanisms are fast (millisecond timescales) because the switchover to a back-up path is a local decision taken by the router upstream from the point of failure.
2. *Control of path taken by the traffic.* With IP multicast, it is difficult to control the path taken by the traffic. The multicast tree that is built is a shortest-path tree, the path being determined by the IGP. Some implementations allow the use of static multicast routes to override this behavior, but it

is a cumbersome process.¹ RSVP-TE allows control of the path taken by the traffic, according to where bandwidth resources are available or user-defined constraints. Rather than having a shortest-path tree, which minimizes latency, the user may want a minimum-cost tree (also known as a Steiner tree) which minimizes the bandwidth utilization. The difference between a shortest-path tree and a minimum-cost tree is discussed in Section 6.3.2.1.1.

3. *Bandwidth guarantees.* IP multicast protocols (such as PIM, or Protocol Independent Multicast) do not have the ability to perform bandwidth reservations and hence there are no guarantees that resources will be available for the traffic to reach its destination. Even if they did have the mechanisms to perform bandwidth reservations, the path of the multicast tree is fixed, so if the required bandwidth resources were not available along that path, there is no way to change the path of the tree. RSVP-TE, in contrast, has mechanisms for reserving the bandwidth and the path computation can take bandwidth availability into account.
4. *Control over receivers permitted to join the tree.* With IP multicast, there is no overall control over the extent of the tree or the receivers allowed to join it, and receivers can splice themselves on to any existing tree, unless prevented from doing so through the use of tools such as PIM Join filters. In contrast, with P2MP TE, the set of receivers to which the tree extends is determined at the ingress node (e.g. through configuration).

6.3 P2MP LSP MECHANISMS

This section examines the forwarding and control plane mechanisms associated with P2MP LSPs. First we discuss how data are forwarded along a P2MP LSP. This is independent of the signaling protocol used to create the P2MP LSP. Then we discuss the two control plane mechanisms by which P2MP LSPs can be created: the RSVP-based scheme and the LDP-based scheme.

6.3.1 Forwarding plane mechanisms

A P2MP LSP [RFC4461] [RFC4875] has a single ingress router and multiple egress routers. This is illustrated in Figure 6.1. PE1 is the ingress router for the P2MP LSP. The egress routers are PE2, PE3, PE4 and PE5.

As can be seen in the figure, PE1 creates two copies of each packet arriving from the data source. One copy having the MPLS label value L1 is

¹ Alternatively, one can use a different IGP topology for multicast traffic to that for unicast traffic, but this does not give control over the path followed by multicast traffic with per-flow granularity.

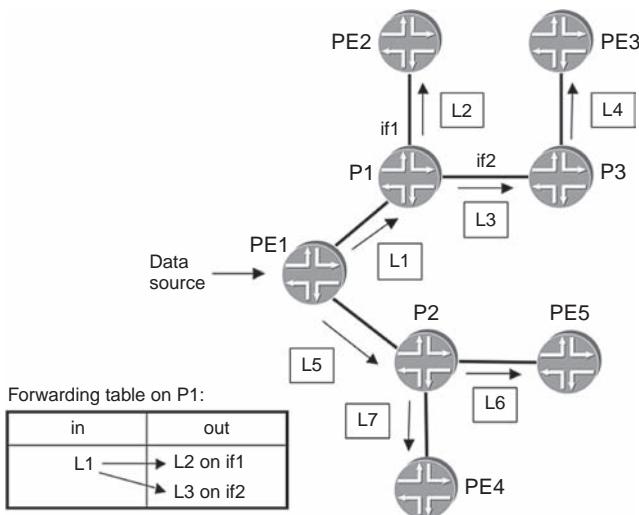


Figure 6.1 P2MP LSP forwarding operation

sent to P1 and another copy having the label value L5 is sent to P2. Routers PE1, P1 and P2 are called branch nodes. As can be seen, replication of MPLS packets occurs at these nodes.

Let us consider the packet forwarding process on P1. For each incoming packet belonging to the P2MP LSP, P1 makes two copies, one of which is sent to PE2 and the other to P3. Let us look at this process in more detail. Packets arrive at P1 having label L1. Looking at the inset of the figure, which shows the forwarding entry corresponding to the P2MP LSP installed on P1, it can be seen that P1 has an entry in its forwarding table for label L1 saying that one copy of the packet should be sent out on interface if1 with label L2 and another copy should be sent out on interface if2 with label L3. Hence P1 is performing a packet replication process in the MPLS domain. The copy of the packet arriving at P3 is forwarded to PE3. No branching occurs at P3, so P3 is just a transit node for this particular P2MP LSP, rather than a branch node.

A key advantage of the P2MP scheme is its bandwidth efficiency. For example, let us suppose that a flow of 100 Mbps is forwarded using the P2MP LSP. On the link between PE1 and P1, only 100 Mbps of bandwidth is used, rather than 200 Mbps if PE1 had to send separate copies of the traffic to PE2 and PE3. As with point-to-point LSPs, the flow of traffic in a P2MP LSP is unidirectional, so no traffic can flow from the egress routers to the ingress routers along the P2MP LSP.

To summarize, the key property of P2MP forwarding is the ability to construct a distribution tree that replicates packets at the branch points.

This is done based on the forwarding information maintained by those branch points. How is this information built? To answer this question, we need to turn our attention to the control plane mechanisms.

6.3.2 Control plane mechanisms

This section describes the control plane mechanisms underpinning P2MP LSPs. First we describe how RSVP creates a P2MP traffic-engineered LSP and discuss how the path computation can be performed. Then we discuss how LDP can create (non-traffic-engineered) P2MP LSPs.

6.3.2.1 Use of RSVP for P2MP traffic engineering

One of the design principles behind the P2MP scheme was to minimize the changes to RSVP-TE needed to accommodate P2MP operation. This section describes how a point-to-multipoint LSP is signaled using RSVP-TE and the changes that were made to RSVP-TE to achieve this. It is useful to refer to the Foundation chapter (Chapter 1) and the Traffic Engineering chapter (Chapter 2) of this book as a reminder of how (point-to-point) traffic engineering works. Figure 6.2 shows the same network as in Figure 6.1 and illustrates how the point-to-multipoint LSP that was shown in Figure 6.1 is signaled by RSVP. It should be noted that the ingress of the P2MP LSP is assumed to know the identity of the egress nodes. The way in which the ingress acquires this information is outside the scope of RSVP-TE, but could be via manual configuration. In the case where a P2MP LSP is being used as infrastructure within a VPN, the egress nodes are discovered through BGP. The figure shows the flow of RSVP Path messages (solid arrows) and Resv messages (dotted arrows). The label values associated with the Resv messages in the diagram (L1, L2, etc.) are those contained in the Label Object in the Resv messages. Bear in mind that as with point-to-point LSPs, downstream label allocation is used. Therefore, the control messages (the Resv messages) containing the label for each link shown in Figure 6.2 travel in the opposite direction from the actual MPLS data packets.

A key point to note is that from the control plane point of view, a P2MP LSP is regarded as a set of point-to-point LSPs, one from the ingress to each of the egress nodes of the LSP. Each of the LSPs within the set is known as a sub-LSP. Recall that for normal point-to-point traffic engineering, an LSP is signaled by sending Path messages that flow from the ingress to the egress and Resv messages that flow from the egress to the ingress. The Path messages contain an Explicit Route Object (ERO) that determines the path followed by the LSP and the Resv messages at each hop contain the label to be used for forwarding along that hop. In the point-to-multipoint

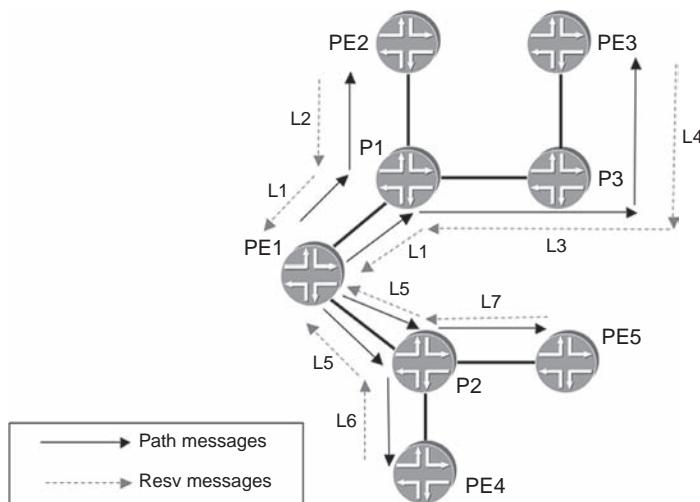


Figure 6.2 RSVP-signaling operation for P2MP LSP

case, each sub-LSP is signaled using its own Path and Resv messages, the Path messages containing the ERO of the sub-LSP in question. The Path and Resv messages contain a new object, the P2MP Session Object, so that the routers involved know which P2MP LSP a particular sub-LSP belongs to. This knowledge is essential for creating the replication state in the forwarding plane. A branch node must realize that two or more sub-LSPs belong to the same P2MP LSP in order to treat them correctly.

Let us see how this works in the example network shown in Figure 6.2. The P2MP LSP has four egress nodes, so it is composed of four sub-LSPs, one from PE1 to PE2, another from PE1 to PE3, and so on. Because each sub-LSP has its own associated Path and Resv messages, on some links multiple Path and Resv messages are exchanged. For example, the link from PE1 to P1 has Path messages corresponding to the sub-LSPs to PE2 and to PE3. Let us examine in more detail how the P2MP LSP in Figure 6.2 is signaled, looking at sub-LSP PE1 to PE3 whose egress is PE3:

1. A Path message is sent by PE1, the ingress router, containing the ERO {PE1, P1, P3, PE3}. This can contain a bandwidth reservation for the P2MP LSP if required.
2. PE3 responds with a Resv message that contains the label value, L4, that P3 should use when forwarding packets to PE3. Similarly, the Resv message sent on by P3 to P1 contains the label value, L3, that P1 should use when forwarding packets to P3.
3. In a similar way, for the sub-LSP whose egress is PE2, P1 receives a Resv message from PE2 containing the label value, L2, that P1 should

use when forwarding packets to PE2. P1 knows that the Resv messages from PE2 and P3 refer to the same P2MP LSP, as a consequence of the P2MP Session Object contained in each.

4. P1 sends a separate Resv message to PE1 corresponding to each of the two sub-LSPs, but deliberately uses the same label value for each, L1, because the two sub-LSPs belong to the same P2MP LSP.
5. P1 installs an entry in its forwarding table such that when a packet arrives with label L1, one copy is sent on the link to PE2 with label L2 and another copy on the link to P3 with label L3. If a bandwidth reservation is being created for the P2MP LSP, the shared explicit (SE) reservation style is used. This ensures that when the Resv messages are sent from P1 to PE1 corresponding to the two sub-LSPs, no double-counting occurs of the bandwidth reservation.
6. PE1, knowing that the two Resv messages received from P1 refer to the same P2MP LSP, a consequence of the P2MP session object contained in each, forwards only one copy of each packet in the flow to P1, with the label value L1 that had been dictated by PE1 in those two Resv messages.

The section of the P2MP LSP from PE1 to PE4 and PE5 is set up in an analogous way to the section from PE1 to PE2 and PE3.

In addition to the scheme described above, in which each sub-LSP is signaled using its own Path message, the RFC [P2MP TE] also discusses another scheme in which each Path message contains details of all the sub-LSPs, including explicit routes for each. However, the existing implementations use the scheme described above.

An interesting question is: what should happen if it is not possible to bring up all of the sub-LSPs belonging to a P2MP LSP? This could be because one of the egress routers is down or there is a loss of connectivity to one or more egress routers due to link failures in the network. Should the entire P2MP tree be torn down? The RFC that covers the requirements for P2MP-TE [RFC4461] leaves this decision to the local policy in the network, because for some applications a partial tree is unacceptable while for others it is not. For example, for an application such as broadcast TV distribution, the typical requirement is that the P2MP LSP should still stay active so that the reachable egress nodes still receive traffic.

In some networks, it may be necessary for a P2MP LSP to cross nodes that do not support P2MP operation. This could happen at the time of the initial deployment of P2MP capability in the network, when some of the nodes support it and other legacy nodes do not. This is a problem because the RSVP messages travel hop by hop, so a sub-LSP will not be established if a node sees an unsupported object (e.g. the P2MP session object) in the RSVP message.

If a node does not support P2MP operation in the control and forwarding planes, a workaround is to use LSP hierarchy (see the Foundation chapter of this book, Chapter 1, for an explanation of LSP hierarchy). In this scheme, sub-LSPs pertaining to a P2MP LSP are nested within an outer point-to-point LSP, so that the transit nodes of the outer LSP are not aware that they might be carrying a P2MP LSP. Naturally, such nodes cannot act as ingress, branching or egress nodes of a P2MP LSP, which may mean that the overall path taken by the P2MP LSP is further from optimum than if those nodes could support branching. As an example of the use of LSP hierarchy, let us refer to Figure 1.9 of the Foundation chapter. Suppose that P2 does not support P2MP operation. It is required to set up a P2MP LSP for which PE1 is the ingress node and PE4, PE5 and PE6 are the egress nodes. The three corresponding sub-LSPs are nested within the core LSP that passes through P2. Hence P2 is unaware of the existence of those sub-LSPs. P3 acts as a branching node so that the traffic is received by the three receiving PEs. Note that the same core LSP can also be used to carry normal point-to-point LSPs at the same time. Another scenario is where P2 is semi-compliant with P2MP TE in that it supports the P2MP control plane, but does not support the branching operation in the forwarding plane. In this situation, it is not necessary to use LSP hierarchy as P2 can process the RSVP messages associated with the three sub-LSPs, but the network administrator needs to bear in mind that the node cannot be expected to act as a branch node.

So far we have looked at the signaling aspect of the P2MP setup and assumed that the ERO is known at the head end. Next we will look at some of the challenges of computing the path of a P2MP-TE LSP.

Path computation in P2MP traffic engineering

It is interesting to explore the path computation of a P2MP-TE LSP. The task is to perform a computation of a P2MP tree taking into account the criteria that define an optimum path from the point of view of the user. For example, if the main requirement is to minimize the latency experienced by the traffic, a shortest-path tree would be appropriate. If, on the other hand, the requirement is to minimize the bandwidth utilization, a minimum-cost tree (Steiner tree), as measured in terms of bandwidth utilization, would be appropriate.

Figure 6.3 compares the path of a P2MP LSP in the shortest-path tree case to the minimum-cost tree case, with the assumption that each link in the network has equal cost and latency and that any link in the network can meet the bandwidth requirement of the LSP. In each case, PE1 is the ingress node and PE2, PE3 and PE4 are the egress nodes. In the case of the shortest-path tree, each egress node is two hops from the ingress node, and the total bandwidth utilization is six units, because the P2MP tree structure uses six

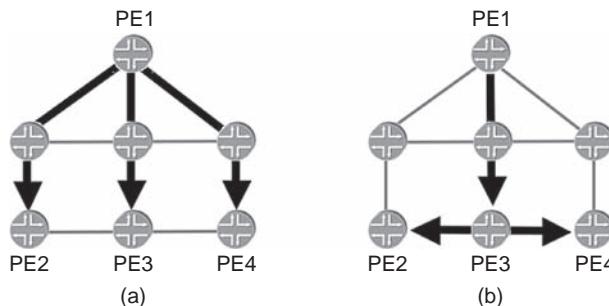


Figure 6.3 Path of P2MP LSP: (a) shortest-path tree and (b) minimum-cost tree

links in total. In contrast, for the minimum-cost tree case, the bandwidth utilization is only four units but with the downside that two of the egress nodes are three hops from the ingress node rather than two. A variation not shown in the figure is a delay-bounded minimum cost tree in which the minimum cost tree is computed for which the propagation delay to any egress is less than a specified maximum.

The path of the P2MP LSP is determined by the branch points. As seen in the discussion above, the placement of the branch points is determined by the network topology, available resources and the objective of the computation (in the case above, whether the tree is shortest path or minimum cost). Additional constraints, such as whether a particular LSR is capable of performing branch functions, or a limit on the maximum number of branch points for a particular LSP at a particular node, may further complicate the computation.

This freedom to define the path of the P2MP tree according to the user requirements contrasts to traditional IP multicast, in which there is no such flexibility: the tree is a shortest path tree [either rooted at the source or the rendezvous point (RP)] and there is no way of changing that. As seen from the discussion above, the path computation for P2MP LSPs can be significantly more complex than for P2P LSPs, both for the initial setup and the reinitialization. For this reason, use of path computation elements (PCEs) [RFC4655], discussed in Chapter 5 in the context of inter-domain path computation, can be an attractive option. Extensions to the PCE communication protocol (PCEP) [RFC5440] for handling computation of P2MP paths are currently being defined in [PCE_P2MP].

As with point-to-point LSPs, potential methods of determining the path of a P2MP LSP are as follows:

1. Manual configuration.
2. Online computation by either the ingress node or a PCE.

3. Offline computation by a PCE, ahead of time. Such computation is appropriate for long-lived LSPs (such as the ones used in transport networks), or as a background activity for network reoptimization.

The considerations about which to use are similar to those discussed for point-to-point LSPs in Chapters 2 and 5. An additional factor to consider is that in some applications of P2MP TE, application level redundancy is sometimes used. This is done by having two P2MP LSPs carry the same datastream. The two LSPs originate at separate ingress routers and follow diverse paths through the network to the receivers, to prevent loss in case of a failure along the path of one of those LSPs. In such cases, it is often easier to use a PCE to compute the paths of the LSPs, as it can be difficult to ensure that paths of the two LSPs do not overlap if the paths are computed by two different ingress routers.

The amount of computation required to calculate an optimum P2MP tree depends on which type of tree is required. In the case of a shortest-path tree, the path to any egress node is independent of the location of other egress nodes, so the computation of the shortest path tree can be decomposed into the computation of each individual sub-LSP. However, in the case of a minimum-cost tree and the delay-bounded minimum cost variant, the optimization problem is more complex, as the path of a sub-LSP to an egress node depends on the location of other egress nodes. In fact, the optimization problem can be shown to be NP-hard (nondeterministic polynomial-time hard). As a consequence, depending on the size of the tree, there may need to be a tradeoff between identifying the optimum tree, which might take an unacceptably long time, and identifying an acceptable, but not necessarily optimum, tree in a shorter period of time. In order to achieve the latter, there exist approximate algorithms that reduce the optimization task from one of NP-hard complexity to one of polynomial complexity.

An interesting question is what to do if one wishes to add or remove a branch from an existing P2MP LSP. In the case of the minimum cost tree (and its delay-bounded variant), should the branch simply be spliced on to or removed from the existing tree, without changing the path taken to any of the egress points already present? This may mean that the tree as a whole is no longer the optimum. Or should the entire tree be reoptimized?

The answer may depend on the application and how often egress nodes come and go. Although make-before-break procedures analogous to those for point-to-point LSPs exist for P2MP LSPs, as with the point-to-point case, there is the possibility of transient reordering of traffic. For example, returning to Figure 6.3(b), let us suppose that PE2 and PE3 are no longer to be required to be egress nodes of the P2MP LSP. If the path to the remaining egress node PE4 is reoptimized from PE1–P2–PE3–PE4 to PE1–P3–PE4,

the first packets to travel along the new path may reach P4 before the last packets to travel along the old path. Whether this is an issue or not depends on whether the application is sensitive to mis-sequencing of packets. Hence the best course is for implementations to give some degree of choice to the user, e.g. by allowing the user to request a recomputation of the tree on an on-demand basis or on a periodic basis.

Taking this approach one step further, doing a network-wide reoptimization for all LSPs in the network, can achieve far greater improvements in optimality. Such computation, and in particular the computation of the migration steps from the old configuration to the new one, would be particularly heavy and require a dedicated PCE.

Whether computed online or offline, by the ingress LSR or by a PCE, whether timer-based or event-driven, the P2MP LSPs discussed so far were signaled using RSVP-TE. In the next section, we will see how LDP can be used for setting up P2MP LSPs.

6.3.2.2 *LDP signaling for P2MP LSPs*

So far we have seen how P2MP LSPs can be created using RSVP. However, many MPLS deployments currently use LDP as the label distribution protocol. For such networks, if P2MP LSPs are required but the service provider does not need the traffic engineering advantages of RSVP-signaled P2MP LSPs, the possibility of using LDP as the signaling mechanism for P2MP LSPs is attractive.

Recall from the Foundations chapter (Chapter 1) that LDP-signaled LSPs are initiated by the egress router. The label propagation is initiated by the receiver and is propagated hop by hop throughout the entire network. All LSRs in the network maintain the forwarding state towards the receiver following the IGP path, and any LSR can act as an ingress to this LSP. In effect, a multipoint-to-point LSP is built with several senders and one receiver. The goal when setting up P2MP LSPs, in contrast, is to have a single sender and several receivers. [P2MP-LDP] describes how to modify LDP to accommodate this scheme, called mLDP (multicast LDP).

One of the fundamental questions is who initiates the signaling of the LSP. In previous sections, we saw that in the RSVP case, the signaling of a P2MP LSP is initiated by the ingress router. However, in the LDP case, requiring the ingress router to initiate the LSP setup requires fundamental changes in the way labels are distributed and therefore is not an attractive option. Instead, the problem of discovering the source and destinations can be decoupled from the actual signaling of the P2MP LSP via LDP. (The discovery problem is also decoupled in the RSVP case, in that the source learns the identity of the receivers by some means outside of RSVP.) This allows the LDP solution to be developed to be receiver initiated rather than sender initiated if required.

Assuming that the receivers know that they must establish a P2MP path towards the sender, the second fundamental question is how to identify the P2MP LSP. Similar to the RSVP case, this information is necessary to be able to install the correct forwarding state at the branch nodes. Clearly the ingress router of the LSP must be identified. The ingress router alone is not enough, because several P2MP LSPs may originate at the same ingress. Thus, it is necessary to identify not just the source but also the tree. LDP does not need to be aware of the semantics of the tree identifier; from its point of view the identifier is opaque. To set up the LSP, a label must be assigned by the receivers and associated with the entity of {source, tree identifier}. We will call this the P2MP forwarding equivalence class (FEC).

Recall that LDP LSPs follow the IGP. As we saw in the Foundations chapter, for a FEC corresponding to an IP address, this is accomplished by using for forwarding only those labels received over sessions that lie in the IGP path for that IP address. In the case of with P2MP FECs, the procedure is different. The rule for distribution is to advertise a label only towards the neighbor that lies on the IGP best path towards the source. Thus in the regular LDP case, the receiver of the label determines the best path towards the egress, but in the P2MP case, the sender of the label determines the best path towards the ingress.

Figure 6.4 shows an example of how a P2MP LSP is signaled by LDP. PE4 is the ingress router of the P2MP LSP and PE1 and PE2 are the egress routers. PE2 advertises a label, L2, for the P2MP FEC only towards P1, and not towards P3, because P1 lies in the best path towards the ingress. The P2MP FEC contains the address of PE4, the ingress of P2MP LSP to be built and the P2MP tree identifier. PE1 advertises a label L1 for the same

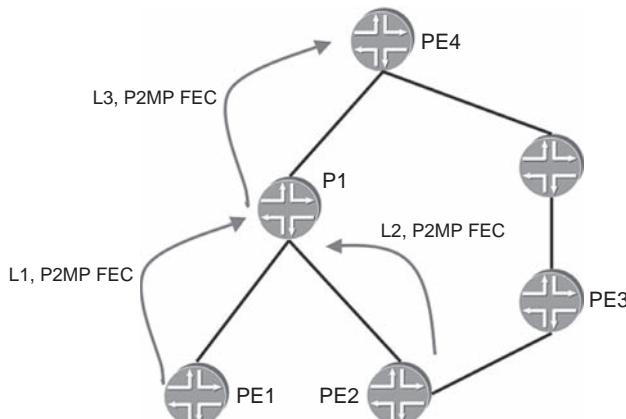


Figure 6.4 Signaling of a P2MP LSP using LDP

P2MP FEC towards P1. At node P1, the labels L1 and L2 are identified as belonging to the same P2MP FEC. As a result, a single label L3 is advertised towards the source, PE4, and the forwarding state is installed to replicate packets arriving with label L3 on each of the interfaces towards PE1 and PE2 with labels L1 and L2 respectively.

In this way, the signaling for the P2MP LSP can be done from the receivers towards the source. Similar procedures have been set in place to define the behavior for label withdrawals.

Comparison of RSVP and LDP signaling for P2MP LSPs

In the Foundations chapter, we discussed the differences between RSVP and LDP signaling for unicast LSPs. We showed that the number of LSPs grows as the square of the number of PEs in the RSVP case and grows in proportion to the number of PEs in the LDP case. In the unicast case, LDP LSPs tend to require less configuration than RSVP LSPs although schemes such as RSVP autobandwidth and RSVP automesh reduce the RSVP configuration overhead.

In the P2MP case, the differences between LDP and RSVP are not analogous to the unicast case. In the P2MP case, each P2MP LSP is required to serve a particular source and a particular set of receivers, so the amount of forwarding state and the total number of P2MP LSPs in the network would be the same for the RSVP and LDP case.

Regarding configuration, the configuration overhead is similar for LDP P2MP LSPs or RSVP P2MP LSPs. In each case, the leaves of the P2MP LSP need to be identified. In the RSVP case this would imply listing the leaves on the ingress router. In the LDP case, it would imply applying some configuration to each egress router to identify it as a member of a particular P2MP LSP having a particular ingress router as the root. If BGP autodiscovery is being used, for example in the context of L3VPN or VPLS where P2MP LSPs are being used as infrastructure, then the configuration is minimal for both the LDP and RSVP cases.

The control plane overhead can be less for the LDP case than the RSVP case, as LDP runs over TCP and so does not require the periodic refreshes needed in the RSVP case. Also, in the LDP case, the control plane state tends to be less, especially near the ingress. For example, in Figure 6.1, in the LDP case, PE1 is only aware of the directly connected nodes, P1 and P2, and is not aware of the leaves downstream from those nodes. In the RSVP case, because of the sub-LSPs created towards each of the leaves, PE1 is aware of each of the leaves of the P2MP LSP and has state associated with each. In practice, these differences may not matter, and for certain types of traffic, such as broadcast TV, the traffic engineering, admission control and traffic protection requirements mean that RSVP is the best choice.

6.4 LAN PROCEDURES FOR P2MP LSPS

One of the main goals of P2MP LSPs is to minimize the bandwidth used to distribute the content from the source to all the receivers. Thus, one of the fundamental requirements is to send every packet at most once over any given link. Let us take a look at an interesting problem that arises when dealing with multiaccess links, e.g. Ethernet. Figure 6.5 shows a simple network topology where source S is required to send traffic to three destinations, R1, R2 and R3. The destinations are connected to three transit routers, P1, P2 and P3, which are all on the same LAN.

To achieve the optimum bandwidth utilization, S sets up a P2MP LSP to the three receivers, according to the procedures described so far. During the setup of the branch LSPs, each of the routers P1, P2 and P3 assigns a label and advertises it to P0. As a result, a single packet is sourced at S towards P0, but three separate copies are sent by P0 towards P1, P2 and P3, although these routers are connected to a shared media and a single packet could have reached all three of them. Indeed, if the three routers had assigned the same label, replication at P0 would not be necessary, and a single packet could be sent over the LAN.

Unfortunately, there can be no guarantee that P1, P2 and P3 assign the same label because they each assign the labels independently from their global label space. One possibility in principle could be to implement a scheme to coordinate between P1, P2 and P3. Alternatively, one could devise a scheme in which router P0 is given control over the label allocation. This approach is referred to as 'upstream label allocation' and is documented in [RFC5331] and [RFC5332]. Before examining the LAN

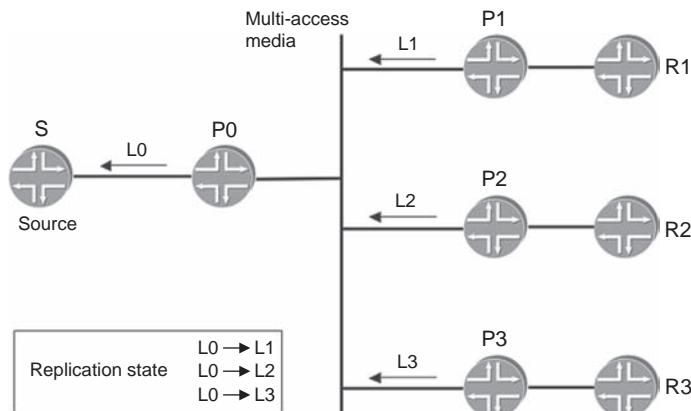


Figure 6.5 Label allocation on a multiaccess network

scenario further, the following section discusses the principles behind upstream label allocation.

6.4.1 Upstream label allocation

Before examining the implications of reversing the control over the label allocation, let us stop for a moment and revise some terminology. Recall from the Foundations chapter (Chapter 1) that the routers are referred to according to their location relative to the traffic flow. For example, in Figure 6.6, traffic flows from left to right, from Ru1 towards R2. Router R1 is performing downstream label allocation because it is assigning a label that it expects router Rd to use. Thus, the allocation is done by a router that is ‘downstream’ of the router that is actually going to put the label on the packet. Downstream label allocation is the scheme that is used by both RSVP and LDP today. Upstream label allocation is the scheme that was proposed as a solution to the multiaccess media problem in the previous paragraph. The label is assigned by the same router that is putting the label on the packet. In Figure 6.6, router Ru1 advertises label L1 for FEC 1.1.1.1 to router Rd, meaning that Ru1 intends to send labeled traffic destined to 1.1.1.1 using label L1. (Although we use the LDP notation, the same is applicable to RSVP.)

If you look carefully at Figure 6.6, the first problem with upstream label allocation becomes immediately evident. Router Ru2 advertises the FEC 2.2.2.2 and by coincidence chooses the same label, L1, that Ru1 had chosen for FEC 1.1.1.1. This can happen because Ru1 and Ru2 assign the labels independently from their global label space. When the labeled traffic is received, how can Rd determine if it is destined towards R2 or towards R3? Clearly, Rd must be able to identify the neighbor from which it receives

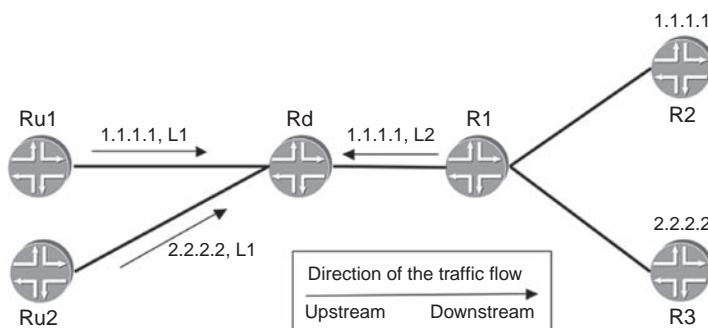


Figure 6.6 Illustration of an issue with upstream label allocation

the traffic, because the label has meaning in the context of that particular neighbor. In the example from Figure 6.6, the obvious answer is to use the incoming interface to distinguish between the two neighbors, but the general answer may be different under different situations.

Returning to our LAN example shown in Figure 6.5, using the incoming interface to distinguish between neighbors does not work, since a labeled packet arriving on the LAN interface on router P2 could have come from any of the other three routers attached to that LAN. In this scenario, an additional label known as the ‘context label’ is used. Each upstream router uses a unique value for the context label, as it is algorithmically derived from the IP address assigned to each router’s interface on the LAN. Procedures have been defined in for RSVP [RSVP-UPSTR] and LDP [LDP-UPSTR] to communicate the value of the context label. In the data plane, the outermost label is the context label and the next label is the upstream-assigned label denoting a particular P2MP LSP. If a node is the upstream router for multiple P2MP LSPs crossing the LAN, each would have the same context label but a different inner label. The value of the context label allows a downstream router to identify which router sent the packet onto the LAN. Hence, the label lookup of the inner upstream-assigned label can be carried out in the context of the label space of that upstream router. For example, suppose P0 allocates a context label of value L5. Then, for the particular P2MP LSP shown in the figure, it allocates an upstream label value of L4. Each packet arriving at P0 from the source is encapsulated with an inner label value of L4 and an outer label value of L5. Only one copy of each packet needs to be sent onto the LAN. P1, P2 and P3 each identify the packet as coming from P0 on the basis of the context label and then perform a label lookup on label L4 in the context of labels advertised by P0 and hence forward the packet to their respective receivers.

So far in this chapter, we have discussed how P2MP LSPs are set up. Another important aspect is OAM of P2MP LSPs. This topic is discussed in the MPLS Management chapter of this book. In the next section, we describe how P2MP LSPs can be used.

6.5 COUPLING TRAFFIC INTO A P2MP LSP

The previous sections described how a P2MP LSP is created. Let us now examine how traffic can be coupled into a P2MP LSP at the ingress node. We consider three categories of traffic: Layer 2 traffic, IP traffic having a unicast destination address and IP traffic having a multicast destination address. All three categories apply to video applications, because for each there exist examples of commercially available video equipment that encapsulate video flows into packets of that format.

6.5.1 Coupling Layer 2 traffic into a P2MP LSP

One application for P2MP LSPs is to carry Layer 2 traffic such as ATM. For example, some encoders encapsulate digital TV signals into ATM AAL1 frames. With a native ATM network, point-to-multipoint VCs are often used to distribute the traffic to multiple destinations. When using an MPLS network, P2MP LSPs provide the analogous function, allowing the Layer 2 traffic to be distributed to multiple receivers in a bandwidth-efficient manner.

An existing implementation achieves this by using a point-to-multipoint version of the Circuit Cross Connect (CCC) [CCC] scheme described in the Layer 2 Transport chapter (Chapter 12). In this scheme, a binding is created, through configuration, between an incoming Layer 2 logical interface (e.g. an ATM VC or an Ethernet VLAN), known as an attachment circuit, and a P2MP LSP at the ingress router. Similarly, at the egress routers, a binding is created between the P2MP LSP and the outgoing Layer 2 logical interface. Note that because CCC depends on RSVP signaling, this scheme applies only to P2MP LSPs that are signaled by RSVP. The detail of how the Layer 2 frames are encapsulated for transportation across the MPLS network is exactly the same as for the point-to-point CCC case described in the Layer 2 Transport chapter. For example, in the ATM case the user can choose how many ATM cells should be carried by each MPLS packet.

At the time of writing, an alternative scheme called Layer 2 Multicast VPN (L2mVPN) for carrying Layer 2 traffic using P2MP LSPs was under discussion in the IETF. This is discussed in more detail in the Layer 2 Transport chapter of this book.

6.5.2 Coupling IP unicast traffic into a P2MP LSP

Another type of traffic is a flow of packets having an IP unicast destination address. This would be case in a scenario where the source generates a stream of IP packets that have a unicast destination address but nevertheless need to be distributed to multiple video receivers. Each receiver would typically be directly connected to one of the egress nodes of the P2MP LSP. The coupling of the IP traffic into the point-to-multipoint LSP at the ingress router could be carried out using a static route, with the P2MP LSP as the next hop. At the egress routers, if the destination address of the packet is on a subnet to which the egress router is attached, the packet is automatically routed correctly. Alternatively, a static route could be used to direct the packet to the appropriate output interface. Although this scheme may sound odd in that multiple receiving hosts are configured with the same IP address, and potentially multiple subnets around the network

are configured with the same address and mask, the scheme is useful for expediency because some commercially available video-to-IP encoders currently generate packets having a unicast IP destination address.

6.5.3 Coupling IP multicast traffic into a P2MP LSP

In this section, we discuss methods by which IP multicast traffic can be coupled into a P2MP LSP. Early deployments used static routes on the ingress PE to map multicast streams into the appropriate P2MP LSP. It is these scenarios that we discuss in this section. More recently, schemes to dynamically map IP multicast streams into P2MP LSPs have been developed, using a BGP control plane for discovery of multicast receivers. These are discussed in more detail in Chapters 10 and 11.

Let us first take the case where no multicast protocols are in use, such as IGMP or PIM, but the application, such as a video encoder, generates packets with a multicast destination address and the receivers are configured to receive packets with that destination address. This scheme would be applicable in a scenario where the multicast source is directly connected to the ingress router of the P2MP LSP and the receivers are directly connected to egress routers of the P2MP LSP. In this case, a static route at the ingress router can be used to direct the packet into the appropriate P2MP LSP. At the egress nodes, again a static route is used to direct the packet to the appropriate receiver.

Another variation in the IP multicast case is a hybrid one in which P2MP LSPs provide a core distribution capability but multicast trees formed through PIM procedures are used for local distribution beyond the egress routers of the P2MP LSPs.

The P2MP LSP tree is fixed but the local PIM trees are formed dynamically using normal PIM procedures in response to IGMP reports generated by receivers wishing to join particular multicast groups. This scheme might be appropriate for IPTV scenarios where all the channels are distributed by the P2MP LSP to local head ends, but multicast group membership determines the onward distribution of those channels from a local head end (i.e. from a P2MP LSP egress point) to a multicast receiver. In this situation, the PIM Joins triggered by IGMP reports received by the router attached to the receiver only extend as far as the egress router of the P2MP LSP.

Figure 6.7 illustrates such a scheme. A P2MP LSP extends from the ingress router, PE1, to the egress routers, PE2, PE3 and PE4. Each egress PE is attached to a local distribution infrastructure. The egress PEs and the local distribution routers (e.g. R8, R9 and R10 in the case of PE2) have PIM enabled. PE1 is attached to the sources of the multicast groups G1, G2, G3 and G4. The P2MP LSP distributes traffic belonging to these

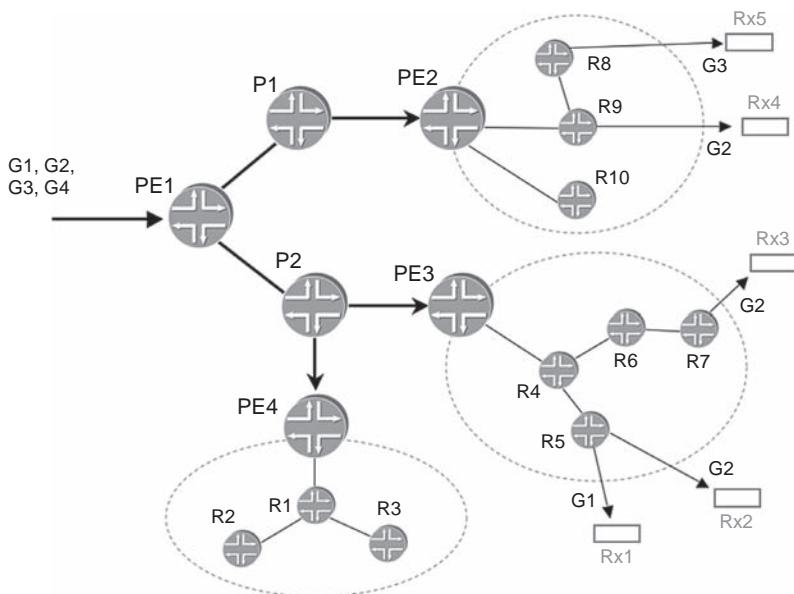


Figure 6.7 Illustration of a hybrid scheme with P2MP LSP in the core and PIM islands at the edge

multicast groups to the egress PEs. Let us suppose receiver Rx1 (which could be a set-top box) needs to receive multicast group G1. It sends a corresponding IGMP message to R5. This triggers R5 to generate a PIM Join which propagates hop by hop in accordance with normal PIM procedures to PE3 (but not beyond). This results in traffic for group G1 to be forwarded by PE3 towards Rx1.

6.6 MPLS FAST REROUTE

A key attraction of P2MP LSPs signaled using RSVP is that MPLS fast reroute can be used for traffic protection, giving low failover times. In contrast, in normal IP multicast, the failover mechanisms are relatively slow (on the order of seconds), which is unacceptable for applications such as real-time video. In the Protection and Restoration chapter (Chapter 3), the following variants of fast reroute were described in the context of point-to-point LSPs:

1. Link protection
2. Node protection

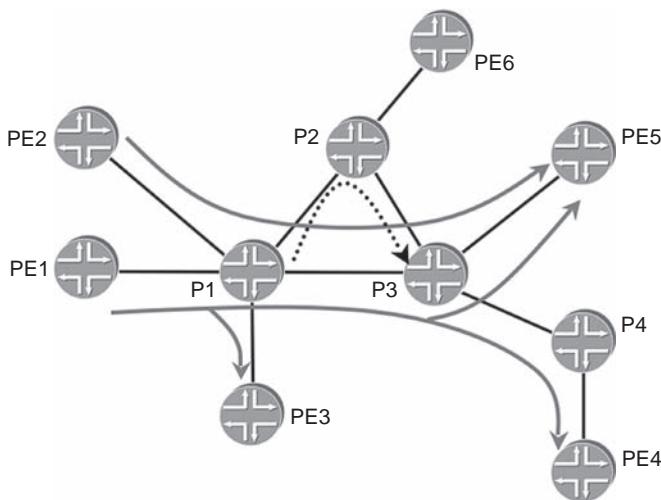


Figure 6.8 Illustration of link protection for a P2MP LSP

In principle, either of these variants could be used in the point-to-multipoint case. The most straightforward case, for implementation and operations, is the link protection case, because the overall topology of the P2MP LSP in terms of the location of branch nodes remains unchanged. In this case, the task of protecting sub-LSPs related to P2MP LSPs is the same as that involved in protecting normal point-to-point LSPs. In the node protection case, the protection paths end downstream of the next-hop node that is being protected, which would result in the location of branch points changing if the node being protected is a branch node. In the facility protection schemes, where a single bypass tunnel protects multiple LSPs, the same bypass tunnel can be used to protect point-to-point LSPs and sub-LSPs of P2MP LSPs. This is illustrated in Figure 6.8. In the figure, there is a P2MP whose ingress is PE1 and whose egress points are PE3, PE4 and PE5. There is also a point-to-point LSP from PE2 to PE5. The link between P1 and P3 is protected by a bypass tunnel that follows the path P1-P2-P3. This is shown as a dotted line in the figure. If the link between P1 and P2 fails, the bypass tunnel protects the PE2-PE5 point-to-point LSP and the P2MP sub-LSPs from PE1 to PE4 and PE5.

An interesting situation would arise if the original P2MP LSP also has a sub-LSP following the path PE2-P1-P2-PE6 in addition to the ones shown in the diagram. In that case, when the link P1-P2 fails, there would be duplicate traffic traveling along the P1-P2 link – one copy traveling along the original P2MP LSP and the other copy traveling along the bypass

tunnel. A solution to this problem has been proposed [P2MP BYPASS] in which a P2MP bypass tunnel is used instead of point-to-point bypass tunnel. In this situation, the bypass tunnel would have one egress point at P2 and another at P3. In this way, if the link between P1 and P3 breaks, P1 stops sending traffic along the original sub-LSP to PE6 and instead sends traffic only onto the P2MP bypass LSP. In this way, P1 only needs to send one copy of each packet on the P1–P2 link. P2 replicates each packet, merging one copy with the original sub-LSP to PE6 and sending the other copy towards P3.

6.7 INGRESS REDUNDANCY FOR P2MP LSPs

In deployments in which a high degree of availability is required, redundant ingress PEs are used for resilience. This is illustrated in Figure 6.9. PE1 is the ingress for P2MP LSP X and PE2 is the ingress for P2MP LSP Y. The paths of the two P2MP LSPs do not have any links or nodes in common.

Sources S1 and S2 send identical traffic to PE1 and PE2, respectively. One option is for each ingress router to always send the traffic onto its respective P2MP LSP. This is sometimes known as ‘Live-Live’. In this way, two identical traffic feeds arrive at each receiver. In the majority of cases in which Live-Live is used, application level selection is performed

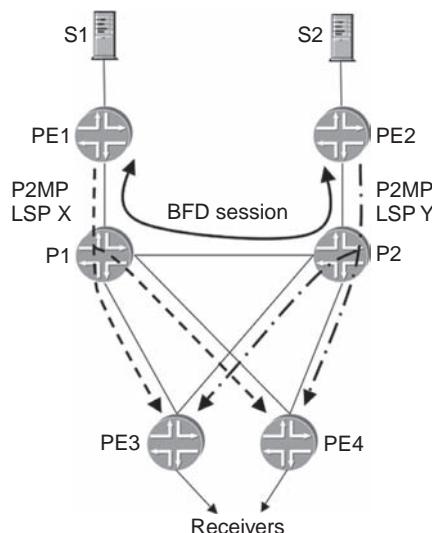


Figure 6.9 Ingress PE redundancy for P2MP LSPs

at the receiving device. For example, in the case of video distribution, the receiver, rather than being a router, is a video-layer device that reconstructs the video streams from the two duplicate feeds and selects one of them to pass downstream. If traffic on the chosen feed no longer arrives or becomes corrupted, the video-layer device instead selects the other feed. If the equipment is designed correctly, this can be done in a seamless way, so no interruption occurs to the application if there is a network failure that disrupts one of the feeds.

When using IP multicast end-to-end, it is difficult to construct the Live-Live scheme as it is not easy to ensure complete path diversity between the two feeds along the entire path from each ingress router to the corresponding egress routers. However, such diversity is straightforward to achieve using P2MP MPLS traffic engineering. Techniques such as link coloring (affinity groups) or appropriate EROs are used to achieve this. If the network has two distinct planes in the core of the network, then the links in one plane can be assigned one color (say blue) and the links on the other plane another color (say red).

For some applications, sending the same data twice into the network is regarded as too expensive in terms of bandwidth consumption. For such cases, a Live-Standby scheme is used. In such a scheme, only one ingress PE sends a particular multicast stream into the network at any time.

For such cases, some implementations have a scheme to choose a designated forwarder for the traffic. In order for each ingress PE to detect that the other is up, a BFD session is created over a point-to-point LSP between the two PEs. Typically, there would be a pre-existing point-to-point LSP between the PEs anyway for the purpose of carrying unicast traffic, which could be used to carry the BFD session. While both PEs are up, the designated forwarder is the one with the lower IP address. Let us suppose this is PE1. As a result, PE2 does not forward traffic into the P2MP LSP, unless it detects through BFD timeout that PE1 has gone down. In that case, it takes over and begins to forward traffic into the P2MP LSP.

The advantage of using BFD in this way is that it eliminates the need to define new control plane message exchanges in order to elect the designated forwarder. The concept is generic in nature and could be extended to other MPLS applications requiring redundancy.

An alternative approach to constructing the Live-Live and Live-Standby schemes is to use Next-Generation Multicast VPN (NG mVPN) as the foundation. This is discussed in detail in Chapter 11.

6.8 P2MP LSP HIERARCHY

In the Foundations chapter, we discussed how MPLS has important hierarchy properties to aid scaling. We discussed how in the data plane,

the hierarchy is achieved by label stacking, for example in the following scenarios:

1. Traffic from multiple VPNs can share the same transport LSP. An inner VPN label is used to distinguish between VPNs and an outer label is used to denote the transport LSP.
2. LDP-signaled LSPs can be tunneled through an RSVP-signaled LSP. An inner label denotes the LDP LSP and the outer label denotes the RSVP LSP.
3. RSVP-signaled LSPs can be nested inside other RSVP-signaled LSPs using the RSVP LSP hierarchy scheme. An inner label denotes the inner RSVP LSP and an outer label denotes the outer RSVP LSP.

Similar concepts are applicable in the case of P2MP LSPs. Traffic from multiple VPNs can share the same P2MP LSP, with inner labels being used to distinguish traffic from the different VPNs. This is discussed further in Section 6.9.2.

P2MP LSPs can be nested inside other P2MP LSPs in order to reduce the amount of state in the core of the network. Figure 6.10 shows an example. R1 is the ingress of P2MP LSP X and P2MP LSP Y, and R3 and R4 are the egress routers.

In order for a P2MP LSP (the ‘inner P2MP LSP’) to be nested inside another P2MP LSP (the ‘outer P2MP LSP’), the following conditions must be met:

- (i) The inner and outer P2MP LSP must have the same ingress router.
- (ii) Each egress node of the inner LSP must also be an egress node of the outer LSP.

As can be seen in Figure 6.10, P2MP LSP X and P2MP LSP Y can be nested inside P2MP Z (denoted by the wide grey lines) as they meet the two conditions listed above. In this way, R2 is unaware of the existence of P2MP LSPs X and Y, either in the data plane or the control plane.

The inner and outer LSPs can all be RSVP-signaled LSPs, by analogy with the RSVP LSP hierarchy case discussed in the Foundations chapter. Alternatively, the inner LSPs can be LDP signaled and the outer LSP can be RSVP signaled, by analogy with the LDP over RSVP scheme discussed in the Foundations chapter. Either way, in the forwarding plane, a label stack is used, with the inner label denoting P2MP LSP X or P2MP LSP Y and the outer label denoting P2MP LSP Z. Again by analogy with the point-to-point case, R2 leaves the inner label untouched – it is only aware of P2MP LSP Z.

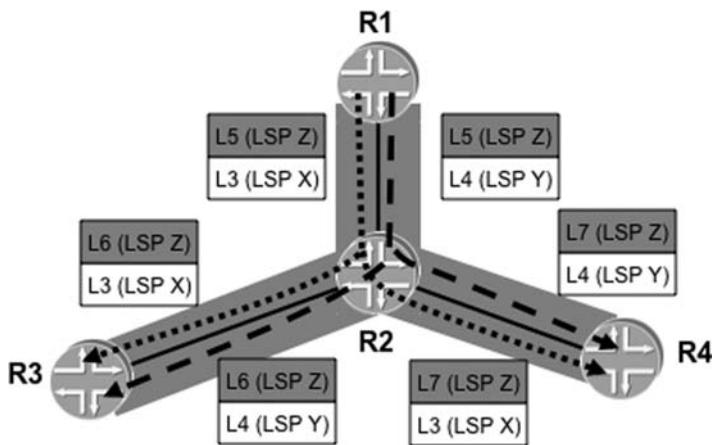


Figure 6.10 LSP hierarchy for P2MP LSPs. The dotted line denotes P2MP LSP X. The dashed line denotes P2MP LSP Y. The wide grey line denotes P2MP LSP Z

6.8.1 P2MP LSP hierarchy forwarding plane operation

Let us examine the label operations associated with P2MP LSP X shown in the diagram, analogous operations occur for P2MP Y:

- R1 pushes label L3 onto each packet. This is the label associated with P2MP LSP X.
- R1 then pushes label L5 onto the top of the label stack. L5 is the label associated with P2MP LSP Z into which P2MP LSP X is nested.
- R2 replicates each packet it receives on P2MP LSP Z. It makes one copy, swaps label L5 for label L6 and sends the packet to R3. It makes another copy, swaps label value L5 for label L7 and sends the packet to R4. Note that it does not touch the inner label associated with P2MP LSP X.
- R3 and R4 pop the outer label and perform a lookup on the inner label L3² and forward the packet accordingly.

An interesting issue occurs with the P2MP hierarchy. If the outer P2MP LSP Z were not being used, R3 and R4 would each signal to P2 the value of the labels they expect for P2MP LSP X in the usual way and in all probability would choose different label values. However, when using P2MP hierarchy, the packet associated with P2MP LSP X must arrive at R3 and R4 with same label, L3. This is because the label is applied by R1

²In this example, we are assuming that PHP is not used on P2MP LSP X.

and only one copy of each packet is sent to R2, which performs replication without changing the inner label value. The question is, how is this inner label allocated? If R3 and R4 were to allocate the label value, they would need some mechanism to agree on what label value to use. Instead, by analogy with the LAN case described earlier in this chapter, upstream label allocation is used, with R1 dictating the value of the label. This results in the need for R3 and R4 to perform context-dependent lookup on the label of the arriving packet. The lookup of the label value on R3 and R4 must be carried out in the context of P2MP LSP Z, as the label value may coincide with a value allocated by another node for some other LSP. For example, R4 may have a P2MP LSP to R3 and R1 in which other P2MP LSPs are nested. R3 needs a way to distinguish between packets received from R4 and from R1 belonging to different inner P2MP LSPs if they have the same label. The way this is achieved is by not using PHP on the outer LSP, P2MP LSP Z. This means that R3, for example, can tell from the outer label value L6 that a packet has arrived on P2MP LSP Z and so can perform a label lookup on the inner label in the context of P2MP LSP Z.

6.8.2 P2MP LSP hierarchy control plane operation

Let us now look at how the label values are exchanged in the control plane. The label values for P2MP LSP Z are distributed in the usual way, using downstream label allocation. For example, R3 dictates to R2 that value L6 is to be used for P2MP LSP Z on the link R2 to R3. In order to exchange label values for P2MP X and Y, R3 and R4 each have a directed LDP or RSVP session with R1. This is analogous to the LDP over RSVP and the RSVP LSP hierarchy schemes described in Chapter 1. The difference in the P2MP case is that upstream label assignment is used, with R1 dictating the label value to be used for P2MP LSPs X and Y. When R1 communicates the label value to be used, it also communicates the identity of the outer P2MP LSP, in the form of the P2MP session object, into which the inner LSP will be nested. This information is carried in new LDP TLVs or RSVP objects that have been defined in [LDP-UPSTR] and [RSVP-UPSTR], respectively. This lets R3 and R4 know that they need to perform inner label lookup in the context of P2MP LSP Z.

6.9 APPLICATIONS OF POINT-TO-MULTIPOINT LSPs

This section describes some of the main applications of P2MP LSPs. We first discuss how P2MP TE is being used for the purposes of broadcast TV distribution. We then describe proposals for how P2MP LSPs can be used

as infrastructure tools to enable service providers to carry their customers' multicast L3VPN traffic and VPLS multicast traffic more efficiently. These proposals are discussed in more detail in the multicast L3VPN chapters (Chapters 10 and 11) and VPLS chapter (Chapter 13).

6.9.1 Application of P2MP TE to broadcast TV distribution

An interesting application of P2MP TE is for professional real-time broadcast TV distribution [IBC2004] [IBC2007] [MPLS-VID]. This application should not be confused with Internet video streaming applications, which typically involve the sending of low-bandwidth video streams to end users' PCs without any quality guarantees. In contrast, professional real-time broadcast TV distribution requires exacting performance guarantees from the network. Customers of such a service are TV broadcast companies who transport real-time video between studios, from a studio to head ends of distribution infrastructure (terrestrial, cable or satellite) or from an outside broadcast location to studios. The ability to offer broadcast TV distribution services is attractive to service providers because of the high-value revenue streams that can be generated. The demand for such services is likely to grow as the number of TV channels increases as a consequence of the extra capacity available through the growth of satellite, digital terrestrial and cable infrastructure.

Traditionally, such networks have been based on TDM transport (in the form of PDH or SDH, or SONET) or on ATM. However, there is increasing interest in moving to IP/MPLS networks for the following reasons:

1. As well as transport of real-time video, broadcast TV companies and production houses increasingly demand the ability to transport non-real-time video with file transfer using IP-based protocols (e.g. FTP), as opposed to the traditional method of physically transporting a tape between one location and another. Cost savings can be achieved by using the same network for the real-time and non-real-time transfers. A packet-switched network is more suitable for this than a TDM network because of the statistical multiplexing advantages offered when dealing with the bursty data flows associated with the transfer of the non-real-time video.
2. Higher interface speeds are available for IP/MPLS networks than for ATM networks.
3. It is easier to build a shared network on which multiple TV broadcast companies can book bandwidth. The lead times for making bandwidth available to new customers or for existing customers requiring extra

capacity are much less than for TDM-based networks. The service provider can go one step further than in item 2 above. Rather than building a dedicated network for the purpose of broadcast TV distribution, this can be just one service among many carried over an IP/MPLS network.

The transport of broadcast quality real-time video places stringent requirements on the network, even more so than voice transport. The nature of the application is that there is no opportunity to resend data that failed to reach the destination and even very short-lived interruptions to the data flow can have a noticeable impact. The key requirements are as follows:

1. *Bandwidth guarantees.* Once a booking for a particular video flow has been accepted, the traffic must be transported without loss of data. There cannot be any contention for the bandwidth from other data flows.
2. *Low delay variation.* The tolerance of the flow to delay variation depends on the nature of the decoding equipment, but on the order of milliseconds is a typical target.
3. *High network availability.* The disturbance to the datastream must be minimal in the event of link failure or failure of components within the network equipment. Hence a high degree of component redundancy and schemes for rapid recovery from link failures are very desirable.
4. *Distribution from a single source to multiple destinations.* It is a common requirement for particular real-time video flows to be transported to multiple destinations. It is important to be able to add or remove a destination corresponding to a particular flow without interruption to the flow of data to the other destinations.

Let us see how the use of P2MP TE on an MPLS network allows the above requirements to be met. The requirements for low delay variation and bandwidth guarantees can be met as follows. If the network is to be shared with other traffic, on each link the real-time video packets are placed into a dedicated queue that has high scheduling priority. This means that the latency experienced by the packets in that queue is minimized, as long as the queue is not over-subscribed. Oversubscription of that queue is avoided by using traffic engineering mechanisms: bandwidth is reserved on each P2MP LSP and admission control is performed so that the sum of the bandwidth reservations does not exceed the bandwidth available to that queue. If the video traffic is the only form of traffic in the network that requires bandwidth guarantees and admission control, then RSVP-based traffic engineering can be used as described in the Traffic Engineering chapter (Chapter 2), with the maximum available bandwidth being set to the size of the queue assigned to the real-time video. If other forms of

traffic also require bandwidth guarantees and admission control, RSVP-based DiffServ Aware Traffic Engineering can be used, as described in the DiffServ Aware Traffic Engineering chapter (Chapter 4).

The service provider can make the most efficient use of bandwidth when meeting the customer's requirement of distributing the traffic to multiple destinations by building P2MP LSPs in the form of minimum-cost trees. Bandwidth efficiency is especially important, bearing in mind that the bandwidth of a single uncompressed standard definition video exceeds 300 Mbps and that of an uncompressed high definition video exceeds 1.5 Gbps. In some cases, compression, e.g. based on MPEG-2, is used to reduce the bandwidth requirement. The requirement of being able to add or remove egress points of a P2MP LSP without affecting traffic traveling to the other egress points of that LSP can be met through a careful router forwarding plane design and implementation.

The use of fast-reroute mechanisms for P2MP LSPs means that the disturbance to traffic is minimized should a link in the network fail, although even when using fast reroute a visible disturbance can be noticed on the TV screen. Note that this is also the case when using SONET or SDH protection when carrying a video over a TDM network rather than an MPLS network. This sensitivity to short interruptions is in contrast to voice, where an interruption of a few tens of milliseconds would be unnoticed.

For the most critical broadcast TV traffic, application-level redundancy is sometimes used, in conjunction with the Live-Live scheme discussed in Section 6.7. In this scheme, two copies of the real-time video stream are sent into the network, the replication required to create the two copies being carried out within the video domain. The two streams follow different paths through the network such that they do not share the same fate (e.g. not following the same fiber or duct). At the receiving end, the two streams are sent to a receiver that is capable of seamlessly switching from one stream to the other should the first stream be interrupted. This scheme increases the end-to-end availability of the video flow because traffic remains uninterrupted in the event of a single failure of a link or component within the network. An alternative to this scheme is the automated ingress PE failover scheme previously discussed in Section 6.7. When using a pair of P2MP LSPs to transport the traffic, the fact that the path of each branch of each of the two LSPs is under the full control of the user makes it straightforward to ensure that the two LSPs do not share the same fate.

Codec equipment is commercially available that can convert a video feed into a stream of IP packets or ATM cells and vice versa. Either of these traffic types can be coupled into a P2MP LSP, as described in Section 6.5 of this chapter.

When used for the application of broadcast TV distribution, in many cases each P2MP LSP is dedicated to a single video stream. Some P2MP

LSPs may be relatively short-lived, e.g. existing for perhaps a couple of hours to transmit footage from a sports event to multiple destinations. This is in contrast to traditional point-to-point traffic engineering in service provider networks, in which a typical LSP is very long-lived (on the order of months or years) and would carry a large number of end-to-end traffic flows.

6.9.2 Application of P2MP LSPs to L3VPN multicast

An application of P2MP LSPs is an ingredient of a next-generation (NG) scheme [VPN-MCAST] for carrying IP multicast traffic belonging to Layer 3 VPN customers. As discussed in the Layer 3 VPN chapter of this book (Chapter 7), currently Layer 3 VPNs are the largest deployed applications of MPLS today. Some Layer 3 VPN customers, in addition to using IP unicast applications in their enterprise, also use IP multicast applications. Hence there is a need for service providers to transport this traffic between sites on behalf of their customers. In Chapter 10, we discuss the BGP/MPLS-based solution for carrying L3VPN multicast traffic, of which P2MP LSPs are an important component. The P2MP LSPs are used to distribute multicast traffic between the PE connected to the site containing the source and the PEs connected to the sites containing the receivers.

Depending on the requirements of the service provider and the customers, the P2MP LSPs can either be LDP-signaled or RSVP-signaled. Using RSVP as the signaling protocol for the P2MP LSPs in the multicast VPN scenario brings several advantages:

1. Ability to do bandwidth reservations for the P2MP LSPs and thus ensure that enough resources are available for the traffic carried by the transport tunnels.
2. Precise control over the path followed by each of the sub-LSPs within the P2MP LSP.
3. Flexible optimization options in the core. For example, the service provider can choose to optimize bandwidth consumption by creating a minimum cost tree or can optimize latency by creating a shortest-path tree, as discussed earlier in this chapter.
4. Traffic protection using MPLS fast reroute. The ability to protect multi-cast traffic using MPLS fast reroute is especially useful if the L3VPN customers already enjoy such protection for their unicast traffic and expect equivalent traffic protection in the multicast case.

Traffic from multiple VPNs may be aggregated onto the same P2MP inter-PE tunnel, essentially creating a hierarchy of P2MP tunnels. Traffic is identified as belonging to a particular VPN through the use of a VPN label.

Because all PEs attached to the receivers must agree on the same label, this is an upstream-assigned label.

The context of the upstream-assigned label is provided by the tunnel over which traffic is aggregated. This is illustrated in the example in Figure 6.11. Assume that four VPNs, VPN A, VPN B, VPN C and VPN D, exist in the network where aggregation is used. For VPNs A and C, the source is in a site attached to PE1 and the receivers to PE3 and PE4. For VPNs B and D, the source is in a site attached to PE2 and the receivers are in sites attached to PE3 and PE5. Note that in this example, two different P2MP trees are created, one rooted at PE1 (LSP1) and the other rooted at PE2 (LSP2). Traffic from VPNs A and C is aggregated onto LSP1 and traffic for VPNs B and D onto LSP2. Assume that by chance both PE1 and PE2 pick label L1 as an identifying label, PE1 for VPN A and PE2 for VPN B. If penultimate hop popping (PHP) is used for LSP1 and LSP2, the leaf PEs advertise label 3 as the value for labels L11, L12, L22 and L21. As a result, traffic would arrive at PE3 with label L1 only. In such a case, there is no way to distinguish whether the traffic belongs to VPN A or to VPN B. To avoid this problem, the P2MP transport tunnel, which is rooted at the ingress PE, could provide this context if traffic arriving at the egress could be identified as having arrived on a particular tree. When the tree is instantiated by a P2MP LSP, this identification is not possible if PHP is used, as explained above. However, the identification can be easily achieved if PHP is disabled. For this reason, support for aggregation in the multicast VPN case requires PHP to be disabled on the P2MP transport tunnel.

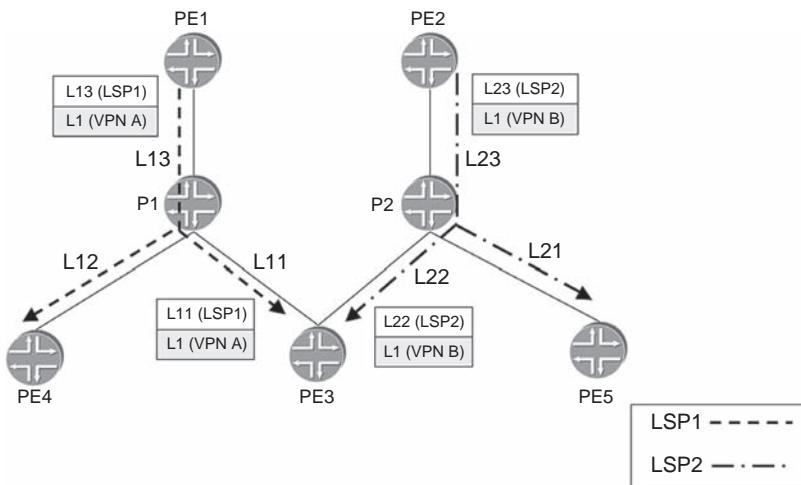


Figure 6.11 Label operations when multiple VPNs share the same P2MP LSP

6.9.3 Application of P2MP LSPs to VPLS

In some Virtual Private LAN Service (VPLS) implementations, multicast traffic arriving at a PE from a customer site is sent to all PEs having members of that VPLS instance attached. Ingress replication is used by the PE. This is wasteful of bandwidth, because in many cases links within the core of the network carry multiple copies of the same packet, each destined to a different egress PE. The advantage, however, is that no multicast state is required in the core of the network. The current scheme may be fine if the volume of multicast traffic is relatively low, but if not then it could be advantageous to the service provider to use a more bandwidth efficient scheme. This scheme is described in [VPLS-MCAS]. As with the NG scheme for L3VPN multicast discussed above, P2MP LSPs are a key component, being used to perform the distribution of multicast VPLS traffic. More details of this scheme are discussed in the VPLS chapter of this book (Chapter 13).

6.10 CONCLUSION

This chapter has discussed a solution to a missing piece of the converged network jigsaw puzzle, namely point-to-multipoint LSPs. Previously, MPLS has not interacted comfortably with multicast, typically coexisting via a ‘ships in the night’ approach. The advent of P2MP-TE means that multicast traffic can enjoy the traffic engineering advantages already offered by MPLS in the unicast case, such as bandwidth guarantees and fast-failover mechanisms. As a consequence of the ‘grand unification’ of the two worlds of MPLS and multicast, MPLS networks are now being used for professional broadcast TV distribution, a very exacting application that was previously difficult to support on an MPLS network.

6.11 REFERENCES

- [CCC] K. Kompella, J. Ospina, S. Kamdar, J. Richmond and G. Miller, *Circuit Cross-connect*, draft-kompella-ccc-02.txt (expired draft)
- [IBC2004] M. Firth, *Challenges in Building a Large Scale MPLS Broadcast Network*, in International Broadcasting Convention (IBC 2004), Amsterdam, September 2004
- [IBC2007] M. Firth, *Media MPLS Around the World*, in International Broadcasting Convention (IBC 2007), Amsterdam, September 2007

[L2mVPN]	R. Aggarwal, Y. Kamite, F. Jounay, <i>BGP Based Virtual Private Multicast Service Auto-Discovery and Signaling</i> , draft-raggarwa-l2vpn-p2mp-pw-02.txt, work in progress
[LDP-UPSTR]	R. Aggarwal, J. L. Le Roux, <i>MPLS Upstream Label Assignment for LDP</i> , draft-ietf-mpls-ldp-upstream-07.txt (work in progress)
[PCEP_P2MP]	Q. Zhao and D. King, <i>Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths</i> , draft-ietf-pce-pcep-p2mp-extensions-10.txt (work in progress)
[RSVP-UPSTR]	R. Aggarwal, J. L. Le Roux, <i>MPLS Upstream Label Assignment for RSVP-TE</i> , draft-ietf-mpls-rsvp-upstream-05.txt (work in progress)
[MPLS-VID]	A. Rayner, <i>Real Time Broadcast Video Transport over MPLS</i> , Paper D1-13, in MPLS World Congress, Paris, February 2005
[P2MP BYPASS]	J.L. Le Roux (ed.), R. Aggarwal, J.P. Vasseur, M. Vigoureux, <i>P2MP MPLS-TE Fast Reroute with P2MP Bypass Tunnels</i> , draft-ietf-mpls-p2mp-te-bypass-01.txt (expired draft)
[P2MP-LDP]	I. Minei (ed.), K. Kompeella, I. Wijnands (ed.), B. Thomas, <i>Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths</i> , draft-ietf-mpls-ldp-p2mp-08.txt (work in progress)
[P2MPWC]	R. Aggarwal, <i>Point to Multipoint MPLS TE Solution and Applications</i> , Paper D1-12, in MPLS World Congress, Paris, February 2005
[RFC4461]	S. Yasukawa (ed.) <i>Signaling Requirements for Point to Multipoint Traffic Engineered MPLS LSPs</i> , RFC 4461, April 2006
[RFC4655]	A. Farrel, J. Vasseur and J. Ash, <i>Path Computation Element (PCE) Architecture</i> , RFC 4655, August 2006
[RFC4875]	R. Aggarwal, D. Papadimitriou, S. Yasukawa (eds), <i>Extensions to Resource Reservation Protocol-Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)</i> , RFC 4875, May 2007
[RFC5331]	R. Aggarwal, Y. Rekhter and E. Rosen, <i>MPLS upstream Label Assignment and Context Specific Label Space</i> , RFC 5331, August 2008

[RFC5332]	T. Eckert, E. Rosen (ed.), R. Aggarwal and Y. Rekhter, <i>MPLS Multicast Encapsulations</i> , RFC 5332, August 2008
[RFC5440]	J.P. Vasseur and J.L. Le-Roux, <i>Path Computation Element (PCE) Communication Protocol (PCEP)</i> , RFC 5440, March 2009
[VPLS-MCAS]	R. Aggarwal, Y. Kamite and L. Fang, <i>Multicast in VPLS</i> , draft-ietf-l2vpn-vpls-mcast-06.txt (work in progress)
[VPN-MCAST]	E. Rosen and R. Aggarwal, <i>Multicast in MPLS/BGP IP VPNs</i> , draft-ietf-l3vpn-2547bis-mcast-10.txt (in the RFC editors' queue, soon to become an RFC).

6.12 STUDY QUESTIONS

1. Describe some of the advantages of RSVP P2MP traffic engineering compared to IP Multicast.
2. Describe the topology options for P2MP trees.
3. Compare and contrast the merits of LDP and RSVP for the signaling of unicast LSPs. Then compare and contrast the merits of LDP and RSVP for the signaling of P2MP LSPs.
4. Imagine a situation where a router, X, is both a transit node and an egress node for the same P2MP LSP. What issue might arise if penultimate hop popping is used on the P2MP LSP?
5. Apart from the situation described in Question 4, describe some other situations in which PHP should not be used with P2MP LSPs.
6. Describe one difference in the way in which a FEC is advertised by LDP in the unicast case compared to the multicast case.

Part Two

7

Foundations of Layer 3 BGP/MPLS Virtual Private Networks

7.1 INTRODUCTION

BGP/MPLS IP VPNs, referred to in short as MPLS L3VPNs or simply L3VPNs throughout this book, are one of the most widely deployed applications enabled by MPLS. When talking about MPLS, it is not fast reroute or traffic engineering that springs to mind, but rather VPN support. In fact, traffic engineering and fast reroute are most often thought about in terms of the benefits that they can provide in the context of a particular service. Perhaps the most popular service is provider-provisioned IP VPNs and the L3VPN solution described in this chapter is the way this service is realized in MPLS networks. For many providers, L3VPNs is the major and sometimes the only driver for deploying MPLS in the network.

VPNs existed long before MPLS. The success of L3 BGP/MPLS VPNs is owed to the scaling and simplicity advantages that the combination of BGP and MPLS brings to VPN scenarios. The L3 BGP/MPLS VPN solution was extended to the Layer 2 space as well, as we will see in the chapters discussing Layer 2 Transport and VPLS (Chapters 12 and 13).

In this chapter we will see how the MPLS VPN solution emerged, introduce its basic principles and concepts and shed light on some of the design decisions taken. We assume the reader has a basic understanding

of both MPLS and BGP. In the next chapters we will look at more advanced topics that arise in the context of L3VPNs. Readers familiar with the BGP/MPLS VPN concepts and basic operation can skip over this chapter and go directly to the advanced topics (Chapters 8, 9, 10 and 11).

7.2 THE BUSINESS DRIVERS

In the simplest scenario, a customer has geographically dispersed sites and requires connectivity between them, in order to run his day-to-day business. The customer does not want to invest in the infrastructure for connecting the sites, nor in the effort of administering this infrastructure. In a competitive world, he or she would rather concentrate on the core business and outsource the task of providing connectivity between sites to the networking expert, the service provider.

From the customer's point of view, the goal is to achieve connectivity with minimum hassle. First of all, connecting the dispersed sites should have the same QoS and privacy guarantees as a private network, and should not require changes to the way the customer's network is configured or run. For example, the customer should be able to use a private address space if he or she chooses. Secondly, the operations that affect connectivity should be easy. For example, adding connectivity to a new site, changing the connectivity between sites or increasing the bandwidth between sites should not require many configuration changes and should be achievable at short notice. Finally, the solution should not require complex routing configuration at the customer's sites.

From the provider's point of view, the goal is to fulfill the customer's expectations while maximizing profits. To fulfill the customer's expectations, the provider must be able not just to provide connectivity but also to extend the service easily and allow customers to use private (and thus possibly overlapping) address spaces. To maximize profits, the provider must support a large number of customers, as well as to be able to support a wide range of customers with respect to the numbers of sites, from customers with a handful of sites to customers with thousands or even tens of thousands of sites. Furthermore, the provider must be able to provide customers with value-added services that can be charged at a premium. Finally, the resources used in providing the service must be shared among the customers.

Based on these goals, let us see why the solution is called a Virtual Private Network (VPN). First, it is a network because it provides connectivity between separate sites. Second, it is private because the customer requires it to have the same properties and guarantees as a private network, both in terms of network operations (addressing space, routing) and in terms of traffic forwarding. Third, it is virtual because the

provider may use the same resources and facilities to provide the service to more than one customer.

In the real world, it is seldom that the goals are crisp and clear from the beginning. What happens instead is that a solution is developed for a given problem. As experience is gained from existing deployments the drawbacks of the solution become apparent and more requirements are added to the ‘goals’ section, yielding a new and improved solution, in an iterative process. Thus it makes sense to look back at the VPN solutions that existed before the BGP/MPLS solution, as they will help us to understand how the current MPLS VPN model emerged and will highlight some of its advantages. In our discussion, we will concentrate on VPNs for which the service provider (SP) participates in the management and provisioning of the VPNs. This type of VPN is known as a provider provisioned VPN (PP VPN).

7.3 THE OVERLAY VPN MODEL

The overlay model is the most intuitive VPN model. If it is connectivity that the customer wants, what can be simpler than connecting the customer sites via point-to-point links between routers at the various sites? The point-to-point links could be Frame Relay or ATM circuits, leased lines or IP-over-IP tunnels such as Generic Route Encapsulation (GRE) or IP Security (IPSec). What is provided is a virtual backbone for the customer’s network, overlaid on top of the provider’s infrastructure. Designated routers at the different customer sites (the customer edge routers, or CE routers) peer with each other and exchange routing information, thus allowing traffic to flow over the links between the different sites.

In this model, the provider is oblivious of the internal structure and addressing in the customer’s network and provides only a transport service. Provisioning the circuits between the customer sites assumes knowledge of the traffic matrix between sites. However, in most cases it is not the traffic matrix that is known but the average traffic sourced and received, thus making it difficult to estimate the bandwidth required. After the circuits are set up, the bandwidth that is not used is wasted, making the solution expensive. One more interesting note on provisioning involves the case where Frame Relay or ATM is used. In this case, increasing the bandwidth between sites may require provisioning of new circuits, which can take a long time to set up.

In the overlay model, the VPN service is provided by the customer routers. A VPN where the intelligence and control are provided by CE routers is called a CE-based VPN. When customers are responsible for configuring and running the CE routers, they are in fact designing and running their own VPN, a task they may not always have the expertise or

desire to be involved in. As a result, the provider may take over the management of the customers' virtual backbone (thus providing a managed VPN service). However, managing the networks of many VPN customers requires managing a large number of CE devices and places a burden on the provider, thus limiting the number of customers that he can service.

Regardless of who manages the customer routers, a model where routers at the customer sites exchange routing information with each other has limitations. Let us take a look at a scenario where there are many sites and a fully meshed configuration. In such a scenario, the number of routing peerings can be very large. This can be a scaling problem for the IGPs due to the large amount of information that may be exchanged when routing changes. Another limitation concerns the amount of configuration that must be done when a new site is added to the VPN. Obviously, the customer router at the new site must be configured to peer with the routers at the other existing sites. Unfortunately the routers at the existing sites must also be reconfigured to establish peering to the new site.

The overlay model achieves the fundamental goals of a VPN. It provides connectivity between customer sites, allows the use of a private address space and ensures the privacy of the traffic between the sites. The functionality is provided by the CE routers and comes at a cost: difficulty in evaluating the bandwidth requirements between sites in cases where the bandwidth must be preprovisioned, the need to manage a large number of customer routers, complex configuration when adding a new site and the need for a large mesh of routing peering.

7.4 THE PEER VPN MODEL

The problems of the overlay model stem from the fact that customer routers peer directly with each other. The peer model attempts to overcome the drawbacks of the overlay model by lifting the requirement for direct routing exchanges between the customer routers. Instead of peering with each other and forming an overlay on top of the service provider's network, CE routers now peer only with directly attached PE routers. As a result, the large mesh of routing peerings between CE routers disappears. From the customer's point of view, routing becomes very easy. The burden of managing the route distribution between the customer sites is now passed on to the provider and the intelligence moves out of the CE routers into the PE routers.

Moving from a CE-based solution to a PE-based one has other benefits as well:

1. Adding a new customer site to a VPN requires configuration of the CE and PE for the new site only, rather than configuration of all the customer's CEs.

2. The number of points of control in the network (i.e. the number of intelligent devices that make meaningful routing decisions) does not necessarily increase for each new customer site added (assuming that more than one CE can attach to the same PE and that the CE can simply run static routing).
3. A single infrastructure is used to service all VPN customers.
4. The exact traffic matrix between customer sites is not required in order to provision bandwidth between customer sites. Instead, it is enough to know the amount of traffic flowing in/out of a site, since the provider's infrastructure is used to carry the traffic.
5. Increasing the amount of bandwidth between sites requires increasing the bandwidth between the CE and PE, rather than upgrading several circuits or leased lines.
6. Simple routing from the CE point of view. Each CE advertises to the PE reachability information for the destinations in the site to which the CE belongs. Optimal routing between the CEs is ensured by the fact that the routing protocols in the provider network ensure optimal routing between the PEs to which these CEs attach.
7. Different routing protocols can run within each one of the different customer sites.

Clearly the PE-based solution is very attractive, assuming it can meet the connectivity and privacy requirements of a VPN; traffic must flow between sites of the same VPN, but is not allowed between sites of different VPNs. Thus the requirement is to constrain the flow of traffic. This can be done either by constraining the traffic at forwarding time or by constraining the distribution of routing information (which implicitly constrains the destinations to which traffic can flow). Let us take a look at two of the early PE-based solutions, as they will highlight some of the problems that MPLS solves in the VPN context.

One of the earliest PE-based VPN solutions ensured traffic isolation between VPNs by constraining traffic at the forwarding time using access lists on the CE-PE links. Access lists operate at the forwarding time on IP packets and allow/disallow forwarding based on fields in the IP header such as source and destination addresses. While conceptually intuitive, a solution based on access lists quickly becomes unmanageable in practice. Implementing complex intersite access policies becomes a challenging task because it is driven by source/destination addresses. As the number of sites and the number of customers grow, the number of access lists increases. In some vendor's implementations, processing large numbers of access lists impacts the forwarding performance of the routers, thus making the solution even less attractive. One last but crucial point is that since access lists operate based on source/destination address information, the model assumes distinct addressing spaces in each VPN. Therefore, the

access-list-based solution cannot service a customer network that uses a private address space, limiting the usefulness of the solution.

The logical next step was to get rid of the access lists by ensuring that traffic arriving at a PE is only destined for one particular VPN. This can be accomplished by connecting every VPN customer to its own dedicated physical or virtual PE router. A dedicated PE router is not enough to ensure that traffic cannot be forwarded from one VPN to another. One must also make sure that there is no routing state on this PE that would allow traffic to be forwarded towards a PE that services a different VPN. The second early PE-based solution used constrained route distribution (based on BGP communities) coupled with dedicated virtual PE routers to ensure traffic isolation between VPNs. In this model, the PE accepts and installs only routes belonging to the VPNs that it services. This model is the basis for the current BGP/MPLS-based VPN solution, and the mechanisms for constrained distribution of routes will be discussed in detail in later sections. However, in its early incarnation, this model also suffered from the limitation regarding private address spaces because of the way routes were advertised in BGP.

An important thing to note about both the early PE-based solutions discussed so far is that forwarding in the provider's network is based on the IP header. Therefore the routers in the provider's network must know how to forward traffic to all destinations in all the VPN customer sites. Forwarding based on IP is a fundamental difference between the early PE-based VPNs and the BGP/MPLS VPN solution. Let us take a look at the impact IP forwarding has on the solution:

1. The use of private address spaces is precluded. Forwarding is done based on the IP header, and there is no way for a router in the middle of the provider's network to differentiate between traffic belonging to different VPNs.
2. The default route cannot be used in the customer VPNs, since there is no way to differentiate between default routes belonging to different customers.
3. The scalability of the solution is limited. The state that must be maintained in all the provider's routers is equal to the number of customer destinations from all VPNs. In a model where provider routers must maintain state for all the VPN routes, the maximum number of customer routes that can be serviced is limited by the number of routes that the provider's core routers can support. The VPN service cannot be scaled beyond the forwarding table size of the core routers.

Clearly, a tunneling scheme would be beneficial in this case, as it would shield the provider's core routers from having to carry the routing information for all customer routes.

7.5 BUILDING THE BGP/MPLS VPN SOLUTION

The BGP/MPLS VPN solution is built on the peer model described in the previous section. This should not come as a surprise, for two reasons. First, we have seen that PE-based VPNs have attractive properties such as simple routing from the customer's point of view and easy addition of new VPN sites. Second, we have seen that early PE-based solutions were limited by the fact that traffic traveled as IP in the core. Tunneling would eliminate this limitation, and MPLS can provide the necessary tunnels.

The BGP/MPLS VPN model was first published in informational RFC 2547 [RFC2547], documenting a solution developed at Cisco. Following the success of 2547 VPNs there was a desire from some service providers to make it into an IETF standard. A new working group was started in the IETF, called ppvpn (for provider-provisioned VPNs). One of the work items of the group was to standardize MPLS VPNs, and the internet draft that resulted from this work was named 2547bis. In the industry today, BGP MPLS/VPNs are often called 2547bis for this reason although in the meantime it has been standardized as RFC 4364 [RFC4364]. The ppvpn Working Group undertook work in both the L2 and L3 spaces, and was later split into the l2vpn and the l3vpn Working Groups [L3VPN] [L2VPN].

In the following sections, we will build the BGP/MPLS VPN solution step by step, hopefully shedding light on some of the design decisions taken. Before we can start, let us remember the goals the VPN solution is trying to achieve:

1. Isolation of traffic between the different VPNs.
2. Connectivity between customer sites.
3. Use of private address spaces in each site.

7.5.1 VPN routing and forwarding tables (VRFs)

Isolation of traffic between the different VPNs means that a customer in one VPN should not be able to send traffic to another VPN. Figure 7.1 shows two customer VPNs, belonging to customer 'white' and customer 'grey'. Each PE has sites from both VPNs attached to it. Assume that each PE is using a single routing/forwarding table. Use of a single table problematic both in the case when the two VPNs use overlapping address spaces (as shown in the figure) and in the case where they use distinct address spaces. If the two VPNs use overlapping address spaces, then forwarding information cannot be installed for both in the common table, as there is no way of distinguishing the destinations in the two VPNs.

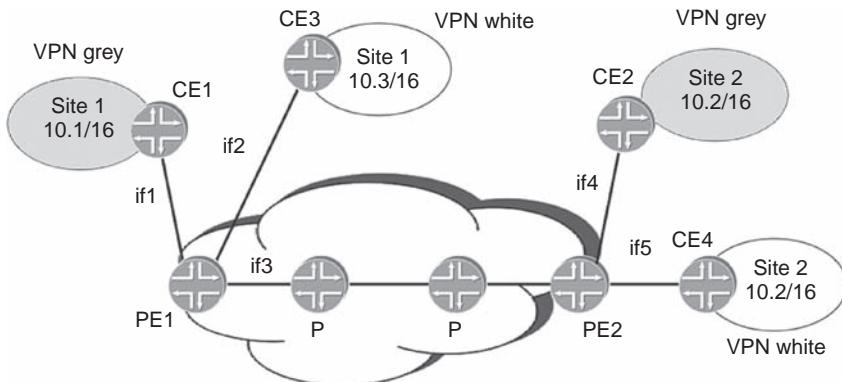


Figure 7.1 A simple network with two customers

If the VPNs use distinct address spaces, it is possible for a host in the VPN white site to send traffic to the VPN grey site, by simply sending IP traffic to the destination in VPN-grey; when the traffic arrives at the PE, the destination address is found in the routing table and the traffic is forwarded to VPN-grey.

Both these problems can be solved by attaching each customer site to its own virtual or physical PE device. Remember, though, from the description of the peer model that the PEs carry the burden of customer route distribution. Increasing the number of PEs with each new customer site is not scalable from either the routing or the network management point of view.

A more scalable solution is to use per-VPN routing and forwarding tables (VRFs), thus maintaining separate information for each VPN. These tables are in addition to the global routing/forwarding table used for non-VPN (Internet) traffic, and they contain routes for the customer's destinations both at the local site and at remote sites. How does the PE know which VRF to use when an IP packet arrives from the customer site? The solution is simple: associate each interface with a particular VRF through configuration. The term 'interface' here does not necessarily mean a physical interface; it could also be a logical one, such as ATM VCI/VPI, FR DLCI or Ethernet VLANs.

For example, in Figure 7.1, the interface if1 connecting PE1 to the CE1 is associated with the VRF for customer grey's VPN and the interface if2 connecting PE1 to CE3 is associated with the VRF for customer white's VPN. When an IP packet arrives over the CE1–PE1 interface if1, the destination of the packet is looked up in the VRF for customer grey and when it arrives on the CE3–PE1 interface if2, it is looked up in the VRF for customer white. When an IP packet arrives over an interface that is not associated with any

VRF, the lookup is done in the global table. In later sections we will see how traffic arriving over core-facing interfaces such as if3 is handled.

The use of multiple forwarding tables at the PE is a necessary condition for allowing support of overlapping address spaces. However, multiple forwarding tables do not automatically ensure that traffic cannot be forwarded from one VPN to another. If the forwarding table for VPN-white were to contain information for destinations in VPN-grey, there would be nothing to prevent a host in VPN-white from sending traffic into VPN-grey. Thus, it is necessary to control the information that is installed in each VPN. This is accomplished by constraining the distribution of routing information, thus constraining the knowledge about reachable destinations.

7.5.2 Constrained route distribution

There are two approaches to constraining routing information per VPN. The first approach is to run separate copies of the routing protocol per VPN, very much like the overlay model, except that the routing peering is between PE routers rather than CE routers. This is not an attractive option from a management and scaling point of view, as the number of routing protocol contexts and the complexity of the routing peerings grow with the addition of each new VPN.

The second approach is to carry all VPN routes in a single routing protocol in the provider's network and constrain the distribution of VPN reachability information at the PEs. This is the method employed in the BGP/MPLS VPN solution, where BGP is the protocol carrying the VPN routes. Here are a few of the properties that make BGP the natural choice in VPN scenarios:

1. Has support for route filtering using the community attribute; thus it can do constrained route distribution. Can attach arbitrary attributes to routes, so the community paradigm can be easily extended.
2. Has support for a rich set of attributes, allowing control of the preferred routing exit point.
3. Can carry a very large number of routes; thus it can support a large number of customer routes.
4. Can exchange information between routers that are not directly connected; thus the routing exchanges can be kept between the PE routers only.
5. Can carry label information associated with routes (we will see later on why this is important).

6. Has support for multiple address families (we will see in the next section why this is required).
7. Can operate across provider boundaries.

7.5.3 VPN-IPv4 addresses and the route distinguisher (RD)

We have seen that BGP has attractive properties as the routing protocol for carrying the VPN routes across the provider's network. However, a BGP speaker can only install and distribute one route to a given address prefix, which is problematic when carrying VPN addresses that are from private (and thus possibly overlapping) address spaces.

The solution is to make the private addresses unique by concatenating an identifying string called the route distinguisher (RD) to the IP prefix, in effect creating a new address family (the VPN-IPv4 address family). The BGP multiprotocol (MP) capability allows BGP to carry prefixes from multiple address families [RFC2858]. This is why sometimes, in the context of VPNs, BGP is referred to as MP-BGP. The address family (AFI) and subsequent address family (SAFI) used for encoding the VPN-IPv4 address family are 1 and 128 respectively. Sometimes when discussing VPN configuration, VPN-IPv4 is referred to as SAFI 128.

An interesting thing to note is that VPN-IP addresses only need to be known by routers in the provider's network, and only by those routers actually involved in exchanging routing information for VPN destinations. The customer is unaware of the existence of VPN-IP addresses. The translation between customer IP routes in a particular VPN and VPNIP routes distributed between provider routers is performed by the PE routers. Before advertising a customer VPN route in BGP, the PE router attaches to it the appropriate RD for the VPN site, transforming it into a VPN-IP route. When receiving a VPN-IP route, the PE converts the route back to plain IP by removing the RD. The association between VPNs and RDs that must be applied to routes belonging to the VPNs is determined through configuration.

Since the RD's task is to make the VPN prefixes unique, it is important to ensure that the RDs themselves are unique. The structure of the RD is driven by the requirement that a service provider should be able to allocate unique RDs without the need for coordination. The RD is an 8-byte quantity consisting of three fields: a two-byte type field, an administrator field and an assigned number field. The type field determines how the other fields are to be interpreted. There are two options for the RD: a combination of a 2-byte AS number and a 4-byte locally assigned number and a combination of a 4-byte IP address and a 2-byte locally assigned number. Both the AS

number and the IP address must be such that they ensure uniqueness of the generated numbers, if several providers are cooperating for providing the VPN service. For example, this can be done by using the AS number assigned to the particular network or an IP address from the public space assigned to the network. In itself, the RD does not contain any information regarding the prefix to which it is attached. In particular, it does not convey information regarding the VPN to which the prefix belongs or the site from which the prefix originates. Since no meaning is associated with the RD, the association of RDs to routes is constrained by two factors: (a) the need for uniqueness and (b) the ease of configuration, management and tracking of RDs. One can imagine different RD allocation policies, with varying degrees of granularity for the RD scope. The most commonly used ones, as a compromise between achieving uniqueness and using a small number of RDs in the network, are using one RD per VPN per PE or using one RD per VPN. Some vendors recommend the use of one RD per VPN, though this is not technically necessary. This in turn creates the perception that the RD somehow helps identify the VPN, when in fact all it does is ensure uniqueness of VPN routes carried in BGP. Using a separate RD per VPN per PE can make troubleshooting easier if the RD is picked in such a way that it can unambiguously identify the PE that originated the route advertisement. It also makes it easy to handle the scenario of overlapping VPNs, where, for example, a particular site can be accessed from two VPNs,¹ since there is no confusion about how to build the RD for the routes in the common site.

To summarize, regardless of how RDs are allocated, their purpose is always the same: to make the VPN routes unique. This is necessary because all VPN routes are carried in the same routing protocol, and BGP can only distribute one route for a given prefix.

7.5.4 The route target (RT)

Let us now go back to the original problem: how to constrain the distribution of VPN routing information between the PEs, thus constraining the routing knowledge and defining which destinations are reachable from each VPN site. The requirement is broader than simply separating the routing information per VPN, for two reasons: (a) customers may require arbitrary and complex connectivity models between their sites and (b) support for overlapping VPNs means that the same route must be present

¹ An example for such a scenario is a case of two companies that partner with each other and therefore require common access to a resource such as a database. The VPN site where the database resides belongs to the VPN of both companies. This scenario will be discussed in detail in the next section.

in several VPN routing tables. In fact, what customers want is a flexible way to define policies that determine the connectivity between different sites. What is therefore needed is a way to do route filtering. For BGP/MPLS VPNs, this is done using BGP extended communities.

A BGP speaker can mark a route by attaching one or more communities to it [RFC1997]. The communities allow the receiver of the advertisement to filter the routes that it wishes to accept. One of the goals of the community attribute is to allow the service provider to allocate values that are locally significant, without the need for external coordination (a similar philosophy to the one we saw in the previous section on RD allocation). The BGP community attribute is a 32-bit string, where the first 16 bits represent the AS number allocated to the provider and the last 16 bits represent a locally assigned number. Since AS number assignments are unique, each provider can manage his or her own number space and define up to 2^{16} distinct values. This means that if a provider uses communities for route filtering in VPNs, he or she is limited to at most 2^{16} customers. Furthermore, the provider must make sure that the values used for VPNs and the values used for other policy functions do not clash. To overcome this limitation, extended communities were introduced [RFC4360]. Extended communities are just like communities, except that they use 32 bits for the locally assigned portion, thus allowing definition of 2^{32} distinct values. Because they provide a structured number space, the extended communities do not clash with communities used for providing other policy functions in the network. In the context of VPNs, the extended community used for route filtering is called the route target (RT). The RT is what accomplishes the constrained route distribution between PEs and ends up defining the connectivity available between the VPN sites.

Here are a few important properties of the RT:

1. One or more RTs can be attached to the same route.
2. Attaching an RT to a route can be done with arbitrary granularity: the same RT can be attached to all the routes in a particular site, or different RTs may be attached to each route. Determining which RT to use is defined through configuration. The use of policy language allows a flexible definition of matching criteria.
3. Up to 2^{32} RTs are available in each AS.

So how does the route distribution work? Let us take a look at the example in Figure 7.2, where sites from two VPNs (white and grey) are attached to the same pair of PEs. The requirement is to provide connectivity between the sites in each VPN. Here are the steps, for advertisements from PE2 towards PE1:

1. Assume that the PE2 knows which customer routes are reachable in each customer site that is attached to it. It may have received

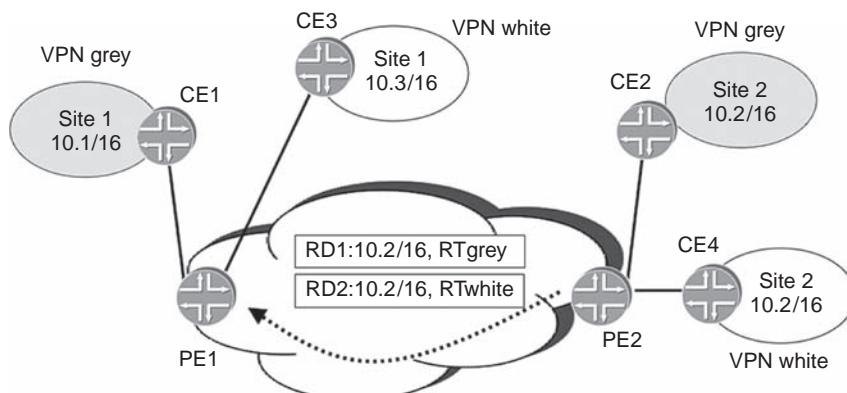


Figure 7.2 Using the RT and the RD

the customer VPN routes from the CE attached to it through a routing protocol or it may have knowledge of these routes through configuration.

2. The goal is to allow other sites in the VPN to forward traffic to these destinations. For this to happen, these routes must now reach the other VPN site; therefore they must be exported into BGP. We have seen in the previous section that the PE translates customer VPN routes into VPN-IPv4 routes before exporting them to BGP by attaching the RD. In addition to this, the route is also tagged with one or more RTs that are determined on a per-VRF basis using policies defined in the configuration. Since these RTs are applied at the time the route is exported, they are sometimes called export-RT. In the example, a single RT is attached to each route.
3. The routes are carried as VPN-IPv4 routes via BGP to the remote PE. The remote PE must decide in which VRF to install this routing information. The decision is done by matching the received routes against locally defined per-VRF import policies expressed in terms of RTs (import-RT). If the route is tagged with any of the RTs specified in the import-RT for a particular VRF, it is stripped of the RD and imported into the VRF and the routing and forwarding state is installed.² In the example, routes tagged with RT-white will be installed in the forwarding table, corresponding to the white VPN.
4. If a routing protocol is running between the PE and the CE, the routes may be advertised to the CE.

²This statement is not entirely accurate. We will see in Section 7.5.6 that one more condition must be satisfied before the forwarding state can be installed.

An interesting question is: what happens to the routes that do not match the RT? There are two options: discard them or keep them. On one hand, it is impractical to keep all advertisements received, because of the scaling limitations such an approach would put on the PE. On the other hand, if these routes are discarded, there is a problem relearning them when the need arises (e.g. following an addition of a new VPN site at a PE or a configuration change to the import-RT). The solution is to discard routes that do not match the RT, but have the ability to ask for them again when the need arises. This is accomplished through the route-refresh capability of BGP, described in [RFC2918]. This capability is negotiated at the session initialization time and allows a BGP speaker to request its peer to resend all the routes it previously advertised to it.³

To summarize, the constrained distribution of routing information is driven by import and export policies defined on a per-VRF basis. These policies are expressed in terms of RTs. A route that is tagged with several RTs is imported in a VRF if any of its RTs match any of the RTs defined in the import-RT of the VRF. Several RTs can be attached to the same route on export and different routes can be tagged with different sets of RTs. In order to ensure that a route is advertised from site 1 to site 2 in a given VPN, the route is tagged with one or more RTs at the time it is advertised by the PE servicing site 1, PE1. The RTs that are attached must be such that the import policy on the PE servicing site 2, PE2, matches it. Thus, the export-RT on PE1 must be such that it contains at least one RT that appears in the import-RT at PE2.

The import and export RTs are the central building block of VPNs, because they express the policies that determine the connectivity between customer sites. Here are a few examples of how the RT can be used to implement some of the common VPN connectivity models.

Full mesh

All sites can communicate with all others directly, creating a ‘full-mesh’ topology. A single RT is used for both the import and the export policies at all the VPN sites; thus all sites end up installing a state for routes from all other sites.

Hub and spoke

Sites can communicate with one another indirectly through a designated site (the hub), creating a ‘hub-and-spoke’ topology. This is useful in the case where it is desired to pass all intersite traffic through a firewall, and

³ The route-refresh capability is not VPN-specific; it was originally added to BGP to support nondisruptive routing policy changes.

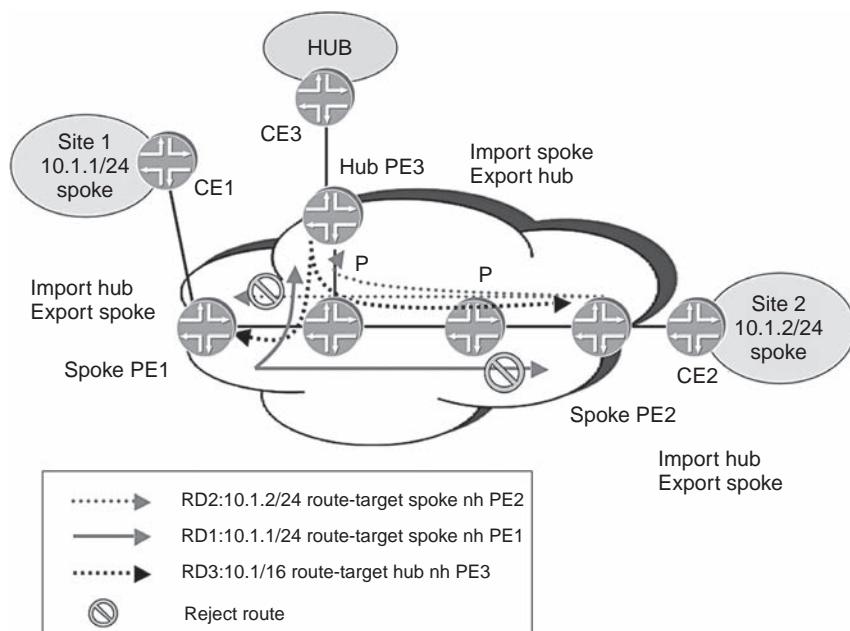


Figure 7.3 Hub and spoke

the firewall resides in a particular site. This topology can be implemented through the use of two route targets, RT-spoke for the spoke sites and RT-hub for the hub site, as shown in Figure 7.3. On export, routes from the spoke sites are tagged with RT-spoke and routes from the hub site are tagged with RT-hub. The import policy for the hub site is to accept routes tagged with RT-spoke, thus learning the route information for all spoke sites. In addition, the hub site readvertises all routes it learned from spoke sites, or a default route, tagged with RT-hub. The import policy for the spoke site is to accept RT-hub only, thus learning the reachability information for all the other spoke sites through the hub. For example, the advertisement for 10.1.1/24, originated by PE1, is rejected at the spoke site serviced by PE2. However, the 10.1/16 advertisement, originated at the hub and tagged with RT-hub, is accepted. The result is that spoke sites do have connectivity to other spoke sites, but the traffic passes through the hub.

Overlapping VPNs – extranets

Designated sites from different VPNs can access a common site in one of the VPNs, in effect overlapping two VPNs. This scheme is often referred to as extranets. For example, a common resource (such as a database or ordering

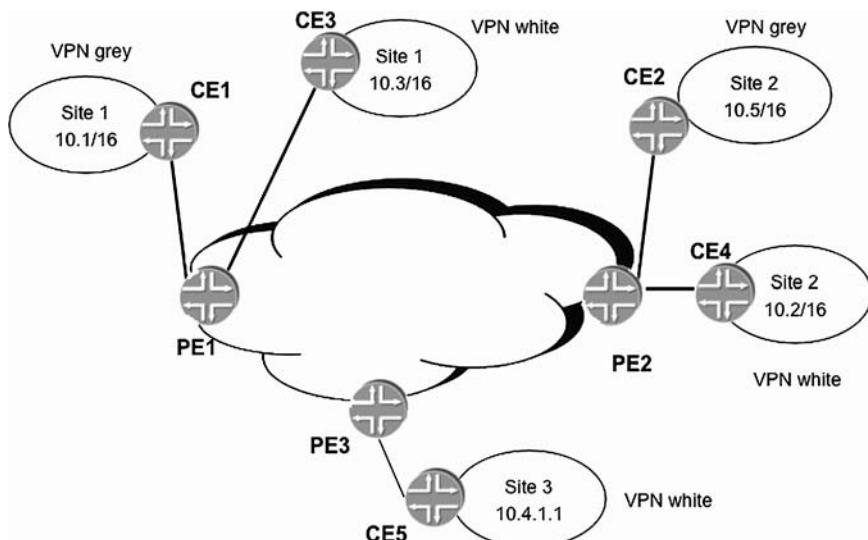


Figure 7.4 Providing access to a common resource from selected sites in two VPNs

system) is used by two companies. The database resides in the VPN of one of the companies, but designated sites from the other company may access it. In Figure 7.4, users in both the white and grey VPN can access a database located in site 3 of the white VPN, with the address 10.4.1.1. The import and export RTs for each VPN are RTwhite and RTgrey respectively. A simple way to achieve the desired access is to tag the route for the common resource with the RTs of both VPNs (RTwhite and RTgrey).⁴ Note that this approach allows connectivity to the shared database from anywhere in the two companies. To provide more selective access the solution is to tag the route with a special route target on export and to import routes with this RT only in those sites that have access to the common resource. In this example, assume that access to the shared database is only allowed to users in site 1 of VPN white and site 1 of VPN grey. The route could be tagged on export with a new RT, RTcommon. The import policy for VPNwhite at PE1 would be modified to accept routes tagged with either RTcommon or RTwhite. Similarly, the import policy for VPNGrey at PE1 would accept routes with either RTcommon or RTgrey. Advertisements for 10.4.1.1 arriving at PE2 would be discarded, as none of the import policies there accepts routes with RTcommon, so none of the users of site 2 in VPNGrey could access the database.

⁴ The connectivity back from the database to the requesting entities in the different sites is achieved by importing the prefixes of the possible requestors in site 3's VRF.

Management VPNs

The goal of a management VPN is to allow the VPN provider to access all the CEs that it services, for network management purposes. A common way to accomplish this is to create a new VPN providing connectivity to all the CEs. The assumption is that because the CEs are managed by the provider, they all have unique addresses. The VRF corresponding to the management VPN has an import policy with a single route target, RT-management. In each of the customer VPNs, the routes corresponding to the PE–CE interfaces are exported with the RT-management route target, in addition to any other route targets necessary in the context of the particular VPN. Thus, only the routes for the PE–CE interfaces are imported in the management VPN. This is a very good example of how only a subset of the routes in a site can be tagged with a particular route target in order to achieve connectivity to only a subset of the destinations.

The route target is a flexible and granular way of constraining the distribution of routing information among sites. With the routing information in place, the next step is to see how traffic is forwarded between sites.

7.5.5 The solution so far – what is missing?

VPN routes are distributed as VPN-IP prefixes between the PEs using BGP. The next-hop of such a route is the address of the advertising PE. Thus, in order to reach the destination, traffic must first be forwarded to the advertising PE. The traffic must be tunneled between PEs, for two reasons: (a) the P routers have no information on the VPN routes and (b) the BGP information is for VPN-IP addresses, which are not routable.

In Figure 7.5, let us take a look at what happens when CE1 in site 1 of the grey VPN sends traffic to host 10.2.1.1 in site 2 of the VPN. The IP packet arrives at PE1 over the CE1–PE1 interface. As a result, PE1 looks up the destination in the VRF for customer grey and determines that the packet must be forwarded to PE2. Let us assume that there is an MPLS tunnel between PE1 and PE2 and that the packet is sent over this tunnel. Upon arrival at PE2, the question is: which VPN does this packet belong to (or over which customer-facing interface should the traffic be forwarded)?

The problem is similar to the one we saw when introducing the VRF concept: when traffic arrives at the PE from the customer site, namely in which VRF should the lookup be done? In that case, the solution was to pick the VRF based on the interface over which the packet arrived from the customer site. Could a similar approach be used for demultiplexing traffic arriving from a remote PE?

In order to use the same paradigm as the ‘incoming interface’ approach, one must maintain several tunnels between the PEs, one for each VPN. At

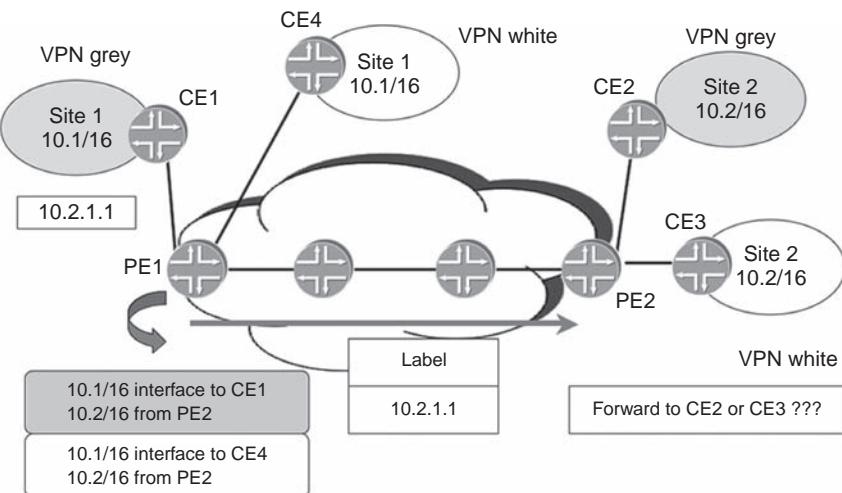


Figure 7.5 Forwarding traffic – what is missing?

the ingress PE (the PE where the VPN traffic enters the provider network), the packets are mapped to the tunnel corresponding to the VPN to which they belong. At the egress PE (the PE where the VPN traffic leaves the provider network), the packets are forwarded to the correct VPN based on the tunnel on which they arrived. However, does this mean that the per-VPN state must be maintained in the core of the provider’s network? The answer is ‘no’. In the next section we will see how the per-VPN tunnels are created and associated with each VPN and how they are carried transparently over the provider’s core.

7.5.6 VPN label

All that is needed in order to create a VPN tunnel with MPLS is to associate a label (the VPN label) with a VPN route. At the forwarding time, the VPN traffic is labeled with the VPN label at the ingress PE and sent to the egress PE. Based on the VPN label, the egress PE can demultiplex the traffic to the correct VPN.

Setting up and maintaining separate tunnels per VPN can only scale if the following two conditions apply:

1. The distribution of the VPN tunnel information is automatic and does not require manual intervention.
2. The P routers do not have to maintain a separate state for each one of the PE–PE VPN tunnels.

The first condition is satisfied by using BGP to distribute the VPN label along with the VPN route information, as explained in the section discussing BGP as a label distribution protocol in the Foundations chapter (Chapter 1).

The second condition is ensured by the label stacking properties of MPLS, which allow the creation of a hierarchy of tunnels, as described in the section discussing hierarchy in Chapter 1 (Section 1.3.1). This is accomplished by stacking two labels: the VPN tunnel label (as the inner label at the bottom of the stack) and the PE–PE tunnel label (as the top label or outer label). Forwarding is always done based on the top label only, so the P routers need not maintain any state regarding the VPN tunnels. The VPN tunnel label is used for controlling forwarding at the PE. This is illustrated in Figure 7.6, where two VPN tunnels corresponding to two different VPNs are nested within the same transport tunnel in the core.

To summarize, a VPN label is advertised along with each VPN-IP route exchanged using BGP. The next-hop of the VPN-IP route is the advertising PE. A PE–PE MPLS tunnel provides connectivity to the BGP next-hop of the VPN route and the VPN label controls forwarding at the egress PE. Three questions arise from this model:

1. What is the policy for VPN label allocation?
2. How is forwarding done at the egress PE?
3. What are the requirements from the BGP next-hop of a labeled VPN-IP route?

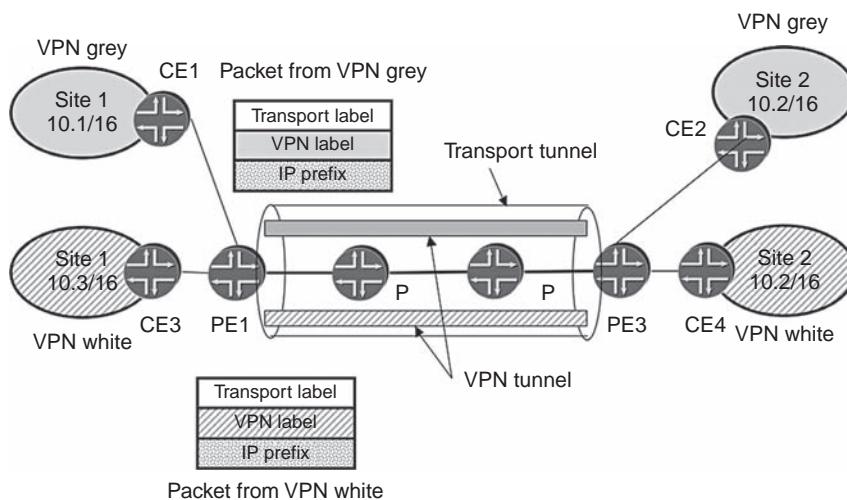


Figure 7.6 VPN tunnels nested within a transport tunnel

Let us try to answer each of these separately below.

1. What is the policy for VPN label allocation?

The purpose of the VPN label is to demultiplex VPN traffic arriving at the PE. From this point of view, any allocation policy, ranging from a separate label per route to a single label per site, fulfils the functional requirement. For most scenarios, one label per site provides the required functionality with the minimum amount of state. At the other extreme, one label per route can provide per-destination statistics and good visibility into the CE–CE traffic matrix. The cost is the extra state maintained, which can make troubleshooting difficult, as we have argued in the Foundations chapter in the context of LDP. Therefore, unless the functionality is needed, an approach that creates less state is preferable.

2. How is forwarding done at the egress PE?

From an implementation point of view, forwarding at the PE can be one of the following two options:

1. An MPLS lookup on the VPN label to determine the appropriate VRF, followed by an IP lookup in that VRF.
2. An MPLS lookup based on the VPN label, in which case the label provides an outgoing interface.

In the first case, the VPN label is used to identify the correct table for the lookup; in the second case, it is used to identify the correct outgoing interface. The end result is always the same: an IP packet is sent towards the correct VPN site. Most vendors support both types of lookup and let the user choose between the two through configuration. The reason is because most forwarding engines only look at the portion of the packet that contains the information for the type of forwarding they perform. For example, if one wanted to set the DiffServ behavior of the traffic based on the IP header, the lookup would need to be done as IP rather than MPLS. The same holds true for any other features where the IP header information is required, such as applying firewall filters or doing accounting based on the IP header.

3. What are the requirements for the PE–PE connectivity?

The BGP next-hop of a labeled VPN-IP prefix distributed by BGP is the address of the PE that originated the advertisement (the egress PE). The intention is to forward the VPN traffic as MPLS, labeled with the VPN label. Thus, the requirement is to have a tunnel to the egress PE, which is capable of forwarding MPLS traffic. The process of finding an MPLS path to the BGP next-hop is often referred to as ‘resolving’ the BGP route. If the tunnel

exists, the route is considered ‘resolved’ or ‘active’, meaning that traffic can be forwarded to its destination and that the route can be readvertised into other routing protocols.⁵ When the tunnel goes away, the route becomes ‘unresolved’ or ‘inactive’ and cannot be used for forwarding. If previously readvertised into a different routing protocol, the route must be withdrawn. From an implementation point of view it is important to note that the process of resolving and unresolving routes should be event-driven rather than based on a timer that scans the state of the LSPs. This is because the time it takes to update the routing state affects the convergence time and ultimately the time during which traffic may be blackholed.

MPLS tunnels are the most intuitive way to forward labeled traffic towards a destination. However, they are not the only option. The IETF defined extensions for carrying MPLS traffic in IPSec, GRE [RFC4023] and L2TPv3 tunnels [RFC4817], thus allowing providers to offer VPN services even over networks that do not support MPLS. This is particularly useful during network migration to MPLS or in cases where providers do not want to deploy MPLS in the core of their network. The development of the extensions to carry MPLS in other tunneling mechanisms is proof of the widespread acceptance and success of the BGP/MPLS VPN solution.

Let us take a look at both the routing and the forwarding state created by the VPN routes for the network in Figure 7.7. For simplicity, we will look at the routing exchanges in one direction only, from PE2 towards PE1, in order to see the forwarding state that is used when forwarding traffic from PE1 towards PE2 (similar exchanges happen in the opposite direction as well):

1. Assuming that a dynamic routing protocol is running between PE2 and CE2 (attached to it), PE2 receives a route advertisement for prefix 10.2.0.0/16 from CE2. If no routing protocol is running, PE2 has this routing information configured as a static route. In Figure 7.7 BGP is assumed to be on the CE–PE link.
2. PE2 attaches an RD, e.g. 65000:1, to the route and allocates a VPN label for this prefix, e.g. label 100. PE2 then exports the labeled route into MP-BGP, with the appropriate RT for customer grey’s VPN.
3. PE2 creates the forwarding state binding the VPN label (100) to the outgoing interface leading to CE2.
4. PE1 receives the MP-BGP advertisement for the VPN-IP route 65 000:1:10.2.0.0/16 with BGP next-hop of PE2 and label 100.
5. Based on the RT, PE1 determines this route must be installed in the VRF for customer grey. PE1 strips off the RD and installs a route to

⁵ Only when the route is resolved does the forwarding state get installed in the VRF. Similarly, only resolved routes are eligible to be readvertised from the PE to the CE.

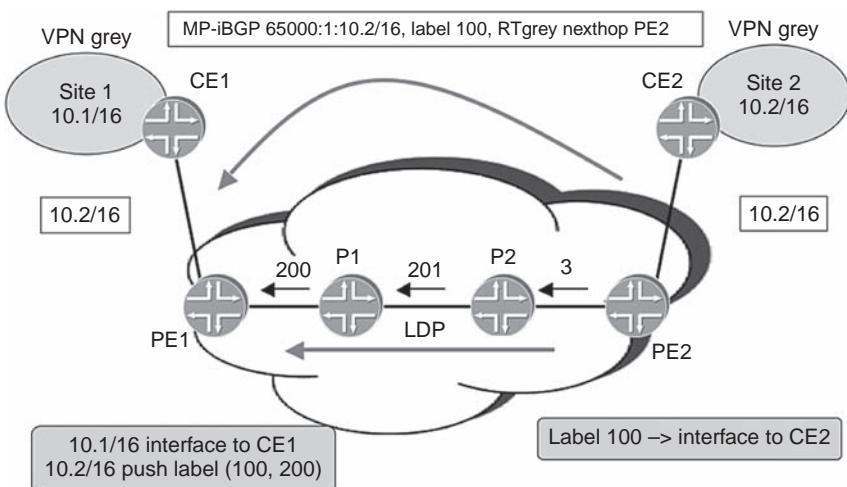


Figure 7.7 The routing and forwarding state created by VPN routes

10.2.0.0/16, with label 100 and next-hop PE2. In order to be able to forward traffic to this destination and make the route active, the ‘next-hop PE2’ information must be translated to the forwarding state. It is not enough for PE2 to simply be reachable from PE1; PE2 must be reachable via an MPLS path. This is necessary because labeled traffic will be sent to PE2. Assuming that an MPLS tunnel exists between PE1 and PE2, set up by LDP, and that the LDP label is 200, the forwarding state is: destination 10.2.0.0/16, push a label stack of (100, 200).

6. Assuming that a routing protocol is running between PE1 and CE1 (attached to it), PE1 advertises the route towards CE1.

Figure 7.8 shows how the state created is used at the forwarding time:

1. PE1 receives an IP packet from CE1 with destination 10.2.1.1.
2. Based on the interface on which the packet is received, the route lookup is done in customer grey’s VRF. The two-label stack (100, 200) is pushed on the packet and the packet is sent towards PE2.
3. The packet arrives at PE2 with a single-label stack of 100 (the LDP label was popped one hop before PE2, at P2 because of penultimate hop-popping).
4. PE2 has forwarding information binding label 100 to the outgoing interface to CE2 and the packet is stripped of its label and forwarded towards CE2.

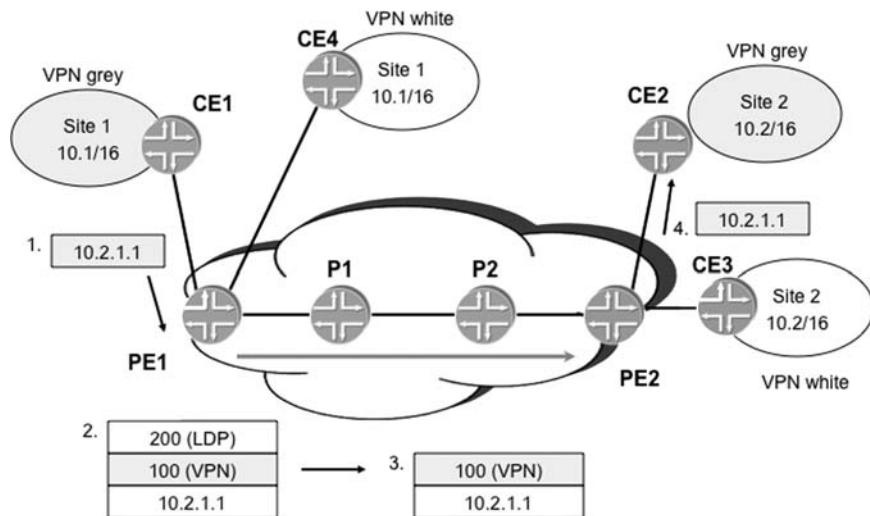


Figure 7.8 Forwarding traffic in a VPN

To summarize, the VPN label allows the PE routers to demultiplex VPN traffic arriving at the PE. BGP provides an automated way to distribute the VPN label by attaching it to the VPN-IP routes. The VPN tunnel information is hidden from the P routers and multiple VPN tunnels are carried inside a single PE-PE tunnel.

7.6 BENEFITS OF THE BGP/MPLS VPN SOLUTION

BGP/MPLS VPNs allow the customer to offload routing between the sites to the provider and enable the service provider to offer value-added services to its customers, such as firewall and authentication. The BGP/MPLS VPN approach allows the provider to leverage the infrastructure to service multiple VPN customers, rather than managing a virtual backbone for each customer. The PE-PE MPLS tunnels are used to carry traffic for multiple VPNs and multiple applications. By hiding the VPN information from the core of the network, the complexity is kept at the PE routers and the service can grow by adding more PE routers when needed.

The property of MPLS that is most powerful in the context of BGP/MPLS VPNs is tunneling. Tunneling using MPLS enables:

1. Building a hierarchy of routing knowledge. Tunneling makes it possible to forward traffic to addresses that are not known in the middle of the network, thus shielding P routers from any VPN knowledge.

2. Identifying traffic as belonging to a particular VPN at the egress point from the provider's network.
3. Providing straightforward and low-cost protection against packet spoofing.

The BGP/MPLS VPN solution builds on existing protocols and technology, only extending the protocols where necessary. It is a great example of the inventing versus reusing paradigm discussed in earlier chapters. The principles discussed in this chapter form the foundation of other MPLS applications, such as L2VPNs and VPLS. However, before exploring them, let us finish the discussion on L3VPN by looking at a few more advanced topics.

7.7 REFERENCES

[L2VPN]	http://ietf.org/html.charters/12vpn-charter.html
[L3VPN]	http://ietf.org/html.charters/l3vpn-charter.html
[RFC1997]	R. Chandra, P. Traina and T. Li, <i>BGP Communities Attribute</i> , RFC1997, August 1996
[RFC2547]	E. Rosen and Y. Rekhter, <i>BGP/MPLS VPNs</i> , RFC2547, March 1999
[RFC2858]	T. Bates, R. Chandra, D. Katz and Y. Rekhter, <i>Multi-protocol Extensions for BGP-4</i> , RFC2858, June 2000
[RFC2918]	E. Chen, <i>Route Refresh Capability for BGP-4</i> , RFC2918, September 2000
[RFC4023]	T. Worster, Y. Rekhter and E. Rosen, <i>Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)</i> , RFC4023, March 2005
[RFC4360]	S. Sangli, D. Tappan and Y. Rekhter, <i>BGP extended communities attribute</i> , RFC 4360, February 2006.
[RFC4364]	E. Rosen and Y. Rekhter, <i>BGP/MPLS IP VPNs</i> , RFC4364, February 2006.
[RFC4817]	M. Townsley, C. Pignataro, S. Wainner, T. Seely, J. Young, <i>Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3</i> , RFC4817, March 2007

7.8 FURTHER READING

[MPLS-TECH]	B. Davie and Y. Rekhter, <i>MPLS Technology and Applications</i> , Morgan Kaufmann, 2000
[RFC4110]	R. Callon and M. Suzuki, <i>A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)</i> , RFC4110, July 2005

7.9 STUDY QUESTIONS

1. In which cases would a customer prefer an overlay VPN model over a peer VPN model?
2. In the MPLS/VPN model, a single BGP session carries the routes from all VPNs between two PEs. If the approach of creating separate BGP instances across the core had been taken, what would have been the number of sessions required to support 1000 VPNs, each with 100 sites (assume no route reflectors are used, and 100 PEs)?
3. At first glance, it would seem that an RD allocated with per-VPN granularity would be sufficient to identify the VPN. What would be the limitations of such an approach?
4. Describe the RT and RD allocation of two VPN customers, A and B, that wish to provide an overlapping VPN as follows. All customers of VPNA are allowed to access the server whose address is 10.1.1.1 at site 1 of VPNA. Customers in site 1 of VPNB are also allowed to access this server, but no other customers of VPN B are allowed to access it.
5. To resolve a VPN route, a tunnel capable of transporting MPLS traffic is required to the BGP next-hop. What happens when this tunnel goes away?

8

Advanced Topics in Layer 3 BGP/MPLS Virtual Private Networks

8.1 INTRODUCTION

The previous chapter laid out the foundations of BGP/MPLS L3VPN. This chapter explores some of the advanced topics that arise in the context of L3VPNs such as scalability, resource planning, convergence and security. All of these require a network-wide view and analysis. Therefore, it is necessary to first discuss two more important components of the VPN solution: PE–CE routing and route reflectors.

8.2 ROUTING BETWEEN CE AND PE

A key concept in the MPLS/VPN solution is that customer routes are kept in a VPN Routing and Forwarding (VRF) table. The VRF is populated with routes learned from the local CE and routes learned from remote CEs as VPN routes. In the previous sections we saw how customer routes are propagated as VPN-IPv4 routes across the provider's network from PE to PE and added to the appropriate VRF. In this section we will take a closer look at how routes are learned from the local CE.

There are several options for a PE to find out about routes from the CE attached to it: static routing, RIPv2, OSPF and BGP.¹ Regardless of how the PE finds out about the routes, it must install them in the VRF associated with the interface to the CE. Thus, a routing protocol must install routes learned over a CE-PE interface in the VRF associated with that interface. From an implementation point of view, this is accomplished by creating separate contexts for the routing protocols per VRF.

So far we have seen that the basic requirement is to have VRF aware routing. The next question is whether to use static or dynamic routing, and which routing protocol to use. One important thing to note is that this decision is local to each PE. In the same VPN, different methods may be employed at different sites, just as different sites may use different routing protocols within the site. This property is particularly useful in a situation where different sites in the VPN are managed by different administrations and may be running different routing protocols within the site (e.g. the sites of a company following an acquisition).

Let us take a look at some of the factors influencing the choice of the PE-CE routing method:

1. *Limitations of the CE device.* For many VPN deployments it is required to have a large number of CE devices with limited functionality. This is true, for example, of a company with small branch offices, where many sites must access a central server. In this case, complex routing capabilities are not required, but due to the large number of CE devices, price is important. Simple, cheap devices often support just static routing or static routing and RIP. This is one of the reasons why static routing is one of the most popular CE-PE protocols in use today.
2. *The routing protocol running in the customer site.* Running a different protocol on the CE-PE link than in the rest of the site means two things: (a) routes must be redistributed from one protocol to another and (b) the site administrator must deal with two routing protocols instead of one. This is why many times customers prefer to run the same routing protocol on the CE-PE link as the one that is already running within the site.
3. *The degree of trust the provider has with regards to the customer's routing information.* When trust is low, the provider may choose to use static routing towards the customer, to shield himself from dynamic routing interactions with the customer.
4. *The degree of control that a protocol gives to the provider.* BGP allows route filtering based on policy and therefore gives the provider control over

¹ At the time of this writing, a proposal to standardize IS-IS as a CE-PE protocol was under discussion in the IETF. The requirement to allow IS-IS as a CE-PE protocol has not been addressed so far, as most enterprise deployments do not use IS-IS as the IGP.

what routes to accept from the customer. Because it is relatively easy for the provider to be protected from a misbehaving customer, BGP is a popular PE–CE protocol.

Let us examine in more detail some of the issues that come up in the context of PE–CE routing:

1. *The use of static routing.* On one hand, static routing is simple and supported by any device. On the other hand, the hosts in the customer site have no knowledge of the reachability of destinations in other sites. If a transport LSP goes down in the core, the VPN routes resolving over that LSP become inactive at the PE. However, this information is not propagated to the customer site, causing a silent failure from the customer's point of view. This is a problem if the customer site is attached to two or more PEs for redundancy (multihomed) and could have picked a different link to exit the site. For this reason, dynamic routing is a requirement when using multihoming.
2. *Running the same protocol on the PE–CE link as in the customer site.* In principle, there is no reason not to use proprietary protocols, such as EIGRP, assuming that both the CE and the PE support them. However, one thing to bear in mind is that the CE and PE are in different domains and often under different administrations. Running any protocol whose algorithm is highly collaborative will make troubleshooting difficult in such an environment, as the customer and the provider site may be managed by different entities.
3. *Protection against a misbehaving CE.* The provider must protect itself from a misbehaving CE advertising a large number of routes. Using static routing avoids this problem altogether. When dynamic routing is used, the options are: (a) configure an upper limit on the number of routes the provider accepts from the customer or (b) use filtering to only accept a prenegotiated subset of prefixes. The concept of limiting the number of routes can be extended to the entire VRF by setting a limit on the number of routes allowed in the VRF, regardless of whether they are received on the local PE–CE routing session or whether they are received from remote PEs.
4. *Using OSPF as the CE–PE protocol.* OSPF uses link state advertisements (LSA) to communicate topology information and therefore has the following properties:
 - (a) LSAs cannot be filtered. Therefore, all LSAs arriving on the CE–PE link must be received and processed at the PE. This implies that there must be a degree of trust between PE and CE and that the CE will not flood the PE with large numbers of advertisements. This trust must exist because the protocol itself has fewer means

to provide this control to the provider (e.g. when compared to a protocol like BGP).

- (b) The LSA type must be maintained as the OSPF routes are advertised between sites. Remember that the customer routing information is carried from one PE to another as VPN-IPv4 routes. Therefore, it is necessary to maintain all the information needed to create the correct type of LSA at the remote site, while the route is carried from one PE to another as a VPN-IPv4 route. This is done by using extended community attributes in BGP. The details of the operation are in [RFC4577] and [RFC4576].

Another issue can arise if a route received from the MPLS/BGP backbone is sent back into the backbone. The problem can be avoided by using OSPF route tagging. Although the use of OSPF may seem complex, it is an attractive option for customers who already run OSPF in their networks and do not want to deploy one more protocol.

5. *Using BGP as the CE–PE protocol.* When using BGP as the PE–CE protocol, there is an EBGP (external BGP) session between the provider and the customer (assuming that the provider and customers are in different ASs). BGP is attractive in this context because of the high degree of control it can give the provider over what routes to accept from the customer. BGP also has good scaling properties, because there are no periodic advertisements of any kind and only incremental updates are sent. For these reasons, BGP is a popular choice as a PE–CE protocol.

An interesting situation arises in this case due to EBGP's built-in loop-prevention mechanism, which ensures that routes are not advertised back into the AS that first originated them. Loop prevention is implemented by appending the AS number of each AS which advertises the route in the AS path attribute attached to the advertisement and requiring a BGP speaker to reject an advertisement if it finds its own AS in the AS path (this is sometimes called an AS path loop). Figure 8.1 shows how this requirement affects route distribution in a VPN context. BGP routes arrive at PE1 from CE1 with the customer's AS number (65 000) in the AS path attribute. The provider's AS number (1) is attached to the AS path as the route is propagated from PE2 to CE2. If the remote customer site serviced by CE2 uses the same AS number as the customer site serviced by CE1, then the route is dropped, since a loop is detected in the AS path (the customer's AS number already appears in the AS path). This problem can be solved in one of the following ways:

- Use different AS numbers in each site. This solution may work if private AS numbers are used by the customer sites, but is not feasible if the customer sites belong to one of the ASs from the assigned number space.

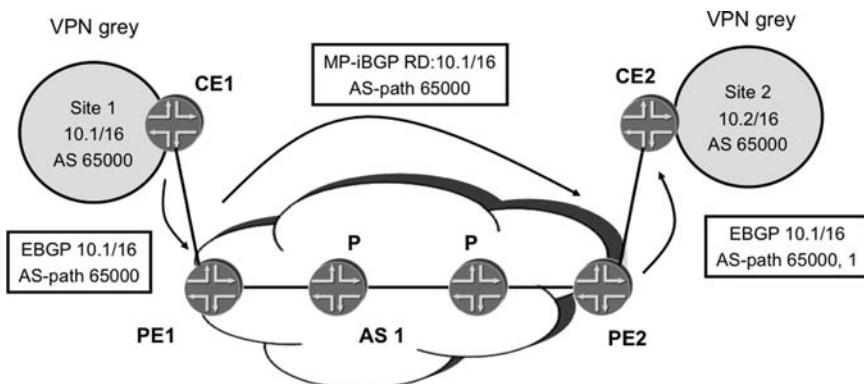


Figure 8.1 Using BGP as the PE–CE protocol may cause problems when the customer sites are in the same AS

- Configure the CE so that it does not enforce the requirement regarding loops in the AS path. This introduces the danger of routing loops caused by other advertisements with AS path loops in them.
- Remap the customer’s AS number to the provider’s AS number as the route is advertised as a VPN route (this is often referred to as AS override). This solution only works if the customer is not readvertising its routes via BGP to other networks from one of its sites, as explained in more detail below.

Let us take a look at a large VPN customer who peers with a service provider in one of its sites and advertises its routes to the Internet. The customer routes learned from remote sites and advertised as VPN routes over the provider’s backbone will have the provider’s AS number in the AS path. However, the customer wants to advertise its routes with its AS number. It makes no sense to include the VPN provider’s AS number in such advertisements.

An elegant solution to this problem is described in [IBGP-PE-CE]. The idea is that BGP attributes received from the customer are stored in a new transitive BGP attribute that functions like a stack. The provider’s BGP attributes are used within the provider’s network. When the route is advertised from the remote PE to the remote CE, the stack is popped in order to discard the provider’s attributes and the original customer attributes are restored. In this way, the customer BGP attributes are carried transparently over the VPN provider’s backbone and the service provider’s attributes, such as the AS number, do not appear in the routes received by the customer.

To summarize, in order to obtain information on CE routes, separate instances or separate contexts of the routing protocols are required per

VRF. The choice of the routing method depends on several factors, among them the CE capabilities, the routing protocol running within the site and the degree of trust that the provider has in the customer's routing information.

8.3 DIFFERENTIATED VPN TREATMENT IN THE CORE

The L3VPN solution described in the Foundations of Layer 3 BGP/MPLS VPNs chapter (Chapter 7) focused on ensuring that traffic from different VPNs is correctly forwarded towards the appropriate CEs once it arrives at the PE devices. In the discussion so far, the service provider network has two important properties: (1) it is oblivious to the VPN membership of the traffic it carries and (2) traffic from all VPNs is forwarded over the same set of transport LSPs in the core. Shielding the routers in the core from per-VPN knowledge and sharing of the core infrastructure for all VPNs provide very good scaling properties to the L3VPN solution.

However, differentiated treatment of the VPN traffic in the core of the network may be desirable in some circumstances. Here are a few examples:

1. *Provide differentiated levels of VPN services, at different price points.* For example, a 'gold' business VPN service would have higher availability than a 'bronze' service. To ensure the appropriate service level, the provider may want to send all 'gold' traffic over protected LSPs in the core, while the LSPs carrying the 'bronze' service would not require such protection.
2. *Isolate one VPN from the rest.* Some customers (such as large banks or government agencies) insist on having a dedicated infrastructure within the VPN provider's network. To be able to leverage an L3VPN solution for such a customer, the transport LSPs used for carrying his or her traffic must be set up over this dedicated infrastructure and must not be available to other VPNs.
3. *Enable services with special requirements.* For example, the service provider may wish to provision VPN voice traffic on a particular set of LSPs that meet certain criteria for latency or failure recovery.

In all the cases above, the requirement for differentiated treatment can be achieved by restricting the set of LSPs which are eligible for use in the route resolution process for a VPN destination (recall from the previous chapter that resolution refers to the process of finding a path to the BGP next-hop of a VPN route).

Let us discuss possible solutions in the context of a network carrying voice and data traffic. Both voice and data destinations are advertised in

BGP, tagged with communities identifying them as ‘voice’ or ‘data’, and voice traffic must be forwarded over LSPs set up with fast reroute. There are two ways for accomplishing this:

1. *Using different BGP next-hops for the different services.* This approach is also referred to as a multiple loopback approach. In this case, voice destinations are advertised with the BGP next-hop NHvoice, while data destinations are advertised with the BGP next-hop NHdata. Transport LSPs with fast-reroute enabled are set up to NHvoice, while the LSPs to NHdata do not have fast-reroute enabled. In this case, the set of LSPs eligible for use in resolution is determined by the address of the LSP endpoint.
2. *Restricting the set of LSPs eligible for use in resolution through policy.* This solution relies on consistent naming of the LSPs, because the set of eligible LSPs is identified by matching against a particular string in the LSP name. For example, all voice LSPs must contain the string ‘voice’ in their name and be set up with fast-reroute enabled. A policy is applied to restrict routes tagged with community ‘voice’ to resolve only over LSPs that contain the string ‘voice’ in their name. This solution requires the implementation of this functionality in the routing equipment (some vendors implement this today).

Note that in all cases, the routers in the core of the provider network are still shielded from having to maintain per-VPN information, thus preserving one of the important scaling properties of L3VPNs.

8.4 ROUTE REFLECTORS AND VPNs

Let us now turn to the last remaining component of the VPN solution: route reflectors. The functional requirements and the tradeoffs regarding the use of route reflectors (RRs) are different in a VPN scenario than in a pure IP (Internet routing) scenario. In both cases the RR provides the following benefits from the configuration management point of view:

- *Reduction in the number of BGP peerings.* A BGP speaker (e.g. a PE) only needs to peer with the RR rather than with all other BGP speakers. Thus, each speaker maintains a constant number of peerings, regardless of the number of BGP speakers in the network.
- *Ease of configuration.* Adding a new BGP speaker only requires setting up a BGP session to the RR, rather than multiple sessions from and to the new speaker.

The differences in the RR use in plain IP service provider networks and VPN networks stem from differences in the routing information carried and how it is advertised, in particular:

1. The routes carried by the PE routers.
2. The number of paths to a particular destination.

In a pure IP scenario, the routing information carried in BGP is the full Internet routing table. Traffic is forwarded as IP and therefore:

1. All routers need to have information for all the destinations.
2. Multiple paths exist for the same destination, because providers typically have several external peerings over which they learn Internet routes.

In this setup, the RR is often used to perform path selection, with the following consequences:

1. Reduction in the number of BGP paths in the network. The RR performs path selection and only advertises to its clients the best path.
2. Reduction in the number of updates generated/processed by each speaker. The RR clients have to process a single update rather than one for each one of the sessions advertising reachability to a particular destination.
3. Requirement for resources on the RR: memory for all BGP paths, CPUs for handling path selection and update processing and forwarding resources to keep the state for all the prefixes. (The RR does path selection based on local criteria; therefore traffic must flow through the RR.)

In contrast, in a VPN scenario:

1. PE routers only need to maintain the routing/forwarding state for the VPNs to which they are connected.
2. Multiple paths for the same destination are not prevalent. Even if the same destination is advertised from several VPN sites, unless the same RD is used in all sites, path selection cannot take place (since the prefixes are treated as distinct).

Here are some observations regarding the use of RRs in a VPN network:

1. It is desirable for PEs to only receive BGP updates for the VPN routes that they will accept and install, rather than all VPN routes. This is in order to conserve CPUs on both the PE and the RR.
2. CPU requirements on the RR. The RR handles the update processing and replication for all route changes in all VPNs that have routes on the RR.

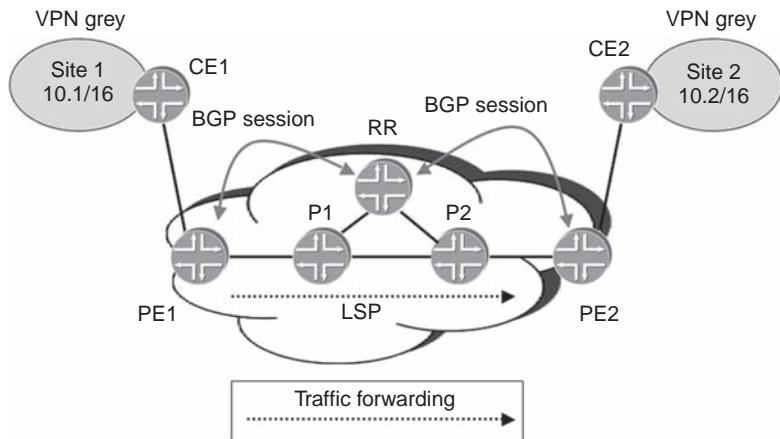


Figure 8.2 RR in a VPN setup, in the case where no traffic is forwarded through the RR

3. There is no requirement to maintain the forwarding state on the RR for all the VPN routes. Assuming distinct RDs, the RR is not performing path selection, so there is no need for traffic to be forwarded through the RR. Many vendors implement this capability in order to conserve forwarding resources on the RR. Figure 8.2 shows a VPN network where the RR is not in the forwarding path. PE1 and PE2 peer with the RR in order to learn the VPN routes, but VPN traffic is forwarded between PE1 and PE2 over the LSP taking the path PE1–P1–P2–PE2. For this to happen, the BGP next-hop must be propagated unchanged by the RR (thus, the advertisement 10.2/16 should arrive at PE1 with next-hop PE2).

However, the RR can become a potential scaling bottleneck, as the one element in the network required to carry the state for all VPN routes. One way to avoid this problem is to partition the VPN routes among several reflectors. A PE router peers only with the route reflector that carries routes from the VPNs in which it is interested. However, this alone is not enough. What if a PE router is required to peer with all the reflectors, because it has customers whose routes reside on each one of the route reflectors? The PE would then receive updates for all the VPN routes in the network. What is needed is a way for the PE to inform the RR which routes it is interested in. In this way the routes are filtered at the RR rather than being propagated to the PE and filtered at the PE.

The solution is for the PE to advertise to the RR the set of RTs for which it is interested in receiving updates (typically, these are the set of RTs used in all the import policies in all VPNs on the PE). As a result, the RR only

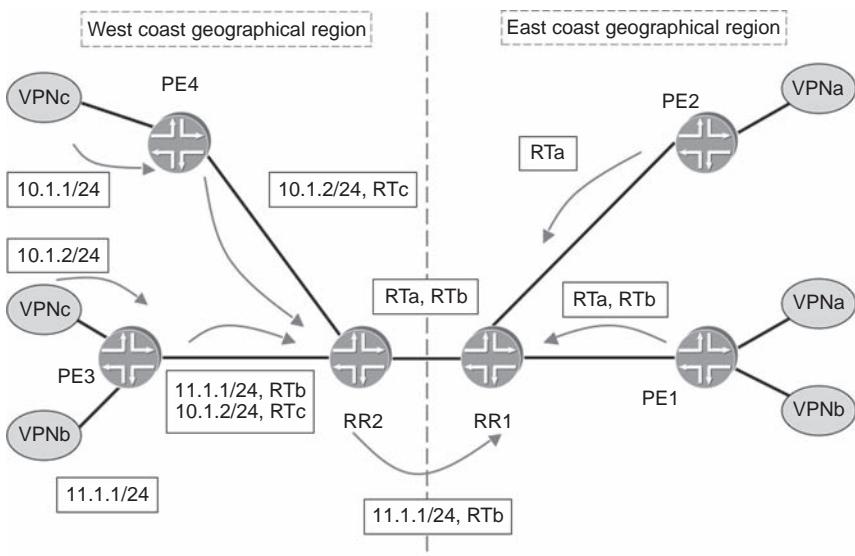


Figure 8.3 Route target filtering

advertises to the PE the routes that are tagged with these RTs, resulting in less work for both the RR, which generates less updates, and the PE, which processes less updates.²

Two mechanisms are available for achieving this goal: outbound route filtering (ORF) [RFC5292] and route-target filtering [RFC4684]. The difference between the two is the scope of the filter advertisements. With ORF, the scope is limited to two directly connected neighbors. For route-target filtering, the filtering information is propagated in the same way as routing information and can cross multiple hops and AS boundaries. ORF was introduced in BGP as a way to minimize the number of updates that are sent between two peers, long before VPNs became popular. However, in the context of VPNs it is useful to be able to propagate the filtering information automatically, which is how route-target filtering came into being.

Let us take a look at an example where route-target filtering is particularly useful. Figure 8.3 shows a US-based network with PEs on both the east coast and the west coast. The network has two route reflectors, RR1 and RR2, one on each coast (for simplicity, let us ignore the extra reflectors used for redundancy). PEs on each coast peer with the local route reflector (in the figure, the solid lines represent connectivity, not necessarily direct links). It is reasonable to assume that most of the VPNs have sites within the same geographical area and that a small number of VPNs have sites on

² One important thing to note is that such a mechanism is not limited to interactions between the route reflector and its clients, and can be used for any BGP peerings.

both coasts. PEs use route-target filtering to inform the RR which RTs they are interested in receiving updates for. For simplicity, Figure 8.3 shows the RT distribution in one direction only, from PE1 and PE2 to RR1 and the route propagation in the opposite direction, from PE3 and PE4 towards RR2 (for simplicity, the RDs are not shown in the route advertisements). The RRs peer with each other and propagate the filtering information they received from their clients. As a result, instead of exchanging all the routes, the RRs only exchange VPN routes for the VPNs that are present on both coasts. In the example, only the routes tagged with RTb are advertised by RR2 to RR1, instead of all the routes advertised by PE3 and PE4 to RR2. RR1 does not receive routes belonging to VPNC from RR2, because RR1 does not have any client PEs that are interested in those routes. This would not be possible with ORF, where the scope of the filtering information is limited to one hop, between the RR and the client.

When hearing about the route-target filtering scheme, people sometimes ask if it would be more efficient for a BGP speaker to simply send all of its routes to each of its peers, rather than having to ‘think’ about which routes each peer needs according to the route-targets received from each. In fact route-target filtering can be implemented in a very efficient manner such that any such overhead is negligible and is far out-weighed by the efficiency savings in not having to send every route to every peer.

In this section we have seen how route reflectors fit in a VPN network. Let us now take a look at the first topic requiring a network-wide view: scalability.

8.5 SCALABILITY DISCUSSION

Scalability is a crucial aspect of any network service. Without scalability, the service becomes the victim of its own success: as the popularity of a service grows, it becomes either technically difficult or economically unsound to provide the service to both new and existing users. When evaluating the scaling properties of BGP/MPLS VPNs one must examine both these aspects and answer the following questions:

1. From a solution point of view, is there a technical upper limit on the maximum number of VPN customers and sites per customer that can be serviced in a given network? In other words, is there a scalability bottleneck in the solution, or can the service be scaled by adding more routers to the network?
2. How does increasing the number of customers impact the cost of the network? In other words, how is the scalability of each of the devices impacted by the growth in the number of customers and routes? At which point must new equipment be purchased?

8.5.1 Potential scaling bottlenecks

The first question is whether there is a scalability bottleneck in the solution. The dimensions in which a VPN service is expected to grow are the number of VPNs, the number of customer sites and the number of customer routes. Growing the VPN service may have implications on:

1. The provisioning burden on the provider.
2. The load on routing protocols.
3. The amount of VPN-related information that must be maintained on the provider's routers.
4. The amount of bandwidth needed to carry the VPN traffic.

What we will show in the following paragraphs is that such limitations do not exist, or can be avoided by splitting the load across several components in the network.

8.5.1.1 The provisioning burden on the provider

BGP/MPLS VPNs are based on the peer model. Thus, adding a new site to an existing VPN requires configuration of the CE router at the new site and of the PE router to which the CE router is attached. The operator is not required to modify the configuration of PE routers connected to other sites or touching the configuration of other VPNs. The PEs learn automatically about the routes in the new site. This process is called 'autodiscovery'. Adding a new VPN to the network does not require changing the configuration of existing VPNs or having knowledge of configuration parameters used for other VPNs.³

Another aspect of provisioning concerns the addition of a PE router to the network. BGP MPLS/VPNs require full meshes of BGP sessions (when no RRs are used) and of PE-to-PE MPLS tunnels between all PEs in the network. Therefore, in principle, when adding a new PE to the network all the existing PEs must be reconfigured to peer with the new PE. However, the provisioning of full meshes is not always a problem. For BGP, the full-mesh problem can be avoided using route reflectors. For MPLS, the full-mesh problem does not exist if the signaling protocol is LDP, as explained in Chapter 1. A full mesh of RSVP tunnels is a concern from a provisioning point of view, but new mechanisms such as the RSVP automesh (discussed in Chapter 15 discussing management of MPLS networks) help to address the issue.

³ The exception to this rule is the allocation of the RD, but even in that case, assuming a combination of IP address and allocated number, the free RD can be picked by looking at the configuration on the individual PEs.

8.5.1.2 *The load on the routing protocols*

Let us examine separately the protocols that run between PE and CE and the protocols that run between PEs. The CE has just one routing peering, regardless of the number of sites in the VPN (assuming that the CE is single-homed). The PE must maintain a peering with potentially each one of the CEs attached to it. As explained earlier in this chapter, this means an individual instance of the routing protocol. When the limit of protocol instances is reached on the PE, a new PE can be added to take over some of the customers. This scaling limitation on the PE affects the cost of growing the network, as we will see in the following section.

Let us now look at the protocol running between PE routers. The provider's PE routers use BGP to distribute the VPN routing information. There are several factors that allow BGP to successfully distribute this ever-growing number of VPN routes:

1. BGP is designed to handle a large number of routes.
2. A PE only needs to maintain information about routes from VPNs that have sites attached to it.
3. Route reflectors reduce the number of BGP sessions a PE must maintain.
4. The route-target filtering scheme described earlier in this chapter can be used to limit the routes sent/received by a PE to only the relevant subset.

8.5.1.3 *The amount of VPN-related information that must be maintained on the provider's routers*

The state maintained on a given router depends on its role in the network:

- P routers do not participate in the routing exchanges for VPN routes, and the VPN traffic is forwarded across them over PE-to-PE MPLS tunnels. Thus, P routers are shielded from any VPN knowledge. The only state that gets created on these routers due to the VPN service is the state related to the PE-to-PE tunnels. This state is a function of the number of PEs and is not related to the number of VPNs or VPN routes. This statement is not necessarily correct for route reflectors, which we will discuss separately.
- PE routers do maintain the state for VPN routes, but only for the VPNs that they service. They also maintain the state for VPN sites, but only for those sites attached to them.
- Route reflectors maintain the state for VPN routes, but we have seen how this problem can be addressed via partitioning in the section dealing with route reflectors.

Thus, no router in the network is required to maintain the state for all VPN routes or sites.

8.5.1.4 The amount of bandwidth needed to carry the VPN traffic

As the number of customers and sites grows, the bandwidth required in the core for carrying the VPN traffic increases. This can be addressed by adding more capacity to the core. When the PE-to-PE tunnels are signaled using RSVP, traffic engineering may be used to obtain better utilization of the existing resources.

To summarize, we have examined the different areas that may be impacted by growing the VPN service and have found that since the BGP/MPLS VPN approach allows distributing the VPN load across several routers in the network, no single router needs to maintain the state for all the VPNs in the network. By splitting the load across multiple routers, BGP/MPLS VPNs can continue to scale as the service grows. Adding new routers implies an additional cost, so the question becomes: how does increasing the number of customers impact on the cost of the network? The following section shows how to determine whether a particular platform is adequate for its role in a VPN deployment and at which point more equipment must be installed. Readers not interested in this type of analysis can proceed directly to Section 8.6, dealing with convergence times.

8.5.2 The cost of growing the VPN network

In order to see how the cost of the network changes with the addition of new customers, one has to analyze at which point it is necessary to add a new router to the network. The requirement to add a new router depends on the scaling limitations of the router. Physical properties such as CPU speed and memory, as well as logical properties such as the convergence time of the routing software, define the capabilities of a router. For a provider building an MPLS/VPN network, it is necessary to express these capabilities in VPN terms.

8.5.2.1 Scalability of PE devices

The number of (logical) interfaces that a PE can support is the first important PE scaling number, because it places an upper bound on the number of VRFs that the PE should be able to support. However, the number of VRFs supported may be smaller than the number of interfaces supported. For this reason, vendors express the scaling properties of potential PE devices by quoting the number of VRFs and number of routes per VRF that the device supports.

For example, a vendor may say that product X supports up to 1000 VRFs with 100 routes each, with RIP as the PE–CE routing protocol. This means that when device X is used as a PE, up to 1000 different VPN sites may be connected to it, and each site may advertise up to 100 routes, assuming that RIP is used for PE–CE route exchanges. When other routing protocols are used, the numbers may change. This should not come as a surprise, since different routing protocols have different resource requirements: a chatty protocol such as RIP requires more resources than plain static routes. It is important to keep in mind that, even for the same vendor, the VPN scaling numbers may depend on the PE–CE routing protocol used. When comparing numbers provided by several vendors, it is important to verify that they assume the same routing protocols on the PE–CE link. In addition, the physical properties of the platform and the software version used also impact the VPN scaling numbers a vendor advertises.

Apart from the number of VRFs and routes that a PE router supports, providers are also interested in the time it takes to bring up a router from an offline state. This time is expressed in minutes from boot-up time until the router is in a fully functional state, routing has converged, the forwarding state has been installed and the CPU utilization has gone down to normal. This time is important, since it determines the service impact the customer sites will experience every time the router is taken offline for maintenance or for software and hardware upgrades. This service impact in turn translates to the SLAs that can be guaranteed to the customer.

An often overlooked scaling dimension of a PE router is the number of routing protocol instances that it can support. If the network design calls for the use of a particular CE–PE routing protocol, the maximum number of instances supported becomes a scaling bottleneck.

The scaling properties of the PE routers in a VPN network determine the load that can be placed on them. This load is a function of the network design, in particular of the number of PE routers used. The fundamental question that a provider is faced with when building a new VPN service is whether the PEs match the load they will carry. In order to answer this question, the requirements on the PE routers must be derived from the requirements on the VPN network. Let us see how this can be done.

Requirements for a network providing a VPN service are expressed in terms of:

1. *Number of customers.* This is the total number of VPNs served by this network.
2. *Number of interfaces (ports) per VPN.* This is the average number of customer circuits or CE devices (assuming a single circuit per CE device) that belong to a given VPN.

3. *Number of routes per CE.* This is the average number of routes injected by each VPN CE device.⁴

The network design determines two parameters:

1. *The number of PEs in the network.* This is either a small number of large PEs or a large number of small PEs. Each approach has its tradeoffs in terms of network manageability and the impact of a PE failure, as well as the scale of the PE-to-PE BGP and MPLS meshes that must be maintained. In reality it is likely that a network would contain both large PEs and small PEs. Large PEs would be used for locations with high customer concentration, while small PEs would be used for locations with small customer concentration.
2. *The number of interfaces in each VRF.* This is a function of two factors:
 - (a) the proximity of the PE device to the customer entry point in the network and
 - (b) the Layer 2 technology used. For example, if Frame Relay is used, the provider can haul circuits belonging to the same VPN to a common PE. By increasing the number of interfaces in each VRF, the provider reduces the total number of VRFs required throughout the network to service a particular customer.

What we will try to do next is to see how these two network design parameters, along with the input of the network requirements, determine the load the PEs must carry. Two issues must be kept in mind when doing such an analysis:

1. The computations are based on averages. Therefore these averages should be maintained within acceptable margins of error.
2. The temptation is to design for the worst case, which results in over-provisioning of the network and increasing its cost. Here is an example of such a design for the worst case. Due to historical reasons, some European countries have a high concentration of economical resources in one or two major cities. For instance, it is safe to assume that almost every VPN in France has circuits in Paris. Thus, assuming a naive analysis, the PE router in Paris would have to maintain the state for all VPNs. At this point, two distinctions must be made: first, the high concentration of customers in the Paris area means more PEs are used

⁴ VPN routing information can be modeled either as a total number of routes per VPN or as the number of routes per CE device. We have chosen the latter for two reasons: (a) it correlates better with PE resource consumption and (b) it is easier to sanity check. One of the most common VPN scenarios is for the CE to interconnect with a branch office that contains a set of networks behind it. In this case there are several routes per CE and the number can be easily validated against the ‘routes per CE’ parameter.

to service them and, second, whatever requirements are extracted for a PE at such a busy spot may not be applicable to other areas in the network.

Let us take an example network and see how the VPN service requirements and the design decisions determine the load on the PE routers. The requirements are: support 10 000 VPNs, with an average of 50 circuits (interfaces) per customer and 200 routes per customer site. The network design assumes 100 PEs and two interfaces in each VRF.

The first question is how many VRFs can be expected on each PE? When doing this computation, one could fall into one of two pitfalls:

1. Assume all PEs have sites from all VPNs (similar to the Paris example we saw earlier) and require support of 10 000 VRFs per PE.
2. Compute the number of VRFs naively by dividing the number of VPNs by the number of PEs, and require $10\,000 / 100 = 100$ VRFs per PE.

In both cases, the problem is that the number of customer sites in each VPN is completely left out from the computation, so the results do not reflect reality.

The number of customer interfaces affects the total number of VRFs in the network. In the example network, each customer has 50 interfaces and the design decision assumes two interfaces per VRF. Therefore, the customer's 50 interfaces translate to 25 separate VRFs, each on a different PE. In this case, a VRF translates to a single customer site ($\text{VRF} = \text{customer_interfaces}/\text{interfaces_in_VRF}$). The number of VPNs in the network must be multiplied by the number of sites in each VPN to obtain the total number of VRFs, yielding $10\,000 \times 25 = 250\,000$ VRFs in the network ($\text{VRF_total} = \text{VPN_total} \times \text{customer_VRFs}$).

Assuming an average distribution of VRFs over PEs, each PE must service $250\,000 / 100 = 2500$ VRFs ($\text{PE_VRFs} = \text{VRF_total}/\text{PE_number}$). Compare this number to the two numbers computed when disregarding the port information: it is four times smaller than the worst case computation and an order of magnitude bigger than the naive computation which implicitly assumed one site per VPN.

By inserting these formulas in a spreadsheet, one can see how different design decisions impact the requirements on the PEs. For example, increasing the number of interfaces in each VRF to five instead of two yields a total of 100 000 VRFs. Thus, only 1000 VRFs need to be maintained per PE. Doing such an analysis can help evaluate different network designs, e.g. when deciding if it is cheaper to haul as many customer circuits as possible to the same PE or to add an extra PE at a particular geographical location, or when comparing the price/performance of different PE devices for a particular design. The same analysis can be applied when increasing the number of customers in the network in order to determine whether new

PE routers must be installed, and how the growth in customer/routes translates to money spent on new PEs.

8.5.2.2 Scalability of route reflectors

We have seen in previous sections that route reflectors have the need for both large memory and a fast CPU. Let us take a look at some of the aspects of evaluating a device for deployment as a route reflector.

One of the most popular metrics service providers rely on is the initial convergence time. Assuming that all PEs are peering with two route reflectors for redundancy, this is the time during which there is a single point of failure in the network. Vendors often include the initial convergence time in the scaling numbers they provide to customers. The initial convergence time is given in minutes and is a function of the number of peers and the number of routes. When comparing convergence times provided by different vendors, it is important to distinguish whether the time is measured until the propagation of all routes has completed, or until the propagation has completed and the CPU utilization has returned to normal.

A more interesting analysis for a potential route reflector is to look at the anticipated load, expressed as the number of updates per second, and at the speed of the update processing on the reflector. The load is a function of the number of PEs peering with the reflector, the number of different VPNs on each PE and the number of VPN route flaps per unit of time. The speed of the update processing is measured as the time between receiving an update from one PE and propagating the update to all relevant PEs. Note that this time is not only affected by the CPU speed but also by software design decisions such as timer-based rather than event-based processing of updates.

Let us revisit the example network from the previous section: 10 000 VPNs, each with sites connected to 25 different PEs. Assuming one route change per minute per VPN, 10 000 updates are received every minute and propagated to an average of 24 PEs (assuming that route-target filtering is applied). This yields the following requirements: process 10 000 updates and generate 240 000 updates every minute. The above analysis is an excellent example of the benefits of deploying route-target filtering. Without route-target filtering, the updates are propagated to all PEs and the number of updates that the reflector must generate increases almost four times, to 990 000. From the same example it is easy to see that decreasing the number of sites by hauling more customer circuits to the same PE reduces the load on the route reflector.

An analysis such as the one above can be applied when evaluating a route reflector deployment to see whether the design can support the estimated load or whether it is necessary to split the VPN routes among several reflectors. Whatever the outcome, it is important to bear in mind

that not using route reflectors at all is also an option. A full mesh of PEs with route-target filtering may be a cheaper option than deploying multiple-route reflectors. By understanding the requirements placed on the different components in a particular network design and the software and hardware features available it is possible to pick the best tradeoffs.

8.6 CONVERGENCE TIMES IN A VPN NETWORK

It is not enough to build a scalable VPN network. One must also make sure that the network meets the customers' expectation: to have the same convergence times as the alternative of buying circuits and running an IGP-based network over these circuits. When discussing convergence for VPN networks, there are two distinct scenarios: (a) a route delete/add in a customer site and (b) a failure in the provider's network, affecting connectivity between customer sites. Let us examine these separately below.

8.6.1 Convergence time for a customer route change

The propagation of a route add/delete in a customer site includes the following steps:

1. Propagation of the route information between the CE and the local PE.
2. Propagation of the route information between the local PE and the remote PEs. This step includes:
 - (a) the update generation on the PE;
 - (b) the processing and propagation of the update at the route reflector (if used);
 - (c) the import of the route into the correct VRF on the remote PEs.
3. Propagation of the route information between the remote PE and the CEs attached to it.

It is important to understand that in order to provide comparable service to an IGP-based network, the above steps must be performed in an event-driven way rather than a timer-driven way. Some implementations use periodic timers to scan for changes and process them (also known as scan timers). Such implementations are open to a maximum delay of the sum of the scan timer intervals, in addition to any processing and propagation delays that are incurred. Typically scan timers are in the orders of seconds, which can add up to tens of seconds of end-to-end delay. Therefore, when evaluating PE devices for a VPN network, it is important to examine the software behavior in this respect in order to gauge the network-wide convergence time.

8.6.2 Convergence time for a failure in the provider's network

In the absence of mechanisms such as fast reroute, a link or router failure in the provider's core can affect the PE–PE LSPs and affect VPN connectivity. The convergence time in this case is defined as the time until the CE routers find out about the fact that a set of destinations has become unreachable. This time is made up of:

- The time it takes to detect of LSP failure and propagate the information to the PE.
- The time it takes to translate the LSP failure into a route withdrawal at the PE.

The LSP failure detection time is largely dependent on the label distribution protocol used. In the Foundations chapter (Chapter 1) we saw a scenario where LDP used in the independent control mode would yield a silent LSP failure that would never be detected. For a protocol such as RSVP, some types of failure would only be detected after the cleanup timer had expired, typically on the order of minutes. Liveness detection mechanisms (such as the ones described in Chapter 15 discussing management of MPLS networks) can help detect LSP failures within a bounded amount of time. The second part of the convergence time depends on whether the remote PE reacts to the LSP going down in an event-driven or in a timer-driven way. As explained in previous sections, for a timer-driven approach, the scan time interval adds to the total convergence time.

To summarize, VPNs are expected to give customers similar service to that of an IGP-based network. However, route propagation and failure detection may take far longer in a VPN. By understanding the interactions that take place, providers can get a better idea of the requirements from the software they deploy, as well as of the tools and the protocol choices they make.

8.7 SECURITY ISSUES

L3VPNs must provide the same security assurances as the alternative of connecting dispersed sites with circuits at Layer 2. The first security concern is the separation of traffic between VPNs. We have already seen that the L3VPN solution has built-in mechanisms for isolation of addressing plans, routing and forwarding. However, since L3VPNs operate over a shared infrastructure, additional concerns arise:

- Can traffic from one VPN 'cross over' into another VPN?
- Can a security attack on one VPN affect another VPN?

- Can a security attack against the service provider's infrastructure affect the VPN service?

Let us examine these separately below and see how the problem can occur and how it can be avoided.

8.7.1 Can traffic from one VPN 'cross over' into another VPN?

One of the most frequent configuration errors is to plug in the CE interface into the incorrect port on the PE. Thus, instead of belonging to VPN A the new site belongs to VPN B (recall that membership in a VRF is based on the interface).⁵ Following such a misconfiguration it becomes possible to send traffic from one VPN to another, especially if the same addressing plan is used in both VPNs. If a routing protocol is running between CE and PE, the problem can be easily avoided by enabling authentication for the routing protocol exchanges. In the case of a misconfiguration, the routing session does not establish and routes are not exchanged between PE and CE.⁶ In Chapter 15 discussing management of MPLS networks we will see a mechanism for detecting this misconfiguration even if no routing protocol is used on the PE-CE link.

Because the PE-CE link extends outside the provider's network and may cross shared-access media it is a natural target for attackers. Securing the routing protocol through authentication prevents an attacker from introducing incorrect routing information. The same can be achieved by setting up routing protocol policy on the PE. For example, the PE may limit the routes it accepts from the CE to the subnets used in the customer VPN. Setting up firewall filters on the CE-PE interface is another popular way to defend against attacks over this potentially insecure interface. For example, the filter would reject any traffic whose source address is not from the address space used in the customer VPN.

8.7.2 Can a security attack on one VPN affect another VPN?

Assuming that one VPN is compromised, other VPNs may be affected if the attack is such that it affects the PEs servicing the VPN. This is the case, for example, if the affected customer floods the PEs with traffic or with route advertisements. The way to protect against such a scenario is to limit

⁵ A similar configuration error can also happen for an L2VPN.

⁶ This works as long as different passwords are used for each session. If a default password is employed throughout, authentication will not detect the misconfiguration.

the PE resources that a customer can consume. We have already seen that the PE can protect itself against a misbehaving CE by limiting the number of routes it accepts from the CE. Similarly, firewall filters on the PE–CE interface can rate-limit the amount of traffic that the CE can send, thus limiting the damage that an attacked VPN can inflict on other VPNs.

8.7.3 Can a security attack against the service provider's infrastructure affect the VPN service?

All VPN traffic is carried across the provider's core over the same infrastructure. Therefore, an Internet attack on the provider's core can impact the availability of the VPN service. To protect this infrastructure, providers conceal the core using two techniques: hiding the internal structure of the core and filtering packets entering the core. The goal is to make it hard for an attacker to send traffic to the core routers. Hiding the internal structure can be accomplished by: (a) using a separate address space inside the core (and not advertising these addresses outside the network) and (b) manipulating the normal TTL propagation rules to make the backbone look like a single hop.

The shared infrastructure over which L3VPNs are built pose additional security challenges. We have seen just a few of the issues that can arise either from innocent misconfigurations or from malicious attacks. The responsibility for preventing such attacks cannot be placed on the provider alone or on the customer alone. Instead, it is shared between the provider and the customer.

8.8 QoS IN A VPN SCENARIO

From a QoS point of view, a BGP/MPLS VPN has to provide at least the same guarantees as a private network. Because it is sold as a premium service with QoS guarantees, the customer expectations from the QoS performance of a BGP/MPLS VPN are high.

As seen in the Traffic Engineering chapter (Chapter 2), QoS guarantees can be readily translated into bandwidth requirements. The question is, how are these requirements expressed? When discussing bandwidth requirements, two conceptual models exist:

1. *Pipe model.* A pipe with certain performance guarantees exists between two specific VPN sites. Therefore, the customer must know the traffic matrix and must translate it into a set of pipes that meet these requirements. For example, for a VPN where branch offices connect to a central site, the amount of traffic between each branch office

and the central site must be known. This approach is difficult to implement in practice because the traffic matrix is typically not known in advance. Furthermore, changes to connectivity (such as two branch offices starting to exchange traffic) require changing the pipe definitions.

2. *Hose model.* The bandwidth requirements are expressed as the aggregate bandwidth going out of and coming into a site. (This is similar to a hose because traffic is ‘sprayed’ from one point to multiple points.) It is much easier to define the bandwidth requirements in this case because the estimate is for the aggregate traffic rather than individual flows and the amount of traffic in/out of the site depends on its size and importance. Furthermore, such a model can easily accommodate changes to connectivity.

Once these requirements are defined, how are they implemented? In an MPLS network, a pipe can be implemented as an LSP and a hose can be implemented as a collection of pipes (a set of LSPs).

Does this mean, however, that separate LSPs must be set up for each and every VPN? The question is further complicated by the fact that within the same VPN several levels of service are offered, e.g. voice and data, requiring different QoS behaviors. Do the resources for each of the service levels come from the resources allocated to the particular VPN or from the resources allocated to the particular service level across all VPNs?

The answer depends on the goals that the provider is trying to achieve and is a tradeoff between the amount of state that must be maintained in the core of the network and the degree of control that the provider has over the different allocations. At one extreme, a set of PE–PE LSPs is shared by traffic from all service levels, belonging to all VPNs. At the other extreme, separate LSPs are set up for each service in each VPN. The number of LSPs that are set up is proportional to the number of PEs in the first case and to the number of PEs, VPNs and services within each VPN in the second case. Typically, providers deploy PE-to-PE tunnels that are shared across VPNs, because of the attractive scaling properties of this approach, but sometimes per-VPN tunnels can be set up, to ensure compliance with particular customer requirements. Refer back to Section 8.3, where we have looked at some of these scenarios and at how the traffic can be mapped to different sets of transport tunnels in the core.

How does the traffic receive its QoS guarantees as it is forwarded through these tunnels? Most providers offer QoS in IP/MPLS networks using DiffServ. In this model, customer traffic is classified at the local PE. When sending the traffic towards the remote PE over the PE–PE LSP, the EXP bits are set to ensure the correct per-hop behavior in the provider’s core. However, one cannot rely on DiffServ alone to provide QoS; it is also necessary to have enough bandwidth along the path taken by the PE–PE LSP to ensure the correct per-hop behavior. This can be done by either

overprovisioning the core or by setting up PE–PE LSPs with bandwidth reservations. In both cases, the assumption is that the traffic sent by the customer stays within the estimates used for the inter-PE LSP. Therefore, many providers police this traffic to ensure compliance.

In the chapter discussing DiffServ Aware Traffic Engineering (Chapter 4), we have seen that LSPs with bandwidth reservations cannot solve the problem of limiting the amount of voice traffic on links (a necessary condition to ensure bounds on jitter). To solve this problem, DiffServ-TE LSPs can be used to reserve bandwidth on a per-class basis. For example, instead of a single PE–PE LSP, two LSPs can be set up, one for voice and one for data traffic. Customer traffic is mapped to the correct PE–PE LSP based on the DiffServ classification.

8.9 IPv6 VPNs

The discussion in the previous chapter assumed that the customer of the L3VPN service uses IPv4. There is increasing interest in being able to support IPv6 customers using the L3VPN model. One driver for this is the fact that the US Government has mandated the use of IPv6 in all of the networks belonging to US Federal Agencies [OMB]. Because some US Government departments buy L3VPN services from service providers, those service providers need to support IPv6 transport over the L3VPN service. There is also interest in the scheme from operators of mobile telephony networks. Increasingly, the transport infrastructure is based on MPLS with VPNs being used to provide separation between the different traffic types. This, coupled with the likely introduction of IPv6 addressing in the future for parts of the mobile infrastructure, including the handsets, makes IPv6 L3VPN a useful internal infrastructure tool for these networks.

The IPv6 L3VPN scheme [RFC4659] was designed to be very similar to the IPv4 L3VPN scheme in terms of mechanisms and operational model. The following mechanisms are common to both schemes:

- A PE achieves separation between different customers by storing their routes in separate VRFs.
- Route targets are used to achieve constrained route distribution.
- Route distinguishers are used to disambiguate overlapping addresses.
- Multiprotocol BGP is used to carry labeled routes between PE routers.
- Route Target Filtering can be used to ensure that a PE only receives routes that it is interested in from its BGP peers.

The infrastructure associated with IPv6 VPN is illustrated in Figure 8.4. An IPv6 routing protocol is used between a PE and the locally attached CE routers so that the PE can populate the VRF with IPv6 prefixes learnt

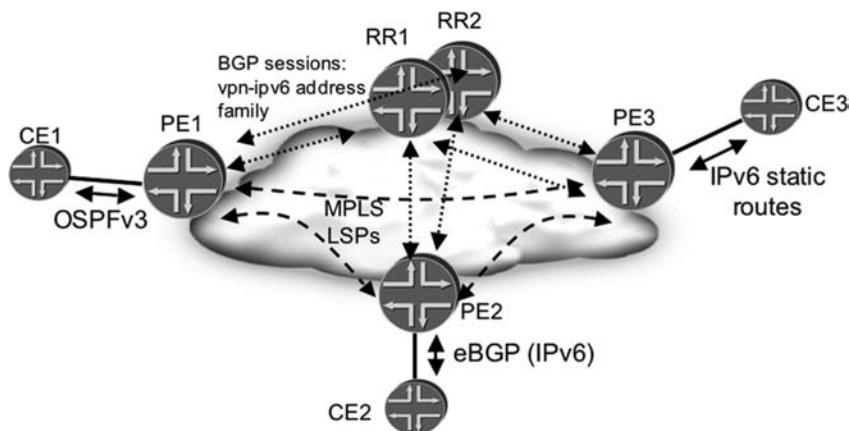


Figure 8.4 IPv6 L3VPN infrastructure. The solid arrows denote IPv6 protocol sessions between a PE and a CE. The dashed lines denote MPLS LSPs. The dotted lines denote BGP sessions

from the CE routers. The protocol used, for example, could be OSPFv3 or BGP with MP v6 support. Alternatively, IPv6 static routes can be configured on the PE and CE. In the forwarding plane, a CE router sends IPv6 packets to the PE router. In order to transport the packets to a remote PE, a transport tunnel is required. As with IPv4 L3VPNs, this can be an MPLS LSP, as illustrated in the figure, or could be a GRE tunnel or IPsec tunnel. In the case where an MPLS LSP is used, the MPLS LSP could be signaled using either the IPv6 version or the IPv4 version of the signaling protocol (RSVP or LDP). The label stack is analogous to that in IPv4 L3VPN, typically consisting of an outer transport label and an inner VPN label. In current deployments, the MPLS LSP is usually signaled using the IPv4 version of the signaling protocol as many vendor implementations do not support IPv6 signaling for MPLS LSPs. In any case, using IPv4-signaled MPLS LSPs means the core routers do not need to be upgraded to support IPv6.

The M-BGP sessions used to carry labeled routes between PE routers (typically via a route reflector) can run over IPv4 or IPv6 although in most current deployments they run over IPv4. In the previous chapter, we mentioned that in order to exchange IPv4 prefixes in IPv4 L3VPNs, the VPN-IPv4 address family is used. This address family has an AFI value of 1 and a SAFI value of 128. By analogy, for the IPv6 L3VPN case, a VPN-IPv6 address family has been defined, having an AFI value of 2 (meaning IPv6) and a SAFI value of 128. Figure 8.5 shows a schematic comparison between the VPN-IPv4 and the VPN-IPv6 cases. In the figure, the format of the BGP next-hop for the VPN-IPv6 case is an IPv4-mapped IPv6 address.

IPv4 VPN case:		IPv6 VPN case:	
AFI = 1, SAFI = 128		AFI = 2, SAFI = 128	
Route Target: Blue VPN		Route Target: Blue VPN	
Next-hop = 192.168.0.4		Next-hop = ::ffff:192.168.0.4	
NLRI:		NLRI:	
Length		Length	
Route Distinguisher:PE1		Route Distinguisher:PE1	
10.1.1/24		2001:db8:11:11::/64	
Label = 100000		Label = 100000	

Figure 8.5 Schematic comparison of the BGP updates for the IPv4 L3VPN case and the IPv6 L3VPN case

This format is used for the case where the transport of the IPv6 traffic is to be over IPv4 (e.g. an MPLS LSP signaled by IPv4).

Because of the similarities between the IPv4 L3VPN scheme and the IPv6 L3VPN scheme, it is relatively straight forward to introduce IPv6 L3VPN service onto a network that already supports IPv4 L3VPN service. Some implementations allow IPv6 and IPv4 VPN service in the same VRF, as illustrated in Figure 8.6.

In the figure, CE1, CE2 and CE3 are in the same VRF. CE3 uses IPv4 only, and so an IPv4 routing protocol is used between CE3 and the PE. CE1 uses IPv6 only, and so an IPv6 protocol is used between CE1 and the PE. CE2 uses IPv6 and IPv4, and so both IPv6 and IPv4 versions of a routing protocol are used between CE2 and the PE. Note that the L3VPN scheme does not in itself provide translation between IPv4 and IPv6, so either CE1 or CE3 have no need to communicate with each other or some translation scheme is provided within the customer domain.

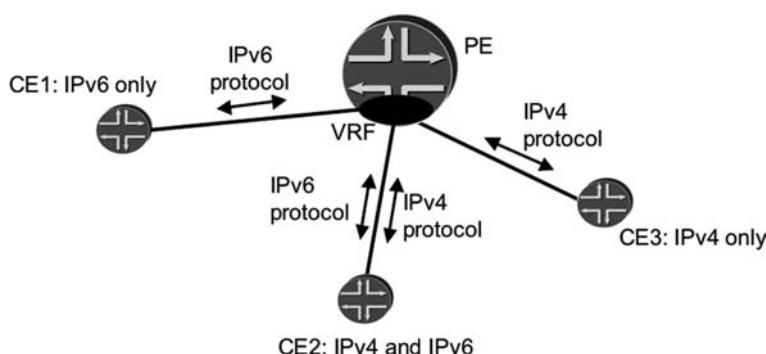


Figure 8.6 Using IPv4 and IPv6 in the same VRF

If desired, the same BGP session can be used to carry both the VPN-IPv4 routes and the VPN-IPv6 routes or separate BGP sessions could be used. Also, the same MPLS LSPs can be used to carry both the IPv4 and the IPv6 L3VPN traffic.

8.10 CONCLUSION

In this chapter we have explored some of the advanced topics that arise in the context of L3VPN: the use of route reflectors, VPN scaling numbers, security issues, QoS and IPv6. However, the discussion so far has been restricted to setups where the VPN spanned a single AS and where the VPN customers were enterprises (implying that each customer has only a small number of routes). In the next chapter we will look at more advanced setups, where VPN customers may themselves be either Internet service providers or VPN providers and where a single VPN may span across several ASs.

8.11 REFERENCES

- [IBGP-PE-CE] Marques, Raszuk, Patel, Kumaki, Yamagata *Internal BGP as PE-CE Protocol*, draft-marques-l3vpn-ibgp-02.txt (work in progress)
- [OMB] US Government Office of Management and Budget Memorandum, <http://www.whitehouse.gov/omb/memoranda/fy2005/m05-22.pdf>
- [RFC4576] E. Rosen, P. Psenak and P. Pillay-Esnault, *Using an LSA Options Bit to Prevent Looping in BGP/MPLS IP VPNs*, RFC 4576, June 2006
- [RFC4577] E. Rosen, P. Psenak and P. Pillay-Esnault, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP VPNs*, RFC 4577, June 2006
- [RFC 4659] J. De Clercq, D. Ooms, M. Carugi, F. Le Faucheur, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*, RFC 4659, September 2006
- [RFC4684] P. Marques *et al.*, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*, RFC 4684, November 2006.
- [RFC5292] E. Chen and S. Sangli, *Address Prefix Based Outbound Route Filter for BGP-4*, RFC 5292, August 2008

8.12 FURTHER READING

[ADV-MPLS]	V. Alwayn, <i>Advanced MPLS Design and Implementation</i> , Cisco Press, 2001
[L3VPN]	http://ietf.org/html.charters/l3vpn-charter.html
[MPLS-TECH]	B. Davie and Y. Rekhter, <i>MPLS Technology and Applications</i> , Morgan Kaufmann, 2000
[MPLS-VPN]	I. Peplnjak and J. Guichard, <i>MPLS and VPN Architectures</i> , Cisco Press, 2000
[RFC2918]	E. Chen, <i>Route Refresh Capability for BGP-4</i> , RFC 2918, September 2000
[RFC4111]	L. Fang, <i>Security Framework for Provider-Provisioned Virtual Private Networks (PPVPNs)</i> , RFC 4111, July 2005
[RFC4364]	E. Rosen and Y. Rekhter, <i>BGP/MPLS IP VPNs</i> , RFC 4364, February 2006
[RFC4381]	M. Behringer, <i>Analysis of the Security of BGP/MPLS IP VPNs</i> , RFC 4381, February 2006

8.13 STUDY QUESTIONS

1. Discuss some of the BGP properties that make it a desirable PE–CE routing protocol from the service provider’s point of view. Give examples of how these properties can be used.
2. Describe what happens when a transport LSP goes down in the core, when a dynamic routing protocol is used between the provider and the customer. How is the interaction different in the case of multi-homing? To avoid a route flap towards the customer, what are the requirements on the PE device?
3. Differentiated VPN treatment in the core can be achieved by either using different next-hops for the different services or by having policies matching on LSP names. What requirements does the next-hop solution impose on the deployment?
4. Describe some of the advantages of using Route Reflectors in L3VPN deployments.
5. Discuss the advantages of route target filtering versus partitioning of prefixes to RRs from a point of view of network growth.
6. Revisit the network in Section 8.5.2.1 and evaluate the CPU load on the PEs if instead of an RR deployment a full mesh of BGP sessions was deployed. What is the impact of a busy CPU on PE on the rest of the PEs in the network, and on the convergence time?

7. Provide one deployment example where route-target filtering provides an advantage and one where it does not.
8. Discuss how MPLS Diffserv can be coupled with flexible resolution of VPN routes over a restricted set of LSPs to provide QoS for a VPN setup.
9. Describe some of the drivers behind IPv6 L3VPNs.
10. Suppose a service provider already offers IPv4 L3VPN service and then decides to offer IPv6 L3VPN service in addition. What does this entail in terms of changes to the infrastructure?

9

Hierarchical and Inter-AS VPNs

9.1 INTRODUCTION

In the previous two chapters we have seen the basic operation of BGP/MPLS L3VPNs, where a service provider offers a VPN service to an enterprise customer as replacement for the mesh of circuits connecting geographically dispersed locations. The problem of connecting geographically dispersed locations is not unique to enterprise customers. Carriers may have similar problems, especially following an acquisition of a new network from a different carrier. In this case, connectivity is needed to the new network and could be (and sometimes is) accomplished by buying L2 circuits. However, just like the enterprise case, it is possible to also connect the remote locations via an L3VPN service.

In the following sections we will see scenarios where the VPN customers are themselves Internet service providers (ISPs) or VPN providers and they obtain backbone service from a VPN provider who acts as a ‘carrier of carriers’ [RFC4364]. An important thing to note in the context of a ‘carriers’ carrier’ scenario is the fact that all sites of a customer who is a carrier belong to the same AS. We will discuss inter-AS solutions in a separate section afterwards, because the concepts introduced in the carriers’ carrier discussion will facilitate the understanding of the inter-AS case.

9.2 CARRIERS' CARRIER – SERVICE PROVIDERS AS VPN CUSTOMERS

The biggest challenge in providing VPN service to customers who are themselves providers is the sheer number of routes that may be advertised from each site: the entire routing table when the customer is an Internet provider and the total number of VPN routes in the customer's network when the customer is a VPN provider. A large number of customer routes from each such customer places a significant burden on the (carriers' carrier) network, both in terms of the memory needed to store the routes at the PEs and in terms of the resources necessary to send and receive advertisements every time any of these routes flap.

In order to be able to scale the solution and support a large number of carrier customers it is necessary to shield the VPN provider (carriers' carrier) from the carrier-customer's routes. The idea is to split the load between the customer and the provider as follows:

1. The carrier-customer handles the route advertisements between the sites, via IBGP (internal BGP) sessions between routers in the different sites. The routes exchanged in these sessions are called 'external routes' because they are not part of the carrier-customer network. They may either be Internet routes or they may be VPN routes belonging to the carrier-customer's own VPN customers. For example, Figure 9.1 shows an ISP as a VPN customer. In this context, prefix 20.1/16 learned over the EBGP peering is considered to be an external route.

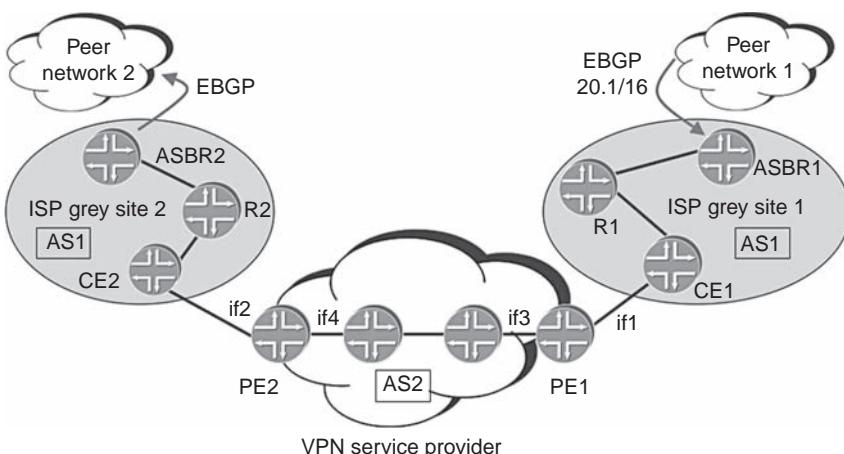


Figure 9.1 An Internet service provider as a VPN customer

2. The carriers' carrier VPN provider builds the connectivity necessary to establish the BGP sessions between the customer sites. In particular, the loopback addresses of the BGP peers in the carrier-customer's sites are carried as VPN routes. These addresses are the BGP next hops for the external routes exchanged in the customer's iBGP sessions and are called 'internal routes' because they are part of the carrier-customer's own network. For example, in Figure 9.1, ASBR1's loopback address is considered an internal route, because it is part of the carrier-customer's network.

This approach is different from the one taken with enterprise customers, where the advertisement of routes from site to site is the responsibility of the provider and where routing peerings between customer sites are not necessary. Note, however, that this fundamentally different approach is catering to a fundamentally different type of customer: one for which routing and route distribution are his or her daily business and who would be running BGP sessions between routers anyway.

To summarize, in a 'carriers' carrier' environment, the backbone VPN provider has two tasks: (a) to facilitate the establishment of the BGP sessions between customer sites by advertising the internal routes and (b) to permit forwarding of traffic to the external destinations learned via these BGP sessions. In the following sections we will see how this is accomplished, using the following principles:

1. Use tunneling to forward traffic across nodes that do not have routing information for the packet's final destination.
2. Use a label to identify the VPN on the PE.
3. Use the next-hop in the BGP advertisement to provide coupling between the VPN routes and the routing to the remote PE.

Let us now look at the two main carriers' carrier (CsC, also known as carrier of carriers or CoC) scenarios. We will first look at the case where the customer carrier is an ISP that only carries Internet routes and does not offer the L3VPN service to its end customers. We will then look at the case where the customer carrier offers L3VPN service to its end customers.

9.2.1 ISP as a VPN customer

Let us take a look at the ISP in Figure 9.1. The ISP has two geographically dispersed sites (belonging to the same AS), from where it maintains external peerings with other ISPs. The ISP buys an L3VPN service in order to connect the two sites. The goal of the ISP is to run IBGP sessions between routers in the two sites and exchange routes learned from the external peers in each one of the sites. In this context, it is very natural that the

backbone VPN provider should not need to carry the ISP's routes, but rather facilitate the establishment of the BGP sessions between routers in the two sites.

In order to establish a BGP session, there must be connectivity to the BGP peer's loopback addresses. This can be easily accomplished by advertising the loopback addresses as VPN routes, using the same mechanisms we have seen in the L3VPN Foundations chapter (Chapter 7). Once these addresses are reachable from both sites, the BGP session can establish. The conceptual model of the route exchanges is depicted in Figure 9.2 for an IBGP session between ASBR1 and ASBR2. The routes that are exchanged as VPN routes are the loopback addresses of the routers between which the IBGP session will be established, ASBR1 and ASBR2. The Internet routes from the customer sites are exchanged over this IBGP session between ASBR1 and ASBR2 and thus are not part of the VPN. The BGP next-hop of such an Internet route is the address of the remote end of the BGP session over which the route was learned.

Figure 9.3 shows the route advertisements that take place. The figure only shows the information relevant to the discussion, in particular:

1. The advertisements are shown in one direction only, from site 1 towards site 2.
2. The figure focuses on a single route learned from an external peering, 20.1/16, learned at ASBR1.

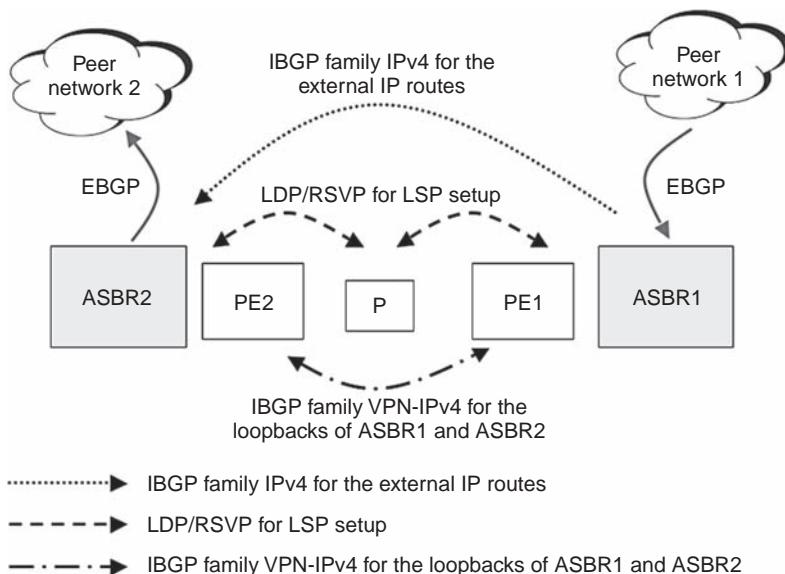


Figure 9.2 Conceptual model of the route exchanges for ISP as a VPN customer

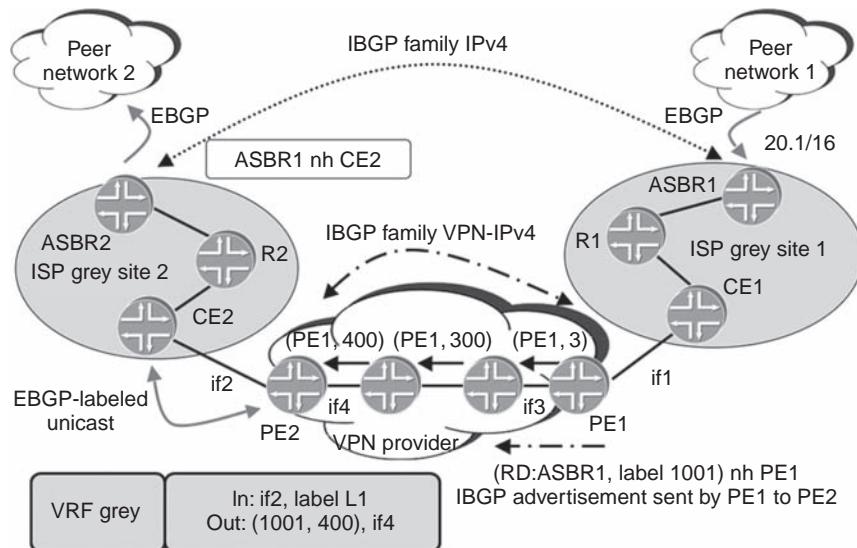


Figure 9.3 The route advertisements for ISP as a VPN customer

3. For simplicity, a single BGP session is shown between the routers in the two sites, namely the session between ASBR1 and ASBR2.
4. The full mesh of IBGP sessions between all the routers in each site is not shown. In particular, remember that although not shown in the picture, IBGP sessions exist on the CEs.

Here are the route advertisements that take place when setting up the BGP session between the ASBRs and exchanging the external routers (we will see later on that these are not the only route advertisements):

1. A label switched path is set up between PE1 and PE2, using either LDP or RSVP.
2. The loopback of ASBR1 is advertised as a VPN route in the same way as we have seen in Chapter 7. As a result:
 - (a) At the PEs, in the VRF associated with this ISP customer, there is a route for ASBR1's loopback.
 - (b) The route for ASBR1's loopback is advertised to CE2 and CE2 advertises it with itself as the next-hop to all routers in site 2.
3. An IBGP session is established between ASBR1 and ASBR2¹ and routes learned from external peers are advertised between sites. In particular, ASBR2 learns the route 20.1/16, with BGP next hop ASBR1.

¹This assumes that the route for ASBR2 is known in site 1. The figure does not show the route exchange in this direction.

4. The route is advertised via IBGP to all the routers in site 2; thus all routers in site 2 have a route for destination 20.1.1.1 with next-hop ASBR1. The route to ASBR1 was advertised as a VPN route and was advertised to all routers in site 2 by CE2. (Thus, the BGP next-hop of the route for ASBR1 is CE2.)

Let us take a look at the solution so far and investigate a problem that happens when attempting to forward traffic from ASBR2 to destination 20.1.1.1. All routers in site 2 have knowledge of this destination, with next-hop ASBR1, which was learned from CE2. The traffic arrives at CE2 and is forwarded as IP to PE2 over the CE2–PE2 interface. PE2 performs a lookup in the appropriate VRF. However, the VRF only contains routes for the loopbacks of the routers in site 1 and does not have an entry for 20.1.1.1, so the packet is dropped.

The problem is that the PE only has knowledge regarding the BGP next-hop of the route, not the route itself. What is needed is a way to tag the traffic so that the local PE (PE2) can forward it to the correct remote PE (PE1). We have already seen in the L3VPN Foundations chapter how MPLS labels are used to tag traffic. The same concept can be applied in this scenario as follows. When PE2 advertises ASBR1's loopback to CE2 it attaches a label L1 to it. This can be done by establishing an EBGP session between PE2 and CE2, with the family labeled-unicast (SAFI 4, also referred to as labeled-inet [RFC3107]). When advertising the labeled route, PE2 installs the forwarding state, swapping the label L1 to the VPN label that it received from PE1. When CE2 receives the advertisement for the labeled route for ASBR1, it installs the forwarding state which pushes label L1 to the traffic destined for ASBR1 before forwarding it to PE2. Figure 9.4 shows the conceptual model of forwarding traffic from CE2. At CE2, label L1 is pushed on incoming IP traffic which has ASBR1 as the next-hop. At PE2, label L1 is swapped for the VPN label advertised by the remote PE (PE1) for the VPN route for ASBR1. The label for the next-hop of this VPN route (PE1) is pushed on top of the VPN label, just as for the normal VPN scenario. Traffic arrives at PE1 with the VPN label and can be forwarded towards CE1.

To summarize, in order to extend the LSP between the CEs, BGP is used on the PE–CE link and advertises a label along with the prefix. Conceptually, what is done is to use BGP to extend the PE–PE MPLS tunnel all the way to the CE, thus shielding the PEs from knowledge about external routes. The LSP is made up of different segments, in this case a BGP-advertised label on the PE–CE segment and the VPN tunnel stacked on top of the LDP/RSPV transport tunnel on the PE–PE segment. The segments are ‘glued’ together by installing the forwarding state, swapping the label from one segment to the label from another segment. This is an important concept that will be applied throughout the remaining sections.

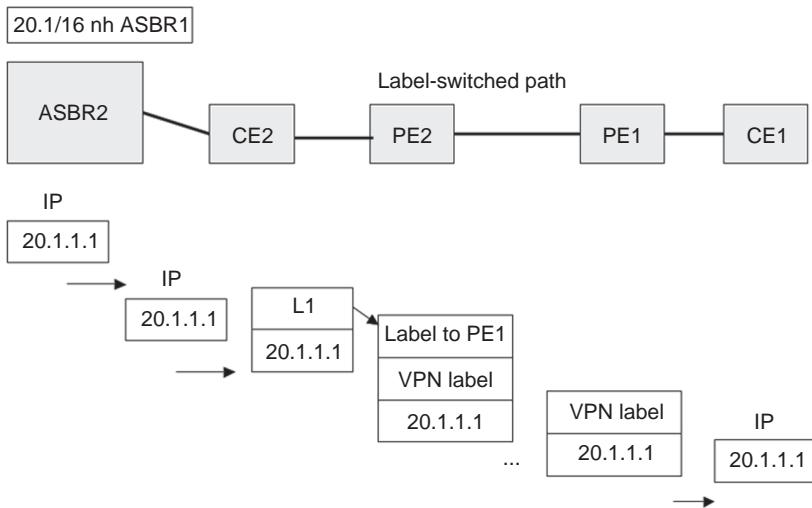


Figure 9.4 Conceptual model of traffic forwarding for ISP as a VPN customer

A potential security issue arises in this scenario if the label advertised in BGP by the PE for a particular route can be guessed (spoofed) by a different customer attached to the same PE. This is only a concern if the forwarding state on the PE (swapping between the assigned label and the VPN label) is not stored on a per-VPN basis. Some implementations solve this problem by maintaining separate MPLS tables per VPN; others may do so by having the PE reject labeled traffic arriving on any interface except the one over which the label was advertised.

Let us summarize the key properties of the solution:

- The ISP's Internet routes are advertised via an IBGP session between routers in the ISP's sites. These routes are external routes, outside the ISP's own network.
- All routers within each customer site must keep the routing state for the external routes and a full mesh of IBGP sessions is required within each site.
- The VPN provider carries as customer VPN routes only routes that are internal to the ISP's network (in the discussion so far, these were loopback addresses, but in practice both loopbacks and interface addresses are exchanged). As a result, the VRF on the PE maintains a small number of routes for the VPN corresponding to this ISP. The routes in the VRF are the BGP next-hops for the routes exchanged over the customer's IBGP sessions.

- Traffic cannot travel as pure IP between CE and PE since the PEs have no knowledge of the customer routes. The BGP next-hop is the glue that ties the customer routes and the forwarding information on the PE.
- MPLS tunnels are necessary between the CEs. The tunnels are made up of several segments, CE-PE, PE-PE and PE-CE, and are glued together by installing a swap state for the labels.
- CEs are required to support MPLS and MP-BGP.
- There is a need to protect the PE from the possibility of label spoofing.

This type of setup is more commonly seen when the backbone provider and the ISP are different divisions of the same company. Another option for providing the same service is to connect the ISP sites at layer 2, rather than implementing the layer 3 solution described above. The arguments for and against each of these approaches are the same as the ones put forth in the comparison of overlay and peer VPN models in the chapter discussing basic L3VPN functionality (Chapter 7).

9.2.2 VPN service provider as a VPN customer – hierarchical VPN

The second type of carrier-customer supported by the ‘carriers’ carrier’ scenario is a VPN provider. For readability purposes, in order to distinguish between the VPN provider who acts as a customer and the one who acts as a provider, we will refer to the providers in Figure 9.5 by the names given to them in the figure. Figure 9.5 shows a VPN provider, provider ‘grey’ with two geographically dispersed sites, sites 1 and 2, both belonging to the same AS. Provider grey services two VPN customers, customer red and customer blue, with sites attached to PEs in both sites 1 and 2. The goal of provider grey is to run IBGP sessions between its PEs and advertise the VPN routes for customer red and customer blue. For this purpose, provider grey buys a VPN service from a ‘carriers’ carrier’ VPN provider. In this context, it is natural that this carrier should not need to carry the routes of provider grey’s customers, but rather to facilitate the establishment of the BGP sessions that exchange these routes. The conceptual model of the route exchanges is depicted in Figure 9.6.

This scenario is very similar to the one we saw in the previous section, with the difference that the routes exchanged over the IBGP sessions between the customer’s PEs are VPN-IP routes rather than IP routes. In order to forward the (labeled) VPN traffic, a label-switched path is required between the customer’s PEs. In the previous section we have seen how such a path can be built between the customer’s CE routers by running a labeled-inet (SAFI 4) EBGP session between the provider’s PE and the

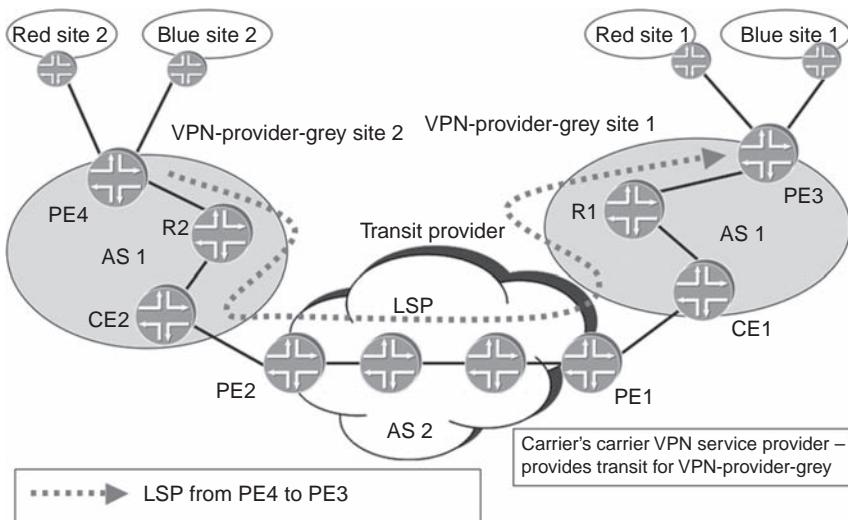


Figure 9.5 A hierarchical VPN where the VPN service provider grey is himself a VPN customer

customer's CE routers. To extend the label-switched path to the customer's PE routers one can take the same approach and run a labeled-inet IBGP session between the customer's CE and PE routers. Figure 9.7 shows a conceptual model of the advertisement for PE3's loopback between provider grey's sites.

The reachability information for PE3's loopback is advertised as a VPN route from site 1 to site 2, as shown in Figure 9.7. At the VPN provider's

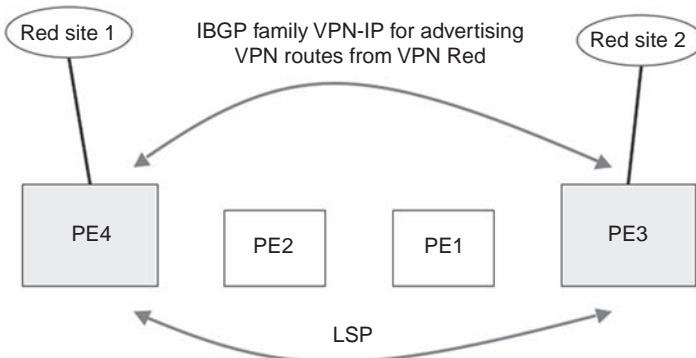


Figure 9.6 Conceptual model of the route exchanges taking place in a hierarchical VPN

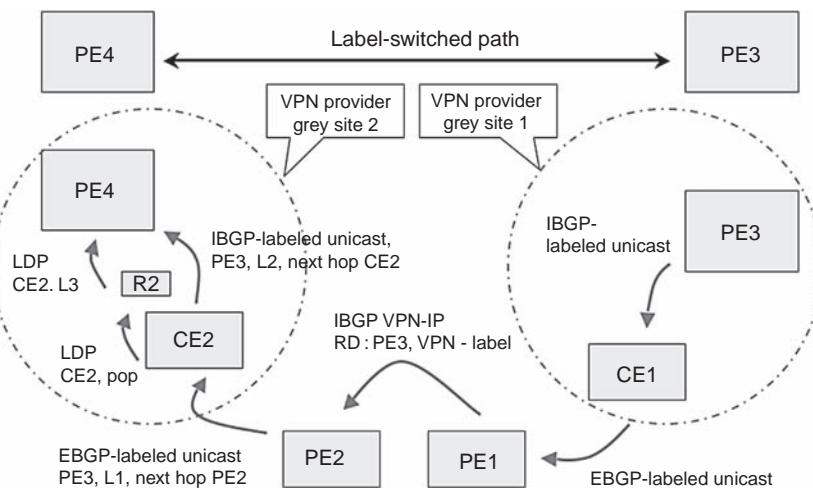


Figure 9.7 Conceptual model of the advertisement of PE3's loopback between the provider grey's sites

remote PE (PE2), the route is advertised as a labeled route to CE2, with label L1, just as in the previous section, and the forwarding state is installed to swap label L1 to the VPN label. At CE2, the route is advertised with label L2 and the forwarding state is installed to swap L2 to L1. At PE4, the route for PE3 is received with label L2 and nexthop CE2. The forwarding state is installed to push label L2 to traffic with BGP next-hop of PE3 and forward it towards CE2. Since this is labeled traffic, it is required that CE2 be reachable via an LSP. This LSP may be built with LDP, RSVP or even BGP. Assuming that LDP is running in site 2, label L3 exists at PE4, advertised by R2 for CE2's loopback. Thus, to forward traffic to PE3 from PE4 two labels must be imposed: the top label is the label for CE2's loopback, L3, and the bottom label is the label identifying the destination PE3, L2.

When traffic arrives from customer red's VPN attached to PE4 it must be forwarded towards PE3, with the following labels imposed: the bottom label is the VPN label that was advertised via the BGP session between PE3 and PE4 (not shown in Figure 9.7) and the top labels are the labels that carry the traffic to PE3, labels L2 and L3. Thus, a three-label stack is pushed at PE4.

So far we have not discussed the BGP sessions shown in Figure 9.7 for site 1. In principle, from a technical point of view, there is no need to advertise PE3's loopback as a labeled route in BGP in the local site (site 1), and therefore the BGP advertisement for PE3's loopback can be an unlabeled IPv4 route. However, note that for the sake of simplicity in Figure 9.7 the routing exchanges are shown in one direction only, from

site 1 towards site 2. However, a symmetrical exchange is happening in the opposite direction from site 2 towards site 1. Thus, a BGP session for the family labeled-inet (SAFI 4) is necessary between PE3 and CE1 for carrying the label for PE4's loopback. In order to maintain a single BGP session rather than two, many vendors recommend advertising a null label for the route in the local site.

At this point let us stop and make the following observations:

1. It is required to run MPLS within the customer's sites. In the example above, it was required to have a label-switched path between the customer CE and PE devices (CE2 to PE4).
2. It is possible to isolate the knowledge regarding the addresses in the remote site to the routers at the edge of the site (routers CE2 and PE4 in Figure 9.7). Therefore, the IGP in site 2 need not carry information about the addresses of routers in site 1.

Note that the solution presented above solves the problem of distributing a label for PE3's loopback by using two label distribution protocols within the remote site, site 2: one protocol (BGP) for PE3's loopback and another one (LDP) for CE2's loopback. An alternate approach is to use LDP only. The idea is to configure LDP to advertise a label for an FEC corresponding to PE3's loopback. Remember from the Foundations chapter (Chapter 1) that in order for the label-switched path to establish, the FEC must also be present in the IGP. Thus, at CE2, PE3's loopback is injected in both LDP and the IGP. As a result, a single label is pushed at PE4 in order to reach PE3. When VPN traffic belonging to customer red is forwarded, it receives a two-label stack, the label to reach PE3 and the VPN label appropriate to the red VPN. Note that LDP is only distributing labels within each site of AS1. BGP is still used between AS1 and AS2.

Using LDP instead of BGP for label distribution has several advantages:

1. If LDP is already running, there is no need for an additional protocol.
2. Fewer labels are imposed at the time the packet is forwarded; in this case, two labels are required instead of three. This used to be an important consideration in older hardware implementations, which had limitations handling deep label stacks.

The main disadvantages of using LDP is that it requires redistribution of routes into both LDP and the IGP, with the following consequences:

- Requires redistribution of the routes for the loopbacks of the remote PEs (BGP peers) into the IGP. These routes are advertised via BGP and providers are wary of redistributions from BGP into the IGP, since a mistake in the redistribution policy can inject a large number of routes in the IGP and cause IGP meltdown.

- The IGP must carry prefixes from a different site, which could impact scaling of the IGP. When the two sites are in different ASs (discussed later in this chapter), the provider requires more control over the routing information injected into one AS from another. This control is readily available with the BGP solution.

The differences from the ISP-as-a-customer scenario in the previous section are:

- MPLS is used within the sites.
- Only the routers imposing the label stack are required to have knowledge of the external routes. VPN routes are exchanged for the BGP next-hops of the external routes. Thus, the exchanges of the VPN routes can happen over sessions between the routers that actually do label imposition.

9.3 MULTI-AS BACKBONES

The previous section showed how a VPN provider can be a VPN customer, and its sites are in the same AS. In this section we will take a look at what happens when the sites are in different ASs. This can be the case when a provider spans several ASs (e.g. following an acquisition) or when two providers cooperate in order to provide a VPN service to a common customer. In the latter case, it is necessary for the providers to agree on the conditions and compensation involved and to determine the management responsibilities. To distinguish the two cases, they are referred to as inter-AS and interprovider respectively.

The problem with multi-AS scenarios is that the routers in the two sites cannot establish an IBGP session to exchange external routes. Instead, an EBGP session must be run. [RFC4364] describes three ways to solve the multi-AS scenario. These methods are often referred to by their respective section number in [RFC4364], as options A, B and C. An important thing to bear in mind when reading this section is that multi-AS scenarios are not targeted in particular at carrier-customers and provide a general solution for VPNs crossing several ASs.

9.3.1 Option A: VRF-to-VRF connections at the ASBR

The simplest method to exchange the VPN routes across an AS boundary is to attach the two ASBRs directly via multiple subinterfaces (e.g. VLANs, or Virtual LANs) and run EBGP between them. Each ASBR associates one of the subinterfaces with the VRF for a VPN requiring inter-AS service and

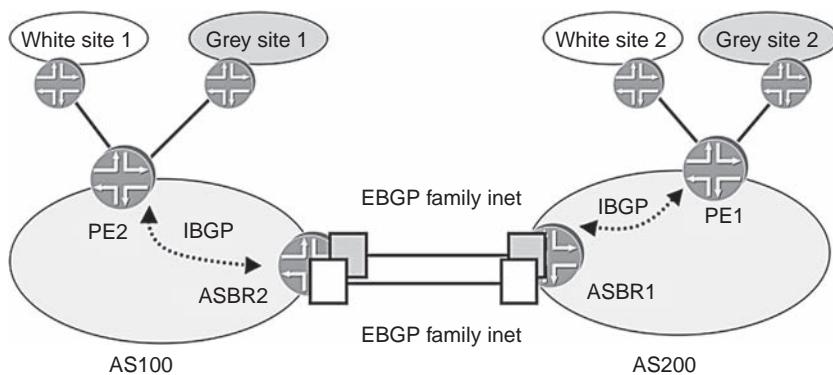


Figure 9.8 Option A, VRF-to-VRF connections at the ASBR

uses EBGP to advertise customer IP routes within each VRF, as shown in Figure 9.8.

This solution requires no MPLS at the border between the two ASs. As far as ASBR2 is concerned, ASBR1 appears to be equivalent to a regular CE site. Similarly, ASBR1 appears to be a regular CE site to ASBR2. The separation of routing and forwarding information is accomplished by using separate VRFs per VPN at the ASBR, with the following consequences:

- The ASBRs are configured with VRFs for each of the VPNs crossing the AS boundaries. There is a separate subinterface associated with each of the VRFs.
- There is per-VPN state on the ASBRs and the provider is required to manage the subinterface assignment for the different VPNs.
- The ASBRs must exchange all the VPN routes from all VPNs crossing the AS boundary.
- Multiple EBGP sessions are maintained (one per VPN).

Despite these less than desirable scaling properties, the solution works and is deployed for situations when the number of VPN customers and the number of VPN routes is small. There are several benefits to the solution. Option A is simple to understand and deploy and is contained by the routers providing the VPN service. Furthermore, it simplifies interworking among providers, because the interconnection between providers is simply an interface. Therefore, this interface becomes the element of control on which policing, filtering and accounting can be done with per-VPN granularity.

9.3.2 Option B: EBGP redistribution of labeled VPN-IPv4 routes

The undesirable scaling properties of option A are caused by the fact that the VPN routes are exchanged as IP routes, so the per-VPN state must be maintained by the ASBRs. Furthermore, every time a new VPN is added that requires inter-AS service, the ASBR must be configured with the correct VRF information. Thus the addition of a new VPN is no longer limited to the configuration of the PE routers and involves ASBR configuration as well.

To avoid keeping the per-VPN state at the ASBR, VPN-IPv4 routes can be advertised instead. The option B solution uses a single EBGP session between the ASBRs regardless of the number of VPNs and advertises labeled VPN-IPv4 routes over it. The routes are exchanged between PE and ASBR via an IBGP session. The conceptual model is shown in Figure 9.9. In order to ensure that unauthorized access to the VPN is not possible, the EBGP session must be secure and VPN-IPv4 routes should not be accepted on any other session except the secure one.

At the end of the routing exchanges, the PEs in the different ASs have received the VPN routes for their customers with the appropriate VPN labels assigned by their peers. However, in order actually to be able to forward traffic between the two customer sites, a label-switched path must exist from one PE to the other, across the AS boundaries.

It is possible to build the necessary LSP by using BGP as a label distribution protocol on the inter-AS link, as we have seen in the carriers' carrier scenario. Let us take the example network of Figure 9.9 and see how this is done for a VPN route advertised from PE1. The assumption is that MPLS is running in each one of the ASs, so label-switched paths exist between

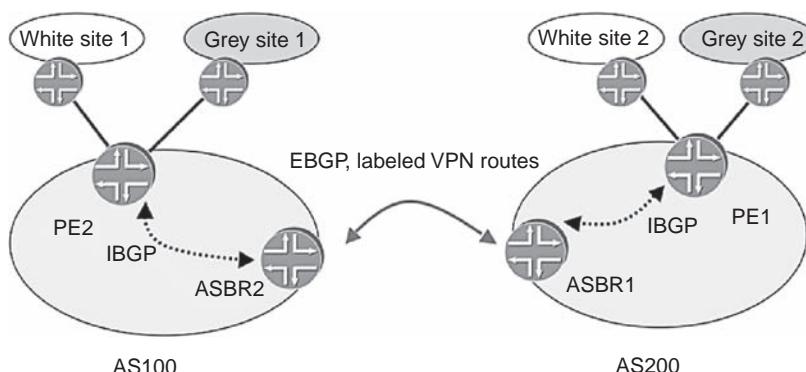


Figure 9.9 Option B: EBGP redistribution of labeled VPN-IPv4 routes

the ASBRs and the PEs. What is needed is a way to stitch between the LSP within each AS and the one-hop LSP over the inter-AS link, by installing the appropriate MPLS swap state. When ASBR1 receives a labeled VPN route from PE1, with the VPN label L1, it allocates a label L2 for it and advertises it to ASBR2 with itself (ASBR1) as the next-hop. At the same time, it installs the forwarding state that swaps between L2 and L1. ASBR2 advertises the VPN route with label L3 in its AS and installs swap state between L3 and L2. When traffic is sent from PE2, for this VPN prefix, it is labeled with L3. At ASBR2, the label is swapped to label L2, and at ASBR1, it is swapped again, this time to label L1 advertised by PE1. Note that in this scenario, the VPN label changes – the label used by the remote PE is not the same as the one that was allocated by the local PE.

A security breach may arise if the label assigned by the ASBR is spoofed by an outsider. To avoid this situation, the ASBR should only forward labeled traffic arriving over an interface over which the label was actually advertised. Another potential security threat results from peering with an unauthorized source who wants to capture the traffic for a particular VPN. To avoid such unauthorized peering, the providers in the two ASs must negotiate and agree upon which routers are allowed to exchange VPN routes and which RTs they will use in the route advertisements.

Let us take a look at some of the properties of this solution:

1. There is no need for per VPN configuration and per VPN interface assignments at the ASBRs.²
2. The ASBRs must keep the state for all VPN routes.
3. It cannot easily do traffic filtering at the IP level for traffic crossing the AS boundary.
4. A single EBGP session must be maintained between the ASs.
5. It requires a PE–PE inter-AS LSP.

9.3.3 Option C: multihop EBGP redistribution of labeled VPN-IPv4 routes between the source and destination AS, with EBGP redistribution of labeled IPv4 routes from one AS to the neighboring AS

The solution in option B still requires that all VPN routes be maintained and advertised by the ASBR. This makes the solution unsuitable for cases

²In practice, if only a subset of the VPNs have sites in both ASs, policies would be configured on the ASBRs to only advertise the routes for the relevant VPNs.

where there are a lot of VPN routes. Option C avoids this problem by using a scheme analogous to the hierarchical VPN scenario discussed in Section 9.2.2: the customer uses a multihop EBGP session between its PE routers to carry external prefixes as labeled VPN-IPv4 routes and the provider provides connectivity to the PE loopbacks by advertising them as labeled IPv4 routes from one AS to another. In this way, the ASBRs do not carry any of the VPN routes. Thus this option is the most scalable of the three options discussed here.

Note that in order for this solution to work, the BGP next-hop of the VPN-IPv4 routes exchanged over the multihop EBGP session must not be changed. Figure 9.10 shows a conceptual model of the route exchanges, based on the network of Figure 9.5. As with the hierarchical VPN scenario, three labels are imposed on the traffic at ingress, unless the addresses of the PE routers are made known to the P routers in each domain, in which case a two-label stack is imposed.

In addition to the security concerns seen so far, a new problem arises in this scenario, if the approach of using the LDP rather than the labeled BGP is used (as explained in Section 9.2.2). Since the addresses of the PE routers are advertised, this means that the IGP in one provider's network carries addresses from a different provider's network. Sometimes, this is viewed as a security concern because the addresses of the PEs in one AS are known to the routers in the core of another network, thus revealing the addressing structure of the remote AS. This makes option C undesirable for interprovider (as opposed to plain inter-AS) setups. Furthermore, in contrast to option A, interworking among providers is seen by some as more difficult because the connection is an end-to-end 'fat pipe' without any VPN context. The LSP between PE3 and PE4 in Figure 9.10 carries

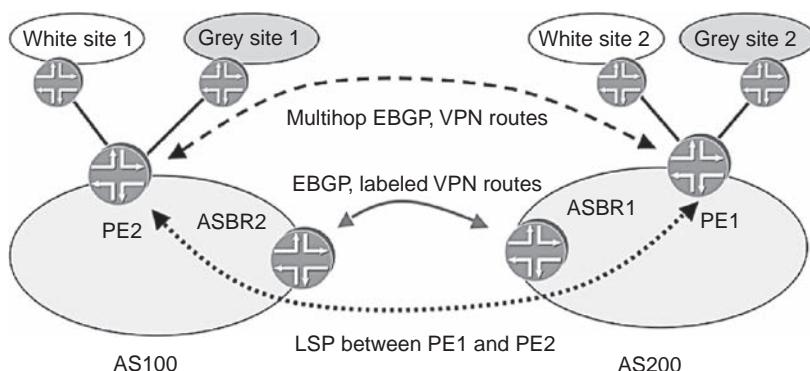


Figure 9.10 Option C: multihop EBGP redistribution of labeled VPN-IPv4 routes between the source and destination AS, with EBGP redistribution of labeled IPv4 routes from one AS to the neighboring AS

Table 9.1 Comparison of the three inter-AS solutions

	Option A	Option B	Option C
State at the ASBR	Per VRF state, all VPN routes in all VPNs are maintained	No per VRF state, but all VPN routes in all VPNs are maintained	Only the addresses of the remote PEs are maintained (the BGP next-hops of the VPN routes)
Per VRF Configuration at the ASBR	Yes	No	No
VPN label	Not used	May change at the ASBR (unless an inter-AS LSP is set up via RSVP)	Remains constant
Requirement to run MPLS across the AS boundary	No	Yes	Yes
EBGP session	Single hop, family IP, multiple sessions (one per VRF)	Single hop, Family VPN-IPv4, single session	Multi-hop, Family VPN-IPv4, single session
Security concerns	None	Prevent label spoofing at the ASBR	Access to all PEs from the remote AS

traffic from all VPNs. Thus, there is no per VPN context and the ability to do policing, filtering and accounting per VPN at the ASBR is lost.

Table 9.1 compares some of the properties of the three options described above.

9.4 INTERPROVIDER QoS

From the customer's point of view, the QoS expectations are the same, regardless of whether the VPN service is implemented by a single provider or by multiple providers. When VPN traffic crosses several domains, it is necessary for each of the domains to enforce its own policy to ensure the desired QoS.

In the Advanced L3VPN chapter (Chapter 8) we have seen how QoS can be provided for BGP/MPLS VPNs by marking customer traffic with

the EXP bits, providing the correct per-hop behavior in the network. The marking is done at the PE and forwarding the traffic in the core is based on the EXP bits. The solution requires mapping DSCP to EXP consistently at the PE-CE boundary. In an interprovider setup, it cannot be expected that the same EXP bits will be used to represent the same per-hop behaviors in both networks. Therefore, it is necessary to remark the EXP bits of the VPN traffic as it crosses the inter-AS link.

The inter-AS link poses other challenges as well. The link is shared between two administrative domains, so it is more difficult to upgrade and can easily become a point of congestion. This problem can be made worse by the fact that often the same link is used to carry Internet traffic between the providers in addition to the VPN traffic. In order to ensure SLAs for the VPN traffic, it is necessary to set up and enforce policies for prioritizing and rate-limiting traffic at the AS boundary. This can be done either at the aggregate level across all VPN customers, or with per VPN visibility, as possible in option A.

However, it is not enough to enforce the customer SLA end-to-end. When a service is shared between two providers it is important to be able to measure, report and troubleshoot the performance consistently in both ASs. SLAs are usually tied to compensation, so in the case of SLA violation it is necessary to be able to determine which provider is at fault. Today, the issue of measuring and reporting is solved on a case-by-case basis using the tools that are available in each provider's network.

9.5 CONCLUSION

The L3VPN solution supports VPNs spanning across several ASs, as well as VPN hierarchies. The key to achieving these in a scalable manner is the use of MPLS tunnels. The tunnels are built by stitching together several tunnel segments using either BGP or LDP. Having covered these advanced deployment scenarios, this chapter concludes the discussion on the support of unicast traffic in BGP/MPLS L3VPN. In the next two chapters, we will look at the evolution of solutions for multicast support in VPNs.

9.6 REFERENCES

- [RFC3107] Y. Rekhter and E. Rosen, *Carrying Label Information in BGP-4*, RFC3107, May 2001
- [RFC4364] E. Rosen and Y. Rekhter, *BGP/MPLS IP VPNs*, RFC4364, February 2006

9.7 FURTHER READING

[ADV-MPLS]	V. Alwayn, <i>Advanced MPLS Design and Implementation</i> , Cisco Press, 2001
[L3VPN]	http://ietf.org/html.charters/l3vpn-charter.html
[MPLS-TECH]	B. Davie and Y. Rekhter, <i>MPLS Technology and Applications</i> , Morgan Kaufmann, 2000
[MPLS-VPN]	I. Peplnjak and J. Guichard, <i>MPLS and VPN Architectures</i> , Cisco Press, 2000
[VPN-TUTORIAL]	I. Minei, <i>BGP/MPLS Layer 3 VPNs</i> , tutorial at Nanog 30, http://nanog.org/mtg0402/minei.html

9.8 STUDY QUESTIONS

1. In the ISP as a customer scenario described in Section 9.2, why do the VPN tunnels extend to CE1 and CE2 rather than stopping at the PEs (PE1 and PE2)?
2. In the hierarchical VPN example in Figure 9.5, the MPLS tunnel between PE4 and PE3 is made up of several segments. What are they and how are they set up?
3. What is the three label stack imposed on the traffic in option C?
4. What are the advantages and disadvantages of setting up an inter-AS RSVP LSP (see Chapter 5) for providing MPLS connectivity between the PEs in the remote ASs in option C (PE1 and PE2 in Figure 9.10)?
5. A service provider has separate LSPs for voice and data in his or her network. This provider is considering interconnecting with a peer in a different AS to provide VPN service to a common customer. How should the interconnect be designed such that he or she can enforce the mapping of the voice traffic arriving over it to the correct LSPs in his domain?

10

Multicast in a Layer 3 VPN

10.1 INTRODUCTION

So far in this book, the L3VPN discussion has focused on providing connectivity for unicast destinations. How is multicast traffic handled? To answer this question, let us first determine what the requirements are, both from the customer's and the provider's point of view. From the customer's point of view, the requirement is simple: be able to forward multicast traffic between senders and receivers in different sites using the same procedures as if they belonged to a single physical network and using private address spaces. From the provider's point of view, the goal is to satisfy the customer's requirement while maximizing his or her own profits. What this exactly means to the provider has changed over time, as L3VPN deployments have grown in both number of sites and of VPNs and as the bandwidth consumed by the multicast traffic has increased many fold. While satisfying the customer's requirement for multicast connectivity among sites, the original solution proposed for multicast VPN (mVPN) was not scalable from the provider's point of view. For this reason, a new approach to multicast support in L3VPN has been developed in the last few years in the L3VPN working group [L3VPN-WG] in the IETF.

The chapter will present both the original and the new approaches and then compare them. Basic familiarity with PIM, P2MP LSPs and unicast L3VPN concepts is assumed for the rest of this chapter. In this chapter,

the terms mVPN and VPN will be used interchangeably when discussing VPNs that support multicast.

10.2 THE BUSINESS DRIVERS

The benefits of BGP/MPLS IP VPNs for both customers and providers have been discussed in Chapter 7. Because L3VPNs provide the customer with the abstraction of a private network, the expectation is to be able to support any type of application that would run in this private network. As applications relying on multicast become more widely deployed in the enterprise, support for multicast becomes a requirement for an increasing number of customer networks which rely on L3VPNs for their intersite connectivity.

The original multicast solution is documented in [DRAFT-ROSEN] and therefore often referred to as draft-rosen. Draft-rosen solved the problem of carrying multicast traffic between customer sites but suffered from various shortcomings, mostly in the scaling area. These problems become most apparent in deployments where (a) a large percentage of the L3VPN customers require support for multicast as well as unicast traffic and (b) the amount of multicast traffic is non-negligible compared to the unicast traffic (thus requiring traffic engineering in the provider's network). When the draft-rosen solution was initially proposed, neither of these two conditions was of concern. However, several factors drove the increase in multicast traffic in L3VPNs, thus creating the need for a new approach:

1. L3VPN deployments became successful and grew in size, increasing the number of VPNs that must be supported.
2. Applications relying on multicast, such as business IPTV, became popular in the enterprise.
3. Service providers started to rely on mVPNs as an internal infrastructure for providing bandwidth-intensive multicast services such as IPTV.

As a result of the growth in (a) the number of VPNs requiring multicast and (b) the multicast traffic volumes, service providers as well as vendors realized the scaling limitations inherent in the original design of multicast support for L3VPN and started working on a new approach. This effort led to the so-called Next Generation (NG) multicast solution for L3VPN, documented in [mVPN-BGP-ENC] and to the development of a framework and architecture document, [VPN-MCAST], that decomposes the problem of multicast support for L3VPN and then provides a variety of mechanisms for each component, including options from both the draft-rosen and the NG solutions.

As explained above, when discussing the business drivers for the development of the mVPN solution, it is not the need for multicast that needs to be examined, but rather the characteristics of the different types of customer multicast applications, as these will clarify the requirements that an mVPN solution must satisfy from the provider's point of view. The provider's requirements for the mVPN solution are documented in [RFC4834].

Let us start by listing three of the applications that require multicast in an L3VPN: (a) live content distribution (such as IPTV or financial market data feeds), (b) non-real-time data distribution (such as data backup or the download of daily catalog or pricing information from a central office to a set of remote branches) and (c) symmetric applications (such as video-conferencing or e-learning).

From this short list, it is immediately clear that the requirements can be widely different. Some applications, such as IPTV, may create a very large number of multicast streams, while others, such as a daily pricing update, may require only a few. In some cases, such as IPTV, the band-width used by each of the streams may be very large, while in others, such as an audio-conferencing application, it could be small. Sometimes, for example when distributing video among production studios, the location of sources and receivers is known ahead of time and static, while other times, such as when using a video-conferencing application, the sources and receivers are not known and sites may join and leave the group at random times. Some applications, such as IPTV, require stringent QoS, while others, such as data backup, can recover from failures of the underlying networks using other mechanisms. Finally, in some cases, such as data distribution, the multicast group spans all sites of the VPN, while in other cases, such as an e-learning application, only a small number of sites may be part of the same multicast group.

Each of these scenarios requires optimization of different parameters: the amount of state maintained in the provider network, or the amount of bandwidth used to provide the service, or the resiliency to failures, or the ability to handle fast join/leave from a group. Because of the wide range of applications and multicast deployment flavors that an enterprise may use in its private network, the mVPN solution must be flexible both in terms of the flavors of multicast protocols that an enterprise may use and in terms of the tuning that the provider should be able to do, for example in order to optimize either bandwidth utilization or state creation in his network. As a secondary goal, an additional requirement is to be able to do all this with the minimum of extra overhead from an operations point of view (e.g. least amount of extra protocols that must be deployed in the core, least amount of training for the operations people). Having seen that 'one size does not fit all' in the multicast world, let us start by looking at the two solutions. Before we do so, let us look at the components of any mVPN solution.

10.3 mVPN – PROBLEM DECOMPOSITION

Figure 10.1 shows the conceptual model of multicast in a VPN. Two VPNs, VPN A and VPN B, are shown. For simplicity, we assume that PIM-SM in the SSM mode is used within each VPN. A multicast source in site S of VPN A is sending traffic to receivers in sites R1 and R2 of the VPN. This delivery of traffic can be decomposed into three stages. First, the traffic must be delivered from the source to PE1. Then, it must be delivered from PE1 to PE2 and PE3, and finally from there to the appropriate CEs, CE4 and CE3. The same decomposition applies when a source in site S of VPN B is sending traffic to receivers in sites R1 and R2 of VPN B.

In order for the source in site S to send multicast traffic to receivers in sites R1 and R2, receivers in sites R1 and R2 must inform the source about their desire to receive the multicast traffic from the source. Propagation of this information from the receivers towards the source could be decomposed into three steps. First, this information must be delivered from the receivers to the PEs connected to the sites that contain the receivers – PE2 and PE3 in our example. Then, this information has to be propagated to the PE connected to the site that contains the source, PE1 in our example.

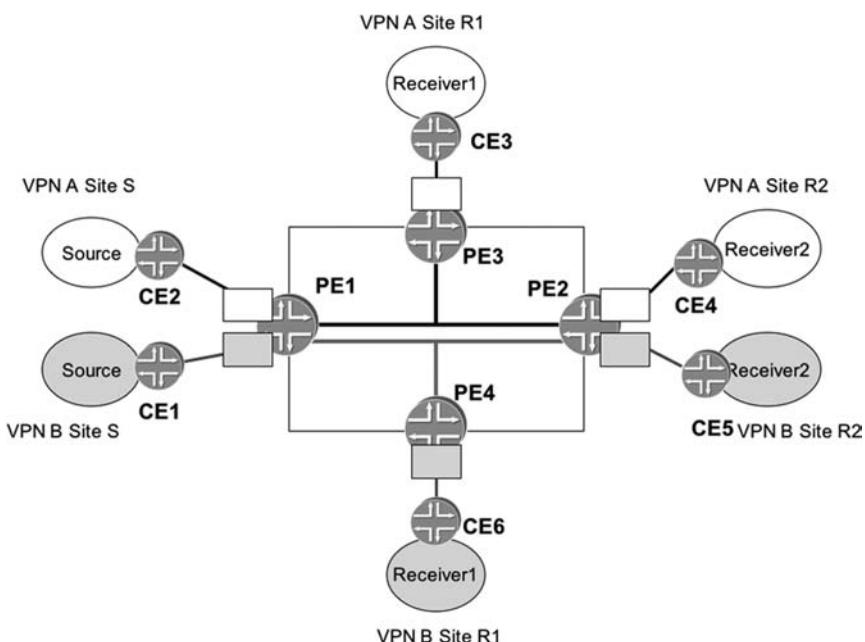


Figure 10.1 Forwarding multicast traffic in a VPN

Finally, this information has to be propagated from that PE to the source itself.

From a routing point of view, the first and the third (last) step in the above decompositions are contained within the mVPN customer domain. Just like in the unicast case, where the routing procedures used by the customers are not changed, the routing within the customer domain, which in this case is based on PIM, must continue to use PIM. However, similar to the unicast case, there is flexibility with respect to the mechanisms used within the service provider network.

The following two mechanisms are required within the service provider's network to allow traffic from the source to reach the receivers in the VPN:

1. A mechanism to exchange mVPN multicast routing information between the PEs servicing different sites of the VPN. This is needed because the receivers must inform the source that they want to receive traffic from it. Assuming PIM is running within the customer VPN, this means that the relevant PIM state must be communicated between the sites and the relevant multicast trees must be established within the customer VPNs between the sources and the receivers.
2. A mechanism to carry multicast traffic from the PE connected to the site that contains the source to the sites that contain the receivers. Conceptually, this is shown in Figure 10.1 by the inter-PE connections in the center of the diagram. As far as the PEs are concerned, a packet sourced at PE1 belonging to VPN A will reach both PE2 and PE3.

Both of the above mechanisms must handle the situation where different VPNs use exactly the same address space, both for unicast and for multicast.

When presenting the two multicast solutions, in the next sections we will look at each of these mechanisms separately.

10.4 THE ORIGINAL MULTICAST SOLUTION – PIM/GRE mVPN (DRAFT-ROSEN)

The original solution for multicast in BGP/MPLS VPNs is often referred to as 'draft-rosen', after the name of the IETF draft [DRAFT-ROSEN] that first described it, or PIM/GRE mVPN, after the technologies used for the exchange of routing information and traffic forwarding. Note that although implementations of this solution were deployed by some of the largest networks in the industry, the draft itself never gained working group status in the IETF. However, the working group draft describing the

architectural solution for multicast [VPN-MCAST] is a superset of draft-rosen, allowing all the functionality available in the original draft.

Although targeted to multicast traffic in a BGP/MPLS VPN setup, the draft-rosen solution departs from the BGP/MPLS VPN model for unicast traffic described in previous chapters. Nevertheless, some of the elements of BGP/MPLS VPNs are reused: the interface between PE and CE in the multicast solution is still based on the VRF concept of BGP/MPLS VPNs, and customer traffic is still tunneled through the provider core between the PEs (the VPN tunnel concept of BGP/MPLS VPNs). Despite these high-level similarities, the PIM/GRE multicast solution is very different from the unicast one. Exactly how different we will see in the remainder of this section, by looking at the mechanisms used for carrying multicast mVPN routing information and data traffic. These will be discussed separately below.

10.4.1 PIM/GRE mVPN – routing information distribution using PIM C-instances

Similar to the unicast VPN case, where the customer's CE routers maintain a routing adjacency only with the local PE router, the distribution of the PIM state among the customer sites is also based on a model where CE routers only require a PIM adjacency with their local PE router (rather than with all other CEs in the VPN). Again similar to the unicast case, the PE must process the PIM messages exchanged with the CE in the context of the VRF to which the CE belongs.

To propagate the PIM information among the PE routers and onwards to other VPN sites, PIM adjacencies are set up among the PEs that have sites in the same VPN. Because the PIM information exchanged among PEs is relevant only in the customer's VPN context, it is necessary to set up these PIM adjacencies among PEs on a per-VPN basis. These per-VPN PIM adjacencies ensure that the necessary multicast trees can be set up within each customer VPN. The need for per-VPN PIM adjacencies is in contrast to the unicast case, where a PE uses a single BGP session¹ to exchange routes belonging to all VPNs present on the PE.

The PIM sessions set up at PE1 are shown in Figure 10.2. Note that PE1 maintains not one but two separate PIM adjacencies with PE2, one for VPNA and the other for VPNB. It also maintains a PIM adjacency with PE3 for VPNA and a PIM adjacency with PE4 for VPNB. These

¹ The notion of a single BGP session is used here in the context of one session carrying routing information for all VPNs between a pair of BGP peers, not in the absolute number of BGP sessions the router must maintain. The latter depends on the BGP deployment, for example whether route reflectors are in use or a full mesh of BGP sessions is set up to all the other PEs.

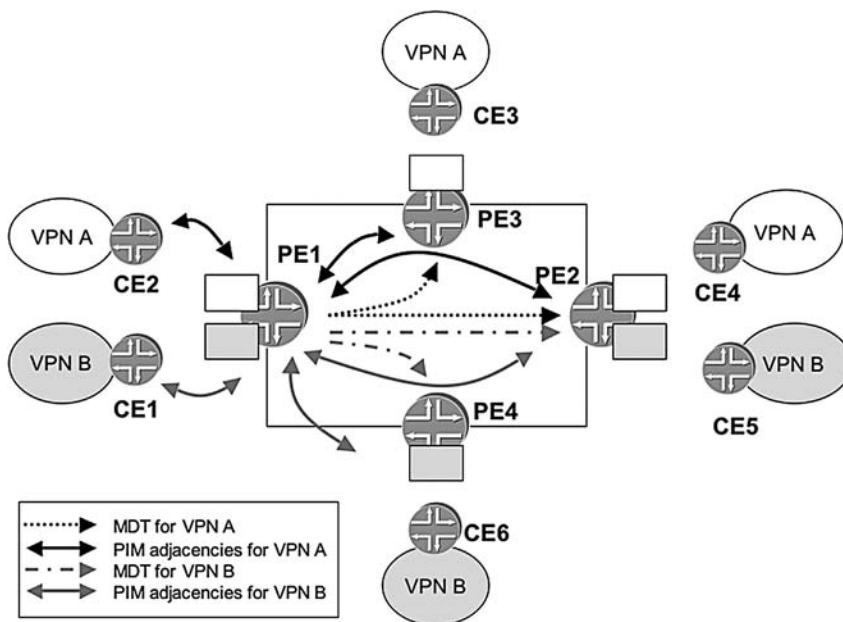


Figure 10.2 PIM sessions for a VPN in the draft-rosen proposal

VPN-specific PIM instances are referred to as PIM C-instance (where C stands for customer).

To summarize, the control plane of the draft-rosen mVPN solution uses PIM to carry customer multicast information across the service provider network. It does so by setting up per-VPN PIM adjacencies called PIM C-instances between the PEs that have sites of that VPN connected to them. In doing so, it diverges from the unicast model in (a) the choice of control protocol used and (b) the need to have per-VPN per PE routing peerings.

10.4.2 PIM/GRE mVPN – carrying multicast traffic across the core using multicast distribution trees

Similar to the unicast case, carrying mVPN traffic across the service provider network is done using tunnels between the PEs. For multicast, these inter-PE tunnels must carry traffic from one PE (the one servicing the customer site with the source) to multiple PEs (the ones servicing the customer sites containing receivers), as previously shown in Figure 10.1. This tunnel is called the multicast distribution tree (MDT). Conceptually, the MDT creates the abstraction of a LAN to which all the PEs belonging to a particular VPN are attached. This property is very important for the

C-instance PIM sessions between the PEs, which can consider each other as directly connected neighbors over this LAN. The fact that the MDT creates the abstraction of a LAN, where all PEs are equidistant from each other, is very different from the BGP/MPLS VPN model, as the latter does not assume that all PEs are equidistant from each other.² Moreover, the details of how the MDT is built and how it is used for forwarding cause the solution for VPN multicast to differ significantly from the BGP/MPLS VPN model.

Let us therefore take a look at some of the design decisions that the MDT forces on the solution:

1. *Separate MDT per VPN.* Conceptually, the MDT provides the same function for the multicast traffic as do VPN tunnels for the unicast traffic. Therefore, it is very intuitive that a separate MDT is required per VPN, connecting all the PEs with sites in the given VPN.³
2. *Using PIM to build the MDT.* The MDT is a point-to-multipoint tree and the PIM protocol is well suited for building such trees. The instance of PIM that is used to build the MDT runs in the provider backbone and for this reason is called a PIM P-instance (where P stands for provider). Note that this is a different instance of PIM than the one used within the VRF to set up the intersite trees. Therefore, two levels of PIM are used, the C-instance for building the trees in each customer's VPN and the P-instance for building the MDTs in the core (note that a single P-instance builds all the MDTs).

The choice of PIM as the signaling protocol, although intuitive in the MDT discussion, forces a series of subsequent decisions as well. PIM uses the multicast address to identify the members in the point-to-multipoint tree it builds. Using PIM to signal the MDT means that the MDT is identified by a multicast address, with the following consequences:

1. A multicast destination address must be assigned to each VPN and manually configured on each PE. This address is called the default MDT group address or P-group address (P stands for provider).
2. Traffic must arrive at the egress PE with a destination address equal to the MDT group address. This address allows the PE to determine which VPN the packet belongs to, the same functionality achieved by the use of a VPN label in the unicast case. In the multicast case, traffic is forwarded through the core using either GRE or IP-in-IP tunnels with the MDT group address.
3. GRE and IP-in-IP are the only tunneling technologies supported.

² See Section 10.6 for more details on this.

³ We will see that more than one MDT can be used per VPN when discussing data-MDTs later in this section.

4. P routers must participate in PIM for the set up of the MDT. Because multicast traffic is forwarded to the MDT multicast address through the service provider network, the P routers in the service provider network must be aware of this address, so they must participate in the PIM exchanges for building the MDT and they must maintain state for the MDTs created.

So far we have seen that each MDT can carry traffic from multiple multicast groups, as long as those groups belong to the same VPN. The detail of how many trees are formed depends on the implementation. Assuming the tree carries traffic for all multicast groups within a VPN, there are two options:

1. For each VPN, there is a single multicast tree rooted at a RP. If there are N VPNs in the network, then there are N multicast trees in the provider's network.
2. For each VPN, there is a multicast tree rooted at each PE. Hence, if there are N VPNs in the network present on each of M PEs, there are $(M \times N)$ multicast trees present in the provider's network. While the first option requires less state on the P routers than the second one, it is also less optimal with respect to the bandwidth consumed.

Note that in the discussion so far, we have always assumed that for a given VPN, each multicast tree extends to all the PEs that service a site of that VPN. This can be wasteful of bandwidth because each PE in a VPN receives all the multicast traffic of that VPN even if it has no sites attached to it that are interested in the multicast groups in question.

One solution to this is the data-MDT scheme in which a multicast group having a high volume of traffic has its own dedicated multicast tree that extends only to those PEs attached to customer sites that contain active receivers for the multicast group in question.

To summarize, the data plane of the draft-rosen mVPN solution uses per-VPN tunnels called MDTs in the service provider network to carry customer multicast data traffic. The MDTs are GRE or IP-in-IP tunnels set up using a separate instance of PIM, the PIM P-instance. The P routers in the service provider network must participate in the PIM P-instance exchanges and maintain state for the MDTs created. In doing so, the draft-rosen solution diverges from the unicast model in (a) the need for P routers to maintain per-VPN state and (b) the technology used for the tunnels.

10.4.3 Properties of the PIM/GRE mVPN solution

Figure 10.3 shows the conceptual model for supporting multicast in an L3VPN, using the draft-rosen solution described above. The draft-rosen

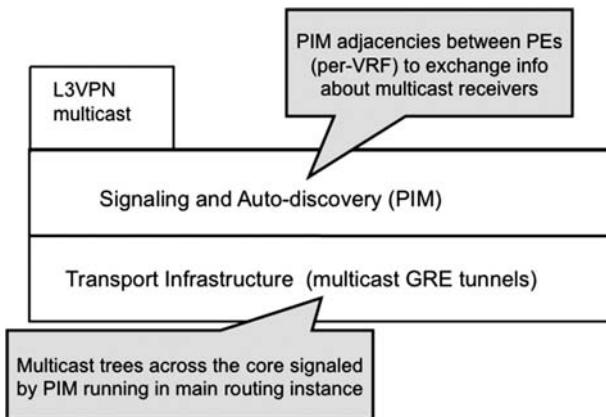


Figure 10.3 Conceptual model for L3VPN multicast using draft-rosen

approach differs from the BGP/MPLS VPN solution used for unicast traffic, both in the control and the data planes:

1. The control protocol used to convey customer routing information between PEs is PIM. In contrast, in the unicast case, BGP is used to carry the customer routes across the provider's core.
2. PIM peerings are required among the PEs that have sites of a given VPN connected to them, and these peerings have to be on a per-VPN basis. In contrast, in the unicast case, the routing information for all VPNs present on the PE is carried in the same BGP peering.
3. The control protocol for building inter-PE tunnels with VPN context is PIM. In the unicast case, BGP is used to carry the VPN label which provides the PE-to-PE tunnels at the VPN level, with either LDP or RSVP-TE as the control protocol for building inter-PE tunnels.
4. Multicast traffic is forwarded through the provider core encapsulated in tunnels set up with either GRE or IP-in-IP. In the unicast case, traffic is encapsulated in tunnels usually set up with MPLS (using either LDP or RSVP-TE as a label distribution protocol).

Let us take a look at some of the drawbacks of the draft-rosen solution.

1. *Large number of PIM adjacencies.* Recall from Section 10.4.1 that separate per-VPN instances of PIM are required to carry the customer routing information among PEs. Thus, assuming that a given PE services N VPNs running multicast, if each of these N VPNs has sites on M other PEs, then the PE must form $(M \times N)$ PIM adjacencies. In a network with 1000 VPNs, each with sites in 100 PEs, a PE servicing CEs from all VPNs must maintain 100 000 PIM adjacencies. This is in addition to the

adjacencies it needs to form anyway to its directly connected CEs. It is interesting to note the contrast with the unicast routing case, where the number of routing adjacencies a PE has to maintain is independent of the total number of sites in the VPNs present on the PE. In the worst case, a PE has a single routing protocol adjacency (a BGP session) with each of the M remote PEs, or when route reflection is in use, a small number of adjacencies with the route reflectors.

2. *Large control plane overhead.* The large number of PIM adjacencies that must be maintained means a large control plane overhead for maintaining PIM neighbor state, as each such adjacency requires a distinct PIM neighbor state, and it is the control plane that maintains this state. Additionally, because the maintenance of each PIM session requires the periodic multicast of PIM hello messages, there is a large volume of control-plane traffic. Referring back to the example above of a network with 1000 VPNs, each with sites in 100 PEs, the PE router servicing CEs from all VPNs would need to process 3300 hello messages per second (assuming default timers of 30 s), in addition to any other PIM traffic (such as Join messages), which are also exchanged periodically.
3. *Limitations of the MDT.* Because the MDT can only be set up with PIM in the original draft-rosen solution and because the tunnels are IP tunnels rather than MPLS tunnels, this approach cannot (a) provide protection, such as fast reroute for the traffic in the service provider core, or (b) create traffic-engineered trees, such as the minimum-cost trees discussed in Chapter 6. The use of IP tunnels rather than MPLS labels makes it more difficult for the service provider to protect against packet spoofing (see Section 5 of [RFC4797] for more on this).
4. *Manual configuration of the MDT group address.* The tunnel multicast destination address tells the receiving PEs which VPN the packet belongs to. For this purpose, the mapping between the multicast destination address and the VPN must be manually configured at each PE. Note that to ensure separation between different VPNs, a distinct multicast address must be assigned per-VPN.
5. *Per-VPN state in the core.* Because P routers participate in the PIM P-instances used for setting up the MDT and because there is at least one MDT per VPN, the amount of state maintained by the P routers is equal to at least the number of VPNs that have multicast support. Recall that in the unicast case, the P routers did not need to maintain any per-VPN routing state. Data-MDTs increase the amount of multicast state in the service provider network even more. Not only do the routers in the service provider network carry the per-VPN state, they also carry the state for those individual multicast groups within a VPN that have their own dedicated multicast tree.

6. *Lack of support for aggregation.* The problem with the amount of state maintained by the P routers in the service provider network lies with the fact that separate multicast trees are required per VPN. Ideally, one would want to be able to carry traffic from multiple VPN customers in a single service provider multicast tree. In such a case, the mapping of customers (S, G) into a tree would be determined by the location of the sources and receivers and would not be constrained by the VPN membership. This is analogous to the aggregation that is happening in the unicast case, where a transport tunnel between two PEs can carry traffic from multiple VPNs across the core.
7. *Challenges in inter-AS/interprovider deployments.* Both the control and the data-plane implementation of draft-rosen place limitations on such deployments. In the control plane, because of the requirement for direct peerings between PEs, PEs in different service providers are forced to (directly) exchange control traffic with each other. In the data plane, because of the way the MDT is set up, the same tunneling technology must be supported across all providers.

Having seen the basic operation and limitations of the draft-rosen solution, let us now take a look at the NG solution for multicast support for L3VPNs.

10.5 NG MULTICAST FOR L3VPN – BGP/MPLS mVPN (NG mVPN)

The NG mVPN approach brings the mVPN solution back into the framework and architecture of unicast BGP/MPLS VPNs by using BGP for the distribution of the routing information among PEs and MPLS for carrying the customer multicast traffic across the service provider network. This section will present the details of how this is done, by looking at various flavors of multicast deployments.

10.5.1 Requirements for support of PIM-SM SSM in an mVPN

Protocol-independent multicast (PIM) spare mode (SM) is documented in [RFC4601]. PIM-SM has two modes of operations: Single Source Multicast (SSM) and Any Source Multicast (ASM). In the SSM mode of operation, PIM-SM provides a service model where there is a single multicast source and multiple receivers. An example of an application using this model is video distribution from one (well-known) source to several receivers.

Although PIM-SM in SSM mode is less widely deployed than PIM-SM in ASM mode, in the context of the mVPN discussion, we will start by presenting it first as a customer mVPN deployment because of two reasons: (1) it is simpler both in the customer domain and the service provider network and (2) the mechanisms used to support it in an mVPN deployment form the foundation for supporting PIM-SM in ASM mode in an mVPN.

To understand what is required to support mVPNs running PIM-SM in SSM mode, let us start by looking at the requirements for PIM-SM in SSM mode in a plain IP (non-VPN) environment:

- The multicast sources must know that there are multicast receivers. This is accomplished by the receivers informing the sources that they are interested in receiving traffic, using PIM join messages.
- There is an assumption that the receivers discover the sources by means outside of PIM.
- There must be multicast forwarding state from the sources to the receivers, so traffic can flow to the receivers.

In a VPN environment, the same requirements must be satisfied, but in addition to these, the usual VPN requirements apply, namely support for (a) overlapping address spaces between different VPNs (for both unicast and multicast), (b) communication of routing information between PEs across the provider network and (c) forwarding of data traffic between PEs across the provider network. In the following sections, we will see the basic control plane and forwarding plane for the support of PIM-SM SSM in the NG mVPN model. Similar to the unicast discussion, we will split the NG mVPN discussion into intra-AS and inter-AS operations and start with the simpler intra-AS case. The inter-AS scenario will be described in Chapter 11, which covers advanced topics in mVPNs.

10.5.2 BGP/MPLS mVPN – carrying multicast mVPN routing information using C-multicast routes

As seen in Chapter 7, in an L3VPN, customer routing information is propagated between two sites in a VPN by advertising it between the remote PEs servicing these sites. In the case of unicast, the information is the prefix reachability, while in the case of multicast, it is the multicast group membership information which PEs obtain from the PIM join and prune messages that PEs receive from the CE routers. One of the major innovations of NG mVPN was to recognize that these join and prune messages are simply customer routing information and thus one could use BGP rather than PIM for carrying this information across the provider network by

encoding it as a special customer multicast (C-multicast) route using BGP Multiprotocol Extensions [RFC4760].⁴

[mVPN-BGP-ENC] defines the encodings and procedures for using BGP to advertise C-multicast routes across the provider network. In a nutshell, this is accomplished by defining a new BGP NLRI (network layer reachability information), the MCAST-VPN NLRI. When a PE receives a PIM join message from an attached CE, indicating the presence of a receiver for a particular multicast group and source in the customer site, the PE converts it to a BGP update containing the multicast group and source. As a result, the information received by the local PE in the PIM join is advertised to the remote PEs as a C-multicast route carried in BGP. At the remote PE, the information carried in the C-multicast route is converted back into a PIM join message that is propagated using PIM to the remote CE. Note that the CE devices are completely oblivious of what mechanism was used to propagate the information among the PEs across the provider network and simply continue to run PIM to the PEs they are attached to.

As with unicast, the BGP advertisement for a C-multicast route is made unique by attaching an RD to it (to deal with a situation where different VPNs use the same address space for unicast or multicast, or both) and is tagged with a route target (RT) extended community that will identify the correct VRF into which to import it.

By default in the unicast VPN case, all VPN routing information for a particular VPN, for example VPN X, is propagated to all PEs connected to sites belonging to VPN X and imported in the VRFs of VPN X present on all these PEs.⁵ In contrast, in the mVPN case, a C-multicast route carrying multicast routing information representing an (S, G) membership originated at a PE connected to a receiver, PE_R, should be propagated only to the PE connected to the site that contains the source, PE_S, and need not be propagated to any other PE. At PE_S, this route must be imported only in the VRF of VPN X, where the information carried by the route will be converted back into a PIM message and propagated to the CE.

To control the import of C-multicast routes into the VRF, each VRF has, in addition to the import RTs used for controlling other types of VPN routing information (such as unicast routes), an additional import RT extended community, called the C-multicast Import RT. An important property of the C-multicast Import RT is that it is unique across all VRFs and all PEs. The

⁴ One may think about this approach as a way to redistribute routes between protocols. The concept of redistribution is well-known and well-understood in the context of unicast routing, where one could redistribute OSPF routes into BGP. What is done here is to extend this concept to multicast routes, by providing a way to redistribute multicast routes from PIM to BGP and back.

⁵ Assuming a deployment where full-mesh connectivity is desired (as opposed to a hub-and-spoke topology for example).

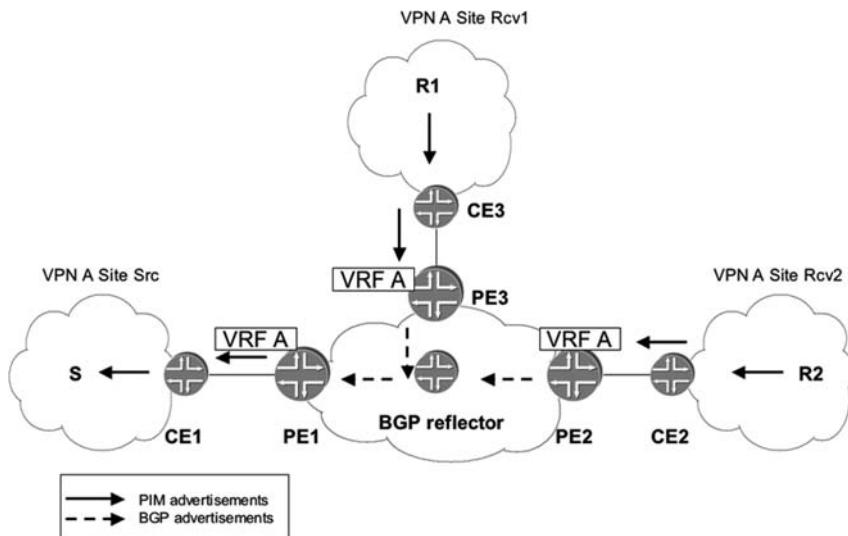


Figure 10.4 Distribution of mVPN routing information

uniqueness property is accomplished by embedding the PE's IP address and a locally assigned number into the RT (the PE assigns a distinct number for each VRF present on the PE). This ensures not only that different mVPNs have different values but also that different VRFs within the same mVPN have different values.⁶ The C-multicast Import RT, which is internally generated by the PE as explained above, is automatically included in the filtering policy that is applied to incoming BGP advertisements. Since the PE that includes in its filtering policy a particular C-multicast Import RT expects C-multicast routes intended for the PE to be tagged with this autogenerated RT, the value of this RT must be known to other PEs. This is done by sending the value of this RT in a new extended community, called VRF Route Import, along with the unicast advertisement of the reachability to the source. In the example above for VPN X, PE_S sends the VRF Route Import constructed from its address and a locally unique number along with the unicast advertisement for S. Based on PE_S's unicast route advertisement for the source S, PE_R will know what RT to attach to its C-multicast routes, such that these routes will be imported into the VRF of VPN X at PE_S.

Let us see the distribution of multicast routing information in the example in Figure 10.4. The figure shows the sites of VPN A from the original example in Figure 10.1. Two receivers, R1 and R2, exist in sites Rcv1 and

⁶ These VRFs are on different PEs.

Rcv2 of the VPN, serviced by PE2 and PE3, respectively. The source S is in site Src, serviced by PE1. Receivers R1 and R2 know about the existence of the source S.⁷ R1 and R2 advertise their interest in receiving multicast traffic from S by sending IGMP messages for a particular (S, G) to their Designated Routers, which in turn send PIM join messages for that (S, G) toward S. Let us continue the discussion from the point of view of R2 only, since the processing for R1 will be identical. The PIM join message for (S, G) is propagated within site Rcv2 that contains R2 towards CE2, which in turn propagates it to PE2. Note that on that PE, the PIM adjacency towards the CE is VRF-aware, and this is why the advertisement is shown in the picture as arriving into the VRF for VPN A.

When PE2 receives a PIM join from CE2, it uses the information carried in this PIM join to construct and originate a C-multicast route as follows:

1. PE2 finds in the VRF associated with VPN A the unicast VPN-IPv4 route to S and extracts from this route the RD and the VRF Route Import extended community.
2. PE2 builds a BGP C-multicast route that carries the source and group (S, G) information from the PIM join message received from the CE, the RD from the VPN-IPv4 route for S and an RT constructed from the VRF Route Import of the VPN-IPv4 route. Note that PE2 does not attach to the C-multicast route the ‘regular’ unicast RT associated with VPN A (the RT used by VPN-IPv4 routes).
3. PE2 sends the C-multicast advertisement to its BGP peers. In Figure 10.4, the advertisement is sent towards the route reflector shown in the middle of the provider network.

The route reflector receives the C-multicast advertisement from PE2 and the one similarly produced by PE3, aggregates them and propagates the aggregated advertisement to PE1. On receipt of the C-multicast route advertisement, the following actions are taken on PE1:

1. PE1 accepts the C-multicast route into the VRF for VPN A because the C-multicast Import RT (which was automatically generated on this PE and automatically included in the import policy for the VRF) matches the RT attached to the route. Any other PE receiving the route ignores it as it does not match *its* C-multicast Import RT, and the ‘regular’ unicast RT for VPN A is not attached to the route either.
2. As a result of accepting the C-multicast route advertisement, PE1 creates (S, G) state in the VRF and propagates the (S, G) join towards CE1 (attached to the site that contains the source S), using PIM as the PE-CE routing protocol. Note that this is the PIM instance running within the

⁷ Recall that the deployment under discussion uses PIM-SM in the SSM mode, which means that the receivers discover the source by means outside of PIM.

VRF on the PE1. (This is why in Figure 10.4 the advertisement is shown as originating in the VRF.)

Having seen how the routing information is distributed among PEs, let us take a look at some of the advantages of BGP for this task:

1. *Leverage of the unicast control plane.* The same BGP sessions and same BGP infrastructure used for the unicast routing exchanges can be used to carry the multicast information. For example, if route reflectors are in place, they can be used for signaling multicast routes as well as unicast routes.⁸ If RT filtering [RFC4684] is used for reducing the number of BGP advertisements, as explained in Chapter 8, it can be applied for multicast advertisements as well.
2. *Reduction in control plane overhead.* BGP only sends incremental updates, while PIM uses periodic complete updates.
3. *Ability to exert tight control over the exchange of control-plane information among service providers for VPNs that span multiple service providers.* This exchange could be confined to a particular set of ASBRs, which provide the place to control the exchange itself.
4. *Decoupling of the control and data plane used by mVPN.* The same control plane can support a variety of data-plane technologies, as there is no dependency of the control plane on data plane.
5. *Consistency with the unicast design.* This is important not just for reducing the operational overhead for the provider but also for reusing solutions from the unicast domain. A good example is support of complex VPN topologies, such as hub-and-spoke or extranets. As seen in Chapter 7, these topologies can easily be built by manipulating RTs and policies, but they are difficult to create when PIM is used as a control-plane protocol.

To summarize, the NG mVPN solution uses BGP to carry customer multicast information across the service provider network. It does so by redistributing PIM information into BGP and encoding it in C-multicast routes carried through the provider network similarly to the unicast case. As a result, the receivers in the customer sites can inform the sender that they are interested in receiving multicast traffic. The use of BGP as the control-plane protocol provides major scaling improvements to the control-plane state maintained on the PE routers, in addition to providing consistency with the unicast VPN solution. Having seen the distribution of routing information, let us take a look at how the data is actually forwarded.

⁸ Alternatively, separate route reflectors could be used just for the multicast routes, to spread the load among route reflectors.

10.5.3 BGP/MPLS mVPN – carrying traffic across the provider network using inter-PE MPLS tunnels

Just like in the unicast case, the PEs identify traffic arriving from the other PEs as belonging to that particular VPN based on the tunnels over which these packets arrive.⁹ In the simple case, assuming there is a single sender in the entire customer VPN and assuming that separate distribution trees are used within the service provider for each VPN, the distribution tree itself can be used to identify the VPN. This is similar to the use of the MDT in the PIM/GRE mVPN solution discussed in Section 10.4.2. However, while the draft-rosen approach restricts the MDTs to be PIM-signaled GRE-based tunnels, the NG mVPN solution allows for a wide range of tunneling technologies in the provider network. Of these, the most interesting is the use of MPLS P2MP LSPs as tunnels for transporting mVPN traffic between the PEs servicing the source and the PEs servicing the receivers. Using MPLS for the inter-PE tunnels is advantageous for two reasons: (1) the same protocol, namely RSVP-TE or LDP, can be used to establish MPLS tunnels for the purpose of carrying multicast traffic as for unicast traffic, resulting in a reduction in the number of protocols in the service provider network, and (2) when RSVP-TE is used as the signaling protocol, traffic engineering and protection can be achieved for the multicast traffic in the service provider network. It is interesting to note that PHP (as described in Chapter 1) must not be used when using P2MP MPLS tunnels in this context. This is because a PE must be able to identify traffic arriving at the PE as being associated with a particular VRF on that PE, and the only information that provides such identification is the MPLS label of the LSP that carries the traffic.

The use of MPLS as a transport technology gives its name to the BGP/MPLS mVPN solution. However, other options for inter-PE tunnels (including PIM-signaled GRE-based tunnels) are not precluded by the NG mVPN solution.¹⁰ The ability to use different tunneling mechanisms in the provider network brings much-needed flexibility to the service provider, whether it is in the context of a migration scenario, a legacy network or the optimization along a different set of parameters, and also makes this solution consistent with the unicast model of BGP/MPLS VPNs.

10.5.4 BGP/MPLS mVPN – inter-PE tunnels – inclusive and selective tunnels

In the discussion in the previous section, we focused on the role of the inter-PE tunnels in identifying the VPN to which the traffic belongs, but

⁹ Recall from Chapter 7 that the VPN labels create a one-hop VPN tunnel.

¹⁰ This is similar to unicast BGP/MPLS VPNs that also support GRE for inter-PE tunnels.

did not touch on which PEs the tunnel extends to or which multicast groups within the VPN use the tunnel. By default, inter-PE tunnels carry all multicast traffic for a particular VPN and extend to all PEs that have sites in the VPN, regardless of whether the sites have receivers for that traffic or not. Such tunnels are referred to as inclusive tunnels, or default tunnels. Inclusive tunnels may be wasteful of bandwidth, because traffic is forwarded to PEs which may end up discarding it. Imagine, for example, that in the network from Figure 10.5, an additional PE, PE4, was connected to a site of VPN A, site 4, which contains only unicast destinations and has no sources or receivers. The inclusive tunnel rooted at PE1 would span PE2, PE3 and PE4 even though PE4 has no interest in receiving any multicast traffic.

In contrast to inclusive tunnels, which carry traffic for all multicast groups in a VPN, selective tunnels carry traffic from just some multicast groups and can extend only to the PEs that have receivers for these groups. If in Figure 10.5, R2 is a receiver for group G1 and PE4 is connected to site 4 of VPN A, which contains receiver R4 for group G1, then the selective tunnel for group G1 would span from PE1 to PE2 and PE4, and not to PE3. This approach may be beneficial for high-bandwidth groups, where bandwidth optimization in the service provider network is required, but comes at the cost of having to create additional state for the extra trees in the service provider network. Note that the data-MDT of draft-rosen is nothing but a type of selective tunnel. In Section 10.5.6, we will see how

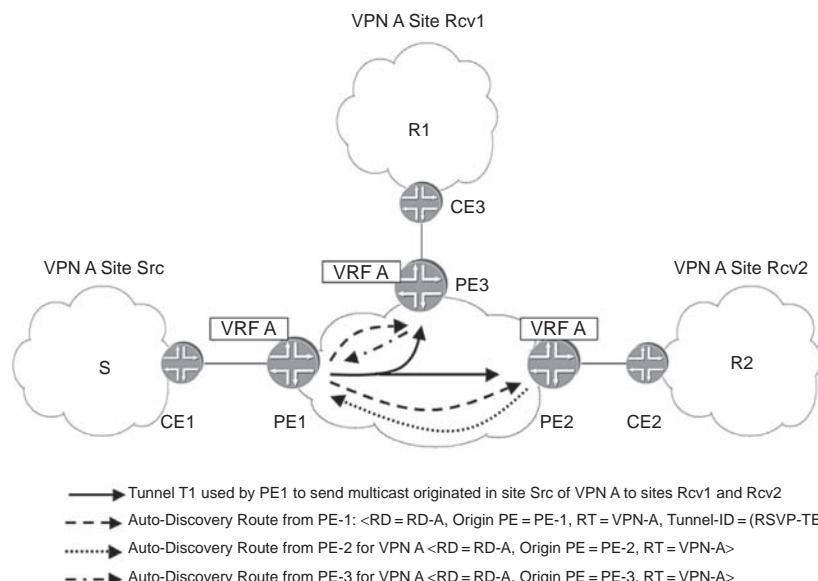


Figure 10.5 Use of autodiscovery routes

it is possible to switch from default to selective tunnels based on dynamic conditions.

10.5.5 BGP/MPLS mVPN – carrying traffic from several mVPNs onto the same inter-PE tunnel

Similar to the unicast case, where the tunnels in the provider network can carry traffic from multiple VPNs, the NG mVPN solution allows the P2MP trees in the provider network to aggregate traffic from multiple VPNs. This is not possible in the draft-rosen approach, where one of the fundamental design decisions was the use distinct per-VPN MDTs. (For data-MDTs, these are distinct MDTs per particular set of (S, G)s within the same VPN.) In other words, draft-rosen does not support the ability to aggregate traffic that belongs to different VPNs into a single tunnel. From the provider's point of view, the primary goal for aggregation is the flexibility of trading off bandwidth efficiency against the amount of multicast state in his network.

Let us take a look at two options that a PE may use for aggregating multiple VPNs present on the PE onto the same distribution tree:

1. *A single shared MDT rooted at the PE.* All VPNs present on the PE share the same distribution tree, resulting in a single tree in the service provider network. This reduces the amount of state in the core of the network, at the expense of potentially wasting bandwidth, by sending traffic to PEs that do not need to receive it. For example, in a network that supports two VPNs, VPN A and VPN B, if routers PE1, PE2 and PE4 are servicing sites that belong to VPN A and routers PE1, PE3 and PE4 are servicing sites that belong to VPN B, then with this option multicast traffic arriving at PE1 from a site in VPN A would be sent to PE2, PE3 and PE4. This would be the case even though PE3 definitely does not need to receive the traffic, not having any site of VPN A attached to it.
2. *Multiple shared MDTs rooted at the PE.* An alternative to a single distribution tree is to have multiple distribution trees, all rooted at the same PE, each carrying a different subset of VPNs. One criterion for sharing trees could be a similar geographical distribution, which implies an overlap in the PEs that service the VPNs in question. Yet another alternative would be to have multiple distribution trees, each carrying a subset of multicast groups. Therefore, from a given VPN, some multicast groups could be carried on one tree, other multicast groups on another tree and so on, regardless of which VPN the groups belong to. An example of where this is useful is if a service provider has POPs in key major cities and other smaller cities. If several VPNs have some multicast

groups that only have members in the major cities, it may be advantageous to have a multicast tree dedicated to serving those particular multicast groups.

Recall from the previous section that the MDT in the provider network was used to identify to which VPN traffic belongs to. In the cases where traffic from multiple VPNs shares the same distribution tree, the receiving PE must be able to identify the VPN to which a packet arriving on a MDT belongs. This is achieved by using a VPN label, as in the unicast case. Because all egress PEs must use the same label, this has to be an upstream-assigned MPLS label [RFC5331], as explained in Chapter 6 when discussing P2MP LSP hierarchy. The context identifying the label is the P2MP tunnel over which traffic is aggregated. The label binding is advertised via a new type of BGP update originated by the ingress PE, as we will see in Section 10.5.6.

The efficient use of aggregation depends on the ability to identify that two (or more) VPNs can be mapped onto the same distribution tree in the provider network. Ideally, a perfect overlap would exist for the two VPNs, but as long as the distribution tree is a superset of the PEs servicing both VPNs, this is not a strict requirement. In that case, traffic may be delivered to PEs that will immediately discard it,¹¹ but even in such a case, it may still be beneficial to maintain the aggregation in the provider network. At which point this benefit vanishes depends on the wasted bandwidth as well as on other network-specific considerations, such as the existence of a small-capacity path to a PE, which may get clogged with unnecessary traffic.

10.5.6 BGP/MPLS mVPN – creating inter-PE tunnels using BGP autodiscovery routes

In the previous sections, we have seen that inter-PE tunnels are used to carry traffic from the PE connected to the site containing the source to the PEs connected to the sites containing the receivers, thus providing the data plane for mVPN. Two things are required to set up the data plane. First, to enable the construction of such tunnels, it is necessary to discover to which PEs the tunnel must extend. This information must be acquired in an automatic way, because both mVPN membership and receiver information are likely to change, for example with the addition of a new site to the VPN or with a receiver leaving a particular group. Second, to allow for the same flexibility provided by the data-MDT in the draft-rosen solution, where some multicast streams are mapped to a separate distribution tree which spans only a subset of the PEs in the VPN, it is desirable to be able to

¹¹ Note that the same may happen for inclusive trees even without aggregation.

specify which (S, G) streams map to a particular inter-PE tunnel (this is a many-to-one binding).

Rather than inventing a new protocol for accomplishing these two tasks, the NG mVPN solution uses BGP for the automatic discovery and distribution of the information required for enabling the setup of inter-PE tunnels.¹² Before describing the solution, let us first see what information is required for building an inter-PE tunnel:

- The set of PEs to which the tunnel must extend. This information is required to enable the setup of the tunnel.¹³
- Information identifying the tunnel. This information enables all PEs in the VPN to discover what tunnels other PEs will be sending traffic on for a particular VPN, and install forwarding state accordingly. This information is also used by the tunnel signaling protocols (e.g. RSVP-TE, LDP) to set up the actual inter-PE tunnels. Several pieces of information may be required: (1) the type and identity of the tunnel used to carry traffic between PEs, (2) the (S, G) streams which may be mapped to this tunnel (if this is a selective tunnel that carries traffic for just some (S, G)) and (3) the upstream-assigned VPN label if aggregation is used. (If aggregation is not used the identity of the transport tunnel can be used to determine to which VPN traffic belongs to.)

To distribute this information with BGP, the MCAST-VPN NLRI is reused for advertising what is known as BGP autodiscovery routes between PEs. Similar to the unicast case, because this information is VPN-specific, the advertisements are made unique by attaching an RD and their distribution is constrained to the right VPN by tagging them with an RT. By default, the unicast RT can be used for tagging the autodiscovery routes, thus creating congruent unicast and multicast topologies, but a different RT could be used to control just the multicast topology. Note that the same BGP mechanisms available for VPN routes, such as RT filtering [RFC4684] and route reflectors can be reused for autodiscovery routes.

Let us now take a look at the actual information that is carried in the autodiscovery route. The membership information is implicitly carried by including the address of the advertising PE and the relevant RT in the autodiscovery route. In addition to the membership information, the autodiscovery route needs to carry information required to set up the inter-PE

¹² In Chapter 13, discussing VPLS, we will see another example where BGP is used for autodiscovery.

¹³ This statement is not entirely accurate, as we will see later in this section. The information required depends on the signaling protocol used. For example, if the tunnel is built with P2MP RSVP-TE, then the ingress of the tunnel must know the identity of the egresses (but not vice versa). If on the other hand mLDP is used, then the egresses must know the identity of the ingress but not vice versa.

tunnel. This information is carried in a new BGP attribute called the P-multicast Service Interface (PMSI) tunnel attribute and includes the type and identity of the tunnel. If aggregation is used, the upstream-assigned VPN label discussed in Section 10.5.5 is also included. The type of the transport tunnel determines the protocol used for signaling it, for example mLDP or P2MP RSVP-TE. The identity of the tunnel depends on the type of the tunnel and is used by a particular tunnel signaling protocol for setting up the tunnel. For example, if the tunnel type is P2MP LDP, the tunnel identifier is the <Root Node Address, Generic LSP identifier> that is carried in the LDP P2MP FEC. Note that the tunnel identifier may be allocated by a PE before the tunnel is actually instantiated. In the case of mLDP, for example, the P2MP LSP is leaf-initiated. In this case, the tree may not yet exist at the time when the root sends out its autodiscovery route, but the root could pre-allocate and advertise a tunnel identifier.¹⁴ Note that a PE connected to a site containing a source for a given mVPN must always generate an autodiscovery route with a PMSI tunnel attribute. However, PEs connected to sites containing receivers need not include the PMSI tunnel attribute in their advertisements, and in fact they do not need to generate the autodiscovery route at all, unless the tunnel type used for the mVPN is P2MP RSVP-TE (as in that case the identity of the leaves must be known to the source in order to enable the set up of the tunnel).

On receipt of an autodiscovery route by a PE, assuming that the import RTs for a particular VRF present on the PE matches the RT attached to the autodiscovery route, the PE imports the route into the VRF and performs the following two types of actions:

1. *Instantiation of the tunnel, if required.* For leaf-initiated trees like PIM or mLDP, the autodiscovery route generated by the root allows the receiver to find out the tree identifier and attach itself to the tree. For root-initiated trees, like P2MP RSVP-TE, autodiscovery routes received by the root allow it to identify the leaves that it should build its tree to. Note that in this case, the tree may be instantiated after the leaf PE has received the autodiscovery route from the root. Similarly, a new branch may be added to it by the root based on the receipt of an autodiscovery route from the leaf-PE.
2. *Creation of the forwarding state for mapping traffic arriving on the tunnel to a particular VRF.* The determination into which VRF to map the traffic is done based on the RT attached to the autodiscovery route. If no label is advertised in the PMSI tunnel attribute, then all traffic arriving on the tunnel is forwarded in the VRF, otherwise only traffic labeled with the VPN label is forwarded in the VRF. An interesting question arises regarding what happens when there are no receivers in a site attached to

¹⁴ The same may be true of RSVP-TE.

a particular PE, but the tunnel extends to all PEs in the mVPN (this could be the case when inclusive trees are used or when a tunnel aggregates traffic from multiple VPNs). Does this mean that it will be forwarded to the customer site? The answer is no, as there will be no forwarding state in the VRF for this multicast traffic (as there was no C-multicast route advertisement sent by the CE in this case).

In addition to the basic functionality described above, autodiscovery routes can provide other features as well. One example is the dynamic creation or teardown of tunnels based on external triggers. For example, it is easy to create a selective tree (the equivalent of a draft-rosen data-MDT tree) when traffic flowing for a particular (S, G) increases to a certain level, by the ingress PE simply generating the appropriate BGP autodiscovery route when the traffic reaches a given threshold. The autodiscovery route contains the identity of the selective tunnel and the (S, G) that is bound to it. Conversely, the tree can be torn down when no longer required by withdrawing the advertisement. In the case of a root-initiated tunnel, such as an RSVP-signaled P2MP, the ingress router needs to know which PEs have interested receivers for (S, G) . This is achieved by those PEs sending corresponding leaf-autodiscovery routes.¹⁵

Let us see the use of autodiscovery route in the example in Figure 10.5. The figure shows the sites of VPN A from the original example in Figure 10.1. Two receivers, R1 and R2, exist in sites Rcv1 and Rcv2 of the VPN, serviced by PE2 and PE3, respectively. A P2MP RSVP tunnel, T1, rooted at PE1, with PE2 and PE3 as leaves is used to carry the customer multicast traffic. To keep this example simple, assume that this tunnel was already set up based on the receipt of autodiscovery routes from PE2 and PE3, which advertised their membership in VPN A.¹⁶ Within the customer sites, multicast data-plane setup is done based on the PIM join messages. PE1 advertises an autodiscovery route for the P2MP LSP, tagged with its unicast RT for VPN A and including in the PMSI tunnel attribute the identity of tunnel T1.¹⁷ PE2 and PE3 receive this route and accept it into the VRF for VPN A. As a result, they install forwarding state mapping traffic that arrives labeled with the (non-null) label associated with the P2MP LSP into the VRF. When the source sends traffic to the receivers, the traffic arrives to PE1 on the data path that was created using the normal PIM procedures. It then gets mapped into the P2MP LSP and arrives at PE2 and PE3, where it gets into VRF A by virtue of the fact that it

¹⁵ In Chapter 16, we will see that leaf autodiscovery routes play an important part in the multicast solution for Seamless MPLS.

¹⁶ These autodiscovery routes did not include the PMSI tunnel attribute, as the sites connected to PE2 and PE3 do not contain any sources for VPN A.

¹⁷ Note that as mentioned before, the tunnel need not exist for PE1 to advertise its identity in the PMSI tunnel attribute.

arrived over the P2MP LSP. From the VRF, the traffic continues to the receivers using the forwarding state set up by the normal PIM procedures. In this case, the receipt of PIM-joins in the VRF for VPN A from the directly attached CE creates multicast forwarding state for traffic arriving into the VRF from the service provider network and which needs to be forwarded in the VRF. If no such state was created (for example because the receivers did not yet send the PIM-joins), the traffic is simply discarded at the PE.

To summarize, using BGP as an autodiscovery mechanism accomplishes two tasks: (1) it provides the information needed for the dynamic creation and teardown of P2MP inter-PE tunnels and (2) it allows the PEs to identify traffic arriving from other PEs as belonging to a particular VPN, thus making it possible to forward the traffic in the appropriate VRF.

10.5.7 Requirements for support of PIM ASM in an mVPN

Having seen the basic operation of BGP/MPLS mVPNs when the VPN customer is using the PIM-SM SSM mode of operation, let us now look at a few more advanced topics that arise in the context of PIM ASM. In the ASM mode of operation, PIM-SM provides a service model where there are multiple sources and multiple receivers for the same group. An example of an application using this model is a video-conferencing service, where many sources come and go and the locations of all these sources must be known to all receivers.

To understand how VPN customers running PIM-SM in the ASM mode can be supported, let us first look at the ASM mode of operation in the context of a plain IP (non-VPN) scenario. There are two important concepts that distinguish PIM-SM in the ASM mode from PIM-SM in the SSM mode. The first one is the concept of RP, the second is the concept of (multicast) domains. In PIM-SM in ASM mode, the discovery of the multicast sources is accomplished by introducing the concept of an RP as a centralized entity that knows about all the active sources. Designated routers connected to active multicast sources register the sources with the RP using PIM register messages. Receivers join an RP Tree (RPT) for the sole purpose of discovering the sources. However, receiving traffic over the RPT may not be ideal, and receivers may switch from the RPT to a shortest path tree (SPT) (typically as soon as the first packet is received from the source). This RPT/SPT interaction introduces a fair amount of additional complexity to the ASM mode of operation.

PIM-SM in the ASM mode defines the concept of a ‘multicast domain’. PIM-SM in the ASM mode supports interdomain operations by using the Multicast Source Discovery Protocol (MSDP) [RFC3618] to exchange

information about active multicast sources among RPs in different domains. Since MSDP exchanges information about active multicast sources, it follows that only (S, G) information is exchanged among domains, even if within each domain both (*, G) and (S, G) information is exchanged. As a result, even if within each domain PIM-SM in the ASM mode is used, interdomain operations effectively look like PIM-SM in the SSM mode.

Given that at the interdomain level PIM-SM in ASM mode relies on the PIM-SM in SSM mode procedures, it follows that if an mVPN running PIM-SM in the ASM mode could be modeled as a collection of multicast domains interconnected by a service provider network, then the same mechanisms described in the previous sections for supporting mVPNs running PIM-SM in the SSM mode could be leveraged to support the ASM deployment. The next section describes how to accomplish this.

10.5.8 BGP/MPLS mVPN – carrying mVPN active source information using BGP source active autodiscovery routes

One can think of an mVPN as a collection of PIM-SM ASM multicast domains that must be interconnected across the service provider mVPN infrastructure, where each domain consists of all the mVPN sites connected to a given PE, plus the PE itself. As mentioned in the previous section, to create such an interconnection in a plain IP network, the information about the active sources is distributed between the RPs in each domain by running MSDP between the RPs.

To interconnect these domains in the context of an mVPN, RPs are placed on the PEs. That is, a PE acts as a customer RP (C-RP) for the multicast domain formed by all the sites of a given mVPN connected to a given PE. Because the C-RP and the PE are the same entity, this mode of operation is referred to as the collocated RP/PE model. (Note that in this model every PE that has sites of a given mVPN connected to it becomes an RP for that mVPN.) A mechanism must be set in place to exchange information about active (multicast) sources among these PEs/C-RPs. This is done using a new type of BGP route, the BGP source active autodiscovery route, carried in the MCAST-VPN NLRI.¹⁸

Figure 10.6 shows the mVPN deployment for VPN A where two sources, S1 and S2, located in sites Src1 and Src2 respectively, send traffic to receivers

¹⁸If MSDP were used to provide this function, it would need to be extended to support (a) overlapping address spaces, (b) constrained distribution of routing information and (c) a hierarchical control plane. However, BGP already has support for these features and only misses the ability to encode information about active sources as a BGP NLRI.

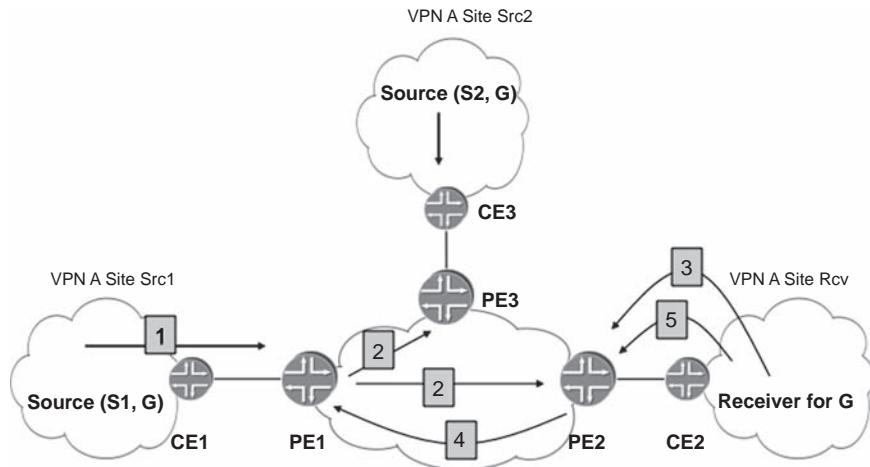


Figure 10.6 PIM-SM ASM collocated RP model – routing information exchange

in site Rcv of the VPN. A PE that acts as a collocated C-RP finds out about the active sources among the sites connected to the PE by using the same procedures as the RP in a plain IP multicast scenario. In the example, PE1 finds out about S1 through the receipt of a PIM register message from the Designated Router connected to S1. This is the exchange labeled 1 in the figure. The PE then advertises this information to the other PEs (who also happen to be C-RPs) using a BGP source active autodiscovery route. In the example, PE1 sends source active autodiscovery routes to PE2 and PE3, the two exchanges labeled 2 in the figure.

As with C-multicast routes, the BGP source active autodiscovery routes carry in addition to the (S, G) information, two additional pieces of information: (1) an RD to make advertisements unique per VPN and (2) an RT to constrain the distribution of the route to all the VRFs of a given mVPN (this RT can be the same as the RT used by unicast). Based on the receipt of source active route for a particular (S, G) from a remote PE, the local PE knows to which PEs to generate C-multicast routes when a Join (*, G) arrives from one of the receivers it services. The PE generates several C-multicast routes, one per each received source active autodiscovery route that has G. In the example in Figure 10.6, when PE2 receives a Join (*, G) from CE2 (exchange 3 in the figure), it generates a single C-multicast route (exchange 4), since it had received only a single source active autodiscovery route so far. When a receiver switches from the RPT to the SPT, the switch is localized to that site. In the example, if R switches to the SPT (exchange 5) by sending a Join (S1, G), no further advertisements need to be propagated in the service provider network. As a result, there is no shift

in the traffic patterns in the service provider network when receivers in an mVPN switch from shared (RP-based) to source-based trees.

The above scheme works as long as an mVPN is willing to outsource its RPs to the mVPN service provider. What about mVPNs that do not want to outsource their RPs to the mVPN service provider? To answer this question note that as long as there is a way to provide one or more PEs with the information about active (multicast) sources of a given mVPN, it really does not matter whether these PEs act as fully functional C-RPs or not. In the scenarios where the RP is maintained within the mVPN customer's network the existing IP multicast mechanisms could be used to communicate information about active sources (S, G) from the mVPN RP to one or more PEs by either (a) running MSDP between the RP and the PE or (b) sending a PIM Register message between the RP and the PE. In both cases, the PE maintains information about the active sources for a given mVPN (just like in the collocated RP/PE model), but the PE does not act as the RP. However, just like in the collocated RP option, the PE generates a BGP source active autodiscovery route based on the sources it knows about. Referring back to the example in Figure 10.6 and assuming that PE1 does not act as the RP, but instead RP1 is maintained within site Src1 of VPN A, only the exchange labeled 1 will differ as compared to the collocated PE/RP model. Namely, instead of S1 sending the PIM-register message to PE1, S1 sends the PIM-register message to RP1, and assuming MSDP is used between RP1 and PE1, RP1 sends an MSDP message to PE1. From this point on, the same exchanges as in the collocated model will take place. When S2 becomes active, the Designated Router connected to S2 sends PIM Register to the RP. Note that even if S2 and RP are in different sites, all what is required to propagate the PIM Register is unicast connectivity¹⁹ (as PIM Register are unicast messages).

So far we covered just the exchange of multicast routing information. But what about exchange of multicast data traffic? It is easy to see that all the mechanisms developed for the exchange of multicast data traffic for mVPNs running PIM-SM in the SSM mode work 'as is' in the context of mVPNs running PIM-SM in the ASM mode.

To summarize, to support PIM-SM in the ASM mode, information about active sources must be advertised between the PEs servicing sites of the mVPN. The PEs find about the active sources in the sites attached to them either by acting as the RP or by communicating with the RP (e.g. using MSDP). To distribute the source information among the PEs in the mVPN,

¹⁹ In this scenario, the source active autodiscovery route is originated by PE that is NOT connected to the site that contains the source. That is not a problem, because the source active autodiscovery route simply indicates that there is an active source and it does not convey information about the site where the source is located. The latter is provided by the (unicast) VPN-IPv4 routes.

a new type of route, the BGP source active autodiscovery route is used. Based on the receipt of this information, the PE will generate separate C-multicast routes towards each one of the PEs servicing the sources, when a Join (*, G) arrives from a receiver that it services.

10.6 COMPARISON OF PIM/GRE AND BGP/MPLS mVPNs

As mentioned in the beginning of the chapter, the L3VPN IETF working group draft on multicast support, [VPN-MCAST] covers both the PIM/GRE (draft-rosen) and the BGP/MPLS (NG) approaches, and deployments of draft-rosen exist in several large networks. Having described both solutions in this chapter, let us compare them by looking at (a) the VPN model implemented by each of the solutions, (b) the control-plane protocol used in each case, (c) the data-plane mechanisms, (d) the operation in an inter-AS scenario and (e) deployment considerations for a service provider. Each of these aspects will be examined separately below.

10.6.1 VPN model used

Some of the most significant differences between the two solutions are rooted in the underlying VPN models used: the Virtual Router model, used by the PIM/GRE solution, and the Aggregated Routing model, used by the BGP/MPLS solution. Therefore, in order to better understand the differences between the two approaches, let us first examine the differences between the two models.

In the Virtual Router model, the exchange of VPN routing information among PEs is accomplished by operating separate instances of routing protocols among the PEs, one instance for each VPN. The exchange of VPN data traffic among PEs is accomplished by setting up VPN-specific tunnels between PE devices, where logically these tunnels are between the VRFs which are within the PE devices. These tunnels are used as if they were normal links between normal routers, and therefore routing protocol data for each customer VPN is also tunneled over them, creating a very tight coupling between the control and data planes.

In contrast to the Virtual Router model, the Aggregated Routing model uses a single instance of a routing protocol for carrying VPN routing information among the PEs, and the routing information for multiple different VPNs is aggregated into this instance. Just like with the Virtual Router model, the Aggregated Routing model uses VPN-specific tunnels set up between PE devices to carry data traffic between the PEs. However, in contrast to the Virtual Router model, these tunnels are used solely by the

data plane, and routing protocol data for the VPN is not forwarded over them. As a result, the exchange of VPN routing information among PEs (control plane) is fully decoupled from transporting VPN user data traffic between PEs (data plane). This, in turn, facilitates support for various tunneling technologies with the same common control plane.

Let us compare the two models in terms of two different properties:

1. *Number of routing adjacencies maintained.* Exchange of VPN routing information in the Virtual Router model requires establishment of a distinct control plane operating across the service provider network for each VPN, which results in requiring PEs to maintain a potentially large number of routing peers and routing adjacencies. (Section 10.4.3 provides an example computation in the context of PIM peerings maintained by the draft-rosen solution.) The Aggregated Routing model greatly reduces the number of routing peers and adjacencies which the PEs must maintain relative to the Virtual Router model, as there is no longer any need to maintain more than one such adjacency between a given pair of PEs.
2. *Support of different tunneling technologies for forwarding traffic.* In the Virtual Router model, there is a tight coupling between the control and the data planes, as the data plane is also used for forwarding the control-plane information. This makes it difficult to support other technologies for setting up the inter-PE tunnels. In contrast, in the Aggregated Routing model there is no such dependency.

It is easy to see that the PIM/GRE solution is nothing but an instance of the Virtual Router model, while the NG mVPN solution (just like unicast BGP/MPLS) is an instance of the Aggregated Routing model. Therefore, the drawbacks of the Virtual Router model are applicable ‘as is’ to the PIM/GRE solution. Likewise, all the benefits of the Aggregated Routing model are applicable ‘as is’ to the NG mVPN solution.

10.6.2 Protocol used in the control plane

More differences between the two approaches are a consequence of the use of PIM and BGP, respectively, as the control-plane protocol for exchanging mVPN customers multicast routing information. As we will see below, they compound the disadvantages of the Virtual Router model and enhance the advantages of the Aggregated Routing model.

The disadvantages of the Virtual Router model are compounded in the draft-rosen solution by the use of PIM as a control-plane protocol due to the following additional factors:

- The need for periodic refreshes. PIM relies on the periodic exchange of the complete routing information. In contrast, the NG mVPN solution benefits from using BGP to exchange mVPN multicast routing information, as BGP is based on the technique of incremental updates, and therefore is more efficient in terms of control-plane resources than PIM.
- The need for direct peerings in the inter-AS /interprovider scenario. The PIM/GRE solution requires PEs in different ASs/providers to have (direct) PIM routing peering, as long as these PEs have at least one mVPN in common. This is one of the direct consequences of following the Virtual Router model. In contrast, the NG mVPN solution allows restricting the exchange of routing information (including mVPN routing information) to only the ASBRs and does not have a direct exchange of routing information among PEs belonging to different ASs/providers.

The advantages of the NG mVPN solution over draft-rosen are enhanced by the use of BGP rather than PIM as a control-plane protocol due to the following factors:

- Support for hierarchical route distribution (hierarchical control plane). BGP has built-in support for hierarchical route distribution using route reflectors. This allows the NG mVPN solution to completely eliminate the PE–PE routing adjacencies and make the number of backbone adjacencies a PE has to maintain into a small constant which is independent of the number of PE devices, which in turn, significantly improves the scaling properties, compared to the Virtual Router model. In inter-AS setups, the ASBRs are also part of the hierarchical control plane of BGP, as the exchange of BGP routes between adjacent ASs is confined to the ASBRs that interconnect these ASs.
- Support for scalability mechanisms for route distribution. In contrast to PIM, which has no built-in scalability mechanisms, BGP has support for several of them, for example (1) mechanisms to constrain the distribution of routes (e.g. using RT constraints), (2) ability to do route dampening and (3) aggregation of routing information at the route reflector.²⁰

10.6.3 Data-plane mechanisms

In the data plane, the NG mVPN solution has two advantages over draftrosen:

1. *Automatic tunnel discovery and tunnel binding.* In the draft-rosen solution, the construction of the MDT relies on manual configuration of the group

²⁰This advantage will become apparent when discussing inter-AS operation in Chapter 11.

address.²¹ In the NG mVPN solution, tunnel discovery and binding is automatically accomplished using BGP.

2. *Support for aggregation.* One of the main drawbacks of the draft-rosen solution is the lack of support for aggregation, as a given MDT cannot carry traffic of multiple mVPNs. In contrast, in the NG mVPN solution, aggregation can be easily achieved as explained in Section 10.5.5. Because the routers in the core of the network participate in the setup of the inter-PE tunnels, lack of support for aggregation increases the amount of both control and data plane state on these routers.

10.6.4 Service provider network as a ‘LAN’

The PIM/GRE solution models the interconnect of Virtual Routers present on the PEs as a LAN, which implies that all these Virtual Routers are equidistant from each other, even in the interprovider scenario where PEs servicing a particular mVPN may be part of different service providers. The LAN model reflects rather poorly the underlying service provider infrastructure. In contrast, the NG mVPN solution provides a straightforward way to reflect the underlying service provider infrastructure in its routing decision, as it does not assume that all the PEs are equidistant from each other and can take into account the inter-PE distance in the VPN route selection procedures.

10.6.5 Deployment considerations

Finally, an important but often overlooked aspect of the comparison is the impact on the service provider operations. The draft-rosen solution requires the deployment of PIM in the service provider network in order to support mVPN traffic. In contrast, in the NG mVPN approach, the same protocols used by the unicast VPN solution are reused, with extensions. Avoiding the deployment of an extra protocol in the network is an important consideration from a point of view of operational expenses for a service provider.

To summarize,²² the PIM/GRE-based solution for multicast support departs from the 2547 model in that it implements the Virtual Router

²¹ [VPN-MCAST] specifies an option to use the BGP autodiscovery mechanism even when the control-plane protocol is PIM. This option is not explained in this chapter. From a service provider’s point of view, the need for an extra protocol (BGP) in a PIM deployment is not ideal.

²² In addition to the dimensions discussed in Sections 10.6.1 through 10.6.5, there are significant advantages to the mVPN technology in the context of inter-AS operations, as we will see in Chapter 11.

model rather than the Aggregated Routing model for VPN support and uses different mechanisms in both the control and the data planes. By doing so, it not only introduces a second set of mechanisms for multicast but also loses a lot of the scalability and flexibility of the unicast L3VPN solution. In contrast, the BGP/MPLS-based approach reuses the unicast L3VPN unicast mechanisms with extensions as necessary, thus retaining as much as possible the flexibility and scalability of unicast.

10.7 CONCLUSION

The L3VPN solution cannot be complete without support for multicast traffic. Although the original PIM-based solution described in draft-rosen could carry VPN customer traffic over a shared provider infrastructure, this approach departed from the L3VPN unicast model and suffered from scaling limitations both in terms of the maximum number of mVPNs it could support and in terms of the efficiency with which it could carry traffic through the provider network. As L3VPN deployments grew in size and as bandwidth-intensive multicast applications such as IPTV grew in popularity, the need for a more scalable way to handle L3VPN multicast traffic became imperative.

As a result, the NG mVPN solution was developed in the IETF and first commercial implementations became available shortly afterwards. The defining features of the NG mVPN solution are (a) carrying PIM join/prune information across the provider network using C-multicast routes in BGP, (b) discovering mVPN membership information using BGP autodiscovery routes, (c) allowing any tunneling technique in the provider network and (d) supporting aggregation of traffic from multiple VPNs onto the same tunnel in the provider network. In a nutshell, the NG mVPN solution uses BGP for the control plane and allows flexibility in the technology used for the tunnels in the provider network, just like L3VPN unicast. This alignment of unicast and multicast enables the reuse of solutions from the unicast domain for multicast, as well as reducing the number of protocols the provider must run in the provider network for enabling L3VPNs. In the next chapter we will explore some of the advanced topics that arise in the context of mVPNs.

10.8 REFERENCES

[DRAFT-ROSEN]

E. Rosen, Y. Cai and I. Wijnands, *Multicast in MPLS/BGP IP VPNs*, draft-rosen-vpn-mcast-08.txt (expired draft)

- [L3VPN-WG] <http://ietf.org/html.charters/l3vpn-charter.html>
- [mVPN-BGP-ENC] R. Aggarwal, E. Rosen, T. Morin, Y. Rekhter and C. Kodeboniya, *BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs*, draft-ietf-l3vpn-2547bis-mcast-bgp-08.txt (in the RFC editors' queue, soon to become RFC)
- [VPN-MCAST] E. Rosen and R. Aggarwal, *Multicast in MPLS/BGP IP VPNs*, draft-ietf-l3-vpn-2547bis-mcast-10.txt (in the RFC editors' queue, soon to become RFC)
- [RFC3618] B. Fenner and D. Meyer, *The Multicast Source Discovery Protocol (MSDP)*, RFC3618, October 2003.
- [RFC4601] B. Fenner, M. Handley, H. Holbrook and I. Kouvelas, *Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Revised)*, RFC4601, August 2006.
- [RFC4684] P. Marques et al., *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*, RFC4684, November 2006.
- [RFC4797] Y. Rekhter, R. Bonica and E. Rosen, *Use of Provider Edge to Provider Edge (PE-PE) Generic Routing Encapsulation (GRE) or IP in BGP/MPLS IP Virtual Private Networks*, January 2007.
- [RFC4834] T. Morin, *Requirements for Multicast in Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs)*, RFC4834, April 2007
- [RFC5331] R. Aggarwal, Y. Rekhter and E. Rosen, *MPLS Upstream Label Assignment and Context Specific Label Space*, RFC5331, August 2008

10.9 FURTHER READING

- [mVPN-STD-OPT] T. Morin, B. Niven-Jenkins, Y. Kamite, R. Zhang, N. Leymann and N. Bitar, *Mandatory Features in a Layer 3 Multicast BGP/MPLS VPN Solution*, draft-ietf-l3vpn-mvpn-considerations-06.txt (in the RFC editors' queue, soon to become RFC)

[mVPN-TUTORIAL]

Y. Rekhter and M. Bocci, *Multicast in MPLS/VPLS Networks*, Tutorial at the MPLS 2007 Conference, October 2007, Washington DC

10.10 STUDY QUESTIONS

1. What are the various options for setting up MDTs in the service provider network in the draft-rosen proposal and how do these map to different PIM modes? Compare the state created in the provider's network in each of these cases to the state that is created by LSPs in an L3VPN unicast deployment.
2. What could be some of the reasons why the increase in the bandwidth used by multicast traffic was one of the drivers for moving away from the draft-rosen solution?
3. The benefits of using RSVP-TE P2MP LSPs for the transport tunnels in the provider network were discussed in Section 10.5.3. What would be some of the advantages of some of the other technologies for instantiating transport tunnels?
4. Aggregation of traffic from multiple VPNs onto a shared distribution tree relies on identifying the VPN to which the traffic belongs through the use of an upstream-allocated label. What restrictions does this model place on the types of transport tunnels that can be used in the provider network?
5. The MCcast-VPN NLRI does not have support for prune routes, only for join routes (see [BGP-mVPN-ENC]). What is the reason for this decision?
6. What is the use of the C-multicast Import RT and why does it require changes to the unicast advertisement of the route for the source?
7. If the P2MP tree from PE1 to PE2 and PE3 in Figure 10.5 was set up using P2MP LDP rather than P2MP RSVP-TE, then routers PE2 and PE3 need not have sent autodiscovery routes. How would the tunnel have been set up in that case?
8. How does the BGP/MPLS mVPN solution support aggregation?
9. What would happen when source S2 becomes active in the example shown in Figure 10.1 and discussed in Section 10.5.8?

11

Advanced Topics in BGP/MPLS mVPNs

11.1 INTRODUCTION

The previous chapter described the original PIM/GRE-based solution for providing multicast, explored its shortcomings, and introduced the basic concepts for the alternative BGP/MPLS mVPN architecture, also referred to as NG mVPN. In this chapter, we discuss advanced topics related to BGP/MPLS mVPNs. We cover inter-AS operations, PIM Dense-Mode (DM), RP discovery, extranets, migration from PIM/GRE to BGP/MPLS mVPNs, and scaling. Finally, we conclude by looking at achieving high availability for video traffic using mVPNs and leveraging some of the BGP/MPLS mVPN mechanisms for delivering Internet multicast over an MPLS core.

11.2 BGP/MPLS mVPN – INTER-AS OPERATIONS

The discussion in the previous chapter was limited to a scenario in which all the sites of a given VPN are connected to the PEs that are in the same AS. However, as we saw in Chapter 9 (Hierarchical and inter-AS VPNs) for the unicast case, sites of a given VPN can be connected to PEs that are in different ASs. To allow operation in such a case two things must happen: an inter-PE tunnel must be set up across multiple ASs, and multicast routing information must be exchanged between the PEs in the different ASs.

Let us see how this is done in the simple context of a deployment spanning two ASs, AS1 and AS2, by looking at the inter-PE tunnel establishment and the multicast information distribution separately. Figure 11.1 shows two ASs, AS1 and AS2, connected at ASBR1 and ASBR2. (For simplicity, a single ASBR is assumed at the interconnect point in each AS, though in practice there will be several for redundancy.) VPN A has four sites, two in each AS. A multicast source S is located in site Src and three receivers, R1, R2, and R3, are located in sites Rcv1, Rcv2, and Rcv3, respectively. Sites Src and Rcv1 are in AS1, and sites Rcv2 and Rcv3 are in AS2. As seen in Figure 11.1, to forward the multicast traffic, an intra-AS tunnel is set up within AS1 from PE1 (servicing site Src) to PE2 and ASBR1. ASBR1, in turn, originates an inter-AS tunnel spanning across the two ASs and consisting of two segments, a single-hop inter-AS segment between the adjacent ASBRs (ASBR1 and ASBR2) and an intra-AS segment within AS2 from ASBR2 to PE3 and PE4.

The intra-AS tunnel within AS1 is set up using intra-AS BGP auto-discovery routes, as explained in Section 10.5.6. The inter-AS tunnel is initiated by the ASBR within the AS in which the source is present, in this case ASBR1. ASBR1 sends an inter-AS auto-discovery route to its peer ASBR2. This route includes the Route Target (RT) from the intra-AS auto-discovery route and the identity of the AS in which this route was generated. Importing the auto-discovery route into the correct VPN is governed by the RT, just as in the intra-AS case. To improve scaling, ASBR1 aggregates all intra-AS auto-discovery routes of a given mVPN

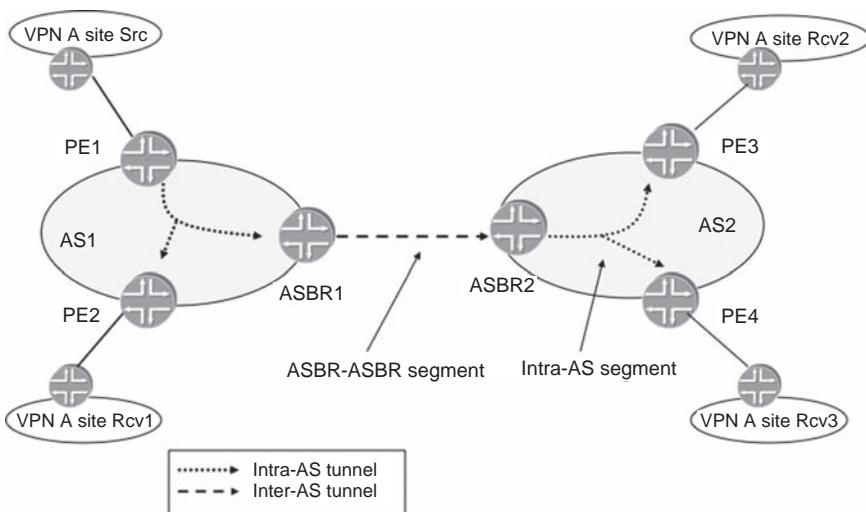


Figure 11.1 Inter-PE tunnel in an inter-AS mVPN deployment

originated within ASBR1's own AS into a single inter-AS advertisement originated by ASBR1, rather than advertising each such intra-AS auto-discovery route individually to other ASs. More precisely, an ASBR that originates a given inter-AS auto-discovery route aggregates the intra-AS auto-discovery routes that carry exactly the same set of RTs. (Note that this set may have just one RT.) The inter-AS auto-discovery route originated by the ASBR carries exactly the same RTs as the routes it aggregates. For example, if inter-AS auto-discovery route AD1 aggregates intra-AS auto-discovery routes AD2...ADn which are tagged with RTx, then AD1 is also tagged with RTx. Note that performing this type of aggregation significantly enhances the scalability of the inter-AS solution, because the number of inter-AS auto-discovery advertisements is much smaller than the number of PEs. For example, assuming in Figure 11.1 M PEs in AS1 and N PEs in AS2 for VPNA, the number of advertisements would be two rather than (M+N). Because the auto-discovery routes are what govern the creation of inter-AS tunnels, this translates directly into a gain in data plane scalability.

An interesting situation arises because of the fact that different ASs can use different technologies for setting up point-to-multipoint tunnels, for example mLDP in AS1 and P2MP RSVP in AS2. Because inter-AS auto-discovery routes are built on the notion of segments (that is, intra-AS and inter-AS segments) that are stitched at ASBRs, different intra-AS segments of a given inter-AS tunnel can be constructed using different tunneling technologies (inter-AS segments are always constructed using BGP). This is accomplished by the ASBRs modifying the type and identifier of the tunnel, which are carried in the PMSE Tunnel attribute of the inter-AS auto-discovery routes. In our example, the intra-AS tunnel in AS1 can be constructed using RSVP-TE, while the intra-AS segment in AS2 can be constructed using mLDP. ASBR1 is going to merge all the intra-AS tunnels of VPN A into the inter-AS segment of the inter-AS tunnel originated by ASBR1.¹ ASBR2, when it re-advertises the inter-AS auto-discovery route for VPN A received from ASBR1 into its own AS, AS2, then specifies in the PMSE Tunnel attribute of the route the type and identifier of the intra-AS segment of that tunnel, as required by mLDP, and also stitches the tail of the inter-AS segment to the head of the intra-AS segment.

So far we have seen how traffic can be forwarded across the two ASs. Let us now turn our attention to the distribution of multicast information. Recall from Section 10.5.2 that C-multicast routes propagate from PEs connected to sites that contain the receivers to the PE connected to the site that contains the source, and the importing of C-multicast routes into the correct VRF is governed by the C-multicast Import RT, whose value

¹The inter-AS segment is created by ASBR2 generating a leaf autodiscovery route containing a downstream-assigned label that it sends to ASBRI.

is carried in the VRF Route Import extended community attached to the VPN unicast advertisement of the route to the source. In the inter-AS case, the distribution of the C-multicast route should be restricted in one more way, namely by restricting it to the ASBR leading towards the AS in which the PE connected to the site that contains the source resides. In this way, the C-multicast route is not unnecessarily propagated into ASs that do not need it.

Therefore, in addition to the C-multicast Import RT, an additional route target must be attached to the advertisement to limit the importing of the C-multicast route only to the ASBRs leading towards the source. This new RT, which we refer to as the C-multicast ASBR Import RT, is automatically generated from the address that the ASBR places in the next hop of the inter-AS auto-discovery routes that it advertises. This RT is also automatically included in the ASBR's filtering policy for BGP advertisements, similar to how the C-multicast Import RT is handled. If the source-AS (that is, the AS that contains the PE connected to the source) is known, the identity of this ASBR can be obtained from the BGP next hop of the inter-AS auto-discovery route, because the route carries the AS number of the AS that originates the route. Thus, to support inter-AS operations, an additional extended community called Source AS, which carries the AS number of the PE that is connected to the site where the source resides, is attached to the unicast advertisement for the source address.

Let us see the distribution of routing information based on the example in Figure 11.2. For simplicity, the C-multicast route is shown only for receiver R2 in AS2. R1, R3, and their respective PEs have been removed to make the figure easier to understand.

- PE1 originates an inter-AS unicast route advertisement for the source S. In addition to the VRF Route Import extended community, whose value is equal to the value of the C-multicast Import RT of the VRF for mVPN A on PE1, this advertisement also contains the Source AS extended community identifying AS1 as the AS in which PE1 (the PE connected to the site that contains the source) resides. This route is propagated to other PEs that have VPN A, including PE3.
- When PE3 determines that it has receivers for (S, G) in mVPN A, it generates a C-multicast route. To do so, PE3 first finds the unicast VPN-IPv4 route to S. In addition to the C-multicast Import RT derived from the VRF Route Import of the unicast route to source S, PE3 also extracts from this route the source AS (AS1 in Figure 11.2), as carried in the Source AS extended community. Then, PE3 finds an inter-AS auto-discovery route for mVPN A originated by an ASBR in the source AS, AS1, and uses the next hop of this route (ASBR2 in this example) to construct the C-multicast ASBR Import RT, which it attaches to the C-multicast route.

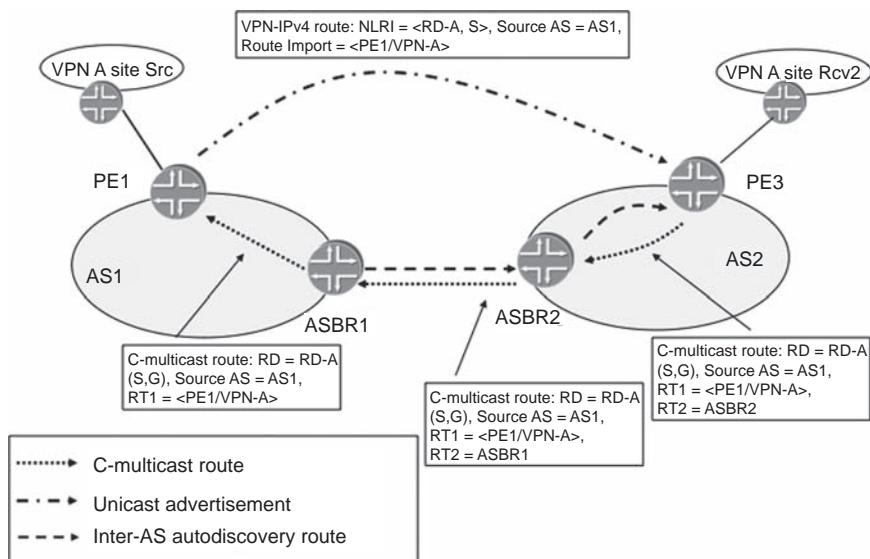


Figure 11.2 Distribution of multicast routing information in an inter-AS scenario

- PE3 places two additional values in the C-multicast route: the value of the source AS and the RD of the inter-AS auto-discovery route (discussed in the previous bullet point). The combination of (RD, Source AS) carried in the C-multicast route allows ASBRs along the path from PE3 to the source AS to correlate this C-multicast route with a particular inter-AS auto-discovery route, because they both carry the same <RD, Source AS> tuple.
- ASBR2 accepts this C-multicast route, because ASBR2's route import policy is such that it accepts all the routes carrying the locally generated C-multicast ASBR Import RT.
- ASBR2 finds an inter-AS auto-discovery route carrying the same AS and RD as the ones carried in the received (and accepted) C-multicast route. From the found inter-AS auto-discovery route, ASBR2 determines that the next hop towards AS1 is ASBR1. Therefore, ASBR2 replaces the RT of the route that contains ASBR2's address with the RT that contains ASBR1's address and sends this route to ASBR1.
- Once the C-multicast route reaches ASBR1, ASBR1 propagates the route to PE1 using the intra-AS procedures.

Note that the flow of the C-multicast route follows the reverse direction of the flow of the inter-AS auto-discovery route originated by ASBR1 in AS1.

It is interesting to compare this solution with the draft-rosen approach. The use of segmented tunnels, as described in this section, allows different tunneling technologies to be used in different ASs and also reduces the number of tunnels set up, because the inter-AS tunnels are per-ASBR/per-VPN rather than per-PE/per-VPN. In contrast, the draft-rosen solution does not support segmented tunnels and requires extending inter-AS tunnels the same way as intra-AS tunnels are constructed.

To summarize, because BGP has built-in support for distributing routing information across AS boundaries and for constraining the distribution of this information, the intra-AS BGP/MPLS mVPN solution lends itself naturally to extensions for the inter-AS case, without making any assumptions about the tunneling technologies used in each AS. The inter-AS solution requires advertising additional information in the C-multicast and auto-discovery routes, but reuses, with only minor modifications, all the elements of the intra-AS solution.

11.3 SUPPORT OF PIM DM IN BGP/MPLS mVPN

The previous chapter focused entirely on PIM sparse mode (SM), which has become the CE-PE multicast protocol of choice and the de facto standard for multicast routing. PIM SM is designed with the assumption of a sparse distribution of interested receivers and follows a model in which multicast data is forwarded to routers that explicitly express interest in receiving it (by sending PIM join messages). In contrast, PIM dense mode (DM), specified in [RFC3973], assumes a dense distribution of receivers and follows a flood model, in which multicast data is broadcast out all interfaces (except the one over which it was received). Router R not interested in receiving this data, multicasts a prune message to its upstream RPF router U to be explicitly removed from the distribution tree. If at a later point R gets more receivers and wants to rejoin the distribution tree, it unicasts a graft message to U. As a result, U adds R back to the tree and replies (again in unicast) with a graft-ack message. With PIM DM, multicast data always flows on SPTs. There are no RPTs and therefore also no RPs.

PIM DM is used by the autorp RP discovery mechanism, because RP discovery assumes a dense distribution of receivers for the RP discovery information and because this information is fairly low-volume traffic. For these reasons, support of PIM DM in a VPN is important despite the fact that PIM DM is not the protocol of choice for multicast routing. Specification of the procedures to support PIM DM as the CE-PE multicast routing protocol is outside the scope of the mVPN standard [VPN-MCAST]. However, the procedures for unsolicited flooded data specified in [VPN-MCAST] do provide a way to support PIM DM as a CE-PE multicast routing protocol.

Let us take a look at the operation of PIM DM in a VPN. Traffic arriving from the CE is flooded to the ingress PE. What happens next depends on whether this traffic is carried over a selective or an inclusive tunnel. In the case of selective P-tunnels, the implementation is simple. As long as an egress PE router has not received a Prune for a customer (S, G), it must join the S-PMSI for the group and flood the data it receives on that tunnel to its CEs. Once the egress PE receives a Prune for (S, G) from all its CEs, it can drop from the tunnel, but it must keep its S-PMSI auto-discovery route in order to later be able to rejoin the tunnel if it receives a Graft message for (S, G) from one of its CEs.

In the case of inclusive tunnels, a differentiation must be made between whether or not DM traffic can be flooded over the inclusive tunnel all the time. If unconditional flooding is acceptable, for example because the volume of traffic is low, the egress PE always receives the data and can decide whether to forward it to the CE based on whether the PE received any Prune message from the CE. If flooding DM traffic on the inclusive tunnel is not acceptable, support of Prune is not as straightforward. In the BGP/MPLS mVPN implementation, prune information is propagated in the form of a BGP route withdrawal. However, because PIM-DM does not use Join messages (and thus did not send a BGP route advertisement in the first place), support of prune in dense mode makes the code dealing with the join/prune mechanism quite complex. In addition, a full-mesh of tunnels is necessary for an egress router to send the prune packet to the ingress router.² However, because the only disadvantage of not supporting the prune mechanism is unnecessary transmission of data on the P-tunnel, and because of the fact that in a DM deployment this would be very unlikely, it is an acceptable compromise to not support the prune/grant mechanism. When discussing the actual need for support of PIM DM in the next section, we will see that this assumption is reasonable.

11.4 DISCOVERING THE RP – AUTO-RP AND BSR SUPPORT IN BGP/MPLS mVPN

PIM SM in ASM mode relies on the existence of a rendezvous point (RP), a centralized entity that knows about all the active sources, as described in Section 10.5.7. Designated routers connected to active multicast sources register the sources with the RP using PIM register messages. Receivers discover the sources by joining an RP Tree (RPT). Thus, for PIM-SM to work properly, all routers in the domain must have a consistent view of the identity of the RP for each multicast group. The RP can be either

² Assuming that the prunes would be sent over the data tunnels.

manually configured on every router or auto-discovered. This section discusses support of RP discovery in BGP/MPLS mVPNs. Two methods for RP discovery do not involve manual configuration:

1. *Auto-RP – a proprietary protocol.* Auto-RP relies on PIM-DM to distribute information about which RP to use to two well-known group addresses. All routers in the domain join one or both of these dense groups and dynamically learn the address of the RPs. PIM-DM support is required, because the control messages sent to these two well-known groups must be forwarded without relying on RPs. One thing to note is that because every PIM router within the mVPN sites must receive data sent to these well-known groups, support of Prune messages is not needed.
2. *PIM bootstrap (BSR) – a PIMv2 extension defined in [RFC5059].* In a nutshell, a single router is elected as the bootstrap router and the candidate RPs periodically unicast their identity to it. The bootstrap router selects the RP for each group range and sends this information to all the routers in a Bootstrap message (BSM), which is sent to the ALL-PIM-ROUTERS well-known address with a TTL of 1 and is regenerated hop by hop. Just as with Auto-RP, because every PIM router within an MVPN sites must receive BSM, support of Prune/Graft messages is not needed.³

To support the auto-RP and BSR mechanisms used within an MVPN, one option is to carry this information, using appropriate new extensions, in BGP, similar to how C-multicast routes are carried in BGP. An alternative option is to transparently transmit the required messages in the forwarding plane. Let us see how this is done:

1. For auto-RP, the only requirement for transmitting messages in the forwarding plane is support of PIM-DM, as described in Section 11.3. You may recall that PIM-DM in mVPNs does not support Prune messages, but this is not a problem because for auto-RP they are also not needed.
2. For BSR, the BSM message is transmitted from the PE connected to the VPN site containing the BSR router to all other PEs. The egress PE routers generate the BSM message for their VPN sites. The only change required is for the PEs to accept incoming traffic destined from the core sent to the ALL-PIM-ROUTERS address and to forward BSM traffic coming from the edge and destined to that address, regardless of whether an explicit join for the group had been done.

Having explained the support for PIM-DM and for RP discovery, we conclude the discussion about the multicast part of mVPN and next turn

³ Note that the BSR election also uses the BSM messages and they must be flooded in the same manner as the BSMs that carry the RP mappings. This must happen in order for the candidate RP routers (CRP) to discover the identity of the bootstrap router so they can unicast their CRP messages to it.

our attention to topics related to the VPN portion of mVPNs, the first of which is support of extranets.

11.5 IMPLEMENTING EXTRANETS IN BGP/MPLS mVPN

This section shows how multicast extranet functionality is achieved with BGP/MPLS mVPNs. The definition of a multicast extranet is that multicast receivers in one mVPN can receive traffic from multicast sources located in other multicast VPNs, and likewise, that multicast sources in one mVPN can send traffic to multicast receivers located in other mVPNs.

One application of multicast extranet is IPTV wholesale, allowing video channels originating at a multicast source in one mVPN to be sent to receivers in multiple other mVPNs, where each mVPN belongs to a different IPTV retailer. Another application is real-time financial information feeds, such as stock price information. Each customer subscribing to such services is located in its own mVPN and receives the information feeds from multicast sources in the other mVPNs that belong to the various information providers.

Figure 11.3 shows an example extranet scenario. Receivers in the grey mVPN are allowed to receive traffic from some or all of the sources located in the black mVPN (for example, source S1 attached to PE2) and the white mVPN (for example, source S2 attached to PE3). However, traffic is not allowed to pass between the black and white mVPNs. In situations in which an egress PE has sites with receivers that belong to different mVPNs, but wants to receive the same multicast flow (originated by a source connected

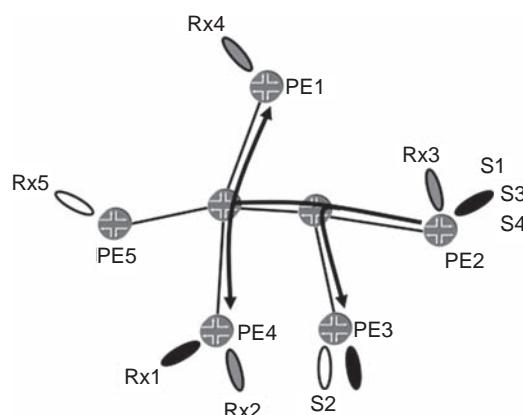


Figure 11.3 Extranet scenario showing the inclusive tree built by PE2 to deliver traffic from sources in black VRF to receivers in the black and grey VRFs

to another PE), it is desirable that the PE attached to the source should have to send a given multicast packet only once to that egress PE. For example, suppose that both Rx1 in the black VRF and Rx2 in the grey VRF attached to PE4 want the same multicast flow from source S1, which is attached to PE2. Ideally, PE2 should send each packet only once, and the packet should then be replicated at PE4 to Rx1 and Rx2. If a receiver Rx3 in the grey site attached to PE2 needs a multicast flow from source S1 in the black site attached to the same PE, the PE needs to be able to pass the traffic from one VRF to the other.

With BGP/MPLS mVPN, the same basic technique of importing and exporting appropriate route targets is used as in unicast VPN extranets [MVPN-EXTRA]. Referring again to Figure 11.3, let us see how multicast traffic from sources in the black and white VRFs is delivered to receivers in the grey VRFs.

- First, the grey VRFs must have (unicast) routes to sources that are located in the black and white VRFs. If PIM-ASM is being used in the C-domain, the gray VRFs must also have routes to the RPs.
- Second, the grey VRFs need to have BGP auto-discovery routes from PE3 for the white mVPN and from PE2 for the black mVPN. This is so that the egress PEs know the type (for example, RSVP-P2MP or LDP-P2MP) and identity of the multicast tunnel on which the ingress PEs send their traffic. These auto-discovery routes can be advertising inclusive trees or selective trees, or there could be a mixture of auto-discovery routes.
- Next, if PIM-ASM is being used in the C-domain, the grey VRFs must have BGP source-active routes pertaining to sources in the white and black VPNs.
- Finally, if RSVP-P2MP is used as the tunnel technology, PEs with black or white source sites attached need to have auto-discovery routes from PEs having grey VPN sites attached. This is because RSVP LSP signaling is initiated from the ingress router, so the ingress needs to know the identity of the egress points. For example, if PE2 is using an RSVP-P2MP inclusive tree, these routes allow it to know that PE1 needs to be added as an egress node to that tree so traffic can be delivered to the grey VPN on PE1.

In some cases, the first three items above are achieved by simply having policies on the PE1, PE2, and PE4 to import routes having the white or black route target into the grey VRFs. Similarly, the last item on the list can be achieved by having one policy on PE2 to import routes having the grey route target into the black VRF and a second policy on PE3 to import routes having the grey route target into the white VRF. In such cases, no additional route targets need to be defined to create the extranet functionality.

Such policies are sufficient for multicast distribution by selective trees. For example, suppose PE3 uses a selective tree to send traffic from S2, located in the white VRF, to group G2. If the interested receivers at a certain point in time are Rx5 in the white VRF on PE5 and Rx3 in the grey VRF on PE2, the corresponding selective tree has PE2 and PE5 as its leaves.

If inclusive trees are being used, there are various ways in which traffic can be mapped to them. Perhaps the most straightforward case is for a PE to use one inclusive tree to deliver traffic from all sources in a given attached VRF to all other PEs having that VRF or associated extranet VRFs attached. In this case, the route-target import policies described above achieve the required functionality. This mode of operation is illustrated in Figure 11.3. PE2 uses the same inclusive tree to deliver multicast traffic originating from its black VRF to receivers located in black VRFs and receivers located in grey VRFs. As can be seen, PE1 is an egress point on the tree because it has a grey site attached, PE3 is an egress point on the tree because it has a black site attached, and PE4 is an egress point on the tree because it has both a black and a grey site attached. Note that if both Rx1 in the black VRF and Rx2 in the grey VRF need to receive the same multicast flow arriving on that tree, PE4 performs the replication locally, thus avoiding the need to send the same multicast packet twice to PE4.

In situations in which some multicast sources in an mVPN are only ever required to send to receivers belonging to that same mVPN, but other multicast sources in that mVPN also send traffic to receivers in other mVPNs, an alternative inclusive tree mapping can be used, if desired. Suppose that multicast traffic from sources S3 and S4 in the black VRF attached to PE2 is needed only by receivers in the black mVPN, but multicast traffic from source S1 could be required by receivers in the black or grey mVPN. In such a situation, PE2 could use one inclusive tree (T-A) for traffic from S3 and S4 and another inclusive tree (T-B) for traffic from S1. Tree T-A would have PE3 and PE4 as its leaves, because these PEs have black sites attached. Tree T-B would have PE1, PE3 and PE4 as its leaves, because these PEs have either a grey site or a black site, or both, attached. This scheme ensures that traffic from S3 and S4 is not sent unnecessarily to PE1, which is not attached to a black VRF and so never needs to receive traffic from S3 and S4.

To achieve this scenario, PE2 originates two I-PMSI auto-discovery routes pertaining to its grey VRF. One of these advertises tree T-A and has route target RT-A attached. The other auto-discovery route advertises tree T-B and has route targets RT-A and RT-B attached. Note that the two auto-discovery routes must have different route distinguishers (RD-A and RD-B, respectively) for them to be regarded as two distinct routes by BGP. In general, PE2 advertises unicast routes with RT-A, including those encompassing sources S3 and S4. However, the unicast route to

S1 is advertised with both RT-A and RT-B attached. The unicast routes can have either RD-A or RD-B as the route distinguisher. The PEs in the network import routes that have RT-A attached into their black VRFs. As a result, PE3 and PE4 import both I-PMSI auto-discovery routes from PE1 into their black VRFs, along with all the unicast routes. However, the PEs import only those routes that have RT-B attached to their grey VRFs. As a consequence, the unicast route to S1 is imported into the grey VRFs, but not the unicast routes to S3 and S4. Also, the I-PMSI auto-discovery route corresponding to T-B is imported into the grey VRF, but not the one corresponding to T-A. As a result, the desired behavior of having the grey VRFs not receive traffic from S3 and S4 is achieved.

In some scenarios, fine-grained control may be needed over which multicast flows originating in another mVPN a given receiver is allowed to receive. A receiver may be allowed to receive only certain multicast groups from a particular source. For example, Rx4 in the grey VRF may be permitted to receive only groups G1, G4, and G6 from source S1 in the black VRF. An efficient way to enforce such restrictions is to configure PIM join filters in the VRF attached to the receiver to discard PIM joins corresponding to a disallowed source or group or source/group combination. This, in turn, prevents the generation of any corresponding BGP C-multicast routes and so avoids creating any unnecessary multicast state in the mVPN.

11.6 TRANSITION FROM DRAFT-ROSEN TO BGP/MPLS mVPNs

This section examines how to migrate from a draft-rosen mVPN deployment to a BGP/MPLS mVPN. Because a given mVPN can contain a large number of sites, it may not be practical for the operator to migrate all the sites to BGP/MPLS mVPN at the same time. Ideally, a method is needed that allows an mVPN to be migrated gradually (potentially over the course of several days or weeks) with minimum traffic disruption. Also, the transition scheme must take into account the fact that multicast flows continuously come and go depending on the join activity in the C-domain, so it is not acceptable to have 'black-out periods' during which new join requests arriving from the C-domain are ignored.

Because one of the main attractions of BGP/MPLS mVPN is its ability to use P2MP LSPs as the provider tunnels, most migrations are done directly from draft-Rosen to BGP/MPLS mVPN with BGP and P2MP LSP data planes. However, in some networks, it may not be possible to migrate immediately to a P2MP LSP data plane, because some of the core routers may not support P2MP LSPs. In such cases, an interim step can be a migration to BGP/MPLS mVPN with an mGRE data plane so at least

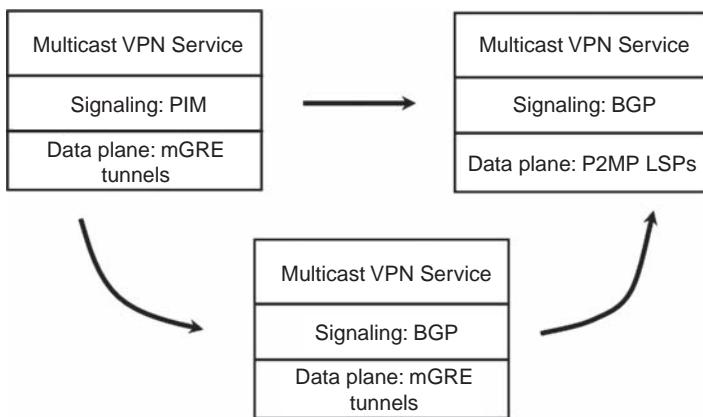


Figure 11.4 Draft-rosen mVPN to BGP/MPLS mVPN migration options

the operator can benefit from the BGP/MPLS mVPN BGP control plane. Migration to a P2MP LSP data plane can be done later, once core routers that support P2MP LSPs have been deployed. These migration options are illustrated in Figure 11.4.

The following principles are important for a scheme that allows gradual migration of an mVPN:

1. An ingress PE should send a given multicast packet onto either a draft-rosen mGRE tunnel or a BGP/MPLS mVPN provider tunnel, but not both.
2. During the migration process, a PE must be able to receive traffic for a particular VPN on draft-rosen mGRE tunnels from some PEs and on BGP/MPLS mVPN provider tunnels from other PEs, depending on how far the migration has progressed.

Let us look at how to migrate the grey mVPN shown in Figure 11.5.

Initially, the mVPN uses the draft-rosen control and data plane throughout. The first step in the migration is to activate the BGP/MPLS mVPN BGP control plane while keeping the draft-rosen control plane active. The BGP control plane requires the BGP MCAST-VPN address family. If this address family is not already present on the BGP sessions (for example, because the grey mVPN is the first mVPN to be migrated), it is added at this stage. Note that adding an address family causes a BGP session to reset, a fact that needs to be taken into account if the MCAST-VPN address family is being added to existing BGP sessions that are carrying other address families. In deployments in which route reflectors are used, to achieve resilience a PE typically has two BGP sessions, each to a different route reflector. In this situation, traffic loss can be avoided by adding the MCAST-VPN address

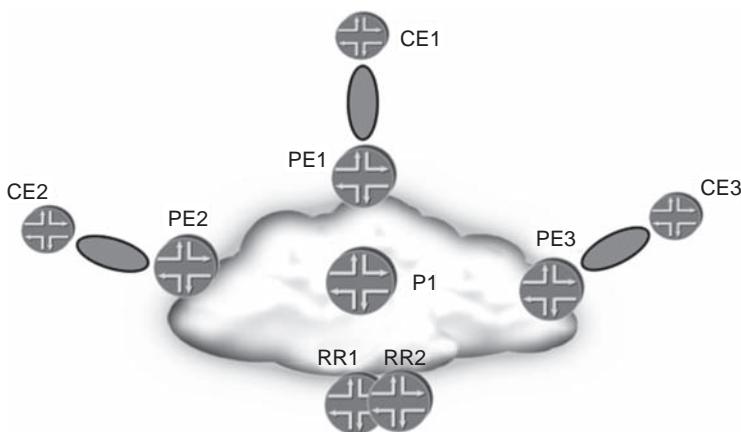


Figure 11.5 Network topology for draft-rosen mVPN to BGP/MPLS mVPN migration discussion

family on a PE to one BGP session at a time. For example, suppose PE2 has a BGP session with route reflector RR1 and another BGP session with RR2. The MCAST-VPN address family can be configured on the session between PE2 and RR1 first. Once that session has re-established and the exchange of routes is complete, the MCAST-VPN address family can be configured on the session between PE2 and RR2. Once all the BGP sessions in the network have the MCAST-VPN address family, the next step is to activate the BGP/MPLS mVPN control plane in the grey VRF on each PE, but without deactivating the draft-rosen control plane.

After these two steps, each PE is running the draft-rosen and BGP/MPLS mVPN control planes simultaneously for the grey VRF. For example, suppose PE3 receives PIM joins for a multicast group G1 from CE3. This request triggers PE3 to send both a PIM join (in the context of the draft-rosen control plane) and a BGP C-Multicast route (in the context of the BGP/MPLS mVPN control plane). If PE1 is attached to a source site for group G1, PE1 forwards the traffic onto its draft-rosen mGRE tunnel, because at this stage in the migration it is still using the draft-rosen mGRE data plane.

The next step is to activate the BGP/MPLS mVPN provider tunnel in the grey VRF on each PE attached to a sender site. Depending on which of the migration paths shown in Figure 11.5 is being used, this can be an mGRE tunnel or a P2MP LSP. Either way, applying this configuration to a PE triggers it to send a BGP auto-discovery route to advertise the identity of the provider tunnel that it will use to send traffic. After this step, the PE uses the BGP/MPLS mVPN provider tunnel in preference to the draft-rosen mGRE tunnel to forward multicast traffic to other

PEs.⁴ For example, if multicast traffic to group G1 is still required by the receiver in the grey site attached to PE3, router PE1 forwards that flow onto its BGP/MPLS mVPN provider tunnel because it has an active C-Multicast route that originated from PE3.

The final stage of the process is to remove the draft-rosen configuration from each of the grey VRFs, because now all the grey VRFs are using the BGP/MPLS mVPN control and data plane.

If an mGRE tunnel is being used as the BGP/MPLS mVPN data plane, migration to a P2MP data plane can be done later by removing the mGRE provider-tunnel configuration from each of the grey VRFs and replacing it with P2MP LSP provider-tunnel configuration.

If an implementation is designed correctly, the traffic impact of such a migration has been shown to be very low, with some modest (sub-second) packet loss occurring during the setup of the P2MP LSP provider tunnel [MPLSWC2010].

11.7 SCALABILITY DISCUSSION

A comprehensive comparison of PIM/GRE and BGP/MPLS mVPN can be found in [mVPN-STD-OPT]. In this section, we focus on comparing the scaling properties of the PIM/GRE mVPN and BGP/MPLS mVPN control planes with respect to how they handle requests for multicast flows from the customer domain. We consider a scenario in which the PE-CE protocol is PIM-SSM, and a default MDT (in the PIM/GRE mVPN case) and inclusive trees (in the BGP/MPLS mVPN case) are used for forwarding.

Suppose that a network contains a total of N PEs that are configured for mVPN service. M of these PEs are attached to sites of the Blue mVPN. One of these PEs, PE1, is attached to multicast source S in a site belonging to the Blue mVPN. Of all PEs in the Blue mVPN, X of them have a receiver in their respective local VRF site that wishes to receive the flow (S, G). In the following paragraphs, we examine the effect on the mVPN control plane created by this scenario.

11.7.1 PIM/GRE mVPN control plane scaling

In the PIM/GRE mVPN case, the arrival of a join request for (S, G) from a CE site triggers the attached PE to send a PIM join for (S, G) onto the

⁴Note that if an mGRE tunnel is being used as the BGP/MPLS mVPN provider tunnel, it can be configured to use the same group address as used for the draft-rosen mGRE tunnel. In that case, physically it is the same tunnel throughout the migration, but control of it passes from the draft-rosen control plane to the BGP/MPLS mVPNcontrol plane at this stage.

emulated LAN for that mVPN. The PIM join is sent over the mGRE default MDT for that mVPN and hence is received by all the other ($M-1$) PEs in the mVPN. Because each of the X PEs attached to receiver sites for (S, G) behaves in this way, each of the PEs in the Blue mVPN, whether attached to the source site or not, receives a total of X instances of the PIM join for (S, G) . As a consequence, the $(M-1)$ PEs not attached to the source site incur some processing overhead associated with the $(X-1)$ PIM joins that each receives. Each one of these PEs needs to inspect each PIM join packet that it receives to determine that it is not the upstream neighbor named in the join packet and whether any changes to its state machines are triggered by the packet.

11.7.2 BGP/MPLS mVPN control plane scaling

In the case of the BGP control plane, the arrival of a PIM join from a VRF site triggers the creation of a BGP C-multicast route. There are four main ways of setting up the BGP control plane, depending on whether route reflectors are deployed and whether route target filtering (described in Chapter 8) is in use. Let us look at each of these cases.

11.7.2.1 Full mesh of BGP sessions, no route target filtering

When there is a full mesh of BGP sessions between the PE routers and route target filtering is not activated, each PE with a receiver in the attached VPN site sends a copy of the BGP C-multicast route to each of its $(N-1)$ PE peers. This means that each PE, whether attached to the source site or not, and whether a member of the Blue mVPN or not, receives X copies of the C-Multicast route, one from each PE attached to a receiver site.

As a result, PE1 receives X C-Multicast routes, and the other $(N-1)$ PEs incur some processing overhead associated with the received X or $(X-1)$ unwanted C-multicast routes that arrive at each, to determine that they do not have a match for the attached route target.

11.7.2.2 Full mesh of BGP sessions, with route target filtering

As described in Chapter 10, a C-multicast route does *not* contain the ‘regular’ route target associated with the all the sites of that VPN. Instead, it contains the C-multicast Import route target that is unique to the PE attached to the source site of the requested multicast flow. When route target filtering is used, each of the X PEs that is attached to receiver sites for (S, G) sends only the C-multicast route to PE1, because only that PE has expressed an interest in receiving routes having that particular community. As a result, although PE1 still receives a total of X instances of the

C-multicast route, as in the previous case, none of the other PEs receive them, thus reducing the processing load on these PEs.

11.7.2.3 BGP route reflectors, no route target filtering

Suppose one or more route reflection clusters are deployed, each with two redundant route reflectors. Each PE is a route reflector client of one cluster and hence has a BGP peering with two route reflectors. Route reflectors have a very useful property in the context of C-multicast routes. If a route reflector receives the same C-multicast route from multiple route reflector clients, it propagates only one copy to each of its peers, because the NLRI of all the C-multicast routes is identical. As a result, each PE, whether a member of the Blue mVPN or not, receives two copies of the C-multicast route, one from each of its two route reflector peers, rather than the X copies in the first case above.

11.7.2.4 BGP route reflectors, with route target filtering

Let us suppose the same arrangement of route reflectors from the previous case is used, except that route target filtering is configured. This means that the C-multicast route is delivered only to PE1, because only that PE has expressed interest in receiving routes having that particular community. In this way, PE1 receives two copies of the C-multicast route, one from each of its two route reflector peers, and none of the other PEs receives any copies.

Table 11.1 summarizes the four scenarios discussed above. As can be seen, the PIM case and the BGP full-mesh case without route target filtering are similar in the sense that PEs other than PE1 receive X instances of the PIM join or BGP update, respectively. In the BGP case, receipt of the extraneous joins can be avoided by using route target filtering; however, in the PIM case this cannot be avoided. Furthermore, combining route reflection and route target filtering in the BGP case ensures that the number of updates received by PE1 is reduced from X to the number of route reflectors of which PE1 is a client (typically two).

It is worth making a few notes about Table 11.1. First, the table considers only the control packets that are initially sent by the X PEs that wish to receive the (S, G) flow. However, in the PIM case, the join state needs to be refreshed periodically. PIM join suppression can be used to ensure that not all X PEs attached to receiver sites need to send refreshes of the join state. (Note, however, that join suppression does not apply to the *initial* PIM join sent by each of the X PEs requiring the flow (S, G) ; it applies only to subsequent refreshes of the join state.) As a result, the number of join messages received by PE1 and the other $(M-1)$ PEs is not as high during a refresh cycle as during the initial creation of the join state. Even so, in

Table 11.1 Scaling comparison for customer multicast routing

Control Plane	Egress PE: route generation	Ingress PE, PE1	PE, (except PE1), with Blue mVPN site attached	PE, (except PE1), no Blue mVPN site attached
PIM	Each egress PE sends 1 PIM join. X total joins sent	Receives X PIM joins.	Each receives X PIM joins	None
BGP full-mesh, no route target filtering	N updates sent, 1 to each of the other PEs $N * X$ sent in total	Receives X BGP updates	Each receives X BGP updates	Each receives X BGP updates
BGP full-mesh, with route target filtering	1 update sent by each egress PE. X sent in total	Receives X BGP updates	None	None
BGP with route reflectors, no route target filtering	2 updates sent by each egress PE. 2X sent in total	Receives 2 BGP updates	Each receives 2 BGP updates	Each receives 2 BGP updates
BGP with route reflectors and route target filtering	Two updates sent by each egress PE, so 2X sent in total	Receives 2 BGP updates	None	None

the BGP case, no refreshes are required at all because BGP is a stateful protocol. Hence, BGP does not incur the ongoing overhead incurred by PIM to maintain the flow state.

11.8 ACHIEVING MULTICAST HIGH AVAILABILITY WITH BGP/MPLS mVPN

The topic of multicast high availability has received a large amount of attention recently because of the mission-critical nature of many multicast applications, for example, Professional Broadcast TV, IPTV, and real-time financial data feeds. In the MPLS Multicast chapter (Chapter 6), we

discussed how MPLS fast-reroute (Section 6.6) can be used with point-to-multipoint (P2MP) LSPs to give rapid recovery from failures in the core of the network. In Section 6.7, we showed how to achieve ingress redundancy for standalone P2MP LSPs, which provides protection in the case of an ingress failure. Two schemes were presented, ‘Live-Live,’ in which two identical streams are sent and one is discarded at the receiver, and ‘Live-Standby,’ in which the traffic is sent by just one of two available sources. At the time of this current writing, the predominant way in which network operators use P2MP LSPs for multicast delivery is in conjunction with BGP/MPLS mVPNs, rather than using standalone P2MP LSPs. For this reason, the following sections explain how the BGP/MPLS mVPN technology can be leveraged to implement the ‘Live-Live’ and ‘Live-Standby’ schemes.

11.8.1 Live-Standby multicast delivery using BGP/MPLS mVPN

Let us examine how the Live-Standby scheme can be built using BGP/MPLS mVPN. Let us refer to the network topology in Figure 11.6.

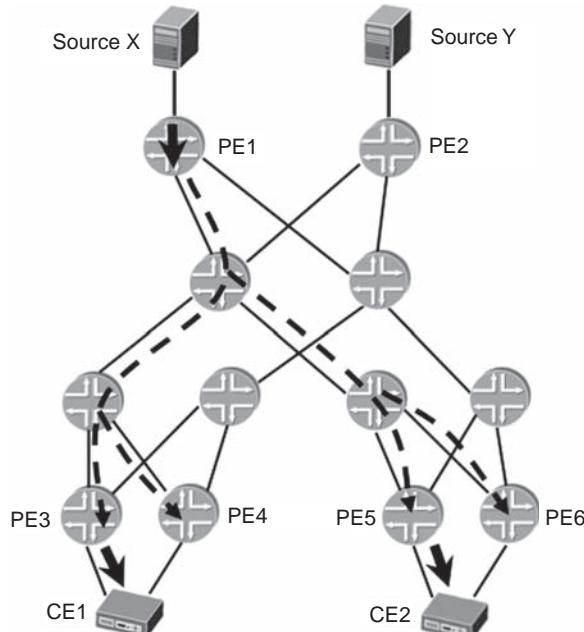


Figure 11.6 Live-Standby scheme for multicast high availability

Sources X and Y are deliberately configured with an identical IP address, S, and send identical multicast data. Sources X and Y and the receiving devices, CE1 and CE2, are all in the same BGP/MPLS mVPN. On PE1 and PE2, the links to X and Y are in the VRF of that mVPN. Similarly, on the egress PEs the links to the CEs are in the VRFs of that mVPN. Assume that inclusive tunnels are used to deliver multicast traffic of that mVPN. As discussed in Chapter 10, an inclusive tunnel delivers a multicast flow to all the PEs in an mVPN if at least one PE has requested that flow. Suppose that OSPF is the unicast routing protocol between the PE routers and the CE routers and that PIM-SSM is the multicast protocol between the PE routers and the CE routers. Also suppose that the IGP metric configured on CE1's uplink to PE3 is lower than the one on its uplink to PE4. Similarly, the IGP metric configured on CE2's uplink to PE5 is lower than the one on its uplink to PE6. This means that under normal circumstances, when CE1 sends a PIM join for (S, G) , it sends it to PE3 because that is the best path towards S.

By definition, in the Live-Standby scheme, only one ingress PE should send a particular multicast flow into the network at any time. BGP/MPLS mVPN caters to this scheme very well because it contains built-in procedures for selecting one particular ingress PE for a given multicast flow. This procedure is known as upstream multicast hop selection. When PE3 receives a PIM join for the flow (S, G) from CE1, it identifies which ingress PEs lie on the path towards the source address, S. PE3 in the VRF has routes to S from both PE1 and PE2, and so needs to select one of those. The default mode of operation is to select the PE configured with the highest IP address. Suppose that PE1 has a higher IP address than PE2. When PE3 generates the BGP C-Multicast route to request the flow (S, G) , it attaches the VRF Route Import extended community that is unique to PE1 so that only PE1 accepts that route. When PE1 accepts the route, it triggers PE1 to send the (S, G) flow onto its inclusive tunnel, which means that the traffic is received by all the egress PEs. PE3, in turn, forwards the traffic to CE1.⁵

Let us now examine some failure scenarios to see how the network restores the traffic. Suppose that ingress router PE1 fails or that the link between it and X fails. When ingress router PE1 fails, PE3 detects a failure via the IGP and so deletes the route to S via PE1 from its routing table. When the link between PE1 and X fails, PE1 sends a BGP update to the route reflectors withdrawing the prefix containing S. PE3 receives that update from the route reflectors. Either way, PE3 now knows that the only

⁵ Note that other means of selecting the ingress PE instead of highest IP address are possible, for example according to which path to S is considered best according to normal unicast selection rules (e.g. BGP local preference). This method can be used as long as all egress PEs are consistent about which ingress PE is chosen for particular flow, as this is a requirement of the live-standby scheme.

path to S is via PE2. This change of path triggers two events. First, PE3 sends the C-Multicast route for the (S, G) flow with the VRF Route Import extended community unique to PE2 attached. Then PE2 sends the (S, G) flow onto its inclusive tree so that it is received by the egress PEs and so traffic can flow again to CE1 via PE3.

In this scenario, the following factors contribute to determining how long the traffic flow is interrupted:

1. Time required to detect the initial failure (for example, for PE1 to detect that the link to X has gone down).
2. Time required for PE3 to learn about this failure (for example, the time taken for PE1 to send the BGP update via the route reflectors and for PE3 to receive it).
3. The time required for PE3 to generate and propagate the BGP C-Multicast route via the route reflectors to PE2, and the time taken for PE2 to install the required forwarding state.

Tests have shown that interruption times can be relatively low (of order hundreds of milliseconds) even if several hundred multicast flows are affected by the failure [MPLSWC2009]. Even so, it is possible to reduce the interruption time even further. An optimisation proposed in the IETF [Morin-Fast] removes the last item in the list above as a contributing factor to the interruption time.⁶ It works as follows. Under normal conditions, that is, when both paths to S via PE1 and PE2 are intact, PE3 sends a C-Multicast route with the VRF Route Import extended community unique to PE1 attached in the usual way. However, in addition, it also sends a C-Multicast route with the VRF Route Import extended community unique to PE2 attached. This route, known as a standby BGP C-multicast route, has an additional BGP community attached to signify that it is the standby route. This procedure lets PE2 know in advance that it is expected to forward traffic for the flow (S, G) if S becomes unreachable via PE1. If PE2 learns, via the routing protocols, that S is unreachable via PE1, it immediately starts forwarding the flow (S, G) originated by Y to the egress PEs, rather than having to be requested by an egress PE to do so. In this scenario, the contributing factors to the traffic interruption time now become the following:

1. Time required to detect the initial failure (for example, for PE1 to detect that the link to X has gone down).

⁶Note that if the source were not directly connected to PE2, for example it is behind a CE router attached to PE2, there would be an additional contribution to the interruption time associated with propagating multicast state from PE2 towards the source (and the time for the multicast traffic to start flowing to PE2). [Morin-Fast] also describes how to eliminate that contribution to the interruption time.

2. Time required for PE2 to learn of this failure (for example, the time taken for the BGP update to be propagated by PE1 via the route reflectors and to be received by PE2).

Let us now look at a different failure scenario, the failure of PE3, or the link between PE3 and CE1. In this scenario, CE1 detects that the link has gone down, and hence CE1 knows that S is no longer reachable via its uplink to PE3. CE1 performs an OSPF SPF run and determines that the best path to S is now via the uplink to PE4. CE1 then sends the PIM join for (S, G) to PE4, which triggers PE4 to generate a BGP C-Multicast route with the VRF Route Import extended community unique to PE1. This, in turn, triggers PE1 to restart sending traffic from the (S, G) flow to the egress PEs. It may be the case that the flow (S, G) is already being forwarded by PE1 on the inclusive tree because that flow has already been requested previously by other egress PEs (for example, PE5). In this situation, as soon as PE4 receives the PIM join from CE1, it can immediately forward the (S, G) flow to CE1. The contributing factors to the interruption time to the traffic are now the following:

1. Time required by CE1 to detect that the link to PE3 has gone down, or that PE3 itself has gone down.
2. Time required for CE1 to perform its OSPF SPF run and to send the PIM join to PE4.
3. Time required for PE4 to process the PIM join, and if the traffic is not already arriving at PE4, the time taken for PE4 to propagate the BGP C-Multicast route via the route reflectors and for PE1 to receive that route.

11.8.2 Live-Live multicast delivery using BGP/MPLS mVPN

Let us now examine how to achieve Live-Live multicast delivery using BGP/MPLS mVPN. As mentioned earlier, in the Live-Live scenario, two ingress PEs send the same flow across diverse paths to each receiving location. At each receiving location, each of the two flows is delivered to a different PE. For example, in Figure 11.7, the flow entering the network at PE1 is delivered to PE3 and PE5, and the flow entering the network at PE2 is delivered to PE4 and PE6. Suppose a failure in the network occurs such that PE3 is not reachable from PE1. In this situation PE2 should *not* be used as the entry point to deliver traffic to PE3, because the feed entering the network at PE2 is already being delivered to PE4. A convenient way to achieve this separation using BGP/MPLS mVPN is to configure two separate mVPNs. One mVPN is configured on ingress router PE1 (with

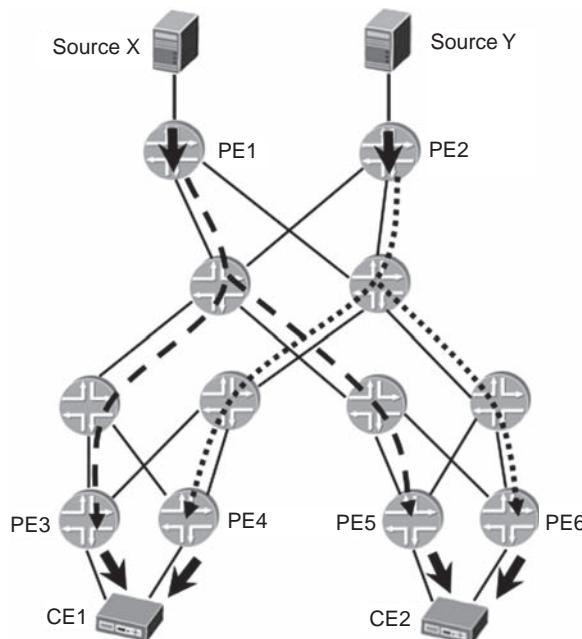


Figure 11.7 Live-Live scheme for multicast resilience

the link to X in the VRF) and on egress routers PE3 and PE5 (with the links to CE1 and CE2, respectively, in the VRF). Similarly, the second mVPN is configured on ingress router PE2 (with the link to Y in the VRF) and on egress routers PE4 and PE6 (with the links to CE1 and CE2, respectively, in the VRF). Thus, from the point of view of PE3, only PE1 lies on the path to source address S, and from the point of view of PE4, only PE2 lies on the path to source address S. Traffic engineering techniques ensure that the P2MP inclusive tree rooted at PE1 and the P2MP inclusive tree rooted at PE2 follow diverse paths through the network.

Typically, the requirement in the Live-Live scenario is to deliver the multicast feeds on a semi-permanent basis. Also, the receiving equipment at the application level is often not capable of participating in dynamic multicast protocols. In such situations, the most convenient way to attract the traffic towards the CE device is to configure a static IGMP entry for (S, G) on the PE interfaces facing the CE. This technique can also be used if the CE device is a multicast router. On PE3, the static IGMP entry triggers a BGP C-Multicast route requesting the (S, G) flow from PE1. On PE4, the static IGMP entry triggers a BGP C-Multicast route requesting the (S, G) flow from PE2. In this way, two duplicate flows arrive at the CE device, one copy via PE3 and the other via PE4. As explained earlier, if the CE

device is performing the flow selection at the application layer, seamless delivery of traffic to the receiving application can be achieved if network failure disrupts one of the two duplicate flows.

Let us examine what happens if the CE device is a multicast-capable router rather than a device that performs selection at the application layer. Suppose that the IGP metrics are such that the best path towards the source address S, from the point of view of CE1, is via PE3. In this case, the version of the flow arriving from PE3 is accepted and passed downstream, while the version of the flow arriving from PE4 is discarded because it is not arriving on the RPF interface. Let us now look at some failure scenarios. Suppose that ingress router PE1 fails or that the link between PE1 and X fails. If ingress router PE1 fails, PE3 detects a failure via the IGP and so deletes the route to S via PE3 from its routing table. If the link between PE1 and X fails, PE1 sends a BGP update to the route reflectors withdrawing its route to S. PE3 receives that update from the route reflectors. Either way, PE3 no longer has the route to source S in the VRF and it informs CE1 via OSPF, which triggers CE1 to perform an OSPF SPF calculation. As a result, the RPF interface to source S is now the uplink to PE4. Traffic is already arriving on that link from PE4 and is passed downstream by CE1. In this scenario, the interruption time to the traffic consists of the following contributing factors:

1. Time required to detect the initial failure (for example, for PE1 to detect that the link to X has gone down).
2. Time required for PE3 to learn of this failure (for example, the time taken for PE1 to send the BGP update via the route reflectors and for PE3 to receive it).
3. Time required for PE3 to generate and propagate the OSPF message to the CE1 so that CE1 learns that S is no longer reachable via PE3.
4. Time required for CE1 to perform its OSPF SPF calculation and change its RPF interface to the uplink to PE4.

Let us now look at a different failure scenario, the failure of PE3, or the link between PE3 and CE1. Either way, CE1 detects that the link has gone down and thus knows that S is no longer reachable via its uplink to PE3. CE1 performs an OSPF SPF run and determines that the path to S is via the uplink to PE4. As a result, the RPF interface to source S is now the uplink to PE4. Because traffic is already arriving on that link from PE4, the traffic can be immediately passed downstream by CE1. In this scenario, the interruption time to the traffic consists of the following contributing factors:

1. Time required by CE1 to detect that the link to PE3 has gone down, or that PE3 itself has gone down.
2. Time required for CE1 to perform its OSPF SPF calculation and change its RPF interface to the uplink to PE4.

11.8.3 Comparison of the Live-Live and Live-Standby multicast high-availability schemes

The Live-Live scheme with application-layer selection can achieve zero interruption to the flow of data to the receiving application if the client equipment is designed correctly. Also, this scheme caters to a wider range of failure modes, because it can detect problems that are not detected by network-layer selection. For example, in video-distribution scenarios, one of the feeds may contain corrupted or absent video data in the packet payloads because of a problem in the source. This issue would not be detected at the network layer. For these reasons, the Live-Live scheme with application-layer selection is used for the most mission-critical applications.

When the Live-Live scheme is used with network layer selection, the interruption time is dominated by convergence operations on the egress CE, because the alternate feed is already arriving there. For example, the IPG convergence time is a contributing factor. Many implementations allow the SPF hold-down timer to be tuned to very low values to reduce this contributing factor.

With the Live-Standby scheme, the convergence times are typically longer than in the Live-Live case, because extra protocol steps are required to draw the traffic to the receiver via an alternate path following a failure in the network. Nevertheless, quite respectable convergence times, on the order of hundreds of milliseconds, can still be achieved [MPLSWC2009]. For some cases, such as mainstream enterprise multicast applications being transported over a service provider's mVPN service, these convergence times are perfectly acceptable and the use of Live-Live is regarded as overkill.

11.9 INTERNET MULTICAST SERVICE USING THE BGP/MPLS mVPN TECHNOLOGY

The BGP/MPLS mVPN solution was developed to support multicast in VPNs. Interestingly, the same technology can be applied to providing Internet multicast service over a service provider's network, independently of VPNs. The scenario is one in which a service provider wishes to provide Internet multicast support over MPLS LSPs (either P2P or P2MP) in the data plane, while keeping the core free of PIM. The two main reasons for pursuing such an approach, one related to the data plane, the other related to the control plane are:

- *Data plane optimization.* As discussed in Chapter 6 (MPLS multicast), forwarding multicast traffic over P2MP LSPs can significantly optimize bandwidth usage for the multicast distribution. But even if P2MP LSPs

are not used, if all non-multicast traffic in the core is forwarded over a mesh of P2P or MP2P LSPs, it is beneficial to forward multicast traffic over the same mesh as well, thus eliminating any multicast-related state from the core, be it either IP multicast state or MPLS P2MP multicast state.

- *Control plane simplicity.* As more and more providers move to try to minimize the number of protocols that need to be supported on their core routers, being able to remove PIM from the core of the network is an attractive option. In fact, it may be the only option if the data plane is a mesh of P2P LSPs. In such a setup, even if PIM can treat the LSPs as interfaces, such a deployment would create a very large set of neighbors and would pose scalability concerns.

The solution to providing Internet multicast over an MPLS uses the same building blocks as the BGP/MPLS mVPN technology. The network is partitioned into two domains, the multicast domain (routers IP-R1 through IP-R4) and the MPLS domain, as shown in Figure 11.8. Some interfaces on the routers at the edge of the domain (border routers, or BRs) are multicast enabled and some are MPLS enabled, such as those on routers BR1, BR2, and BR3. The main idea behind the solution is for the border routers to exchange multicast state with each other using BGP and for them to forward the multicast traffic using MPLS. To achieve this, the border routers do the following:

- Discover the other border routers. They do this by exchanging intra-AS auto-discovery routes. The PMSI Tunnel attribute (see Section 10.5.6) can signal whether P2P or P2MP LSPs are used.

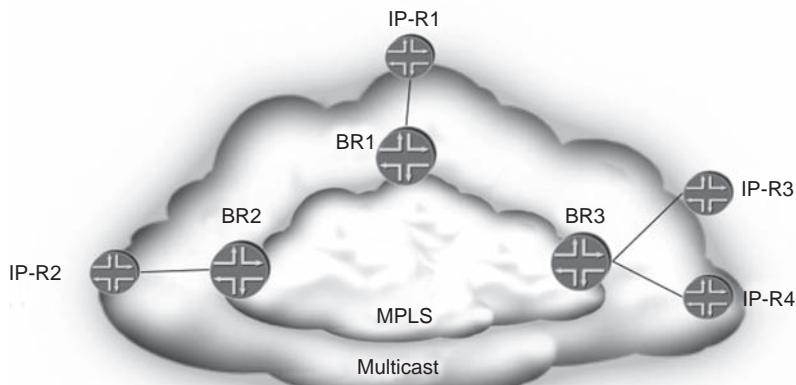


Figure 11.8 Providing Internet multicast over an MPLS core

- Assuming an ASM service model, maintain and distribute information about active sources.⁷ Obtaining the information about active sources is done by either collocating RP with the border router or using MSDP or PIM-Register to communicate this information from the RP to the border router. Border routers distribute this information using BGP Source Active auto-discovery routes and may use BGP Source Tree Join C-multicast routes to inform border routers connected to active sources that receivers exist for (S, G).

Having the border routers do these two tasks sets up the control plane. The flexibility of the mVPN data plane can be leveraged to allow for ingress replication over a mesh of P2P or MP2P LSPs or over a mesh of P2MP LSPs, or over any of the other supported tunneling technologies. Of particular interest is the scenario in which P2P LSPs are used to forward the multicast traffic, because this situation serves as a first step in a transition scenario. When a mesh of P2P LSPs is used, ingress replication must be performed and the relevant border routers to which traffic needs to be forwarded have to be identified. The relevant border routers are identified based on the router address carried in the BGP intra-AS auto-discovery routes, and traffic can thus be replicated at the ingress border router such that it is delivered to all relevant egresses. This description assumes that Internet multicast is carried over an inclusive tunnel. With selective tunnels, leaf auto-discovery routes can be used to discover which border routers have receivers for a particular Internet multicast group.

Thus, delivery of Internet multicast traffic can be achieved over an MPLS core using the same mechanisms used for support of multicast in a VPN. The flexibility of the BGP/MPLS mVPN solution becomes apparent in this application, which has nothing to do with mVPNs at all.

11.10 CONCLUSION

This chapter has explored some advanced topics that arise in the context of supporting multicast over VPNs. The BGP/MPLS mVPN technology supports all multicast features, has excellent control-plane properties because it leverages well-defined VPN scaling techniques, and enables the delivery of advanced services such as Live-Live/Live-Standby redundancy and Internet multicast over an MPLS core.

This chapter concludes the discussion on Layer 3 BGP/MPLS VPNs. In the following two chapters, we discuss Layer 2 VPNs and VPLS, and we see that much of the protocol machinery used for the Layer 3 VPN has been extended to Layer 2 VPNs and VPLS.

⁷ For the SSM service model, there is no need to maintain information about active source.

11.11 REFERENCES

- [MVPN-EXTRA] R. Aggarwal, Y. Rekhter, T. Morin, W. Hendrickx, P. Muley, R. Qiu, *Extranet in BGP Multicast VPN (MVPN)*, draft-raggarwa-l3vpn-bgp-mvpn-extranet-03.txt (work in progress)
- [MPLSWC2009] J. Lucek, A. Stiphout and S. Clarke, ‘Achieving Resilience in Multicast Networks’, Paper D2-06, in MPLS World Congress, Paris, February 2008.
- [Morin-Fast] T. Morin, Y. Rekhter, R. Aggarwal, W. Hendrickx, P. Muley, R. Qiu, ‘Multicast VPN fast upstream failover’, draft-morin-l3vpn-mvpn-fast-failover-04 (work in progress)
- [MPLSWC2010] J. Lucek, *Deploying Next-Generation Multicast VPN*, Paper D2-17, MPLS World Congress, Paris, February 2010
- [mVPN-STD-OPT] T. Morin, B. Niven-Jenkins, Y. Kamite, R. Zhang. N. Leymann and N. Bitar, *Mandatory Features in a Layer 3 Multicast BGP/MPLS VPN Solution*, draft-ietf-l3vpn-mvpn-considerations-06.txt (in the RFC editors’ queue, soon to become RFC)
- [RFC3973] A. Adams, J. Nicholas, W. Siadak, *Protocol Independent Multicast – Dense Mode (PIM-DM):Protocol Specification (Revised)*, RFC3973, January 2005
- [RFC 5059] N. Bhaskar, A. Gall, J. Lingard, S. Venaas, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*, RFC5059, January 2008
- [VPN-MCAST] E. Rosen and R. Aggarwal, *Multicast in MPLS/BGP IP VPNs*, draft-ietf-l3-vpn-2547bis-mcast-09.txt (work in progress)

11.12 STUDY QUESTIONS

1. In Figure 11.1, assume that one more site, site Src2, is attached to PEx in AS-1. Walk through the advertisements sent by ASBR1, once with aggregation (as in the mVPN specification) and once without. Assume mLDP as the transport protocol in AS1. Explain how aggregation helps reduce forwarding state and why this is important for scaling the solution.
2. Describe the sequence of steps required to migrate from a draft-rosen mVPN to a BGP/MPLS mVPN with P2MP LSP provider tunnels.

3. Discuss how route reflectors can reduce the number of C-multicast routes for a particular multicast flow received by an ingress PE.
4. Name some applications of multicast extranets.
5. Why is auto-RP not a good candidate for protocol extensions to support it in an mVPN?
6. Describe the relative merits of the Live-Live and Live-Standby schemes for multicast high availability.

12

Layer 2 Transport over MPLS

12.1 INTRODUCTION

This chapter describes the rapidly growing area of Layer 2 transport over MPLS networks. This is a key component of a multiservice network as it allows service providers to migrate Frame Relay, ATM and leased-line customers to an MPLS network while maintaining similar service characteristics from the customer's point of view. It also enables new Layer 2 service offerings based on Ethernet access. In this chapter, we compare the two main schemes for achieving Layer 2 transport over MPLS, one based on LDP signaling and the other based on BGP signaling. We also discuss Circuit Cross Connect (CCC), which was the precursor to these schemes and is still in use in several service providers' networks.

12.2 THE BUSINESS DRIVERS

Native Layer 2 services have existed for several years, based on Frame Relay or ATM. Often these services are used by an enterprise to build its corporate Layer 2 VPN by interconnecting its LANs over a wide area. Service providers can offer near global reach, either directly or through interconnection agreements with partners. The services are a valuable

source of revenue to service providers, at the time of writing far outstripping revenues from IP services. In these networks, customer sites are interconnected at Layer 2, sometimes in a full mesh but more typically in a hub-and-spoke topology. The role of the service provider is to transport the ATM cells or Frame Relay frames over the wide area, at an agreed bit-rate for each circuit.

As well as being used to carry general LAN interconnection traffic, these services, especially in the ATM case, are sometimes used to carry traffic requiring more stringent SLAs from the network, e.g. with respect to delay variation, such as video traffic or Private Automatic Branch exchange (PABX) interconnections.

In many cases, a service provider can migrate these services to an MPLS network while retaining the same connectivity, as far as the customer is concerned, and maintaining similar service characteristics. In these cases, the presentation to the customer is still over ATM or Frame Relay and a similar service-level agreement (SLA) is offered. For example, in the Frame Relay case, a CIR (committed information rate) is agreed for each circuit and SLA is defined for parameters such as packet loss, latency and delay variation.

Migrating these services to an MPLS network saves the service provider capital and operational expenses compared to running separate networks for Layer 2 connectivity and Layer 3 connectivity. Also one of the schemes discussed later in this chapter greatly reduces the operational burden of provisioning Layer 2 connections within the service provider part of the network, especially in cases where a high degree of meshing is used between customer sites, which leads to a further saving in operational costs.

Another growing application of Layer 2 transport over MPLS is Ethernet services, in which a customer's Ethernet frames are transported between the customer's sites over the service provider's MPLS network. The appeal to the end customer is that Ethernet is the standard Layer 2 protocol used within the enterprise and hence is familiar to the corporate IT staff. Using Ethernet to interconnect their sites over the wide area is a natural extension of the use of Ethernet within their premises. In many cases where customers have been using ATM or Frame Relay services for LAN interconnection, there is no fundamental reason why ATM or Frame Relay should be used as the interconnectivity method. Ethernet has the attraction that it is more flexible in terms of access rates – the service provider can offer, for example, a 100 Mbps Ethernet tail that is rate-limited to the level paid for by the customer. This allows for smoother upgrades in access speed than having, for example, to change from an E1/T1 access circuit to an E3/T3 access circuit. These factors, along with the fact that Ethernet-based equipment tends to be less expensive than ATM or Frame Relay equipment, by virtue of volume, mean that in some cases a customer might migrate from

a native ATM or Frame Relay based service to an Ethernet service in order to reduce costs. Similarly, enterprises using leased-line services for LAN interconnection can reduce costs by switching to Ethernet services. Some of the services being replaced have stringent SLAs, for example ATM CBR services are uncontended, i.e. the bandwidth is guaranteed end to end, and accompanied by tight SLAs on delay variation. MPLS networks are also capable of providing such a service through a combination of control plane techniques such as DiffServ Aware TE and packet scheduling mechanisms that prioritize the CBR traffic appropriately at each hop in the network.

Besides point-to-point Ethernet services, many service providers offer multipoint Ethernet services, known as the Virtual Private LAN Service (VPLS). VPLS is the subject of the next chapter of this book, while this chapter discusses point-to-point services.

Whether a customer can migrate to Ethernet depends on whether local Ethernet access is available, bearing in mind that ATM and in particular Frame Relay (FR) access networks have much higher geographical penetration in many territories, reaching the smaller cities, whereas Ethernet may only be available in larger cities. The incumbent service providers, who tend to own the large ATM or Frame Relay networks, may choose to retain the access part of those networks but migrate the core to an MPLS network, as illustrated in Figure 12.1.

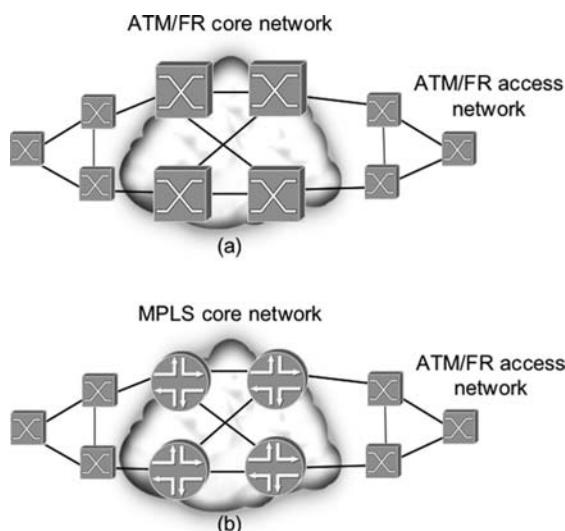


Figure 12.1 Migration of an ATM or Frame Relay core to an MPLS core: (a) prior to migration and (b) after the migration

Apart from these geographical penetration considerations, some customers may be using applications that require ATM in particular, e.g. video codecs or PABXs having an ATM interface, which would preclude them from migrating to an Ethernet service. As a consequence of the factors discussed above, service providers providing Layer 2 services over MPLS are likely to need to support a variety of access media types, including ATM, Frame Relay and Ethernet for the foreseeable future.

12.3 COMPARISON OF LAYER 2 VPNs AND LAYER 3 VPNs

The introduction to the Layer 3 VPN chapter (Chapter 7) discussed the two main models that exist for VPN connectivity: the overlay model and the peer model. BGP/MPLS-based Layer 3 VPNs fall within the peer model. In contrast, when an enterprise builds a Layer 2 VPN, by buying Layer 2 transport services from the service provider they are building an overlay network. Hence the differences between Layer 2 and Layer 3 VPNs are as follows:

1. In the Layer 2 case, no routing interaction occurs between the customer and service provider. In the L3VPN case, the CE and PE router can exchange routes.
2. In the Layer 2 case, the customer can run any type of Layer 3 protocol between sites. The SP network is simply transporting Layer 2 frames and hence is unaware of the Layer 3 protocol that is in use. Although IP is prevalent in many enterprise networks, non-IP protocols such as IPX or SNA are often in use. This would preclude the use of a Layer 3 VPN to transport that type of traffic.
3. Multiple (logical) interfaces between each CE and the corresponding PE are required in the Layer 2 case, one per remote CE that each CE needs to connect to. For example, if the CE routers are fully meshed and there are 10 CE routers in total, each CE needs nine interfaces (e.g. DLCIs, VCs or VLANs, depending on the media type) to the PE, each leading to one of the remote CE routers. In the Layer 3 VPN case, one connection between each CE and the local PE is sufficient as the PE is responsible for routing the traffic towards the appropriate egress CE.

For some customers, L3VPN is the better choice, for others L2VPN, depending on what protocols need to be carried and the degree to which the customer wishes to do their own routing or to outsource it to the service provider. Hence, in order to address the widest possible market, many service providers offer both Layer 3 and Layer 2 services over their MPLS

infrastructure. There exist PE routers that are capable of supporting both types of service simultaneously, in addition to VPLS, which is discussed in the VPLS chapter of this book (Chapter 13).

12.4 PRINCIPLES OF LAYER 2 TRANSPORT OVER MPLS

There are two main approaches to Layer 2 transport over MPLS: one involving LDP signaling [MRT-TRS] [PWE3-CON] and the other based on BGP signaling [KOM-BGP]. In the forwarding plane, these approaches are the same, in terms of how Layer 2 frames are encapsulated for transport across the MPLS network. However, the two approaches differ significantly in the control plane. In later sections, we will discuss how each approach operates and then compare and contrast the two.

A single point-to-point Layer 2 connection provided over an MPLS network is sometimes called a pseudowire, to convey the principle that as far as possible the MPLS network should be invisible to the end customer, in such a way that the two CEs interconnected by the pseudowire appear to be directly connected back to back. An MPLS-based L2VPN is composed of a collection of pseudowires that interconnect a customer's CEs in different locations, in a topology chosen by the customer, for example a full-mesh or hub-and-spoke arrangement.

One of the problems with traditional Layer 2 VPNs is the administrative burden of adding a new site to an existing VPN, and the associated lead-times. If the sites are fully meshed, when a new site is introduced a new circuit must be provisioned between the new site and every other site in the network, and hence extra configuration at every site in the network is required. Indeed, often this administrative burden has forced customers to adopt a hub-and-spoke arrangement. Later in this chapter, we will show how autodiscovery of sites using BGP greatly reduces the administrative overhead associated with traditional Layer 2 VPNs by making it much easier to add new sites to an existing mesh.

Examples of Layer 2 protocol types that can be carried over an MPLS network are as follows:

1. *ATM*. Two main modes exist: a mode in which AAL5 PDUs are transported on the pseudowire and a mode in which ATM cells are transported on the pseudowire. In the latter case, the cells could belong to any AAL type, since the AAL PDUs are not reassembled by the MPLS network.
2. *Ethernet*. The mapping of traffic into a pseudowire can be on a per-VLAN or on a per-port basis. In the per-VLAN case, if an Ethernet connection between the customer CE router and the service provider's

PE router contains multiple VLANs, each VLAN can be mapped to a different pseudowire for transport to a different remote CE.

3. *Frame Relay*. The mapping of traffic into a pseudowire can be on a per-port basis or on a per-DLCI basis. In the per-DLCI case, if a Frame Relay connection between the customer CE router and the service provider's PE router contains multiple DLCIs, each DLCI can be mapped to a different pseudowire for transport to a different remote CE.

The examples above are those that dominate in current deployments, for the reasons discussed in Section 12.2 of this chapter. In addition, the transport of HDLC and PPP frames is also supported by some vendors. Also, there is interest in the transport of TDM circuits [RFC4197] (e.g. E1 or T1) in pseudowires across MPLS networks, using the same control plane mechanisms as for Layer 2 transport.

Figure 12.2 illustrates an example network in which a service provider is using its MPLS network to provide a Layer 2 VPN service to a customer. The three customer sites are fully meshed, so between each pair of sites, a pseudowire is created to carry the traffic across the service provider's network. In the example, the media type chosen by the customer is Ethernet but the same principle applies if the customer were using ATM or Frame Relay instead. The PEs belong to the service provider and the CEs belong to the customer, so the boundary between the service provider and the customer is the set of VLAN access circuits. Each access circuit is sometimes referred to as an attachment circuit (AC). CE1 uses VLAN 100

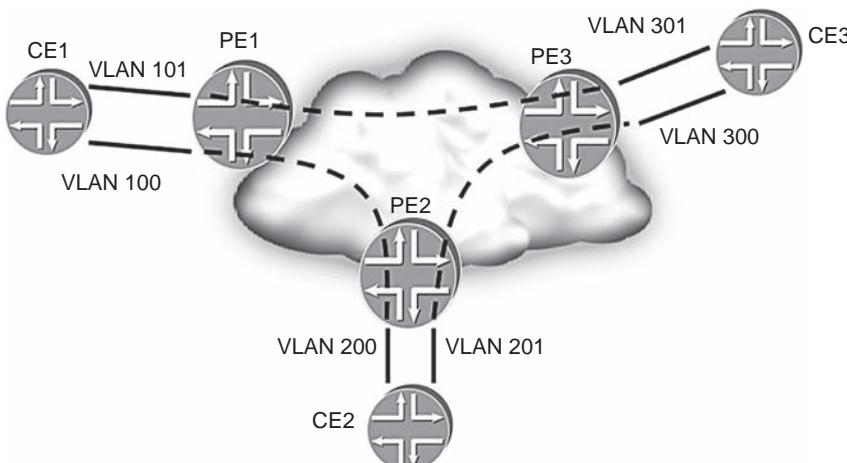


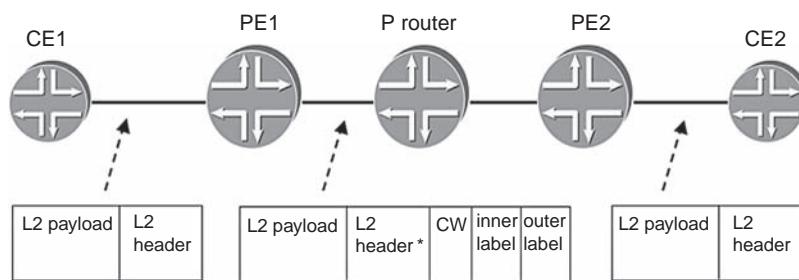
Figure 12.2 Example Layer 2 VPN showing connectivity between three customer sites

to connect to CE2 and uses VLAN 101 to connect to CE3. As far as CE1 can tell, it is directly connected to CE2 and CE3 and is not ‘aware’ of the presence of the service provider network. The ingress PE must send the packet to the appropriate egress PE, from where the packet is forwarded on the appropriate circuit to the receiving CE. For example, if a packet arrives at PE1 from CE1 on VLAN 100, PE1 must forward the packet on the pseudowire that has been created between it and PE2 for that L2VPN and PE2 in turn must forward the packet to CE2 on VLAN 200. The following sections describe how this is achieved, in terms of how the packet is encapsulated for transport across the MPLS network, and the operation of the control plane.

12.5 FORWARDING PLANE

This section describes the operation of Layer 2 transport in the forwarding plane. Note that the encapsulation method is the same regardless of whether the BGP or LDP control plane scheme is in use. The detail of how Layer 2 packets are encapsulated for transport over an MPLS network in a pseudowire is contained in an IETF document entitled ‘Encapsulation methods for transport of Layer 2 frames over IP and MPLS networks’ [MRT-ENC]. This is now a Historic RFC, having been superseded by a series of documents, one per media type, produced by the Pseudowire Emulation Edge-to-Edge (PWE3) Working Group in the IETF. The PWE3 documents contain more encapsulation variants than [MRT-ENC] for some Layer 2 media, but most existing implementations are closer to [MRT-ENC].

Figure 12.3 shows a cross-section through the network shown in Figure 12.2, showing the transport of Layer 2 packets on a pseudowire



* Which parts of the layer 2 header are transported over the MPLS core depends on the layer 2 protocol.

Figure 12.3 Forwarding plane operation of L2 transport over MPLS

between CE1 and CE2. When the Layer 2 frame arrives at PE1 on the attachment circuit, PE1 carries out the following forwarding operations:

1. Parts of the L2 frame that do not need to be transported to the remote PE are removed. For example, in the Ethernet case the FCS (Frame Check Sequence) is removed.
2. In some cases, a 4-byte Control Word (CW) is prepended to the L2 frame. The Control Word can include a sequence number so that the egress PE can detect mis-sequencing of packets. Depending on the media type, the Control Word may also contain flags corresponding to control bits within the header of the native Layer 2 frame. This allows the value of those control bits to be conveyed across the core of the network to the egress PE without having to transport the entire native Layer 2 header.
3. PE1 looks up the value of the MPLS inner label (VPN label) that PE2 expects for the frame and prepends an MPLS header having that label value.
4. PE1 determines how to reach PE2. As with L3VPN, the network operator has a choice of tunneling technologies in the core, including LDP and RSVP-signaled LSPs and GRE and IPsec tunnels. Indeed, the same tunnel can be shared by L3VPN, L2 traffic and VPLS traffic. If an LDP or RSVP-signaled LSP is used, PE1 determines the MPLS label value required to reach PE2 and stacks an MPLS header containing that label value on top of the inner MPLS header. In networks where MPLS transport is not used between PEs, PE1 determines the appropriate tunnel to reach PE2 (e.g. a GRE or IPsec tunnel).
5. PE2, on receiving the packet, examines the value of the VPN label before popping it. If the packet arrived on an MPLS tunnel and PHP is not in use, then PE2 first needs to remove the transport label in order to expose the VPN label. From the VPN label, it determines that the underlying L2 frame must be sent on VLAN 200 to CE2. If the Control Word is present, PE2 may check the sequence number and take appropriate action should the packet be out of sequence. The processing of the sequence number by the egress PE is optional. Actions that a receiving PE can take on receiving an out-of-order packet are to drop the packet or to reorder the packets into the correct sequence. PE2 then regenerates the L2 frame, which may involve determining the values of control bits in the frame header by referencing the corresponding flags in the Control Word. In the example in the figure, the Ethernet frame arrived with a VLAN identifier (ID) of 100. However, CE2 expected a VLAN ID value of 200, so PE2 must rewrite the value of the VLAN ID accordingly. PE2 then forwards the frame on the AC leading to CE2.

Let us now examine the various Layer 2 encapsulations described in the IETF documents. Note that not all implementations necessarily support all of the variants described below.

12.5.1 ATM cell

The PWE3 ATM encapsulation RFC [PWE3-ATM] specifies two modes, the N-to-one mode and the one-to-one mode. In the N-to-one mode, cells from one or more Virtual Channel Connections (VCCs) or from one or more Virtual Path Connections (VPCs) are mapped to a single pseudowire. In this case, the VPI-VCI fields of each cell are preserved when the cell is transported across the core, so that the egress PE knows which VPI/VCI a particular cell belongs to. If desired, the N-to-one mode can be used to transport all the VPCs on a particular port to a remote port in the network.

In the one-to-one mode, cells from a single VCC or a single VPC are mapped to a single pseudowire. In the single VCC case, the VPI and VCI fields are not sent across the core as they can be regenerated at the egress PE. In the single VPC case, the VPI field is not sent across the core as it can be regenerated at the egress PE.

Whichever mode is used, the Control Word can be used to carry the value of the ATM Cell Loss Priority (CLP) bit from the ingress PE to the egress PE. The egress PE can copy the value of the bit into the regenerated ATM cell.

Most implementations today support the N-to-one mode rather than the one-to-one mode. The latter is regarded as optional by the PWE3 ATM RFC [PWE3-ATM] and is not mentioned in [MRT-ENC]. Whichever mode is used, the Header Error Check (HEC) field of each ATM cell is not sent across the core. Instead, it is regenerated by the egress PE.

Either mode allows for multiple ATM cells to be sent in a single MPLS packet across the core. The number of cells that the user wishes to send is a tradeoff between bandwidth efficiency, delay variation and the number of cells that the user can afford to lose should an MPLS packet be lost.

12.5.2 ATM AAL5

In the case where the ATM data to be transported belong to AAL5, it is more bandwidth efficient to reassemble the AAL5 frame at the ingress PE and transport it across the core as a single entity than to transport unassembled cells. In this mode, there is a one-to-one mapping between ATM VCCs and pseudowires.

12.5.3 Frame relay

In the ‘one-to-one mode’, a single Frame Relay DLCI is mapped to a single pseudowire [PWE3-FR]. The Frame Relay header and FCS are not transported. The Control Word, if used, contains a bit corresponding to each of the Frame Relay parameters in the list below:

1. FECN (Forward Explicit Congestion Notification bit).
2. BECN (Backward Explicit Congestion Notification bit).
3. DE bit (Discard Eligibility bit).
4. C/R (Command/Response bit).

The use of these bits is not mandatory, but if used, the ingress PE copies the value of each bit from the Frame Relay frame into the corresponding field in the Control Word, thus allowing the state of those parameters to be conveyed across the core of the network. The egress PE then copies the value of each bit into the Frame Relay frame that it sends to the CE. The Control Word also contains a 16-bit sequence number, although its use is not mandatory.

In addition to the mode described above, the PWE3 Frame Relay RFC [PWE3-FR] also describes a port mode. In this mode, all of the DLCIs on a particular port are transported across the network in a single pseudowire to a particular remote port. This means that, unlike the one-to-one case described above, the Frame Relay address field must be transported across the core. Unlike the one-to-one mode, if the Control Word is used, the fields corresponding to the Frame Relay control bits described above are not used. The 16-bit sequence number can be used, although its use is not mandatory. Note that the port mode is regarded as optional by the PWE3 Frame Relay RFC and is not mentioned in [MRT-ENC].

12.5.4 Ethernet

Two modes of Ethernet transport [PWE3-ETH] exist, one in which the mapping to pseudowires across the core is on a per-VLAN basis and another in which an entire Ethernet port, which may contain multiple VLANs, is mapped to a single pseudowire. The use of the Control Word is optional, but if used there is a 16-bit sequence number that can be used if required. The FCS is stripped off at the ingress PE and regenerated by the egress PE. The Control Word in the Ethernet case is generally regarded as less useful than in the ATM or Frame Relay cases.

12.6 CONTROL PLANE OPERATION

Let us see how the control plane for Layer 2 transport operates. We will examine the LDP-based scheme [MRT-TRS] and the BGP-based scheme [KOM-BGP]. Both approaches have the following characteristics in common:

1. A means for a PE, when forwarding traffic from a local CE via a remote PE to a remote CE, to know the value of the VPN label (inner label) that the remote PE expects.
2. A means for signaling characteristics of the pseudowire, such as media type and MTU. This provides a means to detect whether each end of a pseudowire are configured in a consistent manner or not.
3. An assumption that the pseudowire formed is bidirectional.¹ Hence, if there is a problem with transport in one direction, forwarding is not allowed to occur in the opposite direction.
4. A means for a PE to indicate to remote PE(s) that there is a problem with connectivity, e.g. if the link to a CE goes down.

The two schemes differ significantly in the way in which a PE knows which remote PE(s) it needs to build pseudowires to. In the original LDP-based scheme, this information had to be manually configured on the PEs. The BGP scheme, in contrast, has in-built autodiscovery properties, so this manual configuration is not required. The original LDP scheme was later modified in order to also avoid this manual configuration, by using information discovered by some means external to LDP. One option for the autodiscovery aspect is to use BGP. In the following sections, we will discuss the three cases in turn: the original LDP signaling scheme, the BGP signaling and autodiscovery scheme and the LDP signaling with BGP autodiscovery scheme.

12.6.1 Original LDP signaling scheme

This section describes the original LDP scheme [MRT-TRS] for signaling pseudowires.² This approach is sometimes referred to as ‘martini pseudowires’ or ‘draft-martini’, after the IETF draft that first proposed it. In this scheme, there is no in-built concept of a L2VPN. Rather, the scheme

¹ This is the case for point-to-point pseudowires which we are discussing here. Later on in this chapter, we discuss point-to-multipoint pseudowires which are unidirectional.

² At the time of writing, more implementations and deployments used this scheme rather than the LDP scheme with autodiscovery described later.

is geared to the signaling of individual pseudowires. If an L2VPN is required, in the form of a collection of pseudowires providing connectivity between a set of CE routers, then each of the pseudowires must be manually configured.

A targeted LDP session is created between each pair of PEs in the network (or at least each pair of PEs between which L2 transport is required). On each PE the identity of the remote PE for each pseudowire is manually configured. The PE-PE LDP session is used to communicate the value of the ‘inner label’ or ‘VPN label’ that must be used for each pseudowire. In general, there may be multiple L2 pseudowires between a particular pair of PEs, each pertaining to a different customer, but only one LDP session, so an identifier, known as the VC ID,³ is used to distinguish between the connections being signaled. The same VC ID is configured on each of the two PEs taking part in each L2 pseudowire.

Referring again to Figure 12.2, on PE1 an association (by configuration in the command-line interface) would be created between VLAN 100, a VC ID and an IP address of PE2 (typically the loopback address), which is the address used for the targeted LDP session. Similarly, on PE2, an association would be created between VLAN 200, a VC ID having the same value as on PE1, and the address of PE1. PE1 uses the LDP session to inform PE2 of the VPN label value that PE2 must use when forwarding packets to PE1 on the pseudowire in question; similarly PE2 informs PE1 of the label value that PE1 must use.

Let us look at the information that is communicated over the LDP session, in addition to the inner label value itself. An LDP FEC element has been defined [MRT-TRS] in order that LDP can signal the requisite information. This FEC element has been assigned a type value of 128 and is sometimes called the ‘PWid FEC element’ or ‘FEC 128’ for short. The FEC element contains the following fields:

- VC ID.
- Control Word bit. This indicates whether a Control Word will be used.
- VC type. This indicates the encapsulation type (PPP, VLAN, etc).
- Interface parameters field. This contains information such as media MTU and, in the ATM cell transport case, is an indication of the maximum numbers of concatenated cells the PE can support.

There is no concept of a VPN as such in the LDP-based scheme. The pseudowires are created in a pair-wise manner without the network being ‘aware’ that a set of connections actually form a VPN from the customer’s perspective. Note that this lack of VPN awareness means that if a customer requires that its CEs are fully meshed with pseudowires and an additional

³ This is also known as ‘PW ID’.

CE is added to the network, each PE involved (i.e. each PE having a CE of that customer attached) must be configured with a VC ID corresponding to the new connection. This can cause a large provisioning overhead if there are a large number of CEs in the existing mesh.

The fact that LDP is being used as the signaling mechanism for the pseudowires does not mean that LDP must be used as the signaling mechanism for the underlying transport tunnels used to carry the packets from the ingress PE to the egress PE. In other words, using LDP to signal the pseudowires does not imply that LDP-signaled LSPs must be used to carry the pseudowire traffic. The transport tunnels could be RSVP-signaled or LDP-signaled LSPs or could be GRE or IPsec tunnels.

If LDP is being used as both the signaling mechanism for LSPs (as described in the Foundations chapter of this book) and for the signaling of pseudowires, there is a possibility that unnecessary information is carried over the targeted LDP session created for the signaling of pseudowires. Let us refer again to Figure 12.3. PE2 has a targeted LDP session with PE1 for pseudowire signaling. There is no need for PE1 and PE2 to exchange IPv4 FECs as they are not directly connected – they only need to exchange FECs for pseudowire signaling. Some implementations allow the advertisement of IPv4 FECs over the targeted session to be suppressed, either by default or through a configuration option. Some implementations also allow IPv4 FECs received over a targeted LDP session to be ignored, so, for example, if the implementation on PE1 does not allow the sending of IPv4 FECs to be prevented, at least on PE2 those FECs can be ignored if the implementation on PE2 allows that.

12.6.2 BGP-based signaling and autodiscovery scheme

The BGP-based approach [KOM-BGP] aims to give operational characteristics to the service provider that are familiar from a Layer 3 VPN. As with a Layer 3 VPN, BGP is used to convey VPN reachability information. With a Layer 3 VPN, service providers take it for granted that a new CE site can be added to an existing PE without having to add extra configuration to all the other PEs in the network. This is because the other PEs learn through BGP about the existence of the new site (or rather the routes associated with that site and which PE is attached to that site). This autodiscovery property has been carried through to Layer 2 VPNs in the BGP-based approach. This greatly reduces the operational burden of adding new CEs to an existing L2VPN.

As a consequence of the autodiscovery property, rather than having to manually configure a pseudowire between each pair of CEs, the pseudowires are created automatically. As with the L3VPN, a PE derives the

inner label (VPN label) in order to reach a particular remote CE from information carried in the BGP advertisements.

One difference, however, between Layer 2 and Layer 3 VPNs is that in the L2VPN case, the inner label (the VPN label) used to reach a particular CE depends on the CE that the packet originated from (so that the egress PE can determine which CE the packet came from and hence can forward the packet on the appropriate AC to the receiving CE). In principle, each PE could advertise a list of labels for each attached CE, each label on the list corresponding to one AC. However, in fact a more compact method is used, in which each PE advertises through BGP sufficient information for remote PEs to calculate the label value to use. Without this scheme, either each PE would receive information that it is not interested in (inner label values that other PEs need to use) or the information sent to each PE would have to be tailored to that PE. Using BGP to carry the necessary information allows the reuse of much of the protocol machinery already developed for L3VPNs, such as the use of route distinguishers and route targets. Also, if the service provider offers the L3VPN service, as well as the L2VPN service, the same BGP sessions and same route reflectors can be used to support both services.

Let us look in more detail at the mode of operation of the scheme. For each CE attached to a PE, a CE identifier (CE ID) is configured on the PE. This CE ID is unique within a given L2VPN. Also, each of the ACs between a CE and a PE is associated with a particular remote CE ID. This association is either explicitly made through configuration or implicitly by mapping circuits to CE IDs in the order in which they appear in the configuration. In this way, when a packet arrives from the local CE on a particular AC, the PE knows to which remote CE the packet should be forwarded. The PE obtains the knowledge about the location of a remote CE (in terms of the PE to which it is attached) from the BGP updates originated by other PEs. Each PE advertises the CE IDs of the CEs to which it is attached and also sufficient information for any other PE to calculate the pseudowire label required in order for the packet to be forwarded to the CE by the PE.

Let us have a closer look at the content of a BGP update message:

1. Extended community (route target). As with L3VPNs, this allows the receiving PE to identify which particular VPN the advertisement pertains to.
2. L2-Info extended community. This community is automatically generated by the sending PE. Encoded into the community are the following pieces of information:
 - (a) Control flags, e.g. a flag to indicate whether a control word is required or not.

- (b) Encapsulation type (PPP, VLAN, etc.). This allows the receiving PE to check that the local and remote AC of the L2 connection are of consistent media type.
 - (c) MTU (so that the PE can check that the remote AC is configured with the same MTU as the local AC).
3. Other BGP attributes such as the AS path, etc.
4. The NLRI. This contains the following items:
- (a) Route Distinguisher. As with L3VPNs, this allows ‘routes’ pertaining to different VPNs to be disambiguated.
 - (b) CE ID.
 - (c) Label base.
 - (d) Label-block offset.
 - (e) Circuit status vector (CSV) sub-TLV.

The label base and label-block offset are the information required for a remote PE to calculate the VPN label to use when sending traffic to the CE ID on that PE. Bear in mind that the value of the label by a remote PE depends on which CE it is forwarding traffic from. A PE allocates ‘blocks’ of labels. Each block is a contiguous set of label values. The PE does not explicitly advertise each label within the block. It simply advertises the value of the first label in the block (the label base) and the size of the block (the latter being the length field of the CSV sub-TLV).

In simple cases, there is only one label block whose size is sufficient that each remote CE has a label to use within the block. In such cases, the label value that a remote CE, having a CE ID of value X , must use to reach the CE in question is computed as follows:

$$\text{Label value} = \text{label base} + X - 1$$

For example, let us refer again to Figure 12.2. Suppose that PE1 advertises a label base of 100 000. PE2 and PE3 receive the advertisement, either directly or via a route reflector. Assume that the CE ID of CE1 is 1, that of CE2 is 2 and that of CE3 is 3. When PE3 receives a packet on VLAN 301, it knows (through the configuration) that the packet must be sent to CE ID 1. It looks in its routing table and sees it has an entry for CE1 and sees that the label base is 100 000. The formula above yields a label value of 100 002, which should be used as the inner label. The BGP next-hop of the route is PE1, so it knows that the packet should have an outer label pertaining to the tunnel (RSVP- or LDP-signaled LSP) that leads to PE1.

Sometimes, there may be more than one label block, e.g. if the original label block was exhausted as more sites were added to the Layer 2 VPN. In this case, each block is advertised in a separate Network Layer Reachability Information (NLRI). The first label block corresponds to the CEs having the lowest IDs, the next label block to the next lowest and so on. Note that

in the BGP NLRI, there is a ‘label-block offset’ parameter. This is equal to the CE ID that maps on to the first label in the block. For example, there might be two label blocks, each with a range of 8. The first would have a label-block offset of 1, so CE ID 1 would map to the label base. The second would have a label-block offset of 9, so CE ID 9 would map on to the label base of that block. Let us suppose that the label base of the second block is 100 020. A PE forwarding traffic from CE ID 12 would choose the fourth label in the second block, namely 100 023. The label blocks used in this example are illustrated in Figure 12.4.

It should be noted that the process by which a PE allocates label blocks and advertises them through BGP is fully automated. Hence there is no need for the network operator to be explicitly aware of what is happening.

The circuit status vector allows a PE to communicate to remote PEs the state of its connectivity. Each bit within the vector corresponds to a different AC between it and the local CE. A value of 0 indicates that both the AC in question and the tunnel LSP to the remote PE are up, while a value of 1 indicates that either or both of them are down.

The BGP scheme makes it straightforward for the network operator to create any desired topology for an L2VPN. For example, if a hub-and-spoke topology is required, this can be achieved by only provisioning on each PE that is attached to a spoke CE the remote CE ID of the hub CE. An alternative method to achieve a desired topology is through the manipulation of BGP extended communities. For example, in the hub-and-spoke case, a PE attached to a spoke site can be configured only to accept BGP NLRI originated by the PE attached to the hub CE, by virtue of the fact that the NLRI is configured with a particular extended community.

A key property of the BGP-based scheme is that a PE does not require configuration of the location of any remote CE, in terms of the identity of

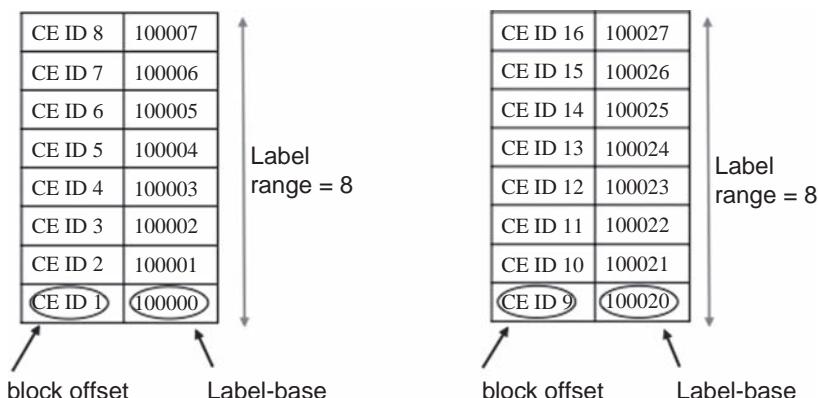


Figure 12.4 Illustration of label blocks and their mapping to CE IDs

the remote PE to which it is attached. This information is learnt through BGP, the BGP next-hop telling a PE which remote PE ‘owns’ the CE having a particular CE ID. This autodiscovery property greatly simplifies the administration of an L2VPN as new sites are added or moved. In addition, if preprovisioning is used, when a new CE site is added to an existing VPN only the local PE needs any additional configuration.

An example of preprovisioning is as follows: assume that a customer orders an L2VPN having 10 sites (CEs), but predicts that over time they may wish to grow to 20 sites. The customer wishes the sites to be fully meshed. Rather than just configuring sufficient circuits (e.g. VLANs, DLCIs, etc.) between each PE and each CE to accommodate the initial 10 sites (i.e. nine circuits), the service provider can provision 19 circuits between each PE and the local CE. Then, when the customer orders a new site, the local PE is configured accordingly and the other PEs automatically learn of the existence of the new site through BGP.

Using BGP as the signaling mechanism also has the advantage that if an entire new PE is added to the SP network and route reflectors are in use, only the new PE and the route reflectors of which it is a client need an additional configuration. An additional advantage is that some developments made for L3VPNs can also be used in the L2VPN case without having to reinvent the wheel. For example, in L2VPN interprovider scenarios, similar techniques can be used as described for the L3VPN in the Hierarchical and Inter-AS VPNs chapter of this book (Chapter 9). For example, if the interprovider option C method is used in the L2VPN case, the label operations are exactly as described in the L3VPN chapter. In fact, the ASBRs are not actually aware of whether they are carrying L2VPN traffic or L3VPN traffic (or indeed VPLS traffic, as discussed in the next chapter). Another advantage of using BGP as the signaling mechanism is that the route-target filtering (RTF) scheme described in the Layer 3 VPN chapter (Chapter 8) can be used to control the flow of L2VPN routing updates as well as L3VPN updates.

12.6.3 LDP signaling with BGP autodiscovery

As discussed in Section 12.6.1, the LDP-based scheme does not have any in-built autodiscovery properties, so each PE needs to be manually configured with the identities of the remote PEs that it needs to set up pseudowires with. In order to reduce this provisioning, the scheme was modified so that L2VPN membership information, discovered by some means external to LDP, enables each PE to know which remote PEs it needs to set up pseudowires with. The pseudowires themselves are still signaled using LDP.

One option is to use LDP signaling in conjunction with BGP autodiscovery [L2VPN-SIG]. On each PE, the following are assigned:

1. An identifier, unique to that L2VPN, associated with each local CE belonging to that L2VPN. This is analogous to the CE-ID used in the BGP signaling and autodiscovery scheme discussed in previous sections.
2. An identifier associated with each L2VPN. This is in the form of an extended community.

This information is carried in BGP, thus allowing each PE to identify which remote PEs it needs to set up a pseudowire with for that L2VPN. Each PE then uses LDP to signal to each remote PE that it has discovered for the creation of the pseudowire. The FEC 128 discussed in Section 12.6.1 is not appropriate, as the VC ID is not used but other information needs to be carried instead. Instead, a different LDP FEC element having type value 129 is used [PWE3-CON]. This is sometimes known as the 'Generalized PWid FEC Element' or FEC 129 for short. The FEC element includes fields that carry the following information to convey the identity of the pseudowire:

1. The L2VPN identifier (the value that is also carried in the BGP extended community).
2. The identifier associated with the local CE site, which is locally configured on the originating PE.
3. The identifier associated with the remote CE site, which the PE has learnt through BGP.

The LDP message also contains the inner label value that the sending PE is expecting for that pseudowire. In this way, the receiving PE now knows which remote PE owns a particular remote CE site and the pseudowire label (VPN label) required to reach it.

12.6.4 Comparison of BGP and LDP approaches to Layer 2 transport over MPLS

Table 12.1 compares the BGP and LDP approaches to Layer 2 transport over MPLS. The key difference is that the BGP scheme has in-built: Autodiscovery has properties similar to those familiar from L3VPNs. This makes provisioning straightforward, both in terms of building the initial mesh when a Layer 2 VPN is first deployed and when new CE sites are added to the mesh over time as, in both cases, the required pseudowires are automatically created rather than having to be individually configured. The original LDP scheme, in contrast, does not have any VPN awareness and requires manual pair-wise configuration of pseudowires between PEs. Hence, the provisioning burden of creating a full mesh of pseudowires

Table 12.1 Comparison of LDP and BGP control plane schemes for Layer 2 transport

	LDP-based scheme	BGP-based scheme
Control plane sessions	Fully meshed	Can use route reflectors or confederations to avoid full mesh
Explicit VPN awareness	No	Yes
In-built autodiscovery	No	Yes
Configuration burden of setting up a full mesh of connections between N sites	$O(N^2)$, unless external autodiscovery scheme is used	$O(N)$
Interdomain capability	Difficult to achieve	Yes, using schemes analogous to those for L3VPN

between N sites is of order N^2 , because on each PE, for each local CE, a connection must be provisioned to each remote CE. Whenever a new CE site is added, every PE in the network requires additional configuration. The use of BGP autodiscovery with LDP signaling removes this provisioning burden. Note, however, that it does not eliminate the need for a full-mesh of LDP sessions between all the PE routers (or at least those with at least one VPN in common). In contrast, in the BGP case, each PE needs to be involved in far fewer signaling sessions as typically it simply has a BGP session with each of two route reflectors. The LDP signaling with BGP autodiscovery scheme is rather cumbersome as it uses two separate protocols to set up an L2VPN, one for the autodiscovery part and one for the signaling part. If BGP is being used for the autodiscovery anyway, it may as well also be used for the signaling part too, thus eliminating the need for LDP to set up an L2VPN.

The use of the BGP signaling and autodiscovery scheme becomes even more attractive if the same PEs are involved in the Layer 3 VPN service as well as the Layer 2 VPN service. In this case, the same BGP sessions can be used for both the L3VPN NLRIIs and the L2VPN NLRIIs, and the same BGP infrastructure can be used, e.g. route reflectors. This is more convenient than having to invoke and maintain separate protocols (BGP and LDP) for the control plane of the two services. Furthermore, if the service provider is offering the VPLS service (the subject of the next chapter in this book), this can also use that same infrastructure if BGP signaling is used for the VPLS service.

12.7 ADMISSION CONTROL OF LAYER 2 CONNECTIONS INTO NETWORK

Sometimes service providers offer Layer 2 transport services based on pseudowires as a replacement for services that have guaranteed bandwidth, such as ATM CBR services or leased lines. Therefore, it may be advantageous to the service provider to also offer bandwidth guarantees for such pseudowire services. Guaranteed bandwidth implies that the traffic does not suffer contention when it crosses the service provider's network. This can be achieved in the following ways:

- Traffic is carried in RSVP-signaled LSPs. Each LSP has a bandwidth reservation, and admission control ensures that an LSP is only set up along a path that has the required bandwidth resources (bandwidth availability on a link or within a particular queue on the link). In order to maintain the bandwidth guarantee, overbooking is not used.
- If other traffic is allowed to use the same resources and is not subjected to admission control, that traffic must be marked to have lower priority than the traffic subjected to admission control (e.g. through EXP marking).
- Policing occurs at the ingress PE, to ensure that the traffic rate entering the LSP does not exceed the bandwidth reservation.

In some cases, there might be several LSPs between a given pair of PEs, in order to spread the load across the network, and a large number of pseudowires using those LSPs, those pseudowires belonging to various end-customers. For such cases, some implementations offer an automated scheme for mapping each pseudowire to a particular LSP, taking into account the bandwidth requirement of the pseudowire and the bandwidth availability of the LSP. If DiffServ TE is in use, the pseudowire could have a bandwidth requirement for a particular class type, or indeed multiple class types, which needs to be taken into account when identifying a suitable LSP to map the pseudowire to. An example of such a bandwidth booking scheme is shown in Figure 12.5. Suppose a pseudowire having 20 Mbps bandwidth at Class Type 1 (CT1) is required between PE1 and PE2.

Figure 12.5 shows the LSPs from PE1 to PE2 and their current bandwidth availability for each class type that they support.⁴ PE1 knows the bandwidth requirement of the pseudowire through configuration and hence knows that LSP Y has insufficient bandwidth available to accommodate the pseudowire. However, it knows that LSP X does have sufficient

⁴ Note that point-to-point pseudowires are bidirectional, so PE2 would also be performing a similar process for the pseudowire traffic traveling from PE2 to PE1. The LSPs from PE2 to PE1 are not shown on the diagram for clarity.

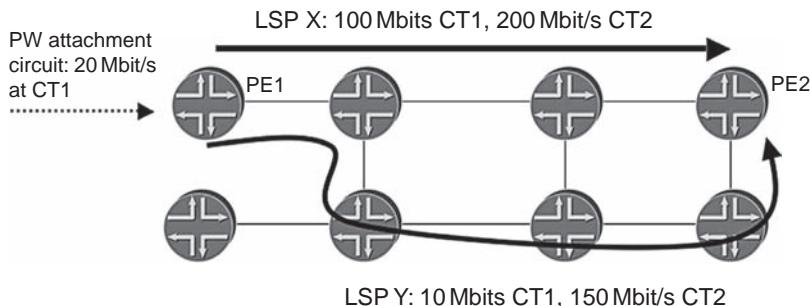


Figure 12.5 Admission control of pseudowire at PE router

bandwidth so it maps the pseudowire to LSP X. Hence, PE1 knows that LSP X has 80 Mbps remaining bandwidth at CT1 for other pseudowires that might be configured in future. The service provider needs to police the traffic arriving on the AC to PE1 to ensure that it does not exceed the bit rate reserved for that pseudowire. If an attempt is made to configure a pseudowire which cannot be accommodated on any of the LSPs, due to lack of bandwidth, all traffic arriving on the AC is dropped, otherwise there would be the danger that packets from other pseudowires using the LSP could be dropped. This admission control scheme ensures that for pseudowires that have been admitted to an LSP at the ingress PE, there is no contention along the path, and so the bandwidth requirement for the pseudowire can be guaranteed. As can be seen, there are two levels of admission control: admission control of an LSP into the network and admission control of pseudowires onto the LSP.

12.8 FAILURE NOTIFICATION MECHANISMS

An important requirement for Layer 2 transport schemes is to provide a mechanism for a PE to indicate to a local CE that there is a problem with the connection to one of the remote CEs. By way of example, let's refer to Figure 12.3. The AC between PE2 and CE2 may be down, or the LSP path from PE2 to PE1 may be down. PE2 needs to inform PE1, and in turn PE1 needs to inform CE1 in order that CE1 no longer attempts to use the circuit.

First of all, how does PE2 inform PE1 of such connectivity problems? As described in a previous section, the BGP scheme provides a circuit status vector so that a PE can advertise to remote PEs the state of its AC circuits and the state of its LSP path to the remote PE.⁵ Earlier versions

⁵ In Chapter 15, we discuss the mechanisms by which a PE can detect failures in the forwarding path between itself and the remote PE.

of the LDP scheme stipulated that a PE should withdraw the VPN label that it advertises to a remote PE if there are problems of this nature. More recently, a Status TLV has been added to the LDP scheme that allows one PE to signal to the remote PE the status of its connectivity. The action taken by PE1 on learning of a connectivity problem depends on the Layer 2 media type.

In the ATM case, operations, administration and management (OAM) cells can be generated by the PE and sent to the local CE. This action tells the ATM CE equipment that there is a problem with the VC or VP in question, so the CE will stop sending traffic on that connection. In the case where pseudowires are provided on a per-VC basis, AIS F5 OAM cells can be sent, and in the case where pseudowires are provided on a per-VP basis, AIS F4 OAM cells can be sent.

In the Frame Relay case, local management interface (LMI) frames can be used in a similar way to the OAM cells in the ATM case.

Ethernet used to be more problematic as Ethernet OAM did not exist until recently. In the absence of Ethernet OAM, if the pseudowires were being provided on a per-port basis, some implementations provided a mechanism to bring an Ethernet port down, in order to make the attached CE aware that there was a problem and to prevent it forwarding traffic to the PE. In the case where pseudowires were provided on a per-VLAN basis, this strategy of course could not be used. Now that Ethernet OAM has become available, Ethernet OAM AIS cells can be generated by the PE and sent towards the CE, in a similar way to the ATM case described above.

12.9 MULTI-HOMING

In some deployments there may be a need to multi-home some or all of the CE sites in order to achieve greater resilience, for example in case a PE fails or an AC goes down. An example is shown in Figure 12.6. CE1 is in the customer's headquarters and so is dual-homed to PE1 and PE2 in order to reduce the probability of the headquarters being unreachable. In this section, we examine how such redundancy can be achieved. We will first look at the case of BGP-signaled L2VPN and then look at the case of LDP signaled L2 connections.

12.9.1 BGP case

Referring again to Figure 12.6, let's suppose that under normal circumstances the network operator wants PE1 to be used in order to forward

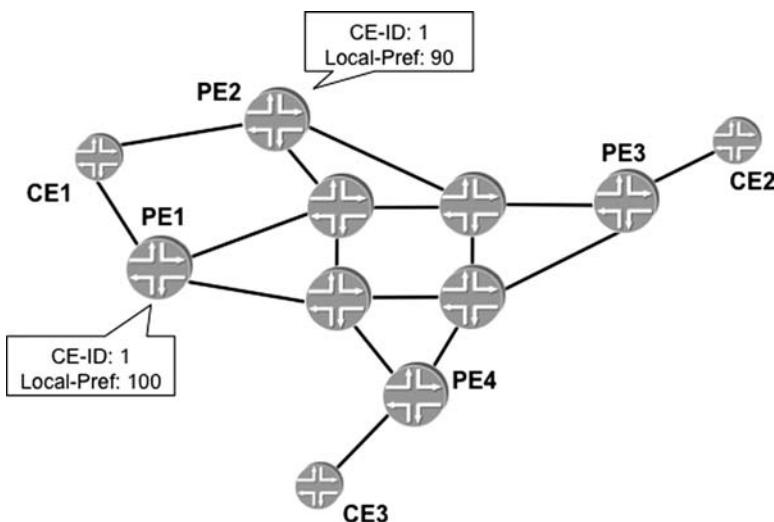


Figure 12.6 Multihoming of site CE1 for the BGP control plane case

traffic to and from site CE1, but automatic failover to PE2 is required in the event of a failure associated with PE1.

The network operator deliberately configures the L2VPN instance on PE1 and PE2 with the same CE ID, which appears in the autodiscovery messages generated by each of the two PEs. As a result, the other PEs in the VPN have two BGP paths to site CE1. In order to select between the two paths, the other PEs apply BGP selection rules. In order to enforce the network operator's requirement to have PE1 as the forwarder for site CE1, PE1 attaches a higher (more favorable) local preference than PE2 to the autodiscovery message. Hence PE3, and any other PEs involved in that L2VPN, choose to install the path advertised by PE1 in their forwarding tables and hence normally send traffic destined for CE1 to PE1. However, when the remote PEs detect a failure associated with PE1, they deselect PE1 as the active path and instead install the path advertised by PE2 in their forwarding tables. For example, if PE1 itself goes down, the remote PEs find out about the failure through the IGP. On the other hand, if the AC between PE1 and CE1 goes down, PE1 advertises this by sending a BGP update with the circuit status vector set accordingly.

As can be seen, the BGP-based multihoming scheme is very similar to how BGP-based multihoming works for IP prefixes, for example in the context of L3VPN. We will also see in the next chapter that a similar scheme applies to BGP-VPLS. This is a good example of a benefit of using the same control protocol for a variety of different service types.

12.9.2 LDP case

In the case of LDP signaled pseudowires, each PE remote to PE1 and PE2 has two pseudowires configured in order to reach site CE1, one to PE1 and the other to PE2. On each remote PE, one pseudowire is designated as primary and the other is designated as backup through configuration. This is shown in Figure 12.7 for PE3 only (similar pseudowires would be set up for PE4 as well, but are not shown in the figure for clarity). PE3 has a primary pseudowire to PE1 and a backup pseudowire to PE2. When the primary pseudowire is up, PE3 can prevent PE2 attempting to use the backup pseudowire in one of the following ways:

- (i) PE3 does not advertise a pseudowire label to PE2, thus forcing the pseudowire into a down state from the point of view of PE2.
- (ii) Some implementations allow the desired status (active or standby) of the pseudowire to be signaled via the LDP control plane, using a particular bit within the Status TLV [REDUN-BIT]. Even when the status is standby, pseudowires labels are exchanged so that pseudowire is ready for forwarding should the primary pseudowire fail.

Let's suppose the primary pseudowire fails at PE1. For example, PE1 itself could fail, and PE3 learns of this through the IGP. Alternatively, the the AC between CE1 and PE1 could fail, and PE3 learns of this through the LDP control plane. PE3 now needs to start using the backup pseudowire to PE2. In case (i) above, PE3 signals a pseudowire label to PE2 in order to activate the pseudowire. In case (ii) above, PE3 signals to PE2 that active status is now desired via the Status TLV.

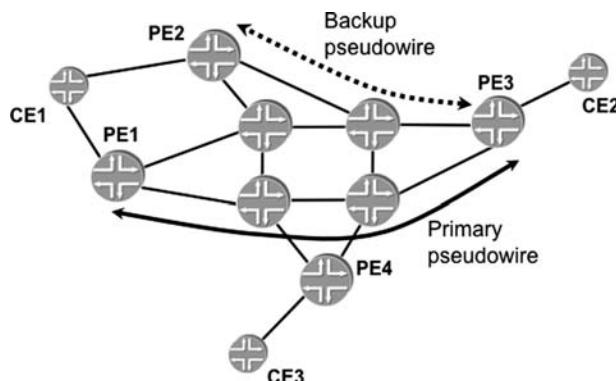


Figure 12.7 Multihoming of site CE1 for the LDP control plane case

12.10 LAYER 2 INTERWORKING

The Layer 2 schemes described so far require both ends of a Layer 2 transport connection, or all the tail circuits of a given Layer 2 VPN, to be of the same Layer 2 media type. This can be a constraint in situations where a customer uses more than one media type, perhaps as a consequence of the prevalence of different media types in the various regions in which the customer is located, or as a consequence of mergers and acquisitions or because the customer is in the middle of migrating from one type of access medium to another. One way of relaxing this constraint is Layer 2 interworking, also known as ‘Layer 2.5 VPNs’. This allows different media types to be used as the tails of the Layer 2 connections, with the proviso that the packets being transported over the connection must be IP packets.

In this scheme, when a packet arrives at a PE from the local CE, the entire Layer 2 encapsulation is stripped off, exposing the underlying IP packet. Note this is unlike the treatment of the Layer 2 frames described in Section 12.5 of this chapter, in which certain parts of the Layer 2 header are retained for transport to the remote PE. The underlying IP packet has the VPN label applied plus any transport labels and is sent to the remote PE. The remote PE extracts the IP packet and applies the appropriate Layer 2 encapsulation corresponding to the local Layer 2 tail circuit.

Only IP packets can be transported using this scheme because there is no way for the receiving PE to know which Layer 3 protocol the packet belongs to (this information having been discarded when the Layer 2 encapsulation was removed from the packet by the ingress PE; e.g. in the Ethernet case the information is carried in the Ethertype field). Hence the receiving PE simply assumes that the packet is IP and sets the relevant field accordingly when it builds the Layer 2 header ready for forwarding to the local CE.

12.11 CIRCUIT CROSS CONNECT (CCC)

This section describes Circuit Cross Connect (CCC)[CCC]. CCC was the first method to be devised and implemented for carrying Layer 2 traffic over a MPLS network and was the precursor to the LDP and BGP schemes discussed so far in this chapter. It is still used by service providers, and indeed is having a renaissance as a method to couple Layer 2 traffic into point-to-multipoint LSPs.

The main difference between CCC and the other schemes described in this chapter is that CCC always uses an RSVP-signaled LSP as the transport tunnel between PEs. Each CCC connection has a dedicated RSVP-signaled LSP associated with it, so unlike the LDP and BGP schemes discussed previously in this chapter, the transport tunnel cannot be shared between

multiple connections. This is fine for small deployments, but if a large number of connections are required between particular pairs of PEs in a network, the number of RSVP-signaled LSPs will be correspondingly large. As a consequence of having a dedicated LSP for each connection, the inner label (VPN label) that is used in the BGP and LDP schemes to identify the connection that a packet belongs to is not required in the CCC case. The Layer 2 media types supported by CCC are the same as for the BGP and LDP schemes.

By default, in most RSVP implementations the egress router declares an implicit null label for the last hop of the LSP, so penultimate hop-popping (PHP) occurs. However, in the case of CCC the egress PE needs to know on which LSP traffic is arriving so that the traffic can be mapped on to the appropriate local Layer 2 interface (bearing in mind that there is no inner label or VPN label). Hence PHP is not used in the CCC case and a non-null label is used for the last hop of the LSP.

A new RSVP object called the Properties Object, carried within the RSVP Path messages, was defined to carry information pertaining to the CCC connection. It contains a Circuit Status TLV, which allows the PE at each end of the connection to convey the status of its PE–CE link to the other PE. Hence if the PE–CE link at one end goes down, the PE at the other end becomes aware of this.

For a point-to-point CCC connection, the connection is bidirectional, so an RSVP-signaled LSP is required in each direction between the two PEs. Configuration-wise, on each of the two PEs, the user creates an association between the local PE–CE interface (VC, VLAN, DLCI, etc.) and the outgoing and incoming RSVP-signaled LSPs corresponding to that connection. If one PE learns from the other PE (via the Circuit Status TLV) that the PE–CE link at the remote end has gone down, it declares the CCC connection down and ceases forwarding traffic on that connection. Similarly, if the LSP in one direction goes down, the CCC connection is declared down in both directions. The various media-specific OAM actions taken by the PE are similar to those described in section 12.8.

12.12 POINT-TO-MULTIPOINT LAYER 2 TRANSPORT

So far in this chapter, we have been discussing point-to-point Layer 2 transport. In some cases, point-to-multipoint Layer 2 transport is required, for example in order to migrate from a legacy ATM infrastructure in which point-to-multipoint VPs or VCs have been used. As discussed in Chapter 6 of this book, some broadcast video codecs generate traffic in the form of ATM cells, so a point-to-multipoint connection is the ideal way to transport these to multiple destinations across an MPLS network, thus emulating the point-to-multipoint VP or VC. Unlike the bidirectional point-to-point

circuits discussed earlier in this chapter, point-to-multipoint Layer 2 transport is unidirectional in nature, transporting Layer 2 frames from an ingress PE to multiple egress PEs. Some implementations support a point-to-multipoint version of the CCC scheme described in the previous section. Also at the time of writing, there were proposals in the IETF for a Layer 2 Multicast VPN scheme with BGP auto-discovery of participating sites. In the following sections, we will discuss each of these schemes in turn.

12.12.1 Point-to-Multipoint CCC

As well as point-to-point connections, CCC can be used to transport point-to-multipoint traffic over P2MP LSPs. By analogy with the point-to-point CCC case described in the previous section, a dedicated RSVP-signaled LSP is used for each CCC connection. In the case of point-to-multipoint CCC connections however, the connection is unidirectional, rather than bidirectional, from the ingress router of the RSVP-signaled P2MP LSP to each of the egress routers. At the ingress PE, the user manually configures the addresses of the leaf nodes of the P2MP LSP to be used for the P2MP CCC connection. Also an association is created between the AC on which the Layer 2 frames enter the ingress PE router and the P2MP LSP. At each egress PE, the user creates an association between the P2MP LSP and the AC on which the Layer 2 frames leave the router. If one or more of the egress PEs goes down (or one or more egress ACs), the P2MP CCC connection remains up so that the remaining PEs can still receive the traffic.

12.12.2 Layer 2 Multicast VPNs

The Layer 2 Multicast VPN scheme (L2mVPN) [L2mVPN] uses similar BGP-autodiscovery techniques to the point-to-point BGP L2VPN scheme discussed earlier in this chapter.

In the case of some L2mVPNs, it may be the case that each site is both a sender site and a receiver site. In that case, each PE is the root of a P2MP LSP having each of the other PEs as leaf nodes. The BGP autodiscovery message that each PE generates contains the CE-ID of the local site and the identity of the P2MP LSP that it uses to forward traffic to the other members of the L2mVPN.

In other cases, some sites of an L2mVPN may only be sender sites and other sites may only be receiver sites. In such a case, different route-targets are used to denote sender sites versus receiver sites, by analogy with the technique used with Layer 3 NG mVPN discussed in Chapter 10.

Another aspect to consider is the mapping of L2mVPNs to P2MP LSPs. One option is for each P2MP LSP to be dedicated to a particular L2mVPN,

by analogy with the CCC case already discussed. Alternatively, aggregation could be used in which an ingress PE maps traffic from multiple L2mVPNs to the same P2MP LSP. In that case, an inner label is required in order to distinguish between traffic belonging to different VPNs sharing the same P2MP LSP. The ingress PE performs upstream allocation of the inner label and uses the BGP autodiscovery message to distribute the label value. This aggregation scheme is analogous to that discussed in the context of Layer 3 NG mVPN in Chapter 10 of this book.

12.13 OTHER APPLICATIONS OF LAYER 2 TRANSPORT

In this chapter so far, we have discussed how the Layer 2 transport mechanisms can be used to supply explicit Layer 2 services to enterprise customers. These Layer 2 transport mechanisms are also used as internal infrastructure tools in service provider networks and as a means for service providers to offer specialist services to other service providers. Some examples are listed below:

1. Layer 2 connections can be used to provide access circuits to other services. For example, if a service provider is offering a Layer 3 VPN service to a customer, rather than providing a traditional leased line connection from the CE to the Layer 3 VPN PE, they may use an Ethernet-based pseudowire between the CE and the Layer 3 VPN PE across an MPLS-enabled metro Ethernet infrastructure. This is discussed in more detail in Chapter 16 of this book.
2. Smaller service providers sometimes have fragmented networks, each based in a particular region or city. They can use Layer 2 transport services bought from larger service providers to provide interconnections between these isolated islands.
3. Some service providers also offer smaller service providers a connection to a public peering exchange by means of an Ethernet pseudowire. In this way, the customer of the service does not need to have a router at the peering exchange yet can still enter into peering agreements with other companies present at the peering exchange. This is illustrated in Figure 12.8. Service provider X has a router, PE1, at a peering exchange. Service provider X supplies an Ethernet pseudowire to service provider Y from PE1 to interface if1 on PE3 at the peering exchange. Service provider X also supplies an Ethernet pseudowire to service provider Z from PE2 to interface if2 on PE3 at the peering exchange. Once the pseudowires are set up, as far as peers A, B and C are concerned, service providers Y and Z are directly attached to the peering exchange. Service providers Y and Z can enter into peering arrangements with A, B and

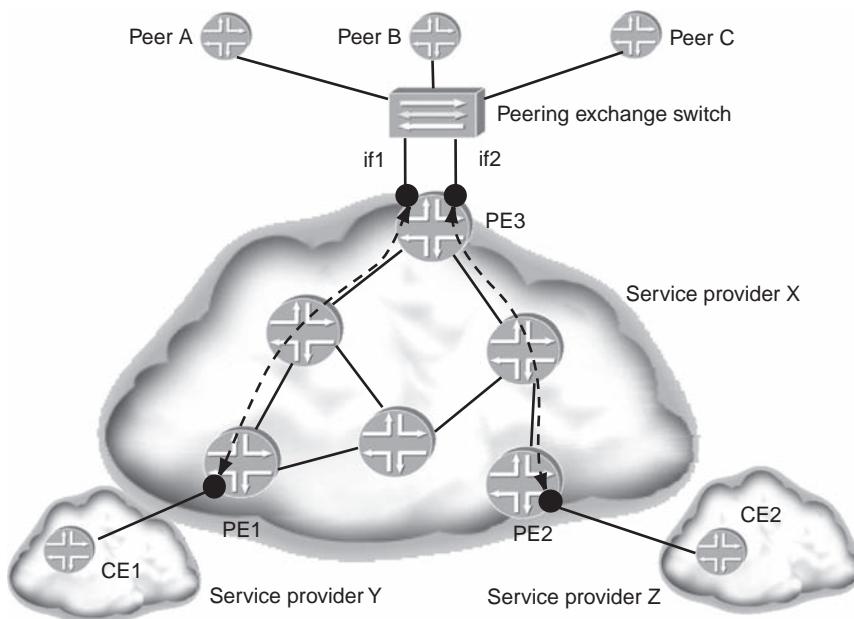


Figure 12.8 Use of pseudowires to connect small service providers to the peering exchange

C and with each other without any involvement from service provider X, since the BGP sessions for the peerings involve peer A, peer B, peer C, CE1 and CE2 but not PE3.

A variation on this scheme is the use of pseudowires to interconnect smaller service providers for the purposes of private peering; e.g. service provider X could provide a pseudowire between CE1 and CE2 to enable service providers Y and Z to peer with each other without having to go through the peering exchange.

4. In mobile telephone networks, certain infrastructure interconnections are provided over ATM. As an alternative to using a native ATM core transport infrastructure to support these connections, it is sometimes advantageous to provide them using pseudowires over an MPLS network. This is especially the case if the service provider offers other services in addition to mobile services such as Internet connectivity or the Layer 3 VPN service as the MPLS network can also be used to support those services. Pseudowires are also becoming a useful tool to allow mobile operators to migrate their Radio Access Networks (RANs) to a packet-switched infrastructure. This is discussed further in Chapter 18.

12.14 CONCLUSION

In this chapter, we have described the mechanisms underpinning Layer 2 transport over MPLS, and compared the two main control plane approaches that are in use.

The ability to transport Layer 2 traffic over an MPLS network is a key ingredient of network convergence, allowing traffic to be migrated from ATM and Frame Relay networks and allowing new services based on Ethernet transport to be created. In the Ethernet case, the service is a natural extension of the technology already used within the enterprise. In the next chapter, we will see how the service provider can go one step further by offering an Ethernet multipoint service using a scheme called the Virtual Private LAN Service (VPLS).

12.15 REFERENCES

- [CCC] K. Kompella, J. Ospina, S. Kamdar, J. Richmond and G. Miller, *Circuit Cross-connect*, draft-kompella-ccc-02.txt (expired draft)
- [KOM-BGP] K. Kompella, B. Kothari, R. Cherukuri, *Layer 2 Virtual Private Networks Using BGP for Auto-discovery and Signaling*, draft-kompella-l2vpn-l2vpn-03.txt (work in progress)
- [L2mVPN] R. Aggarwal, Y. Kamite and F. Jounay, *BGP based Virtual Private Multicast Service Auto-Discovery and Signaling*, draft-raggarwa-l2vpn-p2mp-pw-02.txt (work in progress)
- [MRT-ENC] L. Martini, E. Rosen and N. El-Aawar (eds), *Encapsulation Methods for Transport of Layer 2 Frames over IP and MPLS Networks*, RFC 4905, June 2007
- [MRT-TRS] L. Martini, E. Rosen and N. El-Aawar (eds), *Transport of Layer 2 Frames Over MPLS*, RFC 4906, June 2007
- [PWE3-ATM] L. Martini, J. Jayakumar, M. Bocci, N. El-Aawar, J. Brayley and G. Koleyni, *Encapsulation Methods for Transport of ATM over MPLS Networks*, RFC 4717, December 2006
- [PWE3-CON] L. Martini (ed.), E. Rosen, N. El-Aawar, T. Smith and G. Heron, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*, RFC 4447, April 2006
- [PWE3-ETH] L. Martini (ed.), N. El-Aawar and G. Heron, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*, RFC 4448, April 2006

- [PWE3-FR] L. Martini, C. Kawa and A.G. Malis (eds), *Encapsulation Methods for Transport of Frame Relay over MPLS Networks*, RFC 4619, September 2006
- [L2VPN-SIG] E. Rosen, W. Luo, B. Davie and V. Radoaca, *Provisioning, Autodiscovery, and Signaling in L2VPNs*, draft-ietf-l2vpn-signaling-08.txt (work in progress)
- [RFC4197] M. Riegel, *Requirements for Edge-to-edge Emulation of TDM Circuits over Packet Switching Networks*, RFC 4197, October 2005

12.16 STUDY QUESTIONS

1. In which cases might an enterprise decide to buy L2VPN service rather than L3VPN service?
2. For the case where a pseudowire is used to carry ATM cells, what is the advantage of carrying multiple ATM cells in each MPLS packet?
3. Discuss the advantages that autodiscovery brings when deploying Layer 2 services.
4. Describe some of the similarities between the BGP L2VPN scheme and the BGP L3VPN scheme.
5. Describe the ingredients that can be used to create a guaranteed bandwidth service using pseudowires.

13

Virtual Private LAN Service

13.1 INTRODUCTION

In the previous chapter, we discussed point-to-point Layer 2 transport over an MPLS network. We discussed how the Ethernet case is especially attractive to enterprise customers as it is a natural extension of the technology already used on their own sites. In this chapter, we describe how to take this integration one step further, by enabling the service provider's network to appear as a LAN to the end-user. This scheme is called the Virtual Private LAN Service (VPLS).

13.2 THE BUSINESS DRIVERS

In previous chapters in this part of the book, we have discussed L3VPN and L2VPN services, and compared and contrasted the merits of the two schemes. Both schemes require some degree of networking knowledge on the part of the customer of the service. In the L3VPN case, the customer may be required to configure a routing protocol to run between the CE and the PE, or at a minimum be required to configure a static route pointing to the PE. In the L2VPN case, the customer builds an overlay network with point-to-point connections provisioned by the service provider and needs to run a routing protocol on that overlay network. Thus the degree of expertise required of the customer is somewhat greater than in the L3VPN case. Both of these schemes may be fine for larger

companies that have IT experts available to carry out the necessary designs and configurations.

However, with network-based applications becoming more prevalent in relatively small companies, there is also a need for such companies to have connectivity over the wide area. These companies might have a handful of sites and want to have connectivity between the LANs at those sites. For such companies, it is important to have an easy-to-use service as they may not have the luxury of IT experts that the larger companies have. VPLS achieves this by allowing them to interconnect their equipment over the wide area as if it were attached to the same LAN. Note that the customer plays no part in the emulation of the LAN service – the service provider's equipment does all the work. This is very attractive to the customer as deploying the service can be as simple as plugging an Ethernet switch at each site into an Ethernet port supplied by the service provider. In the case of the L2VPN service described in the previous chapter, multiple VLANs are required between each customer site and the service provider PE (if a full mesh is required), as shown in Figure 13.1(a). In the VPLS case, just one logical interface is required (e.g. a VLAN or an untagged Ethernet port), as illustrated in Figure 13.1(b). This is because the VPLS is a multipoint service, with the service provider's PE router taking care of which remote site, or sites, each frame needs to be delivered to. Like the L2VPN services described in the previous chapter, any Layer 3 protocol can be carried over the VPLS, such as IPX and SNA.

Deploying VPLS allows the service provider to offer service to the small-to-medium enterprise sector that may have been difficult to address using L3VPN or L2VPN services. When the service is provided over native Ethernet (e.g. 100 Mbps or 1 Gbps Ethernet), it is easy for the service provider to offer a range of access rates and associated tariffs with the aid of a policer to enforce the access rate. As well as offering a native Ethernet access connection, other access media that are capable of encapsulating Ethernet frames could be used. This includes Frame Relay, using the method described in [RFC 1490], and ATM, using the method described in [RFC 2684]. Another possibility is to use a SONET/SDH circuit that supports the Generic Framing Procedure (GFP)[GFP] to encapsulate the Ethernet frame. A mixture of access media can be used within the same VPLS service instance, to cater for different types of site that the customer might have. For example, native Ethernet could be used to connect to city offices, but a DSL line with RFC 2684 encapsulation could be used to connect to branch offices.

So far we have discussed VPLS in the context of an explicit service offered by service providers to their enterprise customers. VPLS is also used as an internal infrastructure tool by service providers, for example in the context of MPLS-based Ethernet access networks. This is discussed further in Appendix A.

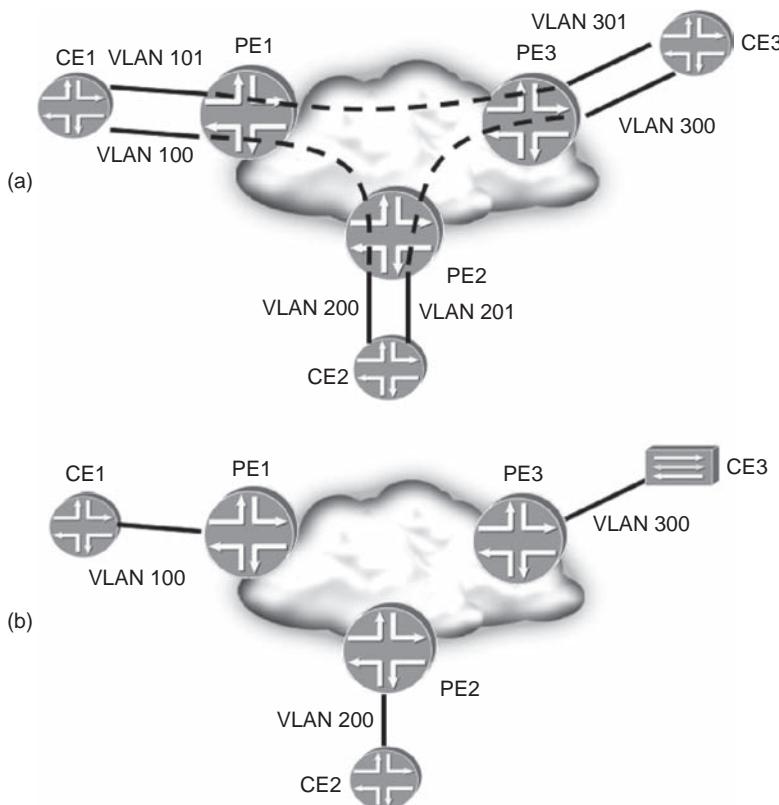


Figure 13.1 (a) L2VPN service connectivity and (b) VPLS service connectivity

13.3 VPLS MECHANISM OVERVIEW

This section gives an overview of the VPLS mechanisms, using the service provider network shown in Figure 13.2 as a reference model. Shown in the diagram are the sites of two VPLS customers, X and Y. Customer X has sites attached to PE1, PE2 and PE3.

Customer Y has sites attached to PE1, PE3 and PE4. From the point of view of each of the customers, the network appears to be a single LAN on to which that customer's, and only that customer's, CE devices are attached. That is to say, customer X belongs to one VPLS and customer Y belongs to another VPLS. This is illustrated in Figure 13.3, which shows the network from the point of view of customer Y. In Figure 13.2, each customer's device, whether a router or a switch, only requires a single Ethernet connection to the SP PE router (e.g. an untagged Ethernet interface

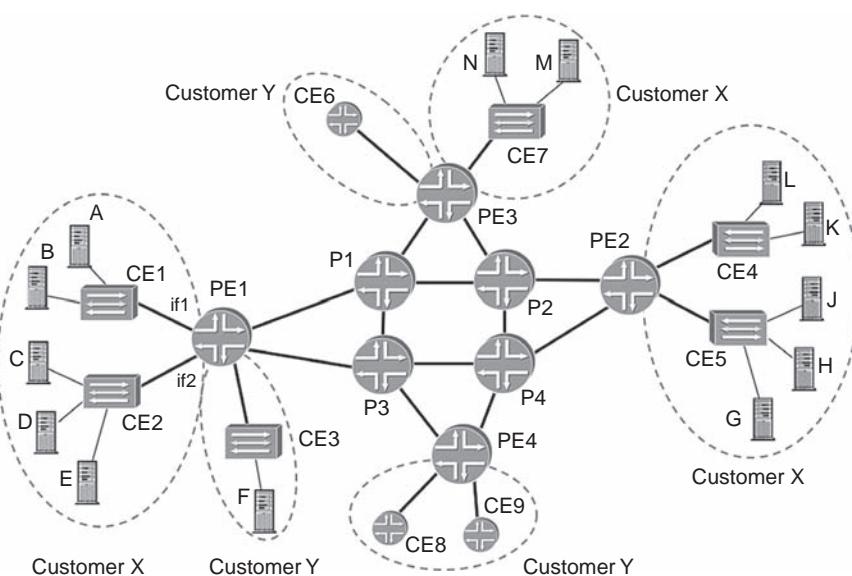


Figure 13.2 Service provider network and sites of two VPLS customers, customer X and customer Y

or a VLAN), because the VPLS is a multipoint service, with the ingress PE taking responsibility of forwarding the frame according to its destination MAC address. For reasons of resilience, a CE can be attached to more than one PE. This is discussed in more detail in Section 13.5.2.2 of this chapter. Each site of customer X contains only a handful of PCs and the CE devices are all Ethernet switches. The sites of customer Y attached to PE3 and PE4 are offices containing a large number of PCs and the corresponding CEs are routers. Customer Y's site attached to PE1 is a small branch office and so a switch is used as a CE on that site. The repercussions of having a switch rather than a router as a CE are discussed later in this section.

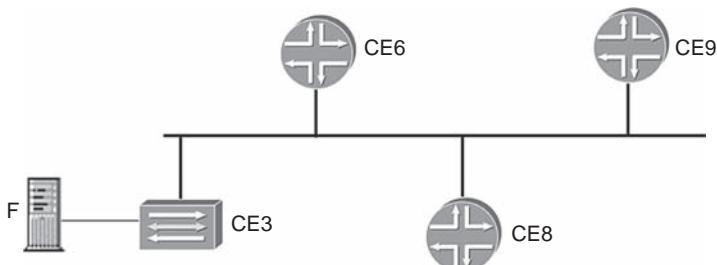


Figure 13.3 Network from the point of view of customer Y

For each VPLS, the PE routers are fully meshed with pseudowires. This is so that a PE receiving a frame from another PE can identify which VPLS the frame belongs to, on the basis of the pseudowire label. In addition, transport tunnels are required between the PEs to carry the pseudowire traffic. As with L3VPN, LDP- or RSVP-signaled LSPs are typically used, but GRE or IPSec tunnels can be used as an alternative. For example, let us consider the connectivity between PE1 and PE3 in Figure 13.2. A pseudowire is required for VPLS traffic pertaining to the VPLS of customer X and another pseudowire for VPLS traffic pertaining to the VPLS of customer Y. In order to send traffic from customer X to PE3, PE1 identifies the corresponding pseudowire label, pushes it on to the Ethernet frame and then applies the tunnel encapsulation (e.g. another MPLS label if the tunnel is an LSP). This forwarding procedure is directly analogous to that for L3VPN or Layer 2 transport, and indeed the same PE-to-PE tunnels can be used to carry traffic from all these services.

The question is, how does each PE discover which other PEs are members of a particular VPLS instance, so that it knows which PEs it needs to build pseudowires to? As with the point-to-point Layer 2 schemes discussed in the previous chapter, there are two main control plane schemes for VPLS. The control plane of one scheme is based on LDP signaling and the control plane of the other is based on BGP signaling. The BGP version of VPLS has inherent autodiscovery mechanisms, which frees the user from having to configure the pseudowires manually. The LDP version of the VPLS has no inherent autodiscovery, so either the pseudowires must be manually configured or some external autodiscovery mechanism must be used. This is discussed in more detail later in Section 13.5.1 of this chapter.

As far as each customer is concerned, an Ethernet frame that is sent into the service provider network is delivered by the service provider to the correct site(s), on the basis of the destination MAC address. It is the task of each PE router to inspect the destination MAC address of each frame arriving from a locally attached site and to forward it to the appropriate destination site. This destination site may be attached to a remote PE or may be attached to another port on the same PE. If the destination site is attached to another PE, the ingress PE must forward the frame on the appropriate pseudowire to the remote PE. This means that the ingress PE needs to know which egress PE to send the frame to.¹ In principle, two ways in which this can be achieved is to have a control plane signaling scheme to carry information about MAC addresses between PEs, or to have a scheme based on MAC address learning. VPLS takes the latter approach,

¹ An analogous situation exists in ATM LAN emulation (LANE). The LANE solution was to use central servers (LAN emulation servers) to provide a control plane function of translating destination MAC addresses to ATM addresses.

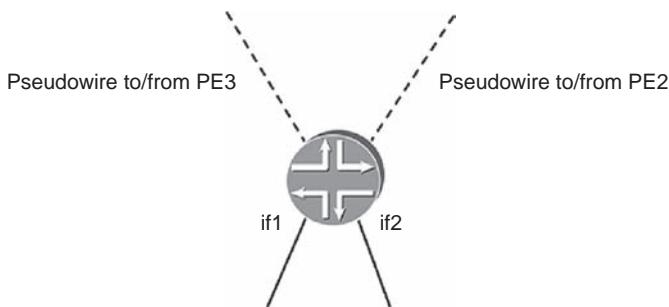


Figure 13.4 Illustration of a learning bridge on PE1 for customer X's VPLS

by having each PE take responsibility for learning which remote PE is associated with a given MAC address.

Each PE is functionally equivalent to a learning bridge, with a separate learning bridge instance for each VPLS. This is illustrated in Figure 13.4 for the case of customer X's VPLS on PE1. As can be seen, the bridge is regarded as having four logical ports, two of which are the local connections to CE1 and CE2. The other two ports are the pseudowires to PE2 and PE3. Note that PE1 is not aware of the detail of the connectivity behind each remote PE. For example, it does not need to know that customer X has two CEs attached to PE2 on separate ports. It simply needs to identify which frames need to be sent to PE2 and PE2 takes care of identifying which local port to forward the frame to. The essence of the learning function is as follows: by inspecting the source MAC address, say A, of a frame arriving on a port, whether an actual local port or a pseudowire from a remote PE, and by creating a corresponding entry in the forwarding table, PE1 learns where to send frames in the future having the destination MAC address, A. Thus with VPLS, neither centralized translation nor advertisement of MAC addresses are required.

It is important to highlight the effect of VPLS on the service provider's PE routers. In the case where Ethernet switches are used as CE devices, the service provider's PEs need to learn the MAC addresses of individual hosts attached to the switches. As MAC addresses do not have a hierarchy, no summarization is possible, so PE routers need to have forwarding table entries for each individual MAC address. This means that if someone plugs their laptop into the office network served by CE2 in Figure 13.2, the effect will be felt by all the PEs that traffic to or from that laptop crosses. This is quite unlike the L3VPN case, where PEs install forwarding entries for subnets, or Layer 2 point-to-point pseudowires, where PEs install forwarding table entries for attachment circuits. (In the L3VPN case, a PE stores MAC addresses in its ARP cache, but only for hosts attached to a directly connected CE switch and not remote hosts.) As a consequence,

the SP may decide to set a limit on the number of MAC addresses, and in practice it may be better for a large customer deployment to have routers as CEs rather than switches. Referring again to Figure 13.2, customer Y's site attached to PE3 is using a router as a CE, so the only MAC address from that site as far as the service provider is concerned is that of the Ethernet interface on CE6 facing PE3, and the service provider is not exposed to the network behind CE6.

Now that we have presented an overview of the VPLS model, let us examine the mechanisms in more detail. First, we discuss the forwarding plane mechanisms. Then we discuss the two control plane schemes for the VPLS in turn, the LDP-based signaling scheme and the BGP-based signaling and autodiscovery scheme. The LDP-based scheme is similar to the LDP-based scheme for point-to-point Layer 2 connections discussed in the previous chapter and the BGP-based scheme is similar to the BGP-based scheme for point-to-point Layer 2 connections. The forwarding plane mechanisms are very similar regardless of which signaling scheme is used.

13.4 FORWARDING PLANE MECHANISMS

In this section, we discuss the forwarding mechanisms associated with VPLS in more detail. The user needs to configure which local ports are members of each VPLS on each PE. Each PE maintains a separate forwarding table for each VPLS. For example, PE1 in Figure 13.2 maintains a forwarding table for the VPLS associated with customer X and another forwarding table associated with the VPLS in customer Y.

A requirement of the VPLS scheme is that, for each VPLS, the participating PE routers should be fully meshed with pseudowires. This means that if a PE needs to send a frame to a remote site, it can send it directly to the appropriate remote PE without the frame having to be processed by an intermediary PE. The advantage of full meshing is that the PE routers do not have to run a spanning tree algorithm to eliminate the possibility of loops (in the same way that Ethernet switches need to run such an algorithm if they are not fully meshed).

We will first discuss how unicast Ethernet frames are forwarded to their destination and then discuss the treatment of broadcast and multicast frames.

13.4.1 Forwarding of unicast frames

Let us examine how unicast Ethernet frames are forwarded from the source to the destination. Referring again to Figure 13.2, let us focus on host A

and host J in customer X's network. The MAC address of host A is A and the MAC address of host J is J. Let us suppose that host A sends a frame with source MAC address A to host J with destination MAC address J. Suppose that PE1 does not know the location of MAC address J. As a learning bridge would do, PE1 floods the frame on all ports except the port on which it arrived (refer back to Figure 13.4 for an illustration of the ports). This means that the frame is flooded to the local port towards CE2. In addition, it is flooded on the pseudowire to PE2 and the pseudowire to PE3. Note that this means that PE1 is creating two copies of the frame, one to send to PE2 and one to send to PE3. This could result in bandwidth being wasted as duplicate payloads could be traveling along the same link within the network, for example both copies might use the link between PE1 and P1. Some implementations as an alternative allow a P2MP LSP to be used by a PE to flood frames to other PEs in the VPLS instance. This is discussed in Section 13.5.5 of this chapter.

Let us consider what happens at PE2 and PE3. PE2 and PE3 know that the incoming frame belongs to customer X's VPLS, by virtue of the pseudowire on which the frame arrived.² PE2 and PE3 each perform a lookup on the destination MAC address in their VPLS forwarding tables corresponding to customer X. If PE2 does not know the location of MAC address J, it floods the frame on its local ports facing CE4 and CE5. Note, however, that it does not flood the frame to any of the other PEs in the network – there is no need to do so, because all the PEs are fully meshed, so each receives a copy of the frame directly from the ingress PE. This split horizon scheme ensures that forwarding loops do not occur (otherwise PE3 might send the frame to PE2 which sends it to PE1 which sends it to PE3 again and so on). Similarly, PE3 sends the frame on to the port facing CE7 (but not CE6, since CE6 does not belong to customer X's VPLS).

Receiving frames with source MAC address A enables each PE to learn the location of A, in terms of the port on which the frame arrived. Thus PE1 puts an entry in its forwarding table creating an association with the port facing CE1. Also PE2 and PE3 put an entry in their forwarding table creating an association between MAC address A and their respective pseudowires to PE1. At this stage, the forwarding table for customer X's VPLS on PE1 is as shown in Figure 13.5. As can be seen, PE1 has an entry for MAC address A pointing to interface if1, which is the interface facing CE1.

Let us now suppose that host J starts sending frames to host A. This is quite likely if host A has been sending frames to host J, since many applications are bidirectional. PE2 has a forwarding table entry for destination MAC address A pointing to the pseudowire to PE1, so it sends the

² Or the P2MP LSP on which the packet arrived, if P2MP LSPs are being used for flooding.

MAC address	Next-hop
A	if1

Figure 13.5 Forwarding table on PE1 for customer X's VPLS, after learning MAC address A

frame on the pseudowire to PE1 (and does not need to flood it). When PE1 receives the frame it learns the fact that frames to host J should be sent on the pseudowire to PE2 and updates its forwarding table accordingly. PE1 already has an entry in its forwarding table for MAC address A so it forwards the frame on if1 to host A.

PE1 has now learnt that frames to J must be forwarded on the pseudowire to PE2, so it no longer needs to flood frames to all the other PEs. The forwarding table entry for MAC address J contains the pseudowire label expected by PE2 for frames belonging to customer X's VPLS arriving from PE1 and the transport tunnel required to reach PE2. The choice of tunnels is the same as in the L3VPN or L2 point-to-point case. In the typical case, the transport tunnel would be an LDP- or RSVP-signaled LSP, but they could also be GRE or IPsec tunnels. If the same PEs are being used to offer all of these services, then the same transport tunnels between a pair of PEs can be shared among all those services.

Let us assume that some time later PE1 has learnt all the MAC addresses in customer X's VPLS. The forwarding table corresponding to customer X's VPLS is as shown in Figure 13.6. As can be seen, some of the entries correspond to hosts reachable via a local interface and some of the entries

MAC address	Next-hop
A	if1
B	if1
C	if2
D	if2
E	if2
G	Push 200, Push 410
H	Push 200, Push 410
J	Push 200, Push 410
K	Push 200, Push 410
L	Push 200, Push 410
M	Push 300, Push 235
N	Push 300, Push 235

Pseudowire label (inner label) to reach customer X's VPLS instance on PE2

MPLS transport tunnel label (outer label) to reach PE2

Figure 13.6 Forwarding table on PE1 for customer X's VPLS, after learning all the MAC addresses in the VPLS

point to hosts reachable via a remote PE. In the latter case, the forwarding table shows the pseudowire label and the transport tunnel label that must be pushed on to the Ethernet frame in order to forward it to the correct VPLS instance on the correct PE. For example, let us suppose that the pseudowire label expected by PE2 for frames arriving from PE1 belonging to VPLS X is 200 and the MPLS tunnel label required to reach PE2 is 410. As can be seen from the forwarding table, any frames that PE1 needs to send to hosts G, H, J, K or L have label 200 pushed on to the frame followed by label 410. Note that PE1 does not need to know the detail of the layout of the customer domain ‘behind’ the remote PE routers. For example, as can be seen from the forwarding table in the figure, PE1 knows that J and K are reachable via the pseudowire to PE2, but does not know (or care) that J and K are attached to different switches.

It should be noted that in the process described above, there was no advertising of MAC addresses using the control plane; the MAC addresses are always learnt. VPLS implementations have mechanisms for MAC aging, so that stale MAC addresses can be removed from the forwarding table. For example, an implementation may choose to remove a MAC address that has not been used for a certain number of minutes. Also, if the size of the table reaches its limit, the implementation may choose to remove the entries that have remained unused for the longest period of time.

13.4.2 Broadcast and multicast frames

Having discussed the treatment of unicast frames, let us discuss the treatment of broadcast and multicast frames. Let us suppose PE1 receives a broadcast frame sent by host B. The frame must be forwarded to all sites of customer X’s VPLS. To achieve this, PE1 floods the frame to PE2 and PE3 and on the port to CE2. In turn, PE2 and PE3 flood the frame to the attached CEs belonging to customer X, but, as a consequence of implementing the split horizon, do not send the frame to any PE.

Most implementations deployed at the time of this writing treat multicast traffic in exactly the same way as broadcast, i.e. the frame is flooded throughout the VPLS. As a consequence, each PE that has a member of that VPLS attached receives a copy of the frame, even though it may not have any interested receivers attached. This may be fine if the amount of multicast traffic is relatively low; otherwise the bandwidth wastage may be of concern to the service provider.

In some VPLS implementations, an ingress PE performs ingress replication in order to send multicast and broadcast frames to the other PEs in the VPLS instance, in the same way that those implementations deal with unknown unicast frames, as discussed in the previous section. For

example, if PE1 receives a broadcast or multicast frame from host B, it sends one copy on the pseudowire corresponding to customer X's VPLS to PE3 and another copy on the pseudowire corresponding to customer X's VPLS to PE2. This could result in bandwidth wastage due to duplicate frames being sent on the same link, if for example the LSPs to PE3 and PE2 both use the PE1 to P1 link.

Other VPLS implementations avoid the problem of bandwidth wastage due to ingress replication by instead using P2MP LSPs to carry broadcast, multicast and unknown unicast traffic. This can result in significant bandwidth savings if large volumes of multicast traffic are being sent within a VPLS instance. The control plane mechanisms by which P2MP LSPs are used within VPLS are discussed in Section 13.5.5. These mechanisms are similar to those used in the Next-Generation (NG) mVPN schemes discussed in Chapter 10 of this book.

If RSVP is used to signal the P2MP LSPs, the user can reserve bandwidth and has precise control over the path followed by each of the sub-LSPs within the P2MP LSP (e.g. to create a minimum cost tree rather than a shortest path tree in order to have the greatest bandwidth efficiency) and can take advantage of MPLS fast reroute for traffic protection.

Another optimization for multicast traffic is IGMP and PIM snooping. This is independent of whether ingress replication or multicast trees are used to forward the multicast traffic. The principle of IGMP snooping is similar to that used in some Ethernet switches. Like other Layer 3 protocol traffic traveling between a VPLS customer's sites, IGMP and PIM packets are simply 'payload' to the service provider's PEs and are carried transparently across the service provider part of the network, because VPLS is a Layer 2 service so there is no Layer 3 protocol interaction between the service provider and the VPLS customer. However, PE routers can inspect ('snoop') the contents of the IGMP and PIM packets in order to determine the location of receivers needing to receive traffic for a particular multicast group [or source and group in the case of Source-Specific Multicast (SSM)]. One level of optimization is for an egress PE router to only send multicast traffic to a CE device if there is an interested receiver behind it. For example, in Figure 13.2 if PE2 through IGMP snooping knows that only CE5 has an interested receiver for a multicast frame arriving from, say, PE1, it does not need to forward the frame to CE4. An additional level of optimization is for an ingress PE only to send multicast frames to PEs that it knows have interested receivers attached. For example, as a result of IGMP snooping, PE1 in Figure 13.2 might learn that only PE3 and not PE2 or CE2 has interested receivers for a particular multicast group whose source is behind CE1. In that case, in order to save bandwidth, PE1 just sends the traffic for that multicast group to PE2. The control plane mechanisms for IGMP and PIM snooping are discussed in Section 13.5.4.

13.5 CONTROL PLANE MECHANISMS

In the previous section, we discussed the forwarding plane mechanisms for the VPLS. Let us now turn our attention to the control plane mechanisms. There are two aspects to be considered:

1. *The discovery aspect.* How does a PE know which other PEs have members of a particular VPLS attached?
2. *The signaling aspect.* How is a full mesh of pseudowires set up between those PEs?

For example, in Figure 13.2, PE1 has members of VPLS X and VPLS Y attached. It needs to know that PE2 and PE3 have members of VPLS X attached and that PE3 and PE4 have members of VPLS Y attached. It then needs a means to signal a pseudowire to PE2 and a pseudowire to PE3 pertaining to VPLS X and a pseudowire to PE3 and a pseudowire to PE4 pertaining to VPLS Y.

As with the Layer 2 point-to-point transport discussed in the previous chapter, there are two alternative schemes for the signaling aspect. One of the schemes is based on LDP and the other on BGP. The LDP scheme is very similar to the LDP scheme for point-to-point transport and the BGP scheme is very similar to the BGP scheme for point-to-point transport discussed in the previous chapter. With regard to the discovery aspect, the BGP scheme has a built-in automated mechanisms for this discovery process (as is the case with L2 point-to-point transport signaled using BGP), so the process is known as autodiscovery. In contrast, the LDP scheme does not support in-built autodiscovery. Therefore in the LDP case, either one must manually configure the pseudowires, in terms of which PE is the destination of each, or introduce some external discovery mechanism.

13.5.1 LDP-based signaling

The LDP signaling scheme for VPLS [RFC 4762] is very similar to the LDP scheme for point-to-point Layer 2 connections described in the previous chapter. LDP is used for the signaling of the pseudowires that are used to interconnect the VPLS instances of a given customer on the PEs. In order to signal the full mesh of pseudowires required, a full mesh of targeted LDP sessions is required between the PEs, or at least each pair of PEs that have VPLS instances in common (unless H-VPLS, or hierarchical VPLS, is being used; see Section 13.5.1.2). In the absence of an autodiscovery mechanism, these sessions must be manually configured on each PE router. Whether or not autodiscovery is being used, the LDP session is used to communicate the value of the ‘inner label’ or ‘VPN label’ that must be used for each

pseudowire. An LDP FEC element carries the necessary parameters to set up the pseudowire. As described in the previous chapter, originally the ‘PWid FEC element’ (FEC 128) was defined for this purpose. Later, this was superseded by the ‘Generalized PWid FEC Element’ (FEC 129); however, most current implementations still use FEC 128.³

Let us see how FEC 128 is used in the context of VPLS. The VC ID, which in the point-to-point case was used to identify a particular pseudowire, is configured to be the same for a particular VPLS instance on all PEs. Hence the VC ID allows a PE to identify which VPLS instance the LDP message refers to.

For example, referring to Figure 13.2, PE1 has an LDP session with PE2, PE3 and PE4. Let us suppose the network operator assigns VC ID 100 to the VPLS of customer X and VC ID 101 to the VPLS of customer Y. Over the LDP session with PE2, PE1 and PE3 exchange pseudowire labels for each of the two VPLSs that they have in common. The VC IDs must be listed manually on each PE, and each VC ID must be associated with a list of remote PE addresses.

Let us look at the information that is communicated over the LDP session, in addition to the label value itself. FEC 128 includes the following fields:

1. *VC ID*.
2. *Control Word bit*. This indicates whether a control word will be used.
3. *VC type*. This indicates the encapsulation type. In the case of VPLS, this would be Ethernet or VLAN-tagged Ethernet.
4. *Interface parameters field*. This contains information such as the media MTU.

It should be noted that just because LDP is being used as the signaling mechanism for the pseudowires this does not mean that LDP must be used as the signaling mechanism for the underlying transport tunnels used to carry the packets from the ingress PE to the egress PE. As stated in Section 13.4.1 of this chapter, the transport tunnels could be RSVP-signaled or LDP-signaled LSPs or could be GRE or IPsec tunnels.

13.5.1.1 Autodiscovery mechanisms

The LDP scheme does not have any in-built autodiscovery mechanisms. As a consequence, each LDP session and each pseudowire must be manually configured or some external autodiscovery mechanism must be used.

³ [RFC 4762] describes the use of FEC 129 in the main body of the document, but also retains a description of FEC 128 in an appendix, in recognition of the fact that many implementations still use FEC 128.

In recognition of the fact that manual configuration is not an attractive option for network operators, work has been carried out in the IETF into potential external discovery mechanisms. In contrast, the BGP scheme for the VPLS discussed later in this chapter has inherent autodiscovery, so no external mechanism is required. Mechanisms that have been proposed for use with the LDP signaling scheme are as follows:

1. *RADIUS*. Using RADIUS to perform autodiscovery has been proposed in the past; however, at the time of writing, there is no work on this topic in the IETF.
2. *LDP*. Extending LDP to provide autodiscovery has also been proposed in the past; however, at the time of writing, there is no work on this topic in the IETF.
3. *BGP [BGP-AUTO]*. Here BGP is only being used as an autodiscovery mechanism; LDP is still being used to signal the pseudowires. The scheme is similar to the LDP signaling with BGP autodiscovery scheme discussed in the context of point-to-point L2VPNs in the previous chapter. For each VPLS of which it is a member, each PE advertises through BGP an identifier which is unique within that VPLS instance. This is sometimes called a 'VSI-ID' (Virtual Switching Instance ID). As in L3VPN, a Route Distinguisher (RD) is used to disambiguate VSI-IDs having the same value but belonging to different VPLS instances. Each VPLS is assigned a Route Target, in the form of an extended community value, which is common to all PEs in the context of that VPLS instance. This acts as a VPLS-ID, so that a PE receiving the BGP advertisement can identify which VPLS the advertisement is associated with and hence import it into the correct VPLS instance. In this way, for each VPLS, a PE knows which other PEs are members of that VPLS. LDP is then used to set up a pseudowire to each of the other PEs. FEC 129 must be used for the signaling, as FEC 128 does not have the capability to carry the required information. The information carried by FEC 129 includes the VPLS-ID, the remote VSI-ID and the local VSI-ID. The LDP advertisement also contains the 'inner label' or 'VPLS label' expected for incoming traffic using that pseudowire. In this way, the LDP peer can identify the VPLS instance with which the pseudowire is to be associated and the label value that it is expected to use when sending traffic on that pseudowire.

13.5.1.2 Hierarchical VPLS

Hierarchical VPLS (H-VPLS) is a scheme that was devised in order to address a significant limitation of the LDP-based signaling scheme, the fact that a full-mesh of LDP sessions is required between PE routers. This results in a large administrative overhead, unless external autodiscovery

is used. Also the number of LDP sessions can become very high if there are a large number of PEs in the network. The H-VPLS scheme removes this restriction, although, as we shall see, at the expense of introducing other issues. As discussed at the end of this section, H-VPLS also gives some bandwidth efficiency improvement when dealing with multicast or broadcast traffic.

In the H-VPLS scheme, instead of a PE being fully meshed with LDP sessions, a two-level hierarchy is created involving 'hub PEs' and 'spoke PEs'. The hub PEs are fully meshed with LDP sessions. Attached to each hub PE are multiple spoke PEs. The spoke PEs are connected to the hub PEs via pseudowires, one per VPLS instance. From the point of view of a spoke PE, it has local ports and a pseudowire 'uplink' port, leading to the parent hub PE. The spoke PE performs flooding and learning operations in the same way as a normal VPLS PE. However, the spoke PEs are not required to be fully meshed with LDP sessions.

Let us look at Figure 13.7 in order to examine this scheme. Let us suppose that all the PEs in the diagram provide a VPLS service, but the service provider wants to avoid the operational overhead of configuring a full mesh of LDP sessions and of adding to that mesh as more PEs are deployed in the future. The service provider could instead choose to implement an H-VPLS scheme by designating P1, P2, P3 and P4 as hub PEs and the PEs PE1 to PE13 as spoke PEs. Rather than having to fully mesh PE1 to

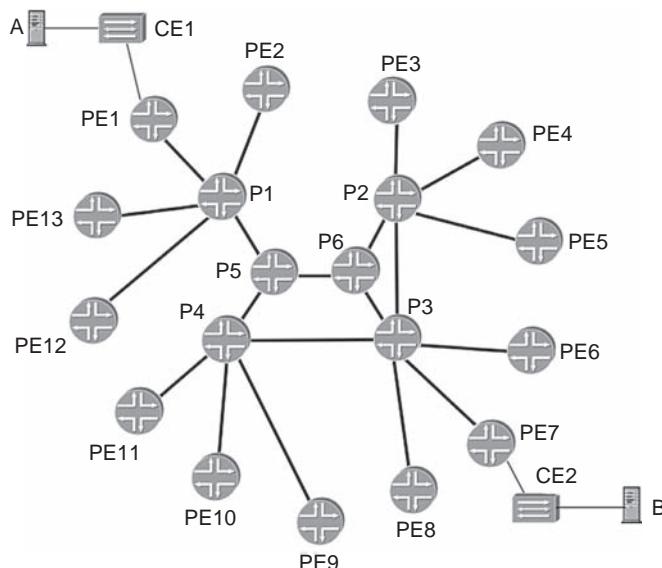


Figure 13.7 Network diagram for H-VPLS discussion

PE13, as would be the case without H-VPLS, only P1, P2, P3 and P4 are fully meshed with LDP sessions in order to exchange pseudowire labels. In addition, there is an LDP session between each hub PE and each of its satellite spoke PEs. A pseudowire is created between each hub PE and each spoke PE for each VPLS instance (i.e. for each VPLS customer attached to the spoke PE).

For example, in the figure, if P1 is the hub PE for spoke PEs PE1, PE2, PE12 and PE13, a pseudowire is created between P1 and each of those spoke PEs per VPLS that each spoke PE supports. The spoke PEs maintain a separate forwarding table for each VPLS and populate it by learning MAC addresses. Let us suppose that PE1 has two customers attached, X and Y. PE1 maintains a separate forwarding table for each of the two VPLS instances it is involved in. From the point of view of the instance pertaining to customer X, it has a logical port which is the pseudowire for that instance to P1 and a local port. It learns MAC addresses over those ports in the usual way. Let us consider the logical ports on the hub PE, P1, for that same VPLS. It has a pseudowire to each of the other hub routers in the LDP mesh and a pseudowire to each of the spoke PEs. It learns and floods across these. If host A needs to send to host B and the location of B is unknown to PE1, PE1 floods on all ports, including the uplink (pseudowire) to P1. P1, if it does not know the location of B, floods on all pseudowires, i.e. the pseudowires to its spoke PEs, PE2, PE12 and PE13, and the pseudowires to the other hub PEs. The other hub PEs in turn flood the frame to their spoke PEs. In this way, the frame is flooded to all locations on that customer's VPLS. If a new spoke PE is added to the network, or an additional customer is attached to an existing spoke PE, a new configuration is only required on that spoke PE and the parent hub PE.

Let us compare the properties of the H-VPLS scheme to a normal one-level VPLS in which the routers PE1 to PE13 are fully meshed with LDP sessions. In the H-VPLS scheme, the hub PEs, P1, P2, P3 and P4, have to learn MAC addresses. In contrast, if the service provider had deployed a one-level VPLS scheme with routers PE 1 to PE13 as the PEs then P1, P2, P3 and P4 are simply P routers and so do not carry any of this information. The number of MAC addresses needed to be stored in a hub PE in the H-VPLS scheme is roughly equal to the sum of the MAC addresses in its satellite spoke PEs. This means that it is important to ensure that the hub PE is capable of storing the expected number of MAC addresses.

Another property of the H-VPLS scheme to examine is the handling of broadcast and multicast traffic. If ingress replication is used to deal with this type of traffic, then the H-VPLS scheme is more bandwidth efficient than the one-level VPLS scheme. For example, PE1 only needs to send one copy of each broadcast/multicast frame to P1 which then floods to the other hub PEs and to its satellite spoke PEs. However, if the bandwidth

inefficiency of ingress replication in a one-level VPLS scheme is a concern, a much better method of curing it is through the use of multicast replication trees, as discussed in Sections 13.4.2 and 13.5.5 of this chapter, rather than the use of H-VPLS. In contrast to H-VPLS, which only saves the bandwidth on the first hop, a multicast replication tree can give optimum bandwidth utilization end to end.

13.5.2 BGP signaling and autodiscovery

The BGP signaling and autodiscovery scheme for VPLS [RFC 4761] is very similar to that for L2VPN and L3VPN. Fundamentally it has the following components:

1. A means for a PE to know which remote PEs are members of a given VPLS. This process is known as autodiscovery.
2. A means for a PE to know the pseudowire label expected by a given remote PE for a given VPLS. This process is known as signaling.

A BGP NLRI has been defined for this purpose, known as the BGP VPLS NLRI. This takes care of the two components above at the same time, the NLRI generated by a given PE containing the necessary information required by any other PE. These components enable the automatic setting up of a full mesh of pseudowires for each VPLS without having to manually configure those pseudowires on each PE.

Like the BGP scheme for L3VPN and L2VPN, on each PE a RD and a Route Target is configured for each VPLS. The Route Target is the same for a particular VPLS across all PEs, and is used to identify which VPLS an incoming BGP message pertains to. This is exactly analogous to the L3VPN and L2VPN cases, in which the RT identifies which VRF or L2VPN instance a BGP advertisement pertains to. As in the L3VPN and L2VPN cases, the RD is used to disambiguate ‘routes’.

On each PE, for each VPLS an identifier is configured, known as a VPLS Edge Identifier (VE ID). Each PE involved in a particular VPLS must be configured with a different VE ID.⁴ BGP is used to advertise the VE ID to other PEs in the network. This, along with other information in the NLRI, provides the means for remote PEs to calculate the value of the pseudowire label required to reach the advertising PE. The key advantage of the scheme is that each PE discovers the identities of all the other PEs in each VPLS without requiring any manual configuration of that information.

The VE ID is somewhat analogous to the CE ID in the BGP signaling scheme for L2VPNs discussed in the previous chapter. One difference is

⁴ An exception is the scheme for multihoming discussed later in this chapter.

that a single VE ID covers all the CEs that belong to a given VPLS instance on a PE, whereas a different CE ID is needed for each CE in a given L2VPN instance on a PE. For example, in Figure 13.2, although PE2 has two CEs attached that are members of customer X's VPLS, it advertises one VE ID that encompasses both (and any other CEs that customer X might attach in the future to PE2). This difference is because, in the L2VPN case, the pseudowire maps to a particular local attachment circuit (VLAN, VC, etc.), whereas in the VPLS case, the pseudowire maps to the VPLS instance pertaining to a particular customer.

Note that, for a given VPLS, a given PE requires that each remote PE uses a different pseudowire label to send traffic to that PE. This is to facilitate the MAC learning process, as described in Section 13.4.1 of this chapter. Knowing which PE sent a frame means that the receiving PE can learn which PE is associated with the source MAC address of the frame. A PE in principle could simply send a list of pseudowire labels required to reach it, one per remote VPLS Edge in that VPLS. However, this could potentially mean having to send a long list of labels if there are a large number of PEs in the network that are involved in that VPLS instance.

Instead, as with BGP L2VPNs described in the previous chapter, the necessary information is communicated in the BGP NLRI to enable each remote PE to calculate the pseudowire label expected by the advertising PE. The advantage of this scheme is that each PE in a given VPLS only needs to generate a small 'nugget' of information (as little as one NLRI per VPLS) to enable any remote PE to know that the PE in question is a member of that VPLS (by virtue of the route target and the BGP next-hop) and the pseudowire label expected by that PE.

Let us look in more detail at the information communicated by BGP to see how this scheme works. As can be seen, the scheme is very similar to the scheme for L2VPNs described in the previous chapter. A BGP update message contains the following items:

1. *Extended community (route target)*. As with L3VPNs, this allows the receiving PE to identify which particular VPN the advertisement pertains to.
2. *L2-Info extended community*. This community is automatically generated by the sending PE. Encoded into the community are the following pieces of information:
 - (a) *Control flags*. These include a flag to indicate whether a Control Word is required or not and a flag known as the Down-bit to indicate that all the attachment circuits to the local site are down.
 - (b) *Encapsulation type* (i.e. Ethernet with VLAN tagging or untagged Ethernet). This allows the receiving PE to check that the local and remote ports are configured in a consistent manner.

- (c) *MTU* (so that PE can check that remote ports are configured with the same MTU as the local ports).
- 3. Other BGP attributes such as the AS path, etc.
- 4. *The NLRI*. This contains the following items:
 - (a) RD. As with L3VPNs, this allows ‘routes’ pertaining to different VPNs to be disambiguated.
 - (b) VE ID.
 - (c) Label base.
 - (d) VE block offset.
 - (e) VE block size.

The label base, VE block offset and VE block size are the information required for a remote PE to calculate the pseudowire label to use when sending traffic to the VPLS in question on the advertising PE. A PE allocates ‘blocks’ of labels. Each block is a contiguous set of label values. The PE does not explicitly advertise each label within the block. It simply advertises the value of the first label in the block (the label base) and the number of labels in the block (the VE block size).

In simple cases, there is only one label block whose size is sufficient that each remote PE has a label to use within the block. In such cases, the label value that a remote PE, having a VE ID of value X, must use to reach the advertising PE is computed as follows:

$$\text{Label value} = \text{label base} + X - 1$$

For example, let us refer again to Figure 13.2. Suppose that PE1 advertises a label base of 100 000. PE2 and PE3 receive the advertisement, either directly or via a route reflector. Assume that the VE ID of customer X’s VPLS instance on PE 1 is 1, that on PE2 is 2 and that on PE3 is 3. When PE3 needs to forward a frame to PE1, it calculates the pseudowire label expected by PE1 by adding its own VE ID (value 3) to the label base of 100 000 and by subtracting 1, yielding a label value of 100 002 to be used as the inner label. The BGP next-hop of the route is PE1, so PE3 knows that the frame should have an outer label pertaining to the tunnel (RSVP or LDP-signaled LSP) that leads to PE1.

Sometimes, there may be more than one label block, e.g. if the original label block was exhausted as more sites were added to the VPLS. In this case, each block is advertised in a separate NLRI. Note that in the BGP NLRI, there is a ‘VE block offset’ parameter. This is equal to the VE ID that maps on to the first label in the block. For example, let us suppose that there are two label blocks for a particular VPLS, each with a range of 8. The first would have a label block offset of 1, so VE ID 1 would map to the label base of that block. The second would have a label block offset of 9, so VE ID 9 would map on to the label base of that block. Let us suppose that

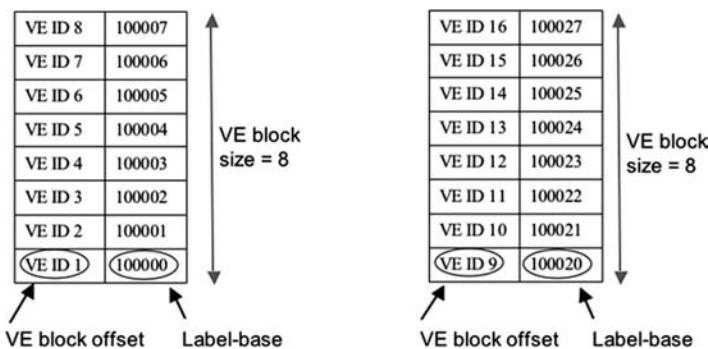


Figure 13.8 Illustration of label blocks and their mapping to VE IDs

the label base of the second block is 100 020. A PE forwarding traffic from VE ID 12 would use the fourth label in the second block, namely 100 023. These label blocks are illustrated in Figure 13.8.

13.5.2.1 Interprovider mechanisms for BGP-signaled VPLS

It is likely that some customers requiring VPLS service are based in multiple geographies. Often it is the case that no single service provider can provide service in all those geographies. Hence there is a need for interprovider capability for the VPLS, in such a way that a seamless end-to-end service can be offered to the customer across multiple ASs.⁵ In the Hierarchical and Inter-AS VPNs chapter of this book (Chapter 9), we discussed three alternative schemes by which interprovider capability can be achieved for L3VPNs. These schemes are known as option A, option B and option C after their section numbers in the IETF draft. The RFC describing BGP signaling for VPLS [RFC 4761] includes an analogous version of each of these for the VPLS case. Many of the mechanisms are the same for the L3VPN and VPLS cases, which saves having to reinvent the wheel. In this section, we examine each of these in turn. In addition to Options A, B and C, there is another option unique to VPLS called Option E. We examine this option in Section 13.7 and then compare all four options.

Option A: VPLS to VPLS connections between ASBRs

By analogy with the L3VPN case, in which a back-to-back connection is made between VRPs on the two ASBRs, in the VPLS case a back-to-back connection is made between the VPLS instances on the two ASBRs. If

⁵ Also, in some cases a service provider network may contain multiple ASs, hence requiring the use of interprovider mechanisms.

there are multiple instances needing to be connected in this way, a separate VLAN is used for each. To the PEs in each AS, the ASBR in that AS acts as any normal PE and hence is involved in the flooding and MAC learning operations in the same way as a normal PE. The fact that the ASBR needs to hold MAC address information is analogous to the L3VPN case, in which the ASBR holds L3VPN routes. To each ASBR, the other ASBR acts as a CE router. One issue with the scheme in the VPLS case is that if for reasons of redundancy multiple inter-AS connections are required between the two ASs between different ASBRs, then a scheme such as a spanning tree protocol would be required to prevent forwarding loops occurring.

Option B: distribution of VPLS information between ASBRs by BGP

In this scheme, an EBGP session between ASBRs is used to advertise label blocks from one AS into another. For each label block advertised by a PE in the same AS as the ASBR, the ASBR creates an analogous label block which is advertised to the peer ASBR over the EBGP session. The label block created by the ASBR contains the same number of labels and maps on to the same VE IDs as the original label block, but the label base can be different. Why is this ‘translation’ of the label block necessary? If the ASBR simply relayed the label block with the same label base as that chosen by the originating PE, the label values could clash with labels that the ASBR had allocated for other purposes. For the same reason, in the L3VPN case, the VPN label may be translated by the ASBR.

In the forwarding table, the ASBR installs a label swap operation between each label value in the original label block and the corresponding label value in the new label block. Note that the analogous scheme for the L3VPN potentially may involve the ASBR holding a large number of L3VPN routes. However, in the VPLS scheme the number of ‘routes’ is small (as few as one per VPLS instance) so the scheme is very scalable in the VPLS case. Note that the ASBR is not involved in any MAC learning and does not need to hold any MAC address information.

Option C: multihop EBGP redistribution of VPLS NLRI between source and destination AS, with EBGP redistribution of labeled IPv4 routes between ASBRs

This is directly analogous to option C for L3VPN, except that the EBGP multihop connection between PEs (or route reflectors) in the two ASs conveys VPLS NLRI rather than L3VPN NLRI. As far as the ASBRs are concerned, the operations are the same as for the L3VPN case – PE loopback addresses are advertised as labeled IPv4 routes to enable an MPLS forwarding path between PEs in the two ASs. Indeed, if the ASBRs are already exchanging labeled IPv4 routes for the purposes of interprovider L3VPN, no change in configuration is required to accommodate the

interprovider VPLS service since the same PE-to-PE MPLS path can be used by both services. The ASBRs are not involved in any of the VPLS forwarding or control plane operations, in the same way that they are not involved in the L3VPN operations in the L3VPN case. At the time of writing, this inter-AS scheme had already been deployed by a pair of service providers.

In Section 13.7, we examine Option E for VPLS and compare the relative merits of the different interprovider schemes.

13.5.2.2 Multihoming

In the diagram shown in Figure 13.2, each customer site was attached to one PE only. In practice, customers may wish to have some or all sites attached to more than one PE for reasons of resilience. In such scenarios, it is important to avoid Layer 2 forwarding loops occurring, bearing in mind that, unlike the case of IP forwarding loops, there is no time-to-live (TTL) mechanism to limit the number of circulations a frame can make. If the customer's CE devices are routers, then loops will not occur. If, however, the customer's CE devices are Ethernet switches, then there is the danger of loops occurring unless countermeasures are taken.

Let us look at Figure 13.9 to see how such loops could occur in the absence of any countermeasures. The Ethernet switch CE1 is homed to PE1 and PE2. Host A sends an Ethernet frame addressed to host B. Let us assume that none of the PEs in the network knows the location of host B.

PE3 floods the frame to all the PEs in the network. PE1 and PE2 each send a copy of the frame to CE1. Let us assume that CE1 does not know the

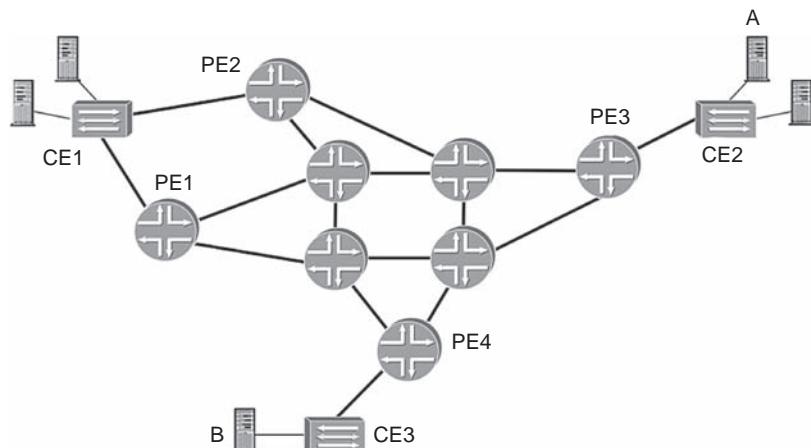


Figure 13.9 Multihoming of the customer's CE

location of host B. If CE1 is not running a spanning tree, then on receiving the frames from the PEs, it floods them on all ports, except the incoming port. As a result, PE2 receives a copy of the frame that had been sent to CE1 from PE1. PE2 floods it on all ports (excluding the incoming one), so PE1 receives a copy of the same frame and floods it on all ports and so on.

One countermeasure is for the customer to run the spanning-tree algorithm on its switches, so that the switches can create a loop-free topology by selectively blocking ports. However, this involves the service provider relying on the customer to implement this correctly. This runs counter to the model used by service providers for other services, in which the mechanisms of the service and the practices adopted by the service provider protect the service provider from mistakes made by the customer. Mistakes made by one customer cannot be allowed to affect other customers or the service provider.

The service provider can protect itself from such errors by only allowing one port to be active at a time. The question is how the PEs to which the customer site is homed know which should be the active port and how other PEs know which port is the active one. This can be achieved in the BGP case in quite a natural and straightforward way. Let us consider the dual-homed case shown in Figure 13.9, although the scheme operates in an analogous fashion for homing to more than two points. The scheme is very similar to the L2VPN multihoming scheme discussed in the previous chapter.

The service provider configures the VPLS instance on PE1 and PE2 with the same VE ID, even though they are attached to different PEs. Each PE creates a BGP NLRI for that VE ID in the usual way. Thus each PE in the network receives two advertisements for the same VE ID. In the same way as, for example, IP path selection, BGP applies selection rules to determine which of the two NRIs it installs in its routing table.

The goal is that traffic from other PEs only exits the network on one of the two ports facing CE1. Let us suppose that by default we wish traffic to exit from PE1. This can be achieved by having PE1 apply a higher (more favorable) local preference than PE2 when advertising the NLRIs. Hence PE3, and any other PEs involved in that VPLS, choose to install the version of route advertised by PE1. PE2 sees the advertisement from PE1 and, seeing that it has a higher local preference than its own version of the route, does not forward any frames on the attachment circuit to CE1, and drops any frames that arrive on that circuit. This is key to achieving loop-free operation. If PE1 goes down, or the port or link facing CE1 on PE1 goes down, then the BGP session between PE1 and its peers (other PEs or route reflectors) times out or PE1 sends a BGP update with the Down-bit set. This triggers the other PEs to install the version of the route sent by PE2 and triggers PE2 to start using its port to CE1. An additional advantage of the BGP path selection scheme (compared to simply relying

on correct use of the Spanning Tree Protocol, or STP, by the customer) is that a remote PE (e.g. PE3) only sends broadcast traffic to the PE with the active port, rather than to both PEs, resulting in less bandwidth wastage in the core of the network.

At the time of writing, there is work in the IETF [MULTIHOME] to extend the use of BGP-signaling as a control plane for multi-homing to LDP-signaled VPLS. Various scenarios are possible. For example, a VPLS could use LDP for the service signaling and BGP for the both autodiscovery and the multihoming. Alternatively, a VPLS might have no autodiscovery at all and uses LDP for the service signaling and BGP just for the multihoming signaling. Another aspect of the IETF work is to allow the multihoming to extend across multiple ASs. In such cases, BGP local preference cannot be used to control PE selection because the local preference attribute, by definition, does not cross AS boundaries. Instead, a VPLS Preference parameter is proposed. This is contained within the Layer2 Info Extended Community and hence can cross AS boundaries.

13.5.3 Comparison of LDP and BGP for VPLS control plane implementation

Let us now compare the LDP and BGP schemes for the VPLS. Many of the differences between the two schemes are analogous to the differences between the LDP and BGP schemes for point-to-point Layer 2 transport discussed in the previous chapter of this book.

13.5.3.1 Control plane signaling sessions

A key difference between the LDP and BGP schemes is the fact that a full mesh of LDP sessions is required between the PE routers, whereas a full mesh of BGP sessions is not required between PE routers. This has two consequences:

- Potentially a PE router is involved in a large number of targeted LDP sessions. This number grows in proportion to the number of PEs in the network and could be the factor that limits how much the network can grow. This is in contrast to the case in which LDP is used as a label distribution protocol for MPLS transport tunnels. In that case, the number of LDP sessions is typically low and fairly constant (one session between each directly connected pair of routers in the network).
- If external autodiscovery is not used, the configuration burden of having a full mesh of LDP sessions is large. When a new PE is added to the network, a new LDP session has to be configured on every existing PE in the network. This is in contrast to the case in which LDP is used as

a label distribution protocol for MPLS transport tunnels. In that case, LDP is very attractive precisely because of the ease of configuration, the configuration typically involving activating the LDP protocol on each core interface in the network. In addition, if md5 authentication is in use, there is the overhead of managing the md5 authentication keys associated with each LDP session in the mesh.

Note that a full mesh of BGP sessions would give rise to similar issues. In order to address this, BGP Route Reflection was developed. It is in widespread use today and enables service providers to run the Layer 3 VPN and Internet service without having to maintain a full mesh of BGP sessions between PE routers. The same route reflectors can also be used for the VPLS service.

H-VPLS, on the face of it, solves the two problems described above, but at the expense of introducing other problems, as discussed in Section 13.5.1.2 of this chapter. This is because H-VPLS is not a mechanism analogous to BGP route reflection. BGP route reflectors, by relaying signaling information, remove the need for direct sessions between PEs without affecting any forwarding plane operations. H-VPLS removes the need for LDP sessions between outer PEs, but in an indirect way, by having spoke PE routers forward frames to a hub PE. Because a spoke PE does not forward frames directly to another spoke PE, it does not need a pseudowire to it and hence does not need an associated LDP session. As described in Section 13.5.1.2 of this chapter, the consequence of this is an increased amount of learning and storage of MAC addresses on the hub PE, approximately equal to the sum of the MAC addresses stored on its spoke PEs.

Let us return to Figure 13.7 to examine how the BGP version of VPLS deals with the problem of a full mesh of control plane sessions. Instead of having a full mesh of BGP sessions between PE1 and PE13, route reflectors can be used. For example, P1, P2, P3 and P4 can be designated route reflectors and PE1 to PE13 can be router reflector clients. In so doing, the P routers do not get involved in having to learn and store MAC address information. They are simply holding and relaying the BGP reachability information received from the PE routers. As a consequence, there is no need to employ H-VPLS in the BGP version of VPLS. In order to deploy a new PE, one simply configures BGP sessions between that PE and its route reflectors.

13.5.3.2 *Discovery of remote PEs*

Another difference between the BGP and LDP versions of VPLS is the way in which a remote PE discovers which other PEs are involved in a particular VPLS. The LDP version of VPLS, as it stands today, does not have any inbuilt discovery mechanism, so the identities of the remote PEs must be manually configured on the routers if an external autodiscovery

scheme is not being used. This means that if a new site is added to a customer's existing VPLS on to a PE that does not already have sites of that customer attached, then all the other PEs that have a site of that VPLS attached must be configured with a pseudowire to that PE. This runs counter to the operational model for L3VPNs, where if a new CE is added to a customer's VPN, only the PE to which it is attached requires any configuration. The BGP scheme for VPLS in contrast, described in Section 13.5.2 of this chapter, has in-built autodiscovery. Therefore the operational model is very similar to that for L3VPN: if a new site is added to an existing VPLS, only the PE to which the site is attached requires any new configuration.

As already discussed in Section 13.5.1.1 of this chapter, external autodiscovery mechanisms have been devised that could be used in conjunction with LDP. One of the schemes is to combine BGP autodiscovery with LDP signaling. However, it is difficult to see the advantage of this scheme compared to using BGP for autodiscovery and signaling and not using LDP at all. Once BGP is used for autodiscovery, the amount of extra information required to convey pseudowire MPLS label information is very small, with one NLRI taking care of both aspects.

As well reducing the operational burden, an autodiscovery scheme reduces the probability of configuration errors being introduced. As already discussed, a premise of the VPLS schemes is that the PE routers are fully meshed with pseudowires for each VPLS instance. If by accident one or more pseudowires are omitted, unexpected behavior can occur within the customer domain, the cause of which can be difficult to pinpoint. For example, in Figure 13.2, let us suppose the pseudowire between PE1 and PE3 corresponding to customer X's VPLS service is missing. If host A sends an ARP corresponding to host M, then it receives no reply. If, however, it sends ARPs for host L, then it does receive a reply. If all the hosts involved were attached to a traditional LAN, then that would lead one to conclude that host M is turned off or the port to it is down, whereas in fact the problem is in the SP part of the network. Let us look at another example. Let us suppose that the CE routers in customer Y's VPLS are running OSPF and the pseudowire between PE3 and PE4 corresponding to customer Y's VPLS service is missing. CE6 is the designated router for the LAN and CE8 is the backup designated router. CE8 does not hear OSPF hellos from CE6 as a consequence of the missing pseudowire. This causes CE8 to take over as the designated router, confusing other CEs in the network, which are still receiving OSPF hellos from CE6.

13.5.3.3 *Interprovider operations*

In Section 13.5.2.1, we discussed the interprovider capability offered by the BGP scheme for VPLS and showed how the schemes are analogous to

those for L3VPN. In the LDP case, the only methods proposed at the time of writing are:

1. '*Brute force*' meshing. If interprovider capability is offered by ASs 1 and 2, a full mesh of LDP sessions is created between all the PEs in AS 1 and AS 2. In addition, all the PEs in the two ASs providing the VPLS service to a particular customer need to be fully meshed with pseudowires. This compounds the operational difficulties with adding a new PE or adding an additional site to an existing customer's VPLS, as extra configuration is required on all the PEs involved in the two ASs.
2. *Using a spoke pseudowire*. A spoke pseudowire (per VPLS) is provisioned between border routers, in order to interconnect the VPLS instances in the two domains. Methods for providing multiple connections for redundancy between different border routers without causing loops are under investigation. This problem is the same as in the option A BGP scheme described in Section 13.5.2.1. However, the option B and option C BGP schemes described in that section avoid the problem by the use of the AS path attribute. LDP does not have the capability to create schemes equivalent to these BGP schemes.

In summary, interprovider VPLS services are more readily created using the BGP version of VPLS, because BGP (by definition) was designed with inter-AS operations in mind.⁶ Also the BGP version of VPLS has the advantage that the machinery developed for L3VPN interprovider operations can largely be reused.

Summary of differences between LDP and BGP schemes for VPLS

The differences between the LDP and BGP signaling schemes for VPLS are summarized in Table 13.1. Although in principle it might be possible to modify LDP to accommodate all these issues, in effect one would be reinventing BGP, so the advantage of doing so is not clear.

13.5.4 IGMP and PIM snooping

In Section 13.4.2, we mentioned that IGMP and PIM snooping can help optimize the transport of multicast traffic in VPLS, assuming that the multicast packets are IP packets. IGMP snooping is relevant when a receiver on a customer site is attached either directly or via Ethernet switches to the PE

⁶ A workaround that indirectly achieves interprovider operation for LDP-VPLS deployments is to deploy ASBRs that are capable of interworking between LDP-VPLS and BGP-VPLS and to use BGP Interprovider Option E across the inner-AS interconnect. This scheme is described in Section 13.7.

Table 13.1 Comparison of LDP and BGP control plane schemes for VPLS

	LDP	BGP
Control plane sessions	Fully meshed, unless H-VPLS is used	Can use BGP route reflection or confederations to avoid full mesh
SP-controlled multihoming capability	No generic in-built solution. Workarounds are to (i) use H-VPLS or (ii) to use an external protocol (BGP) to signal multihoming information	Yes
Commonality with operational model used for L3VPN	None	High
In-built autodiscovery	No	Yes
Configuration burden of setting up mesh of N Pseudowires	$O(N^2)$, unless external autodiscovery is used	$O(N)$
Interdomain capability	Difficult to achieve	Yes, using schemes analogous to those for L3VPN

router, for example host K in Figure 13.2. PIM snooping is relevant when a receiver on a customer site is ‘behind’ a CE router, for example behind CE8 in Figure 13.2. In this case, the IGMP message from the receiver triggers the CE to send a PIM join upstream towards the source. The following description uses IGMP snooping as an example, but the same principles apply to PIM snooping.

Referring again to Figure 13.2, suppose host A is a multicast source, S, for a particular multicast group G. In order to avoid wasting bandwidth, IGMP snooping can be employed to ensure that PE1 only sends the multicast traffic to PEs that have interested receivers for group G behind them. In principle, each PE could perform the snooping on all IGMP messages arriving on the pseudowires from remote PEs as well as on the local attachment circuits. However, this could result in a large control plane overhead on each PE and duplication of effort as all the PEs in a VPLS would need to process each IGMP message passing across a VPLS instance. For example, an IGMP message originating from host K would need to be snooped by the local PE, PE2 and all the remote PEs (PE1 and PE3 in the example, but in a real network many more PEs could be involved). In order to avoid this, an alternative approach is for a PE only to snoop IGMP messages

arriving on attachment circuits from local customer sites. That PE then communicates to the other PEs in the VPLS, using BGP, the identity of the group (and source in the case of SSM) contained in the IGMP message. For example, if PE2 snoops an IGMP message from host K asking to join group G, PE2 communicates that information to PE1 and PE3 through BGP.⁷ In this way, PE1 knows that PE2 has at least one interested receiver for group G. Similarly, PE1 knows that PE3 does not have an interested receiver for group G. Therefore, PE1 only sends the multicast traffic corresponding to group G to PE2 and not PE3. In turn, PE2 knows through its snooping that only host K is interested in group G, and so only sends the frame towards CE4 and not CE5.

13.5.5 Use of multicast trees in VPLS

In Section 13.4.2, we discussed how some VPLS implementations use multicast trees, in the form of P2MP LSPs, in order for an ingress PE to send broadcast, multicast and unknown unicast frames to other PEs in the network. This allows more efficient use of bandwidth in the service provider part of the network by avoiding ingress replication. Also, if P2MP traffic engineering is being used, the operator has control over the topology of the multicast tree, for example allowing minimum cost trees to be constructed if desired.

The scheme has many similarities with the Next-Generation (NG) multicast L3VPN scheme discussed in Chapter 10, for example:

1. BGP is used as the control plane between PE routers. This allows each PE to communicate to the other PEs the identity of the multicast tree that it will use to forward traffic.
2. Similar options exist in both schemes for the building of multicast trees, such as the use of inclusive trees, selective trees and the ability to create aggregate trees.

Interestingly, it has been pointed out [WC2005] that the ingress replication scheme for dealing with multicast traffic in earlier VPLS implementations works at the opposite limit to the draft-Rosen scheme for dealing with multicast traffic in L3VPNs, in that the former has no multicast state in the core but wastes bandwidth as a consequence of using ingress replication for multicast traffic, whereas the latter is more bandwidth efficient but has a potentially large amount of multicast state in the core. However, there is no fundamental reason why the two types of VPN should be handled in a different way. The introduction of the NG

⁷ More details about the BGP procedures are discussed in Section 13.5.5.3.

multicast scheme for L3VPNs and the introduction of multicast trees for VPLS multicast gives the service provider a common set of tools and range of options to cater for the two cases. For example, one service provider might decide not to have any form of multicast tree in the network and use ingress replication for both L3VPN multicast and VPLS multicast traffic. Another service provider might choose to use inclusive trees for L3VPN multicast and VPLS multicast traffic. Yet another service provider might choose to use selective trees for the two cases, and another service provider could choose to use a different scheme for the two cases.

The use of multicast trees in VPLS [VPLS-MCAST] relies on BGP-based procedures. Those procedures could be used in conjunction with either BGP-signaled VPLS or LDP-signaled VPLS, however for simplicity the descriptions in the following sections assume that BGP-VPLS is being used.⁸

In the following sections, we examine in turn how the various types of multicast tree are used in conjunction with VPLS.

13.5.5.1 *Inclusive trees*

One option is for each PE to build a separate inclusive tree for each VPLS of which it is a member. For a given VPLS, the corresponding tree serves all the other PEs involved in that VPLS. This is illustrated in Figure 13.10.

Only the trees having PE1 as the ingress are shown in the diagram for clarity. PE1 creates one tree having PE3 and PE5 as the egress points for the grey VPLS instance and another tree having PE2, PE3 and PE4 as the egress points for the black VPLS instance. As we saw in section 13.5.2, BGP-VPLS procedures involve each PE sending an autodiscovery message with a BGP-VPLS NLRI containing information such as the VE ID and label block information. In order for each PE to advertise the identity of the inclusive multicast tree that it will use to forward traffic to the other member PEs of a VPLS, a PMSI tunnel attribute (analogous to that used in NG mVPN) is added to the BGP-VPLS autodiscovery message. For example, if PE1 plans to use an RSVP-signaled P2MP LSP having P2MP session object X for the black VPLS, it advertises this information in the PMSI tunnel attribute in its autodiscovery message. If PE1 uses an RSVP P2MP LSP, because this is root-initiated PE1 needs to know which other PEs are members of the black VPLS. It discovers this information from the

⁸ During the course of this chapter, we have mentioned several ways in which BGP-based procedures can be used to augment the functionality of a VPLS in which LDP is used for the pseudowire signaling: BGP-based autodiscovery, BGP-signaling for multihoming and BGP-signaling for multicast operations. This makes it difficult to see the point of using a separate protocol, LDP, for the pseudowire signaling in the first place as BGP can be used for that task too.

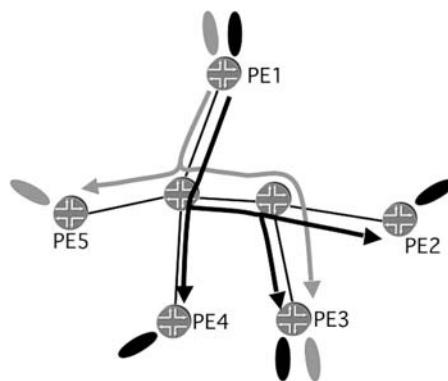


Figure 13.10 Using inclusive multicast trees as VPLS infrastructure

BGP-VPLS autodiscovery message that each of the other PEs generates. In the forwarding plane, PHP is not used. This is so that each egress PE can use the MPLS label on the arriving packet to identify which P2MP LSP the packet has arrived on.

13.5.5.2 Aggregate trees

Another option is for a PE to use the same P2MP LSP to carry traffic from more than one VPLS. This is illustrated in Figure 13.11.

In this case, PE1 is using one P2MP LSP to carry multicast/broadcast/unknown traffic for both the grey and the black VPLS. Compared to the previous example in Figure 13.10, this has the advantage that fewer P2MP

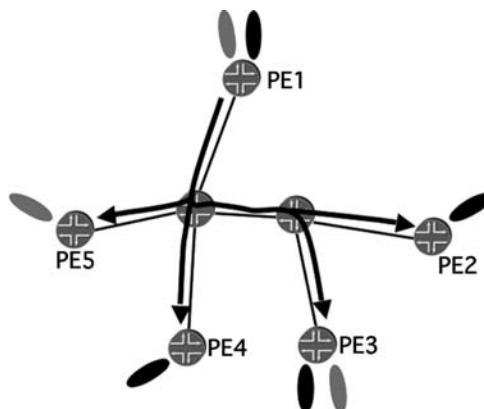


Figure 13.11 Using an aggregate multicast tree as VPLS infrastructure

LSPs need to be built, at the expense of lower bandwidth efficiency. For example, a multicast packet sent by PE1 for the black VPLS would be received by PE5 on the aggregate tree even though PE5 is not involved in the black VPLS. In the case of aggregate trees, an inner label is required, a different label value being used for each VPLS so that each receiving PE can identify which VPLS the packet belongs to. The inner label is assigned by the ingress PE, that is to say upstream label assignment is used. As with inclusive trees, the BGP autodiscovery message generated by each PE contains the PMSI tunnel attribute in order to advertise the type and the identity of the multicast tree being used. PE1 generates separate autodiscovery messages for the black and the grey VPLS, but each of these contains the same tunnel type and tunnel identity information. The PMSI tunnel attribute also contains the upstream-assigned inner label, a different value being advertised in the autodiscovery message for the black VPLS and the grey VPLS. In the forwarding plane, the packet must arrive at the egress PE with both the P2MP LSP label and the inner label. PHP is not used because the egress PE needs to know which P2MP LSP the packet arrived on for the inner label to be meaningful, as different PEs may have coincidentally chosen the same inner label value for different VPLSs.

13.5.5.3 Selective trees

Another option is for an ingress PE to create selective trees to carry a particular multicast group or groups. This is illustrated in Figure 13.12.

Suppose a source behind PE1 is sending a large volume of traffic to a particular multicast group that only has receivers behind PE2 and PE4. PE1 can build a multicast tree to serve that particular multicast group. This is analogous to the use of selective trees in the NG mVPN scheme already discussed in Chapter 10. This is shown by the dotted lines in the figure. As can be seen, the egress PEs are PE2 and PE4 but not PE3, as PE3 does not have an interested receiver for that multicast group. PE1 generates an S-PMSI autodiscovery route in order to advertise to the other PEs in the black VPLS the multicast source and group that uses that selective tree.⁹ The autodiscovery route also includes a PMSI tunnel attribute in order to advertise the tunnel type (e.g. RSVP or LDP) and the identity of the tunnel. PEs can discover the existence of receivers in their local sites by performing IGMP or PIM snooping as discussed in Section 13.5.4. In the case of leaf-initiated tunnels, such as LDP-P2MP, an egress PE can simply splice itself onto the tunnel advertised by PE1 by sending the appropriate LDP message upstream. In the case of root-initiated tunnels (e.g. RSVP-P2MP LSPs), PE1 needs to know which PEs have interested receivers in

⁹ In general, a PE could map multiple multicast groups to the same selective tree if desired.

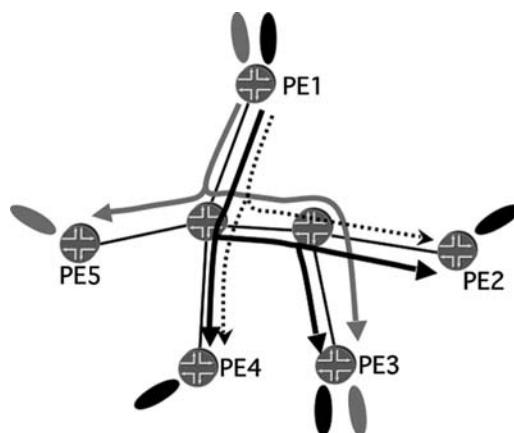


Figure 13.12 Using a selective multicast tree (dashed line) as VPLS infrastructure

their local sites. Those PEs generate leaf autodiscovery routes containing the identity of the snooped multicast streams.¹⁰ In this way, PE1 knows which of the other PEs should be included as leaves of the corresponding selective tree.¹¹

A variation on selective trees is ‘aggregate selective trees’, in which multicast groups from different VPLSs share the same selective tree, for example if there is a large overlap in the set of PEs that are interested in each multicast group. In that case, an inner label needs to be assigned and advertised for each VPLS served by the selective tree, in order for each egress PE to be able to map each received packet to the correct VPLS.

13.5.5.4 Scope of multicast trees

Implementations might allow the user to invoke multicast trees on only some of the PEs in a VPLS, for example if it is known that certain PEs will not be sending significant amounts of multicast or broadcast traffic. This is useful in cases where VPLS is being used as metro Ethernet infrastructure. This is discussed further in Appendix A.

¹⁰ We saw in the previous section that the signaling of inclusive trees is accomplished simply by ‘piggy-backing’ a PMSI tunnel-attribute onto the normal BGP autodiscovery message that is used for BGP-VPLS signaling. In contrast, the S-PMSI autodiscovery routes and leaf autodiscovery routes use a separate address family defined for the purpose called MCAST-VPLS.

¹¹ Similar procedures can be used if an ingress PE uses ingress replication to forward multicast traffic.

In summary, the use of multicast trees in VPLS avoids the bandwidth wastage associated with ingress replication when forwarding broadcast, multicast and unknown traffic. Because of the way the BGP autodiscovery schemes used in VPLS have been modified to deal with multicast trees, the setup of the trees is largely automated so the operational overhead is relatively low.

13.6 LDP AND BGP INTERWORKING FOR VPLS

In earlier sections, we have discussed LDP-based and BGP-based signaling schemes for VPLS. Sometimes there is a need for interworking between these schemes. One example is where a service provider has some PE equipment which only supports LDP signaling for VPLS, which makes it impossible to deploy BGP signaling everywhere.

Let us look at the network in Figure 13.13 as an example. The grey region contains routers that support only LDP signaling for VPLS. The region in white contains routers that support BGP signaling and autodiscovery for VPLS. Router A in the white region supports both BGP and LDP signaling for VPLS. Two customers are shown, customer J whose CE devices are labeled J1, J2, etc. and customer K whose CE devices are labeled K1, K2, etc. As can be seen, both J and K have some sites that are attached to the LDP-VPLS region and others that are attached to the BGP-VPLS region.

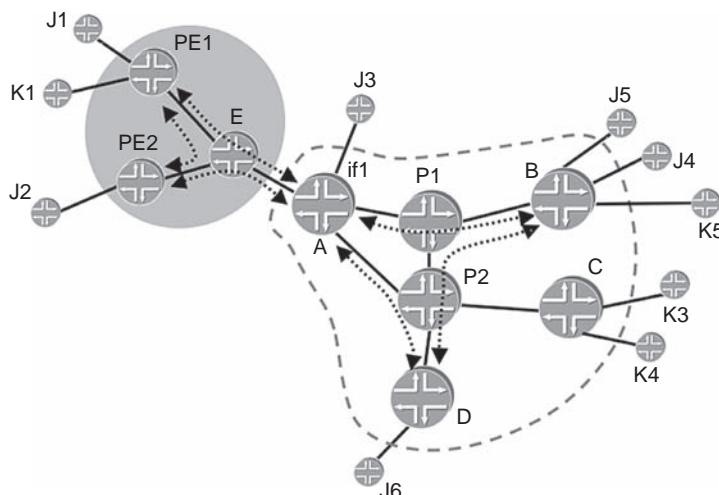


Figure 13.13 Network diagram for LDP-BGP VPLS interworking discussion: one LDP domain and one BGP domain

One method of achieving interworking between the two domains is as follows [INTER]. The equipment in the grey domain simply runs standard LDP-based VPLS signaling. Router A performs an interworking function between the BGP and the LDP domains. The scheme has the following properties:

- No changes are required to the BGP or LDP signaling mechanisms.
- From the VPLS control plane point of view, the PEs in the LDP domain are only aware of the other PEs in that domain and router A which is performing the interworking function.
- Router A maintains a common MAC table per VPLS customer. In order to forward frames between the two domains, router A performs a MAC address lookup on each arriving frame in order to determine the pseudowire on which to forward that frame. This avoids the need to have ‘end-to-end’ pseudowires crossing multiple domains.

Let us see how this scheme is achieved by looking at Figure 13.13 in more detail. In the grey region, routers PE1, PE2 and A are fully meshed with LDP sessions, and there is a pseudowire between each router for each VPLS instance that they are involved in. For example, for customer J, routers PE1, PE2 and A are fully meshed with pseudowires as customer J is attached to both PE1 and PE2. As far as PE1 and PE2 are concerned, A appears to be a ‘normal’ PE performing LDP-based VPLS. From the VPLS point of view, PE1 and PE2 are unaware of the existence of routers in the BGP domain apart from router A. For each VPLS customer, the routers in the core domain run a VPLS instance using BGP-based signaling and autodiscovery. For example, in order to serve customer J, routers A, B and D are configured with a VPLS corresponding to customer J. Through the BGP autodiscovery procedures described in Section 13.5.2, these three routers discover each other as members of customer J’s VPLS and so build pseudowires to each other for that VPLS. The black dotted lines in Figure 13.13 show the pseudowires that are created in the network in order to provide VPLS service to customer J as a result of these procedures. (Note that the pseudowires created to provide VPLS service to customer K are not shown for clarity).

Let us now look at the forwarding plane operations for this inter-working scheme. The key point is that the router performing the inter-working function, router A, maintains one MAC table for each VPLS customer, regardless of where the owner of each MAC address is located. This table is populated as a result of MAC learning operations. For example, for customer J’s VPLS instance, router A maintains one MAC table, with the MAC addresses from all of customer J’s sites. These sites could be in the LDP domain (J1 and J2), directly attached sites (J3) or sites attached to other PEs in the BGP domain (J4, J5 and J6). Similarly, router A

maintains another MAC table containing MAC addresses from all of customer K's sites.

Suppose that J1 sends a frame whose destination MAC address is J5 but none of the service provider routers have learnt that MAC address yet. As far as PE1 is concerned, the other members of customer J's VPLS instance are PE2 and A, so it floods the frame to PE2 and A. PE2 in turn floods the frame to its locally attached site, J2, but not to the other PEs, as per the VPLS split horizon flooding rule described earlier in Section 13.4.1. Let us now examine how router A handles the frame. If router A were following the normal VPLS split horizon flooding rules as described in Section 13.4.1, router A would only flood the frame to the local site J3 and not any of the other PEs in the network. However, this would mean that customer J's sites J4, J5 and J6 would never receive the frame, as their PEs are not meshed with PE1. Therefore, the VPLS split horizon flooding rules are modified in the case of the LDP-BGP interworking scheme by introducing the concept of 'mesh groups'. From the point of view of router A in the context of customer J's VPLS, there are two distinct mesh groups as follows:

1. The routers in the grey LDP domain, namely PE1 and PE2.
2. The routers in the white BGP domain, namely B and D.

Router A must be configured with the fact that for customer J's VPLS, routers PE1 and PE2 are members of one mesh group. Router A already knows that B and D are members of the BGP domain mesh group through BGP autodiscovery.

The split horizon rule is modified as follows: router A does not flood a frame to any member of a mesh group if that frame was received from a member of that mesh group; however, it does flood the frame to members of the other mesh group. Figure 13.14 illustrates router A's perspective of customer J's VPLS, in terms of the logical ports associated with that VPLS.

As a result, router A handles the frame received from J1 as follows:

- Router A must flood the frame to locally attached sites that belong to customer J. It therefore floods the frame to J3.
- Router A must flood the frame to other routers in the BGP domain that are members of the BGP VPLS instance corresponding to customer J. It therefore floods the frame to routers B and D.
- Router A must *not* flood the frame back into the grey LDP domain, as the routers in that domain have already received the frame. Otherwise forwarding loops would occur. For example, PE2 already received the frame from PE1, so A must not flood the frame to PE2. Because PE1 and PE2 are members of the same mesh group, A knows not to flood the frame received from PE1 to PE2.

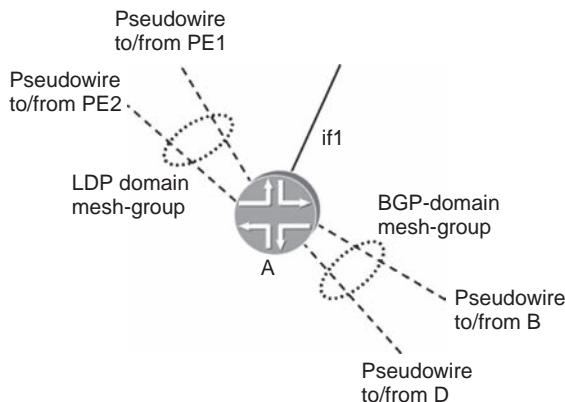


Figure 13.14 Router A's perspective of customer J's VPLS, in network shown in Figure 13.13

In turn, routers B and D continue the flooding operation of the frame originating from J1. D floods the frame to site J6 (but not to B). B floods the frame to J4 and J5. In this way, J5 receives the frame.

As a result of the above process, each VPLS router that the frame has crossed learns the MAC address J1 and updates it in the MAC table accordingly. This means that if subsequently J5 sends a frame to J1, each router knows how to forward the frame in order for it to reach J1 as follows:

1. B performs a MAC lookup and forwards the frame on the pseudowire to A.
2. A performs a MAC lookup and forwards the frame on the pseudowire to PE1.
3. PE1 performs a MAC lookup and forwards the frame on the interface to J1.

The LDP-BGP VPLS interworking scheme can be extended to a scenario where there are multiple ‘islands’ which only support LDP signaling for VPLS and a core which supports BGP signaling and autodiscovery for VPLS. This is illustrated in Figure 13.15.

The grey domains, W, X, Y and Z, are metro regions that only support LDP signaling for VPLS. As in the previous example, BGP signaling and autodiscovery is used in the white domain. This time, routers A, B and C are all performing the BGP-LDP interworking function on behalf of the metro areas to which they are attached – router A for metro areas W and X, router B for metro areas Y and router C for metro area Z.

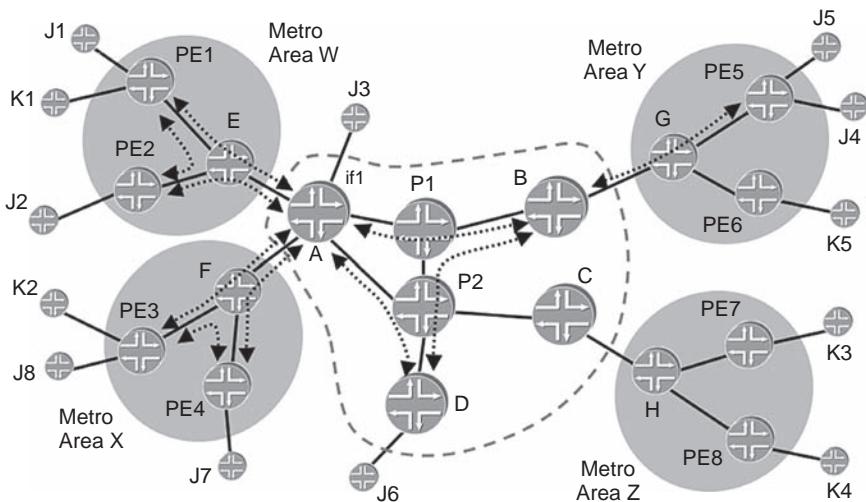


Figure 13.15 Network diagram for LDP-BGP VPLS interworking discussion: multiple LDP domains and one core BGP domain

In each metro area, there is a full mesh of LDP sessions, which includes the interworking router that performs the interworking function for that area. Note, however, that there are no LDP sessions between routers that are in different metro domains. The black dotted lines in Figure 13.15 show the pseudowires that are created in the network in order to provide VPLS service to customer J. (Note that the pseudowires created to provide VPLS service to customer K are not shown for clarity.)

Let us now revisit the example in which J1 sends a frame whose destination MAC address is J5, assuming again that none of the service provider routers have learnt that MAC address yet. As before, as far as PE1 is concerned, the other members of customer J's VPLS instance are PE2 and A, so it floods the frame to PE2 and A. PE2 in turn floods the frame to its locally attached site, J2, but not to the other PEs. This time, from the point of view of router A in the context of customer J's VPLS, there are now three distinct mesh groups as follows:

1. The routers in metro area W, namely PE1 and PE2.
2. The routers in metro area X, namely PE3 and PE4.
3. The routers in the core domain, namely B and D.

Router A must be configured with the fact that for customer J's VPLS, routers PE1 and PE2 are members of one mesh group and routers PE3 and PE4 are members of another mesh group. As in the previous example,

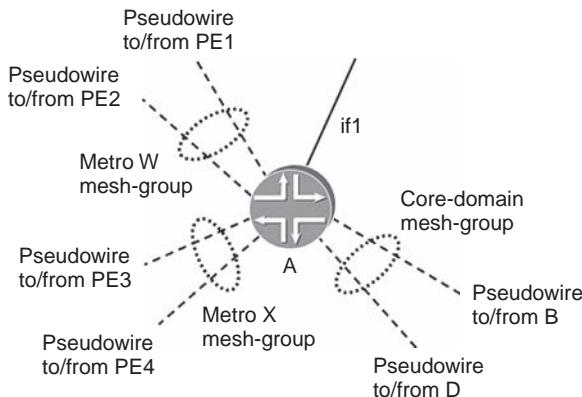


Figure 13.16 Router A’s perspective of customer J’s VPLS, in network shown in Figure 13.15

Router A already knows that B and D are members of the core mesh group through BGP autodiscovery.

Figure 13.16 illustrates router A’s perspective of customer J’s VPLS, in terms of the logical ports associated with that VPLS and their mapping to mesh groups.

As a result, router A handles the frame received from J1 as follows:

- Router A must flood the frame to locally attached sites that belong to customer J. It therefore floods the frame to J3.
- Router A must flood the frame to other routers in the core mesh group that are members of the BGP VPLS instance corresponding to customer J. It therefore floods the frame to routers B and D.
- Router A must flood the frame to other metro domains that it is connected to. Therefore, it floods the frame to PE3 and PE4 in metro area X as they are members of customer J’s VPLS in that metro area.
- Router A must *not* flood the frame back into metro domain W, as the routers in that domain have already received the frame.

In turn, routers B and D continue the flooding operation of the frame originating from J1. D floods the frame to site J6 (but not to B). B floods the frame to PE5 (but not to D). In turn, PE5 floods the frame to J4 and J5. In this way, J5 receives the frame.

As a result of the above process, each VPLS router that the frame has crossed learns the MAC address J1 and updates it in the MAC table accordingly. This means that if subsequently J5 sends a frame to J1, each

router knows how to forward the frame in order for it to reach J1 as follows:

- PE5 performs a MAC lookup and forwards the frame on the pseudowire to B.
- B performs a MAC lookup and forwards the frame on the pseudowire to A.
- A performs a MAC lookup and forwards the frame on the pseudowire to PE1.
- PE1 performs a MAC lookup and forwards the frame on the interface to J1.

As can be seen from the above discussion, the LDP-BGP interworking function involves creating separate meshes of pseudowires in each domain. The routers providing the interworking functions are members of multiple domains. A MAC table on the interworking router provides the glue between the domains, allowing traffic arriving on a pseudowire from one domain to be forwarded on a pseudowire towards another domain. This means that it is important that each interworking router is capable of storing the total number of MAC addresses expected in the VPLSs that it serves.

From the point of view of an interworking router, each attached LDP domain is very similar to a CE site in normal BGP VPLS. As already seen in the examples, one difference is that when an unknown or multicast or broadcast frame is received from a conventional CE site, that frame is flooded on other ports leading to that site. In the case of an LDP domain, when the interworking router receives a frame from a member of the domain, it never floods it to other members of the domain. As far as a remote PE is concerned, an LDP domain is indistinguishable from a conventional CE site. For example, router B is unaware that metro area W exists behind router A. All B needs to know is that the MAC destinations associated with J1, K1, etc. are somewhere behind router A.

In the examples we discussed, each LDP domain was served by only one interworking router. In practice, it is useful to have at least two for resilience. Given the analogy between an LDP domain and a CE site, an LDP domain can be multihomed to more than one interworking router using the same scheme as for PE multihoming described in Section 13.5.2.2. This is illustrated in Figure 13.17, in the context of customer J's VPLS.

There are now two interworking routers serving metro area W, routers A1 and A2. Both A1 and A2 are members of the full mesh of LDP signaling sessions in metro area W. PE1 and PE2 simply view A1 and A2 as conventional PE routers performing LDP signaling for VPLS. As shown in the diagram, the same VE ID is applied to A1 and A2. If B has traffic to forward to a site in metro area W, it applies BGP path selection to choose either A1

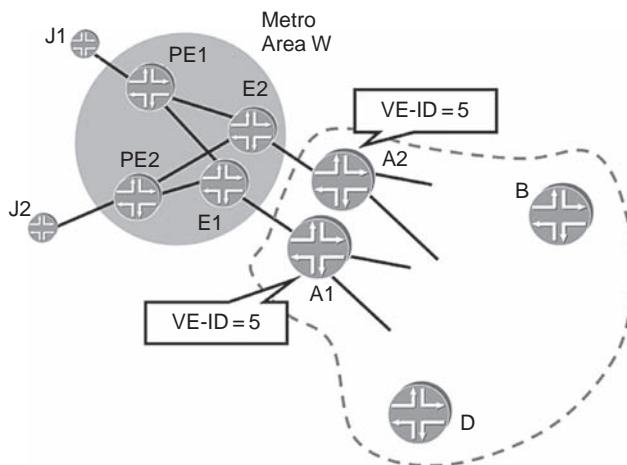


Figure 13.17 Use of BGP VPLS multihoming to provide interworking router redundancy

or A2 as the exit point to that domain. If A1 is the normally selected exit point and A1, or the link to E1, or E1 itself should fail, then B would fail over to A2 as the exit point.

The general concept of using a MAC table as the glue between VPLS domains has applicability in other scenarios apart from those discussed so far. For example, it can be used as a method to achieve inter-AS operation for LDP-signaled VPLS, by having the MAC table on the ASBRs. This avoids the need to fully mesh the PE routers with LDP sessions and pseudowires across the ASs.

13.7 INTERPROVIDER OPTION E FOR VPLS

In Section 13.5.2.1, we discussed Interprovider Options A, B and C for VPLS. We saw that these options are analogous to Options A, B and C for L3VPNs. In this section, we describe Interprovider Option E for VPLS. This option is unique to VPLS because it leverages the MAC table and mesh-group concepts introduced in the previous section, extending them to Interprovider scenarios.

Figure 13.18 shows an example scenario. Customer J's VPLS service extends across AS1, AS2 and AS3. To the PEs in each AS, the ASBR is indistinguishable from a normal PE, and the ASBR and PEs are fully meshed with pseudowires corresponding to customer J's VPLS service. Within each AS, either LDP or BGP signaling can be used, irrespective of what signaling is used in each of the other ASs.

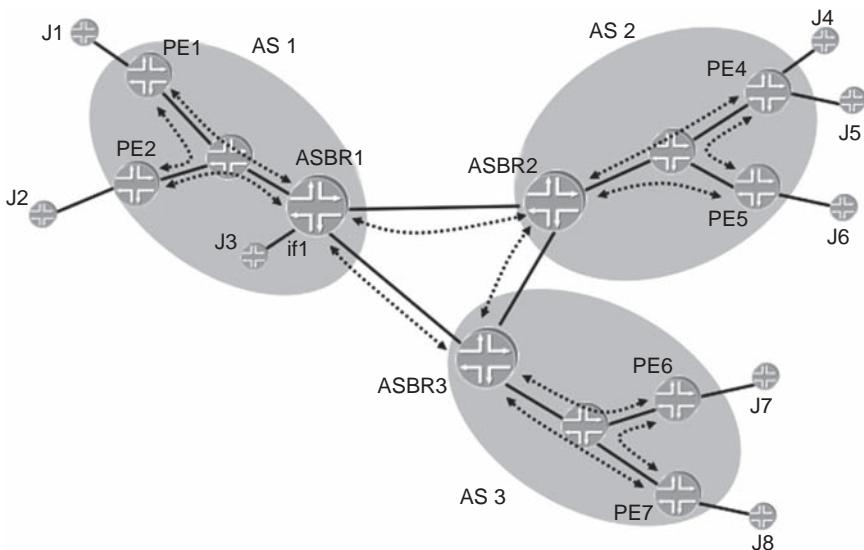


Figure 13.18 Network diagram for VPLS Interprovider Option E discussion

The ASBRs are also fully meshed with pseudowires corresponding to customer J's VPLS. These are created by having a full mesh of eBGP sessions between the ASBRs in order to exchange BGP VPLS NLRIs. Each ASBR advertises a (different) VE ID corresponding to customer J's VPLS on these sessions. As a result, as far as each ASBR is concerned, each of the other ASBRs appears to be BGP VPLS PE with one of customer J's sites attached. This means each ASBR does not have (or need) awareness of the detail of how many of customer J's sites exist in each of the other ASs.

Each ASBR maintains a MAC table for customers J's VPLS with the use of mesh-groups to define the scope of flooding and learning operations. This is very similar to the scheme discussed in the previous section and follows the same rules – a frame received from a member of a mesh-group is never flooded to other members of that mesh-group. Figure 13.19 shows the mesh-groups from the point of view of ASBR1 in the context of customer J's VPLS. As can be seen, ASBR2 and ASBR3 are in one mesh-group, and the PEs involved in customer J's VPLS in AS1, PE1 and PE2, are in another mesh-group. Having ASBR2 and ASBR3 in the same mesh-group is important in order to avoid forwarding loops among the ASBRs. For example, this ensures that if ASBR2 floods a frame to ASBR1 and ASBR3, then ASBR1 does not flood it to ASBR3. This is the reason why all the ASBRs participating in this interprovider scheme for a given VPLS must be fully meshed.

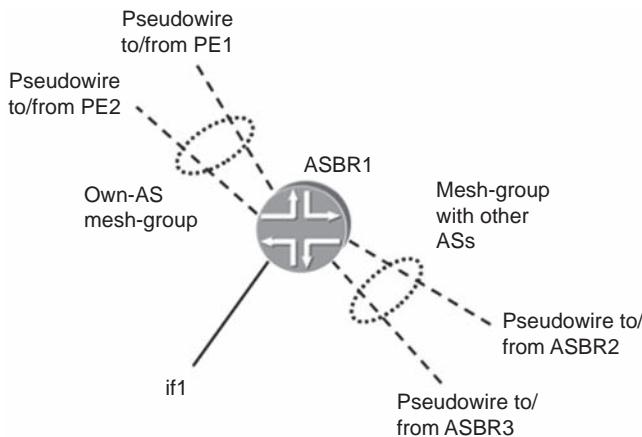


Figure 13.19 Mesh-groups from the point of view of ASBR1

Option E is particularly useful in situations where one or more ASs use LDP VPLS and contain PEs that do not support BGP-VPLS. By deploying ASBRs that support BGP-VPLS (and Option E), interprovider operation can be achieved without having to replace all of the PEs in an AS.

13.7.1 Comparison of interprovider schemes for VPLS

Table 13.2 compares the interprovider options for VPLS. In both inter-provider Option A and Option E, each ASBR contains the sum of the MAC address of all the VPLSs. Both these options require some specific configuration for each VPLS. However, Option A involves more configuration effort as a separate VLAN is required on each interconnect for each VPLS. Option A requires an Ethernet interconnect between the ASBRs – in all of the other options, the interconnect can be of any media type because MPLS is used to forward traffic between the ASBRs. In Option B, the ASBR has no MAC address state but instead has MPLS forwarding state (label swap operations) proportional to the sum of the sizes of the label blocks advertised by the PEs. In principle, Option B requires no specific configuration for each VPLS and so no incremental ASBR configuration is required if a new VPLS requiring interprovider connectivity is deployed. However, in practice, the operator may have route filters to only allow selected VPLSs to be advertised to the peer AS. If so, each time a new VPLS is deployed that requires interprovider service, the filters need to be configured accordingly on the ASBRs. Option C needs the least amount of state on the ASBRs because they only need label state corresponding to the loopback

Table 13.2 Comparison of interprovider options for VPLS

	Option A	Option B	Option C	Option E
Forwarding state at the ASBR	MAC address state	Label block translations	Label per PE	MAC address state
Per-VPLS configuration on ASBR	Yes	No ¹²	No	Yes
Link type between ASBRs	Ethernet	Any	Any	Any
MPLS across interconnect	No	Yes	Yes	Yes
Loop avoidance	STP needed in some topologies	Yes	Yes	Need to fully mesh ASBRs

address of each of the PEs. Also no incremental configuration is required to deploy additional VPLSs requiring interprovider connectivity.

13.8 OPERATIONAL CONSIDERATIONS FOR VPLS

Let us consider some of the operational issues that occur when running VPLS services. Because of the nature of the VPLS service and the way it is more entwined with the customer's network than other VPN services, some of the operational issues discussed below do not have analogies in the L3VPN or L2VPN services. A consideration to bear in mind is that having a VPLS service does not allow the enterprise customer to exceed the best common practices with regard to the scope of a LAN, e.g. in terms of the number of attached hosts. The same scaling issues would occur as with a traditional LAN, where the amount of broadcast traffic becomes excessive if too many hosts are attached.

13.8.1 Number of MAC addresses per customer

A consideration for the service provider is the number of MAC addresses to be stored by each PE, bearing in mind that a PE might be providing a VPLS service to a large number of customers. Although implementations exist in which the number of MAC addresses stored can be large,

¹²In practice, some per-VPLS configuration may be required, see text for details.

there is always an upper limit. The service provider may need to protect themselves against an exhaustion of MAC address capacity by limiting the number of MAC addresses that are stored for each VPLS customer. VPLS implementations exist that allow the service provider to limit the number of MAC addresses on a per-VPLS basis or on a per-interface basis. This is by analogy with some L3VPN implementations that allow the service provider to limit the number of L3VPN routes in a VRF. Being able to control the number of MAC addresses also opens up interesting billing opportunities for the service provider where the customer is billed according to the MAC address limit that they choose to purchase.

13.8.2 Limiting broadcast and multicast traffic

Another operational consideration for the service provider is to consider limiting the volume of broadcast and multicast traffic, bearing in mind that the cost to the service provider of sending such traffic could be high, especially in cases where ingress replication is used and there are a large number of PE members in a VPLS. As a consequence, some VPLS implementations allow the service provider to rate-limit this type of traffic.

13.8.3 Policing of VPLS traffic

If the VPLS service is delivered over 100 Mbps or 1 Gbps native Ethernet ports, service providers may need to police the amount of traffic that the customer sends into their network on each access port. The service provider can offer tiered services in terms of the amount of traffic that the customer is allowed to send, by analogy with many existing L3VPN services.

13.8.4 VPLS with Integrated Routing and Bridging (IRB)

It is often useful for the VPLS PE router to act as the default gateway to a Layer 3 domain for hosts in the local site. This can be achieved by having an IRB interface within the VPLS instance. Frames arriving at the VPLS PE with a MAC address matching that of the default gateway are directed to the main routing instance or a VRF for Layer 3 route look-up.

13.8.5 Learning mode

Some implementations allow the user to choose the scope of flooding and MAC learning within a VPLS instance when multiple VLANs are present.

13.8.5.1 Qualified learning

Let's suppose that VLANs 100, 200 and 300 are coupled into a VPLS instance at each site. If *qualified learning* is in use, frames arriving at VLAN 100 are only ever flooded to VLAN 100, and the MAC learning is in the context of VLAN 100. This mode is convenient if the various VLANs have the same sites in common as they can use the same VPLS instance whilst maintaining separation between VLANs, rather than having to configure a separate VPLS instance for each VLAN. Some implementations allow additional VLANs to be introduced from a customer site without requiring extra configuration on the PE.

13.8.5.2 Non-Qualified learning

With non-qualified learning, the scope of MAC learning and flooding is irrespective of the VLAN ID. Suppose in a VPLS that VLANs 100, 200 and 300 exist at various sites. Frames arriving on VLAN 100 are flooded on VLAN 200 and 300 as well as VLAN 100. Non-qualified learning is useful in situations where different sites in a VPLS have different VLAN IDs for historical reasons – perhaps each site used to be an isolated island and it was only decided to join them together using VPLS at a later date. Using non-qualified learning avoids having to renumber the VLAN IDs to a common value.

13.9 CONCLUSION

In this chapter we have explored the Virtual Private LAN Service (VPLS). VPLS is a valuable addition to a service provider's product portfolio. Because the service is simple for the customer to deploy, the service provider can address a wider range of customers than if they only offered L3VPN and point-to-point Layer 2 services.

Another use of VPLS is an internal infrastructure within a service provider, especially as part of an MPLS-based access network infrastructure for residential broadband. This is discussed in more detail in Appendix A.

At the time of writing, VPLS was becoming popular in enterprise networks as a means of interconnecting data centers in different locations. This is discussed further in the Conclusions chapter.

This chapter concludes the exploration of services enabled by MPLS. Because of the mission-critical nature of some of the traffic carried by these services, the ability to manage and troubleshoot the underlying network infrastructure is important. This is the subject of the next chapter in this book.

13.10 REFERENCES

- [BGP-AUTO] E. Rosen, W. Luo, B. Davie, V. Radoaca, *Provisioning, Autodiscovery, and Signaling in L2VPNs*, draft-ietf-l2vpn-signaling-08.txt (work in progress)
- [GFP] Generic Framing Procedure, ITU-T Recommendation G.7041, 2001
- [INTER] Whitepaper entitled *LDP-BGP VPLS Inter-working*, www.juniper.net
- [MULTIHOME] B. Kothari, K. Kompella, W. Henderickx, F. Balus, J. Uttaro, *BGP Based Multi-homing in Virtual Private LAN Service*, draft-ietf-l2vpn-vpls-multihoming-00.txt (work in progress)
- [RFC1490] T. Bradley, C. Brown and A. Malis, *Multiprotocol Interconnect over Frame Relay*, RFC 1490, July 1993
- [RFC2684] D. Grossman and J. Heinanen, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*, RFC 2684, September 1999
- [RFC4761] K. Kompella and Y. Rekhter (eds), *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*, RFC 4761, January 2007
- [RFC4762] M. Lasserre and V. Kompella (eds), *Virtual Private LAN Services (VPLS) Using Label Distribution Protocol (LDP) Signaling*, RFC 4762, January 2007
- [VPLS-MCAST] R. Aggarwal (ed), Y. Kamite and L. Fang, *Multicast in VPLS*, draft-ietf-l2vpn-vpls-mcast-06.txt (work in progress)
- [WC2005] Y. Rekhter, Paper D1-02, in MPLS World Congress, Paris, February 2005

13.11 STUDY QUESTIONS

1. Describe some methods of improving bandwidth efficiency when forwarding multicast traffic in VPLS.
2. Referring to Figure 13.2, assume that router PE3 has learnt all the MAC addresses of the equipment in customer Y's sites. Work out what router PE3's MAC table looks like for customer Y's VPLS.
3. Referring to Figure 13.13, assume that router A has learnt all the MAC addresses of the routers in customer J's sites. Work out what router A's MAC table looks like for customer J's VPLS.

4. Describe some of the advantages of using an autodiscovery scheme for VPLS.
5. Suppose we have a VPLS for which the PEs use BGP signaling and autodiscovery. There are six PEs, which have VE IDs numbered from 1 to 6 inclusive and there are no multihomed customer sites. Suppose PE1 advertises a label-base of 200 000, a VE block offset of 1 and a VE-block size of 8. Assume that PE1 has VE ID 3. What pseudowire label value is PE1 expecting each remote site to use?

Part Three

14

Advanced Protection and Restoration: Protecting the Service

14.1 INTRODUCTION

Chapter 3, Protection and Restoration in MPLS Networks, described the mechanisms available for providing 50 ms recovery for link and node failures on an LSP. In Chapter 6, we saw how these mechanisms can be extended for P2MP LSPs. In this chapter, we look at various services offered over MPLS networks and see how similar recovery times can be achieved at the service level. We pay particular attention to failures at the egress points of an LSP, because they are not covered by the FRR mechanisms described so far. We also look at various methods for protecting the service, and describe a novel approach for LSP tail-end protection.

Familiarity with MPLS FRR (Chapter 3), LSP nesting and stitching (Chapter 5) and VPNs (Chapters 7 and 8) is assumed for understanding this chapter.

14.2 THE BUSINESS DRIVERS

The first question to ask is why would 50 ms recovery times be required for MPLS services. To answer this, remember that the requirement for 50 ms

failure recovery [GR253-CORE] has its origins in the technologies used for transport.¹ The MPLS applications that we discussed in the previous chapters, such as pseudowires and VPNs, give the customer the equivalent of a physical transport layer, using a virtual infrastructure. Because the applications that run in the customer's context may have very stringent requirements in terms of loss, it is natural that the customer expects the same 50 ms failure recovery regardless of whether the infrastructure is physical or virtual.

The second question is why we even need to examine this problem, given the FRR capabilities of MPLS. The answer is because FRR provides fast recovery for failures of the LSP at the transit, but does not address the failure of the ingress or egress PE or of the PE-CE links connected to them. Having been sold the abstraction of a private physical network, the customer expects the same fast recovery times, regardless of where the failure happens. Regardless of the type of transport technology used, a way to achieve zero or near-zero impact on traffic is to send the data twice over diverse paths (Live-Live protection scheme) and to select one copy at the receiving end at the application layer, as discussed in Chapter 6. However, this requires specialized equipment that is able to seamlessly switch between feeds should one feed fail and thus uses double the network resources. Hence, the scheme is used only for the most stringent applications. The traditional method of dual-homing CEs for redundancy addresses very well the case of the ingress PE failure. However, dual-homing does not provide adequate recovery times when the failure is at the remote site, because of the state propagation required by the current mechanisms (as we see in the following sections). Thus, the problem of service protection is reduced to solving the generic LSP tail-end protection problem.

Having convinced ourselves that MPLS service protection is essential for MPLS services to successfully replace traditional transport, it is important to note that whatever mechanism is chosen to implement it needs to provide the protection at scale. As we see in Chapter 16, which discusses MPLS in Access Networks, MPLS is emerging as the preferred technology for metro and access networks, resulting in a much larger scale of MPLS deployments. The approach for scaling such deployments, as is discussed in detail in Chapter 16, is to partition the network into regions and to use inter-region transport LSPs consisting of multiple intra-region segments to provide the transport. Recall from Chapter 5, 'Inter-domain traffic engineering' that the basic tool for providing scalable inter-region LSPs is to use LSP stitching and nesting. In this setup, the segment's tail end

¹ SONET and SDH standards state a maximum switching time of 50 ms for certain protection schemes. Note that this time is counted from when the switch is initiated, so does not include the time taken to detect the failure.

becomes the Achilles' heel of the solution, because a failure of the stitching point will cause failure of the entire LSP.

In this chapter, we focus on service recovery in general and tail-end protection in particular. We start by looking at the various failure scenarios, then examine existing schemes for protection and their shortcomings, and lastly describe new schemes for achieving fast service restoration.

14.3 FAILURE SCENARIOS

The discussion of service protection is complicated by the fact that there are many combinations of CE connectivity models, failure modes, service characteristics, and control over initiating the recovery, yielding a large number of solutions to explore. Some of these work only for particular service types, failures or deployments, as we see in the next section, in which we describe some of the existing solutions. Let us start by listing the various options:

- *CE connectivity models.* Figure 14.1 shows a set of CEs and PEs that have different levels of connectivity redundancy. CE3 has no redundancy at all, connecting to PE3 via a single link. CE4 has link-level redundancy, being connected to its PE, PE4, via two links. CE1 has PE-level redundancy, being multihomed to PE1 and PE2. PE2 has link-level redundancy, being connected to its CE, CE2, via two links.
- *Failure modes.* Continuing with the example in Figure 14.1, failures can occur in the MPLS core or at the edges. For the purpose of the discussion in this chapter, failures in the MPLS core, either of the P routers or of the links connecting these routers, are not interesting, because they are

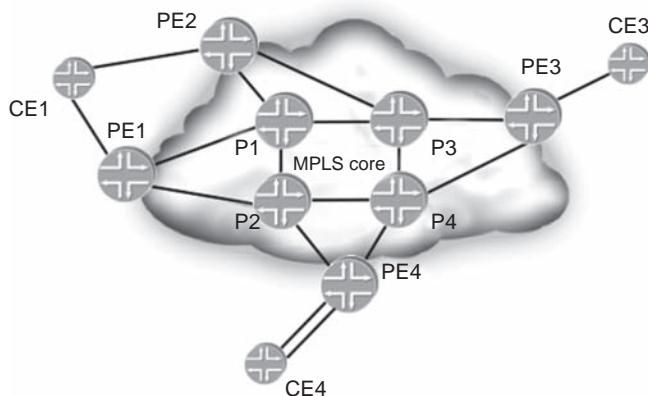


Figure 14.1 CE connectivity options

covered by the MPLS FRR mechanisms discussed in Chapter 3. What we focus on here are failures at the edges, which can be either failure of the PE or of the PE-CE link.² For PE failures, we further distinguish between ‘soft’ and ‘hard’ failures. A soft failure can be a crash of the control-plane software that does not impact the forwarding plane, while a hard one is a non-recoverable event such as a hardware malfunction or power outage.

- *Characteristics of the service.* In general, Layer 2 and Layer 3 services are discussed separately, because they have different characteristics and limitations. An example of such characteristics is a Layer 2 service emulating Ethernet that can leverage Ethernet OAM procedures in its restoration solution in a way that a Layer 3 service cannot. An example of a limitation is the requirement for blocking one of the attachment circuits to prevent loops in VPLS scenarios, described in Section 13.5.2.2 in Chapter 13.
- *Control over initiating the recovery.* Assuming the failure happens at the ‘local’ end of the service, the recovery can be initiated at the remote CE, remote PE or, as we see in one of the local protection schemes discussed in Section 14.5, at the local PE or some node close to the local PE.

Having discussed the different options, let us start looking into the existing solutions.

14.4 EXISTING SOLUTIONS

The discussion about existing solutions makes two fundamental distinctions, the first, between single and dual-homed CEs and the second between Layer 2 and Layer 3 services.

14.4.1 Single homed CE

CE3 and CE4 in Figure 14.1 are examples of single homed CEs, connected to their respective PEs with a single link in the case of CE3 and with multiple links in the case of CE4. Let us discuss link and node failures separately. Obviously, no recovery is possible if a failure occurs in the single-link case (for example, if the link CE3-PE3 breaks), but such a failure would not pose a problem in the multi-link case, because there is redundancy at the link level. At first look, it would seem that the case of a single-homed CE offers no hope of protecting the service in case of a PE failure. However, this may

² CE failures are not interesting in this context, since the CE is the service start/end point.

not necessarily always be true. When the PE failure is due to a software or hardware problem with the control plane, the new non-stop-routing (NSR)³ technologies that equipment vendors have incorporated in their products in the last few years can provide seamless recovery. The idea behind NSR is for a router to have two control planes (routing engines), the primary and the backup. The state is replicated through proprietary mechanisms from the primary to the backup such that the backup is an exact mirror image on the primary and can take over seamlessly in the event of a control plane software crash or hardware failure, without the network ever realizing that such a switch has happened. Note that this technology differs significantly from the graceful restart approach, in which the network neighbors realize a failure has occurred, but maintain the protocol relationships with the failed node for a period of time, rather than tearing them down immediately. Forwarding traffic continues based on stale information as the control plane rebuilds its state with help from the network neighbors. With NSR, the forwarding information is always up to date, and there is no need for cooperation from other network nodes. The NSR technology provides an attractive option to multihoming in an environment in which the main optimization criterion is cost. It is important to bear in mind that NSR only protects against a failure of the control plane, and for this reason, it is in itself not a complete solution. In the rest of this chapter, we do not discuss NSR as a fast restoration mechanism, but rather focus on solutions relying on building redundancy in the network itself.

14.4.2 Dual-homed CE

Let us start the discussion on dual-homed CEs by looking at Layer 2. In Chapter 12, we presented the concept of multihoming for LDP-signalled pseudowires, which uses redundant pseudowires to achieve the necessary protection, as illustrated in Figure 14.2. CE1 is dual-homed to PE1 and PE2. The primary pseudowire from remote end PE3 ends on PE1, and the backup ends on PE2. The concept is equivalent to the path protection scheme for LSP presented in Chapter 3. An alternate path, set up with the required bandwidth guarantees, is available and can be used once the failure is detected. The speed of the recovery depends on the following factors:

- *Speed of the failure detection.* This can be done either through control plane messages (as described in Chapter 12) or by running OAM, either

³NSR is an often overloaded term, because different vendors use the term rather loosely. In this text, NSR refers to the scheme in which the backup routing engine contains a mirror image of the state of the primary routing engine and can take over seamlessly from the primary.

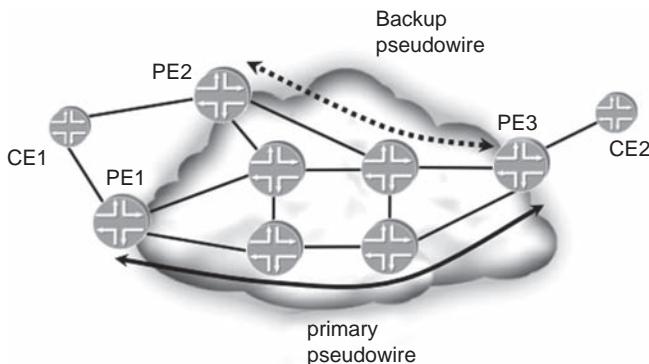


Figure 14.2 Pseudowire redundancy

CE-CE or PE-PE.⁴ An interesting application is the use of Ethernet OAM in cases in which the attachment circuits are Ethernet, and which offers the option to switch to the protection path when there is degradation in the service, not just interruption. For the scheme to work well, OAM runs on both the working and the protect paths.

- *Availability of the backup pseudowire.* The backup pseudowire should be pre-signaled and its forwarding state pre-installed in order to avoid any delays in signaling or in downloading forwarding state. Pre-signaling the backup pseudowire means that resources have to be reserved in the network, which would otherwise be available for other services.⁵

Because of these factors, the above scheme may not be ideal, first because it may not provide the failure detection fast enough, and second because of the extra resources that must be reserved in the network. When BGP is used for setting up L2VPNs, the multihoming scheme is very similar to the L3VPN case and relies on BGP path selection. Because of this similarity, we take a look at a Layer 3 example for describing the existing solutions for a BGP-based service.

For discussing the existing mechanisms to protect Layer 3 we use a simplified topology for a basic VPN scenario, shown in Figure 14.3. There is a single VPN, VPN grey, with two sites, 1 and 2, and LDP is the protocol for setting up inter-PE tunnels. We focus on traffic flowing from site 2 towards site 1. Only site 1 is shown as dual-homed in the figure, to ease the discussion, although in a real deployment all remote sites would be dual-homed. The arrows represent control plane advertisements. Both PE1 and PE2 advertise 10.1/16 as a VPN route, and each of them also advertises

⁴ Depending on the PE-CE connectivity model.

⁵ Assuming bandwidth reservations are used.

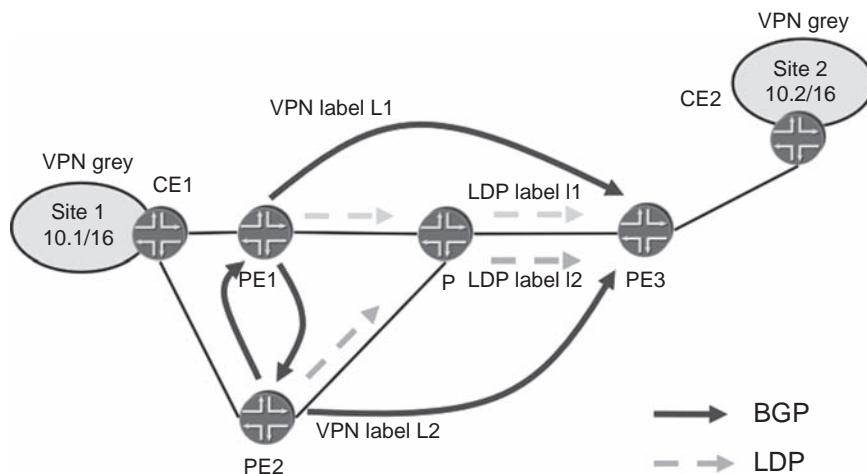


Figure 14.3 Simplified VPN topology and control plane advertisements

the PE's loopback in LDP. As a result, PE3 has LDP LSPs to PE1 and PE2 and has two BGP routes for 10.1/16. PE3 performs path selection, and assuming it chooses PE1 as the more preferred path, it installs forwarding state corresponding to the VPN label received from PE1, L1, and the LDP transport tunnel to PE1, l1, as shown in Figure 14.4. Although the alternate

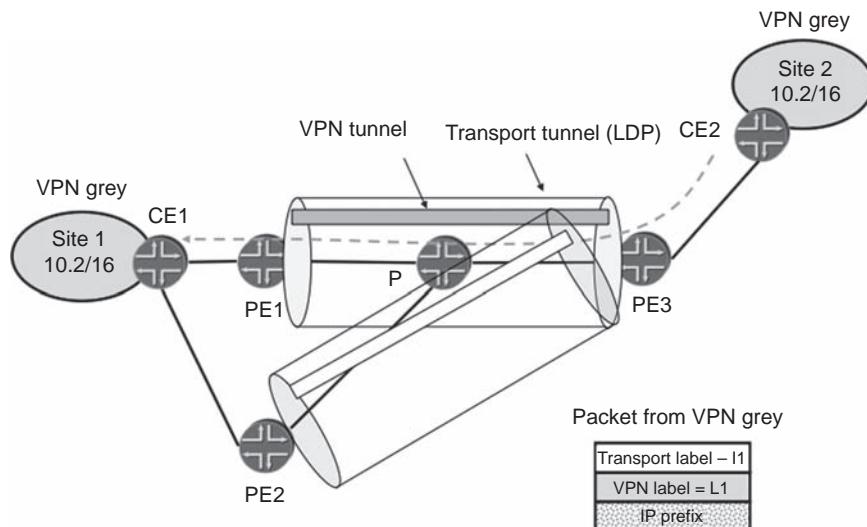


Figure 14.4 VPN example, final forwarding state

path through PE2 is available, it remains unused on the remote PE. The dashed arrow shows the path that the traffic takes.

Let us now look at what happens when either PE1 or the link PE1-CE1 fails. These are both cases of failure at the LSP end point, and the current solution for restoring the traffic is through the action of the ingress router, PE3. To recover from a failure at PE1, PE3 must send the traffic on the LSP to PE2 so that it continues to reach destination CE1. For that to happen, the following events take place:

1. PE3 learns via a routing protocol that the destination CE1 is no longer reachable via PE1. For example, if PE1 itself fails, PE3 learns of the failure via the IGP or via the failure of the LSP to PE1. If the link PE1-CE1 fails, PE3 learns of the failure via BGP (through a route withdrawal).
2. Assuming that PE3 knows that PE2 is an alternate exit point towards the destination CE1, as soon as it learns about the failure, it can reprogram its forwarding plane to send traffic using the VPN tunnel and transport tunnel appropriate for PE2. This is the case in the simple scenario described in the previous section, but may not always be the case in practice. If PE3 is not aware of the alternate exit point, it will blackhole the traffic until it learns about it. Let us take a look at two scenarios in which blackholing can happen.

14.4.2.1 *Blackholing because of route reflector behavior*

Blackholing can happen if route reflectors are deployed and if both PE1 and PE2 use the same RD. Recall from Chapter 7 that there is no restriction regarding choosing an RD, and it is quite possible to choose such a configuration – in fact, some vendors used to recommend to their customers using the same RD per VPN.⁶ In this case, both PE1 and PE2 send their advertisements to the route reflectors. Default route reflector behavior is to advertise only the best path to its route reflector clients, rather than all paths. In the Layer 3 VPN case, if PE1 and PE2 are using different RDs, the advertisements are considered different prefixes by a route reflector, and therefore are both reflected. As a result, PE3 is aware of both exit points from the network, just as described in Figure 14.3. However, if the prefixes have the same RD, they are treated as being equal, and the route reflector selects the best path from its point of view and reflects only that path, for example, the path to PE1. As a result, PE3 is aware of only a single exit point, PE1. A solution to this problem is described in [ADD-PATH],

⁶Such deployments are problematic in terms of convergence and load balancing and for this reason other vendors always recommend deploying unique RDs.

which proposes changes to BGP that allow a route reflector to advertise multiple paths to the same prefix. In this way, if the route reflector and its clients support this new behavior, the reflector could reflect both versions of this path, one with PE1 as the next hop and the other with PE2 as the next hop.

14.4.2.2 Blackholing because of route preference

Another interesting scenario in which the ingress is not aware of the alternate egress is illustrated in Figure 14.5. In this scenario, the ingress never receives the advertisement for an alternate because it was never sent. Suppose that R2 and R3 each set a local preference on prefix X and that the local preference set by R2 is higher (more favored) than the local preference set by R3. In most BGP implementations, the default behavior is that R3 does not send an advertisement for that prefix to its IBGP peers, because it sees the advertisement originating from R2 with the higher local preference, so it knows that R2 is the preferred exit point towards destination X rather than itself. However, this default behavior can have a detrimental impact on convergence times. This is because R1 does not know in advance that R3 is an alternative exit point to reach that prefix. R3 advertises the prefix only when the version from R2 is withdrawn. Some BGP implementations allow this default behavior to be over-ridden through configuration. This allows R3 to advertise the prefix to its IBGP peers even while R2 is advertising it with a better local preference [BEST-EXT].

To summarize, because it is essential for the ingress to be aware of the alternate exit point, solutions have been developed to address scenarios in which additional control plane messages have to be exchanged after the

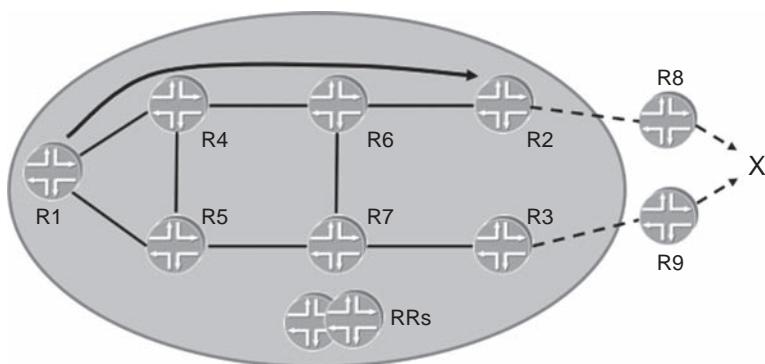


Figure 14.5 Topology for showing how an ingress may not be aware of an alternate egress

failure, to detect the alternates. In the next section, we analyze the existing dual-homing solution.

14.4.3 Analyzing existing dual-homing solutions

Going back to the discussion at the beginning of the previous section, regarding the actions that must happen when a failure happens at the remote point, we can see that in Step 2, it is possible for the ingress to be aware of multiple egress points to the destination. Thus, as soon as it learns of the reachability failure of the currently active exit point, the ingress can start forwarding traffic on the LSP to the alternative exit point. Nevertheless, even if the ingress reacts rapidly once it learns of the failure, the time it takes to learn of the failure in the first place sets a limit on how short the convergence time can be. As we have seen, the length of this time is governed by the generation of routing protocol messages and their propagation. If the IGP is involved, the messages are processed and flooded on a hop-by-hop basis, involving the control planes of all the routers in the path. In the BGP case, the updates may pass through a route reflector, in which case the control plane of at least three routers – the egress, the route reflector and the ingress – play a part in the chain of events required to restore the traffic. The total time taken for such chains of events is typically on the order of a few hundred milliseconds, much longer than the expected 50 ms. Finally, if the appropriate forwarding state is not pre-installed in the forwarding plane, downloading it and starting to forward using it takes additional time. This time can be significant when hundreds of thousands of entries need to be updated.

In the recovery scheme described above, the dominant component in the delay is the time it takes to propagate the failure information, also known as the routing convergence time. For this reason, many vendors equate service high availability with fast routing convergence and focus on speeding up the reaction to and propagation of the failure information. However, the reality is that to restore the service, it is the connectivity that needs to be restored, not the routing state. In fact, to guarantee consistent repair times regardless of the load on the network, it is essential that the solution not rely on the control plane at all. Recall from Chapter 3, ‘Protection and Restoration’, that local repair techniques result in very low impact on traffic, on the order of a handful of milliseconds or so, because the repair is performed closest to the point of failure and without the need for any propagation of routing protocol messages. Finally, to remove the dependency on the time it takes to install the forwarding state, the repair path must be pre-programmed in the forwarding plane. In the next section, we discuss how such local repair techniques can be extended to cater for failures at a network exit point.

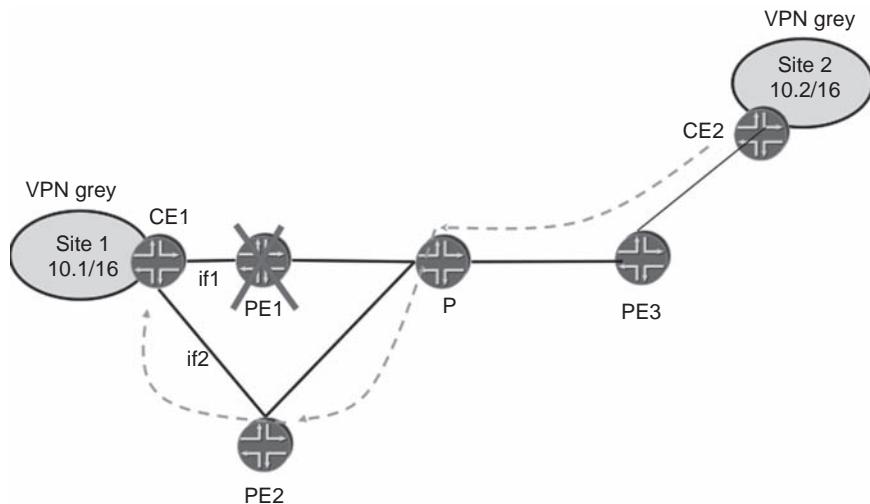


Figure 14.6 Conceptual view of local protection for tail-end failure

14.5 PROTECTING THE EGRESS – LOCAL PROTECTION SOLUTION

To illustrate the main idea behind local protection, let us go back to the simple example of Figure 14.4 and walk through the failure scenario of a crash of PE1, assuming that PE3 has received advertisements from both PE1 and PE2 and that it has established transport tunnels to both. Remember that traffic flows from site 2 towards site 1. In current implementations, PE3 finds out about the crash of PE1 through a routing protocol update. As a result, PE3 reruns its path selection in BGP, and determines that the route from PE2 is now the best, and updates the forwarding state for the route to 10.1/16 in the grey VRF. Until this is done, traffic continues to be sent towards PE1 and gets dropped at router P, the router closest to the point of failure.

The central idea of the tail-end protection solution is to provide local protection at P, the point of local repair (PLR), which is the router closest to the point of failure,⁷ while the ingress router PE3 takes its time converging. This situation is shown conceptually in Figure 14.6: traffic continues to flow from PE3 to PE1 on the same path it would have taken before the failure, but it is intercepted at router P and instead of being forwarded to site 1 via

⁷For the sake of clarity, the discussion so far has been limited to a failure of the PE itself. When the PE-CE link fails, the router closest to the point of failure is the PE router, and it is responsible for forwarding the data to the alternate PE.

PE1, it is now forwarded to site 1 via PE2. There are two requirements for accomplishing this:

1. The PLR, P, has to intercept the packets destined to PE1 and forward them towards the alternate exit point, PE2. To do so, router P must know that PE2 is an alternative to PE1 for reaching site 1 of VPN grey and must have a detour towards router PE2.
2. PE2 has to forward these redirected packets towards site 1 of VPN grey. There are two challenges for router PE2:
 - (a) PE2 has to construct the forwarding state that allows it to successfully forward to site 1 of VPN grey labeled traffic arriving with labels assigned by PE1. To do so, PE2 must be aware of the labels that were assigned by PE1 for destinations in site 1 of VPN grey.
 - (b) PE2 has to be able to determine which labeled traffic that was redirected to PE2 by router P actually belongs to site 1 of VPN grey. This is not a straightforward matter of just looking at the VPN label, because the assignment of VPN labels is done independently at PE1 and PE2, and as a result PE2 may assign to some other VPN site the same label as PE1 assigns to site 1 of VPN grey. To avoid this ambiguity, the lookup on PE2 has to be done within a specific forwarding context, that is, within a specific FIB.

To meet the requirements listed above, the following set of key ideas is applied in building the solution.

1. The first is the concept of a primary/protected PE and one or more backup PEs, called protector PEs. In the example in Figure 14.6, PE1 is the primary/protected, and PE2 is the protector. The roles of primary and protector must be pre-established via configuration. The protector can protect all or just some of the forwarding state of the primary. For example, PE2 can protect just the state for VPN grey, even if CE1 is connected to sites of another VPN, VPN white. The granularity of the protection is at the discretion of the network designer. For now, let us assume that the PLR ‘magically’ knows how to forward the traffic to the protector, using a ‘somehow’ pre-established detour LSP. The details of how this is done will become clear after the requirements on the protector are understood, so we defer that discussion to later in this section.
2. The protector, PE2, constructs a backup forwarding table (backup FIB) to be used for forwarding traffic that was originally destined for PE1. This forwarding table contains mappings of incoming label to the outgoing interface. Because PE1 had advertised VPN label L1 for prefix 10.1/16, the backup forwarding table contains an entry forwarding traffic arriving with label L1 out the interface if2 towards CE1. This table is

built based on the BGP advertisements that PE1 sent out to all its IBGP peers⁸ (and thus also to PE2).⁹ The forwarding data in it is relevant in the context of traffic destined for PE1.

3. When data initially flowing towards primary PE1 arrives from the PLR at the protector PE2, it must be tagged in a way that triggers a lookup in this table. To accomplish this, we use a context label [RFC5331], which identifies the forwarding table in which the label immediately below the context label should be looked up. Both the PLR and the protector PE must agree on the FEC associated with the context. In particular, the PLR has to know which detour LSP terminates on a particular backup FIB on the protector PE.

The way that this LSP is set up from the PLR to the protector PE and the way that traffic is mapped into this LSP are what shapes the local protection solutions for the various deployment and failure scenarios. In the next sections, we examine pseudowire and L3VPN scenarios separately and show the solutions and potential refinements for each case.

14.5.1 Protecting against an attachment circuit failure in a pseudowire scenario – edge protection virtual circuit

For the sake of simplicity, let us start the discussion by looking at the LDP-signaled pseudowire scenario in Figure 14.7. The topology is the same as in the redundant pseudowire discussion described in Section 14.4.2 and the pseudowire is set up from PE3 to PE1, with PE2 being an alternate exit point towards the destination CE, CE1. The failure we focus on is a break in the link PE1-CE1 which affects traffic flowing from CE2 towards CE1.

The PLR in this case is PE1 and the protector is PE2. The roles are set through configuration. Because multiple pseudowires can terminate at a particular PE, there needs to be a way to specify what state is protected and to configure it as well. This state is expressed using a combination of the PW ID and neighbor address (from the pseudowire configuration), and the context-id, which is a loopback IP address configured on both PE1 and PE2

⁸ Implicit in this sentence is an assumption that PE1 and PE2 have an IBGP session with each other. Note that in general this is not the case, because PE1 would have an IBGP session just with its route reflectors. In that case, PE2 would receive advertisements originated by PE1 through the route reflectors. This is important, because PE protection does not require a change to IBGP provisioning.

⁹ Constructing this state is straightforward when BGP is used because all peers see the same advertisements. When LDP is used, for example, for setting up LDP-signaled pseudowires, additional mechanisms must be set in place to distribute the information from the primary to the protector. We discuss the details of this in subsequent sections.

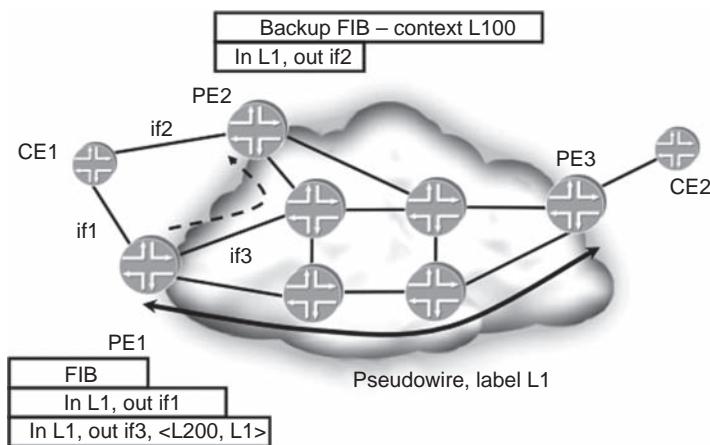


Figure 14.7 Local protection PE-CE link failure in a pseudowire setup

(similar to a virtual address covering both PE1 and PE2). This combination identifies the protected entity. The assumption is that PE1 rapidly detects breakage of the PE-CE link, either through physical level alarms, Ethernet OAM or through other methods such as BFD. After detecting the break, PE1 forwards the traffic to PE2.

Let us see how the correct forwarding state can be constructed at PE2 to accommodate forwarding traffic arriving from PE1. Because the label distribution in the pseudowire case is done with LDP, the labels assigned by PE1 must somehow be communicated to PE2. One way to do this is to set up a targeted LDP session between the protected and the protector PEs, and have the protected PE, PE1, advertise the incoming pseudowire label to PE2.¹⁰ At this point the protector PE, PE2, should have the necessary data to build its mirror forwarding table to use for protection. Referring back to Figure 14.7, if we assume that the pseudowire label used for the pseudowire from PE3 to PE1 is L1, the protector, PE2, installs forwarding state in its backup FIB that forwards traffic arriving with label L1 over the interface if2 to CE1.

To actually perform the lookup in the backup table, protected traffic must arrive at PE2 tagged with a context label which indicates to PE2 that the lookup on the label immediately below the outermost label should be done using the backup-FIB. In the egress protection virtual circuit solution, this tagging is provided by an RSVP-signaled bypass LSP from PE1 to PE2. This LSP is signaled according to [RSVP-OOB], with the UHP flag set in the

¹⁰ Additional information that allows PE2 to correctly associate traffic tagged with the pseudowire label to the correct attachment circuit must also be carried, although it is not discussed in detail here for the sake of simplicity.

LSP_ATTRIBUTE Object, forcing the allocation of a real label (rather than label 3) at the egress PE2 and the creation of a context lookup for traffic arriving with this label.¹¹ In Figure 14.7, this LSP is denoted by a dashed line between PE1 and PE2. The label allocated at the egress of that LSP (at PE2) is L100, and the label PE1 must use to forward traffic on it is L200. Let us take a look at the state that is installed on PE1 and PE2 as a result:

- On PE2, forwarding state is installed that triggers a lookup in the backup FIB when traffic labeled with L100 is received.
- On PE1, the forwarding entry that maps incoming PW label L1 to outgoing interface if1 is enhanced with another next hop, to be used in case if1 fails, which pushes the label L200 on top of the pseudowire label L1 to send the traffic to PE2 on the RSVP LSP.¹²

As a result, when if1 fails, traffic is seamlessly forwarded to PE2, where it is correctly forwarded towards CE1 over if2.

The only remaining mystery is how the connection is made between the RSVP LSP and the protected pseudowire. If PE2 has 10 different backup FIBs and 10 such LSPs, how would PE2 know to look into the right one? The answer is provided by the context identifier mentioned at the beginning of this section. By setting up the LSP with a destination address of the context identifier, PE1 and PE2 have a common understanding of the protection state and can take the appropriate action to install the correct forwarding state. In the simplest setup, this LSP is manually configured on PE1.¹³ Based on the context identifier configured, PE1 and PE2 know to install the correct forwarding state.

The edge protection virtual circuit example showed a simple example for protecting against a PE-CE link failure. In the next section, we look at an L3VPN scenario and see how the protection works for a PE failure.

14.5.2 Protecting against an egress PE failure in an L3VPN scenario

The example in Figure 14.8 shows the L3VPN network discussed earlier in this chapter. The loopback addresses of PE1 and PE2 are 1.1.1.1 and

¹¹ Strictly speaking, it is not necessary to use RSVP-OOB for this purpose. Vanilla RSVP-TE can do the job as long as PE2 assigns a real label (rather than NULL) and uses the knowledge that the destination of the LSP is the loopback address associated with the context-id that identifies the backup FIB, so that the traffic arriving on the LSP is forwarded correctly.

¹² It is important to note that the bypass LSP should not be used for any other purpose. Thus it must be hidden from other applications.

¹³ Note that this is just an example of a very simple setup. Manual configuration is not a requirement on the solution.

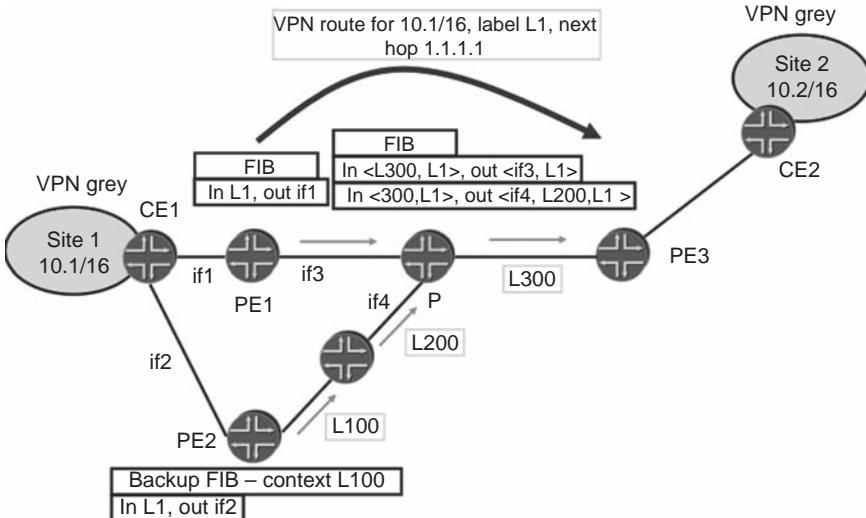


Figure 14.8 Protection of an egress PE in an L3VPN scenario

2.2.2.2, respectively. VPN route advertisements originating from PE1 have a BGP nexthop of 1.1.1.1. In this example, we focus on the advertisement for prefix 10.1/16, with VPN label L1. The transport LSP used for sending traffic from the ingress PE3 towards PE1 is an RSVP LSP for destination 1.1.1.1, and the label allocated by router P for this LSP is L300.

PE2 is the protector of PE1 and is aware through configuration that it protects traffic destined to 1.1.1.1. To fulfil this role, PE2 needs to act as if it were PE1, when PE1 is down, by doing two things. First it must attract PE1 traffic to it, and second it must forward this traffic to the right CE site. Let us see how this is done. The second part is the easy part: PE2 must build and maintain a backup FIB, mirroring the VPN forwarding state at PE1. For every VPN label advertised by PE1, PE2 builds corresponding state in its backup FIB for sending the traffic to the correct CE.¹⁴ In the L3VPN case, building this state is easy to accomplish, because the BGP advertisements for a given VPN originated by a given PE are received by all other PEs in that VPN. In the example, the backup FIB contains an entry mapping traffic tagged with VPN label L1 to the interface if2, leading to

¹⁴ The VPN labels advertised by PE1 are used on PE2 as the incoming labels in the backup FIB. Thus, it is PE1, and not PE2 that allocates incoming labels for the backup FIB on PE2. This particular scenario illustrates one possible case of upstream-assigned labels [RFC5331] in which the VPN labels are assigned neither by the downstream router, PE2, nor by the upstream router, PE3, but rather by some other router, PE1.

site 1 of VPN grey.¹⁵ Having seen how traffic should be forwarded at PE2, let us see now look at how it can arrive there.

To protect PE1, the point of local repair P must create a detour LSP to the protector PE2, and PE2 must make sure that any traffic arriving over this LSP is handled by a lookup in the backup FIB. There are several challenges in building this LSP:

- Figuring out the destination of the detour LSP. Because PE2 is pretending to be PE1 in this setup, it is natural that the detour LSP destination address be the same as the one PE1 uses as a next hop in its BGP advertisements, 1.1.1.1.
- Computing the detour. To be able to compute an LSP to 1.1.1.1, PE2 must advertise this prefix into the IGP. However, because this information is used only when PE1 fails, the advertisement PE2 sends for 1.1.1.1 is tagged with a high metric.
- Signaling the detour. One important feature of the detour LSP is that traffic arriving on it needs to be forwarded based on the backup FIB. Thus, traffic must arrive at PE2 with a real label. This label acts as the context label and ensures that the lookup is done in the backup FIB. To accomplish this, all that PE2 has to do is to realize that the LSP setup request is for a destination that represents a protected PE and to assign a real label rather than label 3. In Figure 14.8, PE2 assigns label L100 to the detour LSP. L100 acts as a context label for the backup FIB. The intermediate router on the way to P assigns label L200 to the same LSP.
- Automatically programming the protection state at P. This can be done by simply leveraging the existing node protection mechanisms for the LSP from PE3 to PE1. When protection is configured, the computation and signaling of the detour are triggered at P. After the setup is complete, a backup next hop is set up at node P, and is used in case the primary fails. The backup state takes traffic coming in with label L300 and forwards it tagged with label L200 out the interface towards PE2. This traffic arrives at PE2 with label L100, and as a result the bottom label L1 is looked up in the backup FIB and traffic is forwarded towards the CE.

In this way, traffic arrives from the PLR to the protector tagged with the correct context label. Because the protection state is pre-computed and pre-installed, the interruption in the event of a failure is similar to the FRR times of normal transport LSPs. The ingress PE eventually redo its path selection and starts sending traffic directly to PE2, but the speed of this move does not impact the end-to-end service.

¹⁵This is done based on the next hop address of the VPN route advertisement.

Although we described local protection in the context of a PE, the same mechanisms can be applied to protect the ABRs or ASBRs in an inter-area or inter-AS setup. Indeed, early work on tail-end protection focused on an inter-AS scenario [EGRESS-PROT].

14.6 CONCLUSION

With the increasing popularity of MPLS services, the use of hierarchy to scale MPLS deployments, and the increasing pressure to keep costs down, there is a dire need for fast and reliable protection of MPLS services from failures at the tail end. The local protection schemes described in this chapter accomplish this goal in the fastest and most scalable way, independent of routing convergence speed. The local protection scheme for LSP tail-end protection relies on extending the use of recently developed architectural concepts such as upstream-assigned labels and context labels, and packaging together existing protocols and technologies, proving once again the flexibility of MPLS technologies. These schemes are still under development at the time of this writing, and although some vendors may be shipping some of them, none of the local protection schemes has yet been documented by standards bodies.

14.7 REFERENCES

- [BEST-EXT] P. Marques, R. Fernando, E. Chen, P. Mohapatra, *Advertisement of the Best External Route in BGP*, draft-marques-idr-best-external-01.txt (work in progress)
- [ADD-PATH] D. Walton, A. Retana, E. Chen, J. Scudder, *Advertisement of Multiple Paths in BGP*, draft-ietf-idr-add-paths-02.txt (work in progress)
- [EGRESS-PROT] Y. Rekhter, *Local Protection for LSP Tail-end Node Failure*, presentation at MPLS 2009, October 2009, Washington DC
- [GR253-CORE] *Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria*, GR-253-CORE
- [RFC 5331] R. Aggarwal, Y. Rekhter, E. Rosen, *MPLS Upstream Label Assignment and Context-Specific Label Space*, RFC5331, August 2008
- [RSVP-OOB] Z. Ali, G. Swallow and R. Aggarwal, *Non PHP Behavior and Out-of-band Mapping for RSVP-TE LSPs*, draft-ietf-mpls-rsvp-te-no-php-oob-mapping-04.txt (work in progress)

14.8 FURTHER READING

- [L3VPN-PROT] J.L. Le Roux, *Fast Protection in MPLS L3 VPN Networks*, presentation at MPLS 2006, October 2006, Washington DC

14.9 STUDY QUESTIONS

1. With the recent advancements vendors have made in improving the speed of routing convergence, is local protection still required?
2. Why doesn't NSR provide a suitable solution for tail-end protection?
3. How are context labels similar to VPN labels?
4. What additional information must be propagated along with the pseudowire label between the primary and the protector to create the correct forwarding state on the protector?
5. Why is it important that the bypass LSP not be used for any routing function in the egress protection virtual circuit solution?

15

MPLS Management

15.1 INTRODUCTION

In the previous chapters we have seen how MPLS is used as a key component for converging multiple services onto the same physical infrastructure. As service providers obtain more of their revenue from MPLS-enabled applications and as the traffic carried requires stricter SLAs, the ability to manage both the MPLS infrastructure and the services running on top of it efficiently becomes more important.

In this chapter we will take a look at some of the unique aspects of managing MPLS networks and services. Some of the topics covered are fault detection and the emerging mechanisms for detecting data plane failures, provisioning challenges and tools for gaining visibility into the network.

15.2 MANAGEMENT – WHY AND WHAT

From the provider's point of view, management is a broad definition of all the aspects that allow him to offer, deploy and bill for a service. This includes provisioning the service, detecting and isolating failures, avoiding downtime and accounting for billing purposes. The availability of tools for accomplishing these tasks and the capabilities of such tools affect the costs incurred for deploying the service and the revenue that can be derived from

it. Good tools can ease the provisioning process, reduce the troubleshooting time when a fault occurs and provide granular accounting.

Management is a broad topic that could easily be the subject of an entire book. In this chapter, we focus on the router functionality that provides the necessary information and the basic tools for managing both the MPLS infrastructure and the services enabled by it. Some of this functionality, such as many accounting features, was added by vendors following customer demand and is specific to a given implementation. Other functionality, such as LSP ping, was defined in the IETF as new protocol machinery that routers must implement.

Because management is such an important piece of any proposed solution, work is done in this area in each and every one of the IETF working groups. In fact, a document cannot advance in the standards track in the IETF without having the appropriate management support, for example, in the form of an SNMP Management Information Base (MIB). In addition to the work in the IETF, the ITU has also produced a large number of standards for MPLS operations and management.

When discussing MPLS management, two questions need to be answered:

1. What are the management functions that must be provided?
2. At which layer must these functions be applied?

The answer to the first question is straightforward. The functions required from any management solution apply to MPLS management as well. In this chapter, we group them in the following categories:

1. Fault detection and troubleshooting functions, such as the ability to detect misrouting of a packet in the network.
2. Configuration functions, such as the ability to avoid misconfigurations or to automate the configuration process.
3. Visibility functions, such as the ability to accurately account traffic or obtain information about a deployed service.

The answer to the second question is more complex. Because MPLS is used at different layers, the management functions must be provided at each of the following layers:

1. Device layer, e.g. the individual links and nodes in the network.
2. Transport layer, e.g. the inter-PE tunnels in the core, set up with LDP or RSVP, used to transport MPLS-labeled traffic.
3. Virtual connection layer, e.g. the per-VPN virtual tunnels created by the VPN labels.
4. Service layer, e.g. the VPN service itself.

Because the same tools and methods are often used for managing different layers, we will discuss MPLS management from the perspective of the functions provided, rather than discussing it from the perspective of how each individual layer can be managed. Let us start with what is perhaps the most important aspect of MPLS management, detecting and troubleshooting failures.

15.3 DETECTING AND TROUBLESHOOTING FAILURES

Detecting the failure is the first step towards fixing it. As MPLS networks start carrying more and more services with strict SLAs, fast failure detection becomes a must. However, given the MPLS fast-reroute mechanisms discussed in the Protection and Restoration chapter (Chapter 3), why are we even discussing failures, let alone their fast detection, in this chapter? The answer is because the local protection mechanisms of MPLS protect against a physical link or a node failure, but other events, such as corruption of a forwarding table entry or a configuration error, can also cause traffic forwarding problems.

When talking about forwarding failures, there are two goals. The first, and most important, is to detect the problem quickly. For a provider, the worst possible scenario is to find out about the existence of a problem from the customer asking why the service is down. The second goal is to automatically recover from the failure. This may mean switching the traffic to a different LSP or even bringing down a service, with the correct indication, instead of blackholing traffic for a service that is reported to be up and running.

15.3.1 Reporting and handling nonsilent failures

From the point of view of failure detection, there are two types of forwarding errors: silent and nonsilent. We will discuss nonsilent failures first, and talk about both detection and fast recovery. Nonsilent failures are the ones that the control plane is aware of, such as the tear-down of a (nonprotected) LSP following a link-down event. If the control plane is aware of the failure, then the problem can be quickly reported. For example, most vendors support sending an SNMP trap when an LSP is torn down, not just for the primary path but also for protection paths. This error indication is important for the operator, who can correlate this information with other events in the network, such as VPN traffic being dropped for a particular customer or the potential failure of protection for an LSP.

The quick detection of the failure in the control plane does not guarantee that traffic will not be impacted. Here are two examples of how this can happen:

1. *Delayed handling of the error.* In a VPN setup, when a link-down event causes the PE-PE LSP to be torn down, the quick detection of the problem at the LSP head end does not necessarily guarantee that the VPN customers will not experience any traffic loss. Recall from the advanced L3VPN chapter (Chapter 8) that if the reevaluation of the VPN routes is timer-based rather than event-driven, the VPN routes will not be immediately re-evaluated, causing blackholing of customer traffic for a bounded amount of time until the reevaluation happens, and traffic is switched to an alternate LSP. Even if the reevaluation of the routes is event-driven, in cases where a large number of forwarding entries must be updated following the failure (to point to a different LSP), traffic loss will still happen until all entries are updated.
2. *Insufficient propagation of the error.* Let us look at a pseudowire service, where a failure of the transport tunnel results in a loss of connectivity between ingress and egress PEs. Assuming that this failure is detected, the PEs may send native indications over the related attachment circuits to notify the endpoints of the fault condition. In such case it is necessary to map the error correctly, using procedures such as those defined in [PW-OAM-MSG-MAP]. To ensure that such mapping is possible, the emulated service must have well-defined error procedures, otherwise the error detected by the PE cannot be correctly translated and propagated over the attachment circuits.¹

To summarize, nonsilent failures are reported in the control plane. The knowledge of the LSP failure can be used by the routers to update the forwarding state and by the operator to address the problem that caused the failure in the first place. However, the fact that the failure is reported in the control plane cannot guarantee that no traffic will be lost.

15.3.2 Detecting silent failures – MPLS OAM

Silent failures are the ones that the control plane is not aware of and are usually caused by a loss of synchronization between the control and data planes. The classic example of a silent failure is the corruption of a forwarding table entry. This is a popular example because some of the early implementations of MPLS suffered from this problem. As the

¹ This used to be the case for Ethernet. However, work is underway in the ITU-T and IEEE to define in-band Ethernet OAM standards.

implementations matured, the problem was resolved, but it remained one of the major concerns for providers because corrupted forwarding entries are particularly difficult to troubleshoot. They usually cause traffic blackholing, but may sometimes lead to traffic misrouting where traffic is incorrectly forwarded. Because the problem manifests itself in the data plane only, it is difficult to detect. For example, traffic to only a handful of destinations may be lost. Finally, this type of failure requires manual intervention to fix, usually rebooting the entire router.

The only way to detect a silent failure is by constantly monitoring the operation of the forwarding plane by sending test traffic. However, at what level should this be done? To answer this, let us take a look at a BGP/MPLS L3VPN service. The options are:

- Transport layer. The inter-PE MPLS tunnels in the core, set up with LDP or RSVP, provide connectivity between the PEs and transport all VPN traffic across the core.
- Virtual connection layer. The per-VPN virtual tunnels created by the VPN labels are invisible in the core of the network. They provide the demultiplexing capability at the PE to steer traffic towards the correct customer site.
- Service layer. Rather than testing the individual building blocks providing the service, this approach tests the service itself, e.g. by sending test traffic between the customer sites in a VPN.

Regardless of which level the polling happens at, the next question is how often should the test probes be sent? There are two factors to take into consideration when answering this question:

1. *The desired detection time.* Clearly, the probes must be sent at intervals shorter than the desired detection time. Just how much shorter depends on how the detection mechanism works, e.g. the time it takes to receive feedback for a given probe and the number of failed probes that are necessary to declare a failure.
2. *The resources spent on detection.* Intuitively, it is easy to understand that a polling-based mechanism uses up forwarding-plane resources, because the probes themselves need to be forwarded along with the regular traffic. However, processing the probes may place a burden on the control plane as well, as will be seen in the following section. Thus, failure detection through polling comes at a cost and there is a tradeoff between quick detection and resources spent on doing so.

The data-plane mechanisms for detecting and pinpointing failures in MPLS networks are collectively referred to as MPLS OAM (operations, administration and management), implying data-plane OAM. When discussing MPLS management in general, it is important to distinguish

between the OAM functions, which operate in the data plane, and other management functions, such as the misconfiguration avoidance schemes that will be discussed in Section 15.4.2, which operate in the control plane. Together, these mechanisms provide a complete set of tools for managing the network. In the next sections, we will take a look at some of the MPLS OAM mechanisms for failure detection, as defined in the MPLS, pwe3 and BFD Working Groups in the IETF.

15.3.2.1 LSP ping

Why define new methods for failure detection in MPLS, instead of just using IP ping for the traffic using the LSP? For example, to check the health of the LSP set up by the LDP from PE2 to PE1 in Figure 15.1, why not simply send periodic IP ping traffic to PE1's loopback address and ensure that this traffic is forwarded over the LSP? The answer is: because such an approach may not detect all failures.

Imagine a probe traveling on the LSP with label L4 on the hop between PE2 and C. Now assume that at node C, the forwarding state is such that the label is popped instead of being swapped to L3. This could happen, for example, because of a corruption in the forwarding state entry, but may also be the result of a legal protocol operation, when LDP-independent control is used (as will be explained in Section 15.4.2.1). In any case, when the traffic arrives at C, the label is popped. The packet continues its journey to PE1 as a pure IP packet and the failure is not detected.

Therefore, what is needed is a way to do two things: (a) validate the forwarding of traffic in the data plane and (b) verify the data-plane state against the control-plane state. The basic mechanism for providing this

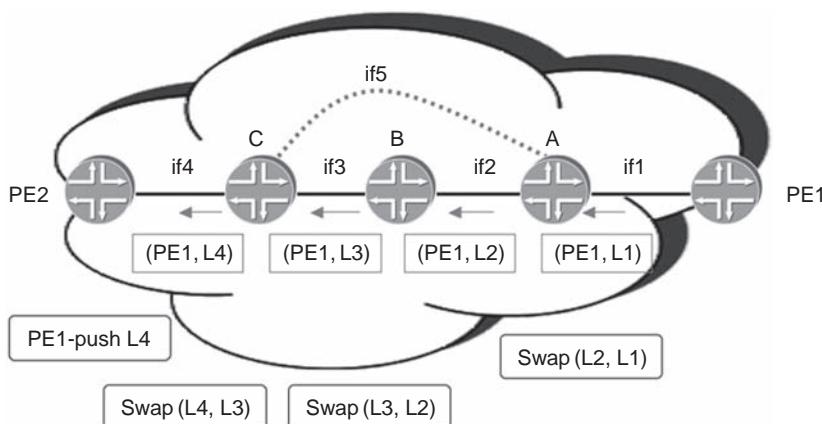


Figure 15.1 Failure of an LDP-established LSP

functionality was defined in the MPLS Working Group and is documented in [RFC4379]. The solution is modeled after the ping utility used in IP for failure detection, and is therefore referred to as LSP ping, or LSPing.² Similar to ping, it uses probe packets, called MPLS echo requests, and expects to receive back acknowledgments, called MPLS echo replies.

The idea behind LSPing is simple. Verify that a packet that belongs to a particular forwarding equivalence class (FEC) actually ends its MPLS path on the LSR that is the egress for that FEC. In the example shown at the beginning of the section, the FEC is PE1's loopback address. In the failure scenario, when the label is popped at C, the MPLS path ends on router C instead of router PE1. Because C is not the egress for the FEC, the check fails and the error is detected. From this description, several requirements become apparent regarding the probes and their handling:

1. The probes must follow exactly the same path as the data packets.
2. The probes must be delivered to the control plane of the LSR on which they ended their MPLS path for verification.
3. The probes must contain enough information about the FEC to allow the receiving LSR to determine if it is indeed the correct egress.

Based on these requirements, the LSPing probe packets are defined as UDP packets as follows:

1. To ensure that the probes follow the same path as the data packets, they are forwarded using the same label stack as the data forwarded over the LSP they are testing.
2. To allow delivery of the probe to the control plane of the egress LSR, the router-alert option is set in the IP header. An interesting challenge arises with regards to the IP destination address. The problem is that the address of the LSR, which is the egress of the LSP, may not always be known. For example, LDP may advertise a label binding for an arbitrary FEC that is not necessarily associated with any address on the router. For this reason, the destination address is a random address in the 127/8 range (remember this is not a routable address). In the next section (Section 15.3.2.2), we will see why it is useful to allow picking the destination address from a range rather than mandating one particular value.
3. To facilitate the check that the egress LSR must perform, the probe contains information about the FEC under test. This information is carried in the payload of the UDP packet and is encoded as a set of TLVs. The information is different based on the type of FEC, so TLVs are

²RFC 4379, which defines LSPing also provides a mechanism for doing hop-by-hop failure localization, similar to the IP traceroute utility. This functionality will be discussed in Section 15.3.3.2.

defined for each of the different types: LDP, RSVP, L3VPN, pseudowire and so on. For example, for an LDP FEC, the prefix, its length and the fact that a binding was advertised for it using LDP is enough. For an L3VPN FEC, the route distinguisher, the prefix and the length are needed. Note that the existence of the different FECs means that LSPing can be used to test different layers of the MPLS network. In a VPN setup, LSPing using the L3VPN FEC tests the VPN tunnel, while LSPing using the LDP FEC can test the LDP LSP that carries the VPN traffic in the core.

4. To inform the originator of the probe of the result of the FEC test, the receiver must know its address. Therefore, the source address of the MPLS echo request probe is set to a routable address on the originator of the probe.

An LSR receiving an MPLS echo request validates it and sends a reply using an MPLS echo reply packet. The reply is also a UDP packet, sent to the source address of the MPLS echo request and forwarded in accordance with the route that is available for its destination address. Thus, the reply packet may travel as pure IP or it may be encapsulated in an LSP, if the path to the destination is through an LSP. The reply contains status information, such as success or failure and the reason for the failure. When the originator of the echo request receives this reply, it matches it against the outstanding requests and reports the appropriate status for the LSP under test. Two assumptions are made in this mode of operation: (a) the egress LSR can forward traffic back to the originator of the echo request and (b) the reply packets will arrive there. These assumptions may not always hold true. For example, if the incorrect source address is used, the egress LSR may not be able to send traffic back to the receiver. Furthermore, because the reply is a UDP packet, its delivery is not guaranteed. When this happens, the reply packets are not delivered to the originator of the LSPing, which will incorrectly infer that there is a problem with the LSP under test. This condition is referred to as a false negative.

15.3.2.2 Properties of LSPing

At this point, let us stop and note a few properties of the LSPing solution:

1. The control plane of the router receiving the probe is involved in the validation of the echo request, with the following consequences:
 - (a) If the control plane is busy and cannot process the echo request in a timely manner, the echo request may time out at the LSP head end, resulting in a false negative.
 - (b) LSPing places a load on the control plane of the egress router. Because only a limited number of probes can be sent to the control

plane and processed there, there is a limit on both the number of LSPs that can be monitored and on the frequency of the probes.

In fact, to avoid a denial-of-service attack on the router control plane using LSPing traffic, [RFC4379] recommends rate limiting the amount of LSPing requests and replies that the router accepts per unit of time. Let us first take a look at the number of LSPing requests arriving at the tail end of a particular LSP. For a point-to-point LSP, such as an RSVP-generated one, this number is directly proportional to the polling frequency at the LSP head end. For multipoint-to-point LSPs, such as LDP-generated ones, the load is more difficult to evaluate, because many routers in the network may be sending LSPing packets to the same egress LSR independently of each other. Let us now do a similar analysis for the number of echo replies being returned to the head end. This number is easy to evaluate for point-to-point or multipoint-to-point LSPs but may be very large for point-to-multipoint LSPs, as we will see in Section 15.3.2.5 describing LSPing for point-to-multipoint LSPs.

2. The echo request tests just one of the possible paths to the destination. When several equal-cost paths can exist, such as the case of LDP, only one of them is tested by the echo request, so an error in a different path may not be detected. To test all the different paths, multiple probes must be sent, such that all the paths are exercised. Routers forward packets along the different equal-cost paths based on the result of a hash function. Therefore, packets with ‘different enough’ destination addresses will typically exercise different paths in the network. This is one of the reasons why the echo request destination address is any address in the 127/8 range. The next question is how to pick the destination address of the probe, such that a particular path is exercised. We will see the solution to this problem in Section 15.3.3.2, which deals with LSP traceroute.
3. The echo request tests the traffic flow in one direction only (from the source to the destination). The return path of the traffic is not tested.
4. The echo reply is sent as UDP; therefore its delivery is not guaranteed and false negatives are possible.
5. The receiver of the echo request may not support the LSPing procedures, causing a false negative.

In light of the discussion above, it is clear that false positives are not possible, making LSPing a very powerful troubleshooting tool. It is also clear that false negatives are possible, raising the question of whether it is at all feasible to use LSPing as a liveness detection mechanism? Assuming that all routers in the network support LSPing and assuming that LSPing replies are never lost, the bottleneck remains the processing capacity of

LSPing requests in the control plane. Thus, the effective use of LSPing for liveness detection depends on the detection time desired. As explained previously, only a limited number of echo requests can be processed by the control plane in any given unit of time. This places a limit on the number of LSPs that can be tested and on the frequency of the probes. To overcome this scaling limitation, the BFD protocol, described in the Protection and Restoration chapter (Chapter 3), was extended for LSP connection verification.

15.3.2.3 Fast failure detection – BFD for MPLS LSPs

Based on the realization that verification of the data plane against the control plane is the main factor limiting the polling frequency for LSPing, the approach taken in BFD for MPLS LSPs is to validate the data plane only. Recall from the Protection and Restoration chapter (Chapter 3) that BFD is a simple hello protocol that can be used to test the forwarding path between two endpoints of a BFD session at high frequency. Two questions immediately arise.

1. How are the LSP endpoints communicated to BFD?
2. Is the mechanism useless if the data plane cannot be verified against the control plane?

The answer to these questions is to use a combination of LSPing and BFD. LSPing is used for bootstrapping the BFD session and for periodically (but infrequently) verifying the control plane against the data plane. BFD is used for doing fast failure detection by exchanging BFD hello packets with high frequency, along the same data path as the LSP being verified.

It is outside the scope of this book to discuss the details of BFD or of the bootstrapping of the BFD session using LSPing. These are described in [MPLS-BFD]. Instead, the important thing to remember is that by combining the fast-failure detection of BFD with the extensive validation capabilities of LSPing, a scalable solution for LSP monitoring is achieved.

As we have seen in the fast-reroute discussion in the Protection and Restoration chapter (Chapter 3), fast detection of the failure is a necessary but not a sufficient condition for fast restoration. As long as operator intervention is required to fix the problem, the restoration time will remain unacceptably long. In the next section, we will explore some of the options for automatic reaction to failures in the MPLS infrastructure.

15.3.2.4 From fast failure detection to self-healing networks

A failure in the MPLS infrastructure (transport LSPs) affects the services that are provided over it. For example, in a VPN setup, failure of a transport

LSP that is not reported in the control plane will cause blackholing of the customer VPN traffic. Such a failure may be the result of a bug, such as the corruption of a forwarding table, or can be a manifestation of correct protocol behavior in the face of a misconfiguration, as explained in Section 15.4.2.1. Rather than silently blackholing customer VPN traffic, it may be preferable to make the service unavailable, for example by withdrawing the VPN route towards the CE, for two reasons: (1) the outage is well understood by both the customer and the provider, and there is no dispute of whether or when the problem started happening; and (2) if the CE is multihomed, the outage may be avoided altogether, as we will see later in this section.

The actions that can be automatically taken upon detection of a failure in the MPLS infrastructure depend first and foremost on the protocol used for setting up the transport LSPs, since LDP and RSVP have very different properties in terms of what actions can be implemented. Another consideration is the need for post-mortem analysis on a failed LSP. Rather than automatically fixing the problem, the operator may want to ‘freeze’ the LSP in the failed state for troubleshooting. Finally, the level of redundancy built into the network also determines what actions make sense. For example, the existence of multiple LSPs towards the same destination, the existence of standby LSPs for RSVP-signaled tunnels or the deployment of multi-homing for a CE site are all important considerations.

One action that can be applied independently of the signaling protocol used is making the LSP invisible to applications such as VPNs. Recall from the Foundations of Layer 3 VPNs chapter (Chapter 7) that an MPLS path must be found to the BGP next hop of a VPN route in order to be able to send traffic to that destination. This process is called the resolution of the VPN route. Only routes for which resolution succeeded are installed in the VRF and consequently advertised to the CE. If, upon detection of a failure, an LSP is removed from the list of ‘acceptable candidates’ for resolving a certain BGP next hop, then one of two things will happen.

(1) If other LSPs exist to the same destination, the VPN route is re-resolved over one (or more) of them or (2) if no other LSPs are valid candidates for resolution, the VPN route becomes unresolved. In such a case, assuming that no multihoming is configured, the route is removed from the VRF and consequently withdrawn from the CE. In the case of multihoming, after the route becomes unresolved, BGP best-path selection is run again and a new route is installed in the VRF. Assuming the process is event-driven, rather than timer-driven, this can be done without propagating a withdrawal towards the CE.

This process is shown in Figure 15.2. Prefix 10.2/16 is advertised from site 2 to site 1 of VPN grey. Assuming that CE2 is initially single-homed to PE2 only, the VPN route is resolved at PE1 over LSP1. The result is installed

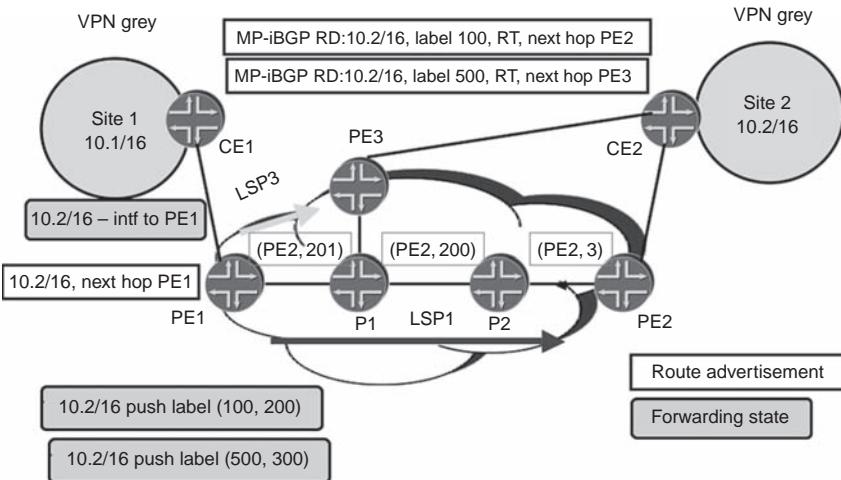


Figure 15.2 Making an LSP invisible to a VPN application

in the VRF corresponding to VPN grey on PE1 and the prefix 10.2/16 is advertised towards CE1. When a failure is detected for LSP1, the VPN route advertised by PE2 becomes unresolved and a withdrawal is propagated towards CE1. Revisiting the same scenario under the assumption that CE2 was multihomed to PE3 and that the advertisement from PE1 had previously been the winner of BGP path selection, the disappearance of the VPN route from PE2 will trigger path selection to be re-run. As a result, the advertisement from PE3 will become the winner and will replace the previous entry in the VRF at PE1. In this case, no change need to be propagated towards CE1.

The process of making an LSP invisible to applications by removing it from resolution is independent of the signaling protocol used. This mechanism is most powerful when redundancy (e.g. through the use of multihoming or multiple core LSPs) is available in the network. In such cases, it allows for leisurely troubleshooting of the failure, because it maintains the ‘broken’ LSP in the network without impacting the service. Even when no redundancy exists, the mechanism is still useful because it can transform a silent failure into a nonsilent one.

Let us now turn our attention to actions that can be taken for LSPs that are RSVP-signaled. In this context, RSVP has two relevant properties: (1) the ability to tear down an LSP from the head end and (2) the existence of secondary paths. The head end is the node that will be aware of the failure of the LSP. Assuming that the failure happened because of the corruption of a forwarding table along the path, tearing down the LSP and resignaling it will likely clear the problem, as this process will remove old state and

install new one. The teardown and re-establishment of the LSP can be automatically triggered by the head end in reaction to the failure detection. This approach has two disadvantages: (1) traffic loss will continue until the LSP is re-established and (2) the failed state is cleared and cannot be analyzed. An alternate approach that avoids these two downsides is to switch from the primary to a secondary path when the failure is detected. Assuming the secondary is a standby path, traffic loss will be minimal and the primary path can be maintained for troubleshooting.

LDP lacks the ability to trigger a reaction at the protocol level from the LSP head end, as its LSPs are egress-initiated. In this case, the only action that can be taken when a failure is detected is to make the LSP invisible to applications (such as VPNs) that could potentially use it. An interesting point that does come up with LDP is the support of ECMP and the desired reaction when failure of a single ECMP path is detected. Two possible approaches are possible: (1) removal of the entire LSP when even a single path fails or (2) removal of only the offending path.

Figure 15.3 shows an LDP LSP to PE1's loopback address, where the path through link D-B fails. If PE3 is the LSP head end, there is no choice but to remove the entire LSP from use of applications such as VPNs. However, if PE2 is the head end, it is possible to remove either the path along interface PE2-D or the entire LSP. Which approach to take depends on why ECMP is employed in this network at all. If the goal is to provide resilience, then enough capacity is provisioned along each of the paths to carry the entire load. In this case, removing just the failed path provides exactly the functionality that the operator envisioned. However, if the reason for ECMP was to do load sharing in a network where none of the paths can carry the entire load, then removing the entire LSP may be the preferable action. Only the operator can determine which of the actions is appropriate in a particular deployment.

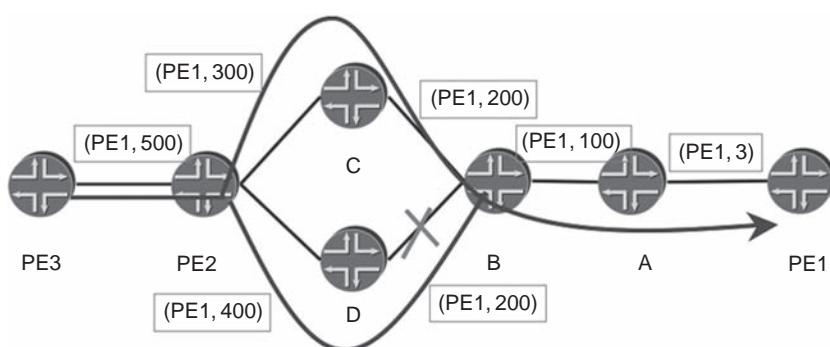


Figure 15.3 Failure of a single ECMP path

So far, we have seen the various automatic reactions that can be implemented and have discussed the fact that they can dramatically reduce the outage time by removing the need for human intervention. But there are hidden dangers in taking this approach. Aggressive failure detection times, coupled with drastic actions, can easily melt down a network. For example, aggressive detection times can yield false positives in the case of LDP, during the normal convergence period following a change in the IGP path. If the reaction to this false alarm is removal of an LSP from resolution, this can trigger a CPU-intensive BGP route re-evaluation. In turn, the high CPU may slow down overall convergence, adversely impacting service availability. For this reason, automatic healing must be used with caution and care must be taken to avoid false positives. The benefit of this technique when appropriately applied is a dramatic reduction in OPEX by reducing the outage time and the amount of operator intervention required.

Having explored the various applications of LSPing for the point-to-point case, let us now turn our attention to point-to-multipoint LSPs.

15.3.2.5 LSPing for Point to Multipoint LSPs

With the introduction of P2MP LSPs, as described in the MPLS Multicast chapter (Chapter 6), it is possible for an LSP to be set up from one head end to multiple tail ends. The IETF has adopted the LSPing extensions described in [P2MP-LSPING] for use in the point-to-multipoint case. Let us take a look at these extensions.

1. *New TLVs.* Clearly, new TLVs must be defined for use in the LSPing echo requests to identify the point-to-multipoint LSP under test. Separate TLVs are defined for RSVP-signaled and LDP-signaled LSPs.
2. *Mechanisms for handling a large number of echo replies.* When an MPLS echo request is forwarded on a P2MP LSP, it is replicated at the branch nodes just like normal traffic and reaches multiple egress routers. Therefore, multiple echo replies are sent back to the head end. Because LSPing traffic is rate limited (to prevent denial-of-service attacks on the router's control plane as explained in Section 15.3.2.2), when the number of replies per unit of time is large, echo replies may be dropped, creating false negatives. To avoid this situation, two approaches are proposed in [P2MP-LSPING].
 - (a) All egress routers respond, but they jitter their echo replies. Because rate limiting is done on the basis of packets / per unit of time, jittering the echo replies ensures that the head end is not overwhelmed by a large number of packets arriving at the same time. The head ends request the egresses to jitter their replies by including the Echo Jitter TLV in the LSPing echo request.

- (b) Only some of the egress routers respond to the echo requests. This is accomplished by specifying the address of the routers from which a reply is expected in an echo request, using a new TLV, the P2MP Egress ID TLV. The echo request is forwarded to all egress nodes, but only the ones that find their address in the P2MP Egress ID TLV respond to it. Note that this mode of operation allows verification of the path to each egress node individually. The assumption is that the egress nodes are known to the head end, which is always the case for RSVP-signaled P2MP LSPs but may not be the case for LDP-signaled P2MP or MP2MP LSPs.
3. *Definition of LSPing failure.* In the P2P case, success and failure are straightforward concepts: if an echo reply with status 'ok' is returned to the head end within the timeout period, the probe is considered successful. For P2MP, such replies must be received from all egresses under test. When the echo requests are targeted at a subset of the egress routers using the P2MP Egress ID TLV, this determination is straightforward. However, when echo requests are sent to all egresses, the head end must find out the set of egresses from which to expect a reply. For P2MP LSPs set up with RSVP, the set of egresses that should receive the request can be determined at the head end, but for LDP-signaled ones, there must be a separate mechanism to discover them.

Although P2MP LSPs pose extra challenges for liveness detection, we have seen that the LSPing mechanism is flexible enough to be extended for use in this scenario as well. Next, we will look at the uses of LSPing in even more challenging situations.

15.3.2.6 VCCV

Virtual circuit connection verification (VCCV) is the connection verification protocol for pseudowires set up using LDP (discussed in the Layer 2 Transport chapter, Chapter 12). VCCV was developed in the PWE Working Group in the IETF [RFC5085]. The natural question is why was a different connection verification mechanism developed? Doesn't LSPing support a TLV for Layer 2 circuits? In fact, VCCV builds on the LSPing solution and actually reuses the Layer 2 circuit TLV defined for LSPing.

Let us take a look at some of the challenges of using LSPing in a pseudowire environment. Recall from the Layer 2 Transport chapter that a CE-facing interface is associated with a pseudowire. When traffic arrives over this interface, it is encapsulated in MPLS, labeled with the pseudowire label and sent to the remote PE. Because labeled traffic is forwarded between the PEs, a transport tunnel is necessary that can carry MPLS. Usually, this tunnel is set up with either LDP or RSVP. Sometimes, a Control Word is prepended to the L2 frame, as explained in the Layer

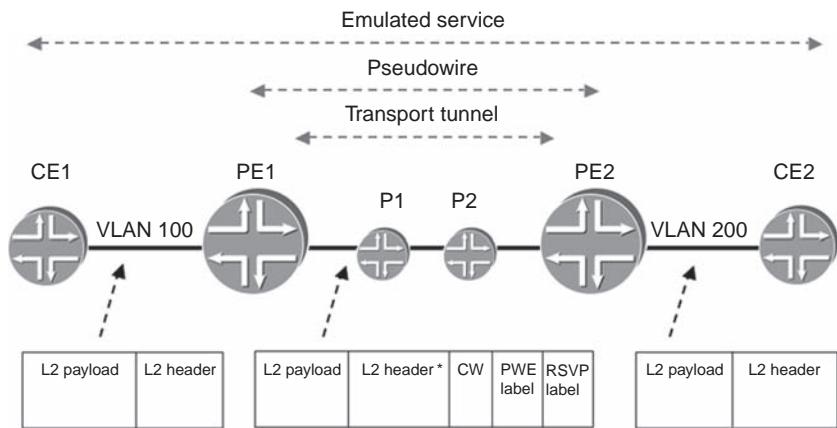


Figure 15.4 Forwarding traffic over a pseudowire

2 Transport chapter. Figure 15.4 shows how traffic is forwarded over the pseudowire.

Figure 15.5 shows what happens if an LSPing echo request is sent from PE1 to PE2 by applying the LSPing procedures described so far. The probe contains a TLV with information allowing PE2 to determine if it is the correct recipient, as described in [RFC4379]. It is encapsulated with the same label stack as the data packets and is sent towards PE1. The inner label is the pseudowire label and the outer label is the transport tunnel label (in this example, RSVP-signaled). When the probe arrives at P2, the RSVP label is popped and the LSPing packet arrives at PE2 with a

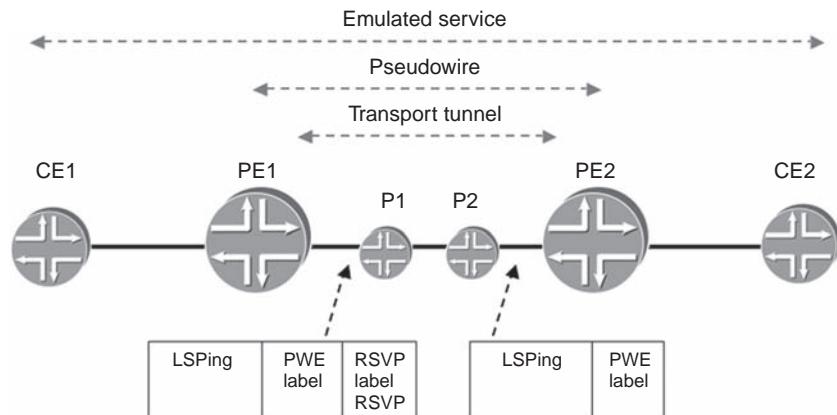


Figure 15.5 Using LSPing for a pseudowire

single label, the pseudowire label. At this point, if PE2 applies the normal forwarding procedures to this packet, the probe would simply be sent over the PE2–CE2 interface, instead of being processed by PE2. Note that this behavior is not unique to pseudowires and would be encountered in an L3VPN setup as well.

Several options are available to fix this problem:

1. Set the TTL expiry to 1 on the inner label. Assuming the RSVP label (transport tunnel label) is popped at P2, when the probe arrives at PE2, the TTL on the inner label expires and the packet is delivered to the control plane at PE2.
2. Insert the router alert label between the pseudowire label and the transport tunnel label. The router alert label is a special label that causes the labeled packet to be delivered to the control plane. In this way, after the transport tunnel label is popped, the top of the stack is the router alert label, which will cause the packet to be delivered to PE2’s control plane. The disadvantage of this approach is that data packets and LSPing packets are forwarded using different label stacks, so the data-plane verification is not as accurate.
3. Insert a special header after the LSPing packet and before the label stack. If forwarding for the data packets on the PE is done in such a way that this header is examined and acted upon before sending the traffic to the CE, then bringing the LSPing packet to the control plane could be driven by evaluation of the header. The idea is implemented by setting a bit in the Control Word to indicate that the packet should be delivered to the control plane rather than being forwarded, and using the Control Word when sending LSPing probes. The resulting header is called the pseudowire associated channel (PW-ACH or simply ACH). (This approach is not shown in the figure.)

Thus, extra steps must be taken to ensure that the LSPing packet is delivered to the remote PE’s control plane. Because several options are available, and because different platforms support different options, no one approach can be mandated. Therefore, it is necessary for the pseudowire endpoints to negotiate which mechanism to use. VCCV defines how this negotiation is done and how the probe packets must be encapsulated, based on the negotiated values, reusing the LSPing procedures.

So why is there no such mechanism available for L3VPNs? Will they not suffer from the same forwarding challenges? The reason is because the goals are different. For pseudowires, the goal is to build a control channel between the two PEs and do both failure detection and failure verification on this channel. This channel is either in-band, when the probes are taken out of the forwarding path based on the Control Word, or out-of-band, when they are taken out of the forwarding path based on the label. One of the goals is to use BFD over this control channel to monitor the liveness of

the pseudowire and bring it down if a failure is detected. Therefore, it is required to determine at the time of the pseudowire setup whether such a channel can be built. The VCCV negotiation can provide this knowledge.

In other cases, such as L3VPN, where LSPing is limited to troubleshooting a failure, the requirement for an indication on whether both ends support the same procedure is not required. This conclusion can be reached easily during the troubleshooting process.

To summarize, VCCV is used for connection verification of pseudowires. Although it is a new protocol, VCCV builds on both LSPing and BFD to provide failure detection and monitoring for pseudowires.

15.3.2.7 Pinging at the service level

The previous sections discussed tools for failure detection at the different layers used for building up a particular service, e.g. the PE–PE transport tunnel built with LDP or the virtual connection between the PEs providing a pseudowire. However, mechanisms such as ping can be used at the level of the service itself. For example, in an L3VPN setup, two routers in customer sites can send ICMP pings to each other to verify connectivity. These ping packets will be forwarded just like any other VPN traffic and can therefore discover if there is any problem on the path between the two sites. This type of check is an example of an end-to-end check of the service itself.

The description above assumes that the ICMP pings are initiated by the CE routers, but they can also be initiated by PE routers, as long as the ping application is VRF aware and can send probes according to the forwarding information for the VRF under test. Note that in this case, only a segment of the service is being tested.

15.3.2.8 Failure detection summary

Fast-failure detection is required to avoid traffic loss following a failure. Nonsilent failures are reported in the control plane, which can act on the failure indication to prevent traffic loss.

Specialized mechanisms for failure detection are required for identifying silent failures, which are not reported in the control plane. The previous sections showed different layers at which failure detection can be applied. To summarize, failure detection can be run at the service layer end to end, at the service layer for just a segment of the service, at the virtual connection layer for the virtual tunnel created by the VPN label or at the transport layer for the PE–PE tunnel over which the VPN tunnel is transported, as shown in Figure 15.6 for an L3VPN setup.

In the following section we will look at mechanisms for pinpointing the exact location of the failure.

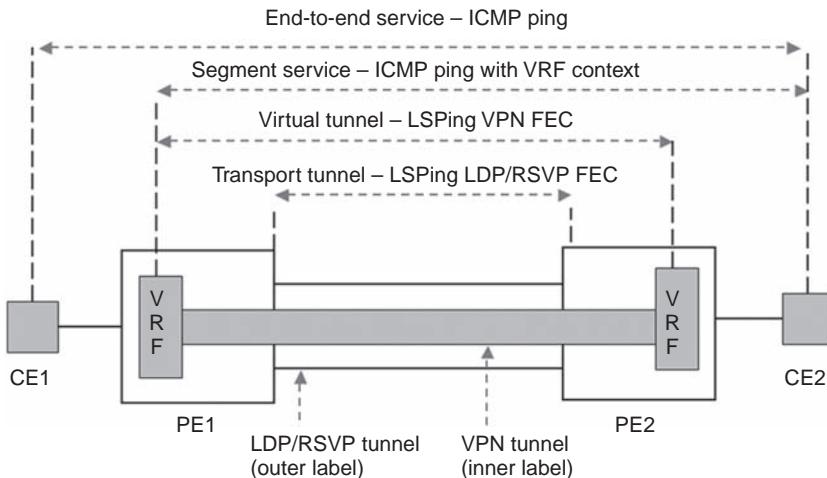


Figure 15.6 Applying failure detection at different layers

15.3.3 Troubleshooting failures

In the previous sections we focused on the mechanisms used for detecting failures. The next step is to see how the location of the failure can be pinpointed. The most popular method for doing so is tracing the path of the traffic in the network. Path-trace can be run at the different layers, similar to how ping can be run at different layers.

15.3.3.1 ICMP tunneling

In Section 15.3.2.7 we saw that ping can be used at the service level within an L3VPN for failure detection. Therefore, it is natural to want to use the traceroute mechanism for the localization of the failure. Traceroute sends a series of probe packets with increasing time-to-live (TTL) values. The hops where the TTL expires return an ICMP message indicating the TTL expiration to the originator of the probe, thus identifying the hops in the path. From this description, it should already be clear that traceroute operation in a VPN setup is not immediately applicable.

Figure 15.7 shows an L3VPN setup where the PE-to-PE tunnel is set up with RSVP. CE1 issues a traceroute for CE2's loopback address. At PE1, the probe is encapsulated with the VPN label and the RSVP label and sent towards PE2. When the TTL expires at P1, an ICMP message must be sent towards CE1. The problem is that CE1 is a private address in a VPN and therefore is not known at P1, so the message cannot be forwarded.

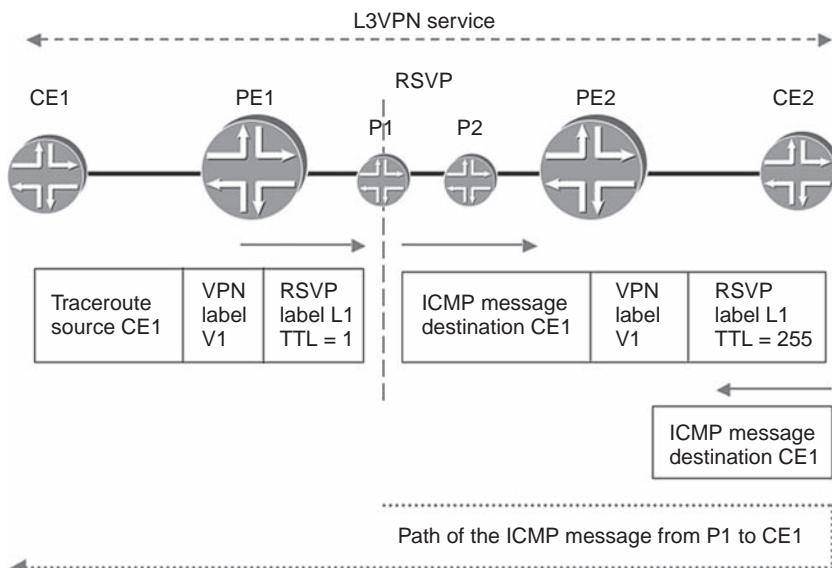


Figure 15.7 ICMP tunneling

An elegant solution to this problem is described in [RFC3032]. The idea is that the destination of the probe is likely to have reachability to the source of the probe. Therefore, if the ICMP message reaches the probe's destination, it can be forwarded from there back to the source. To achieve this, P1 builds the ICMP message indicating TTL expiry at P1 and then copies the label stack from the original packet to this ICMP message (updating the TTL to a high value), as shown in Figure 15.7. This is called ICMP tunneling, because the ICMP message is tunneled all the way to the original probe's destination, CE2. At CE2, the message is looped back to CE1, because CE2 has forwarding information for CE1. The path of the ICMP message is shown at the bottom of Figure 15.7.

ICMP tunneling is essential for implementing traceroute in a VPN setup. However, it can also be used to report other errors such as the ones caused by the need for fragmentation. ICMP tunneling is used for providing the traceroute capabilities at a service level. In the next section we will see traceroute capabilities at the transport tunnel level.

15.3.3.2 LSP traceroute

The LSPping mechanism described in Section 15.3.2.1 also supports a fault isolation mode modeled after the IP traceroute functionality and called

LSP traceroute, or LSPtrace. The idea is not just to report the hops in the path, but also determine if there is a mismatch between the forwarding and control planes. To accomplish this, two conditions must be satisfied:

- The MPLS echo request probe must be processed by each hop in the path.
- Enough information must be available in the probe to allow the transit LSR to determine that it is indeed a transit for the LSP under test.

LSPtrace uses the same packet formats, TLVs and echo request and reply mechanisms used by LSPing. For example, to trace the path of an LDP LSP, an MPLS echo request is sent from the head end, including the LDP FEC TLV, encapsulated with the correct label stack for the LSP, in this case the LDP label. To ensure that the echo request is received by each hop in the path, several such echo requests are sent, with increasing TTL values, just like a normal traceroute. To allow the transit LSR to check the control plane against the data plane, the LSR must know whether it is a correct recipient of the probe. Therefore, the MPLS echo request packet contains, in addition to the FEC TLV for the FEC under test, the list of acceptable recipients of the packet, from the point of view of the upstream LSR.

This list is encoded in the 'Downstream Mapping TLV' and contains identifying information (such as the router ID, label and interface) of all the possible downstream neighbors for the FEC, from the point of view of the upstream LSR. The LSR processing the echo request determines if it is a valid recipient of the traffic if it is listed in the Downstream Mapping TLV, with the correct label and interface information. If the check fails, it informs the head end of the failure in the echo reply, thus pinpointing the location of the problem. If the check succeeds, the LSR extracts all its valid downstream neighbors and labels for the FEC under test and sends this information in the echo reply, encoded in a (per-neighbor) Downstream Mapping TLV. This TLV is sent in the next echo request that will be sent with a TTL greater by 1, and therefore will reach its downstream neighbors.

This process is shown in Figure 15.8 for an LDP-signaled LSP between PE2 and PE1. The label distribution from PE1 to PE2 is shown along the links in the path. PE2 sends an MPLS echo request with a TTL value equal to 1. The request contains an LDP FEC TLV for FEC PE1 and a Downstream Mapping TLV listing neighbor C, as well as the label used by C, L4. When the TTL expires at C, the echo request is delivered to the control plane of router C for processing, along with the label stack with which the packet arrived. LSR C checks if it is one of the routers in the Downstream Mapping. Because the check is positive, C builds a new Downstream Mapping TLV, listing all its valid downstream neighbors for LDP FEC PE1 (in this case, router B with label L3) and sends it back to the head end in the echo reply.

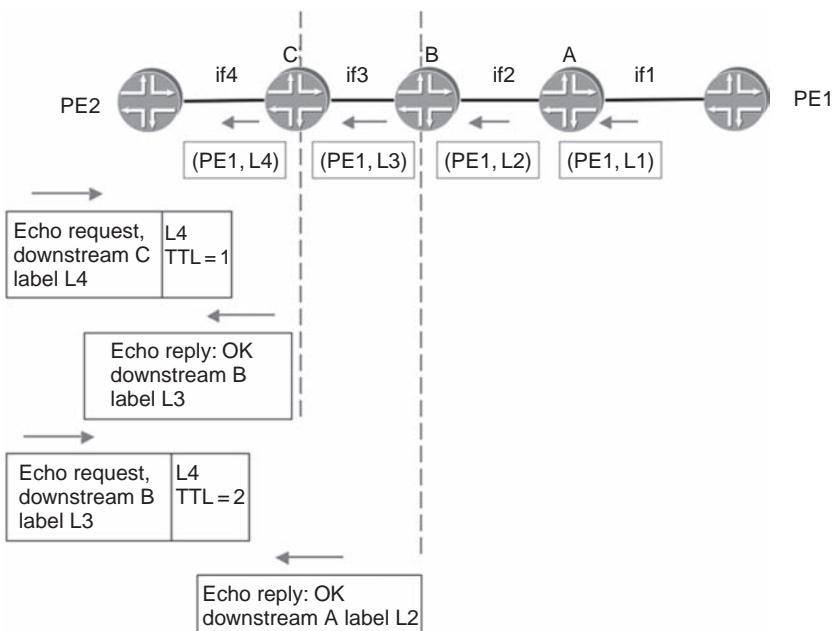


Figure 15.8 LSPtrace

This TLV is included in the next echo request probe sent by the head end, which is sent with TTL equal to 2 and will expire at B. When this probe is processed at B, the same process will be applied and so on (the rest of the steps are not shown in the figure). In this manner, the path of the LSP can be traced and at the same time the location of any failure can be reported.

The example discussed in the previous paragraph presents the simple case where a single downstream is available at each node in the path. Figure 15.9 shows a network where equal cost paths exist. Assume that an LDP LSP is set up in this network for the FEC representing router H's loopback address. Multiple paths exist between the ingress A and the egress H. However, by applying the LSPtrace procedures as described in the previous paragraph, only one of these paths will be traced. Recall from Section 15.3.2.2, which discusses LSPing properties, that echo request packets with different destination addresses will be forwarded differently in the network, because routers pick between ECMP paths based on the results of a hash function applied on the packet header, including the destination address.

If the head end knew what destination address to use, it could force traffic over an arbitrary path in the network. However, because the hash functions are implementation-dependent and vendor-proprietary, only the routers that apply the hash can determine which destination addresses

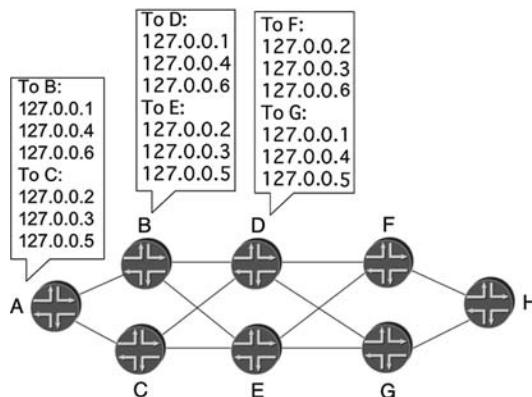


Figure 15.9 LSPtrace for an LDP LSP in a network with ECMP paths

exercise which paths. For this reason, [RFC4379] introduces a way to communicate this information to the head end along with the down-stream neighbor information, using the Multipath Information field of the Downstream Mapping TLV. (In cases where a single path is possible, this information is trivial: any address will exercise this path.) Figure 15.9 shows the addresses exercising different paths for routers B and D. Once A receives this information, it can determine that traffic can be sent to H along the path A–B–D–G–H by using the destination address 127.0.0.4 in the echo request and along the path A–B–D–F–H by using 127.0.0.6. Because such a determination can only be made based on the information provided by LSPtrace, it is necessary sometimes to run LSPtrace in conjunction with LSPping. For example, some implementations automatically couple LSPtrace with LSPping, when the goal is to test all possible paths for a particular LDP FEC. For example, the user may request OAM for a particular FEC. Under the hood, the implementation would first run LSPtrace to determine all the possible paths, after which LSPping would be initiated for state monitoring. From the discussion so far, it should be obvious that there are two assumptions for the correct operation of LSPtrace. Let us discuss them separately below and see under which circumstances they fail and what can be done to ensure correct LSPtrace operation in those cases.

1. *All routers in the path can correctly forward their replies back to the originator of the echo request.* This is somewhat similar to the example we have seen for ICMP tunneling in the previous section, where P routers in the core of the network had no reachability to the CE device which originated the probe. A simple example in the context of LSPtrace is an interdomain LSP providing connectivity across several ASs. For security

reasons, providers do not advertise the addresses of routers inside their domains to other providers. It could be argued that for this reason no traceroute capability should be available at all and the problem need not be solved. This is true, except in the case of a failure, where it is useful to know at least which network the failure happens in, in order to contact the responsible provider for further troubleshooting. [LSPING-INTERAS] presents a solution to this problem by extending the LSPing mechanism to build a stack of ASBR addresses as the trace progresses in the network and then use this stack to forward the echo reply. The main idea is that the ASBR addresses are known, and by forwarding a packet from any router to an ASBR, it can be easily relayed to the previous ASBR in the path and so on back to the head end. In addition to the usual LSPing error codes, the echo reply in this case would contain the AS number and the contact information of the provider responsible for the router where the failure was discovered. So far we have discussed the first assumption in the operation of LSPtrace, namely the ability to forward the reply back to the originator. Let us now look into the second assumption.

2. *When the path is free of errors, the recipient of an echo request can always successfully verify the label stack.* This assumption breaks in cases where tunneling is used. Refer back to Figure 15.8, but now assume that there exists an RSVP-signaled LSP between routers C and A and that the LDP-signaled LSP for PE1's loopback is tunneled over it (as explained in the Foundations chapter, Chapter 1). When PE2 traces the path towards PE1, it does so using the LDP FEC sub-TLV in the echo request. Router C can successfully perform the FEC validation required by [RFC4379], but router B, which is a pure RSVP node with no knowledge of the LDP tunnel, cannot do the same. Rather than returning a failure, one option is for router B to ignore the FEC validation and determine the next-hop information based on the incoming label stack in the echo request. However, this approach is not ideal, because it undermines the usefulness of LSPtrace as a troubleshooting tool and may fail to catch real errors. [LSPING-TUNNELED] proposes an elegant solution to this problem by allowing routers in the path to inform the head end that a different sub-TLV type needs to be used in the FEC stack when sending the next probe, by encoding this information in the Downstream Mapping TLV. In the example above, router C, which is the RSVP tunnel head end, informs router PE1 that for the next echo-request packet an RSVP sub-TLV rather than the LDP sub-TLV should be used and the validation of the echo request can be successfully completed.

To summarize, data-plane failures are difficult to detect and fix. For this reason, specialized tools such as the ones described throughout Section 15.3 were developed in order to discover their existence and pinpoint their

location. However, data-plane failures are not the only source of problems in the network. In the next section, we will look at configuration errors and their impact on the network.

15.4 CONFIGURATION ERRORS

Configuration errors are a common source of problems in network. Their impact can range from a service not coming up to traffic being routed to the wrong destination. There are two ways to deal with configuration errors. The first is to prevent the problem from happening in the first place, by improving the configuration process and by building mechanisms to reduce the amount of configuration needed to deploy a service. The second is to detect and report the misconfiguration, and try to protect the network from its ill effects. Let us discuss these two approaches separately below.

15.4.1 Preventing configuration errors

The basic idea behind preventing configuration errors is simple: fewer and simpler configuration statements means less probability for an error. Here are a few of the techniques used by various commercial implementations to put this idea into practice:

- (a) Minimize the amount of configuration that must be applied to enable a feature. The more configuration statements required, the bigger the chance of an error, especially when the different statements are sprinkled in several places in the configuration file. For example, recall from the L3VPN chapter (Chapter 7) that the definition of a VRF requires both an import and an export policy. For simple any-to-any connectivity, the two are often the same. Therefore, rather than requiring an explicit listing of each, both could be configured in one statement.
- (b) Use intuitive configuration statements. If the configuration is not intuitive, the chance of errors is higher. For an example, refer to Section 2.4.3 discussing link colors in the Traffic Engineering chapter (Chapter 2).
- (c) Avoid configuration when not necessary. For example, the bypass tunnels required for link protection can be dynamically computed and should not require manual configuration.
- (d) Apply the same configuration in multiple places automatically. For example, to enable forwarding for MPLS packets on a large number of interfaces, it should not be necessary to use the same configuration statement multiple times. Instead, it would be better to have a way to apply the configuration to multiple interfaces in one statement.

However, requiring fewer configuration statements in one router's configuration file is not the end of the story. The choices that the operator makes regarding the label distribution protocols and the services deployed directly impact the amount of configuration necessary. Here are a few examples:

- *The number of LSPs required.* In the chapter discussing DiffServ Aware Traffic Engineering (Chapter 4), we saw that LSPs can be set up with reservations for a single class type or from multiple class types. When multiclass reservations are supported, the total number of LSPs that must be set up in the network decreases, and so does the amount of configuration necessary to set them up.
- *Autodiscovery for the BGP-based Layer 2 VPN or VPLS solution.* When using a BGP-based solution, no extra configuration is necessary to identify the other members of the service, as explained in the Layer 2 Transport chapter (Chapter 12). In contrast, when setting up a mesh of pseudowires using LDP, the endpoints of each of the circuits and the targeted LDP sessions must be correctly configured on each box.
- *RSVP as a label distribution protocol.* When a full mesh of transport tunnels is required between all PEs, RSVP requires configuring tunnels on each of the PEs to all the other PEs. When adding a new PE to the mesh, tunnels must be set up from the new PE to all the existing PEs. However, the same tunnels must also be configured towards the new PE from all the existing ones. The problem in this case is not just the fact that a large number of LSPs need to be configured, but also that this configuration is spread over a large number of PEs. Because of this property of RSVP, many deployments prefer to use LDP as the label distribution protocol, as explained in the Foundations chapter (Chapter 1).

In the last two examples listed above, the requirement for extra configuration work is due to the nature of the service or protocol used. Therefore, it makes sense to look for a solution at the same level. For instance, there are ongoing efforts in the IETF to provide autodiscovery capabilities to the LDP-based Layer 2 solutions, as discussed in the Layer 2 Transport chapter (Chapter 12). One thing to note is that many of the proposals rely on deploying another protocol for autodiscovery, which means that more configuration work is required to set up and maintain the protocol. When comparing two competing solutions, one must keep in mind not just the functionality provided but also the cost of deploying the solution, especially on a large scale. In this context, the configuration effort plays a big part.

An interesting solution at the protocol level for a configuration scaling problem has been proposed for the RSVP full-mesh problem in [RFC4972]. Providers want to use RSVP not just for traffic engineering but also for its

fast-reroute capabilities. However, the burden of manually provisioning the full mesh of tunnels and of updating the mesh every time a new PE joins it constitutes a deterrent. The solution is to offload this burden on to the routing protocols themselves.

The idea is simple. Every PE has reachability to every other PE in the network because this information is distributed by the IGP. In principle, a PE could set up an RSVP LSP to any other PE in the network, if it knew that such an LSP was required. Thus, to build a full mesh of LSPs between PEs, all that is needed is for each PE to know who are the other members of the mesh. This can be easily accomplished by assigning an identifier to each group of PEs that must be fully meshed and have each PE advertise which groups it belongs to. Because the IGP distributes reachability information to all PEs and because it is already used for carrying TE information, it is an ideal candidate for distributing this mesh group membership information. The automesh proposal extends OSPF and IS-IS to carry a new TLV, the TE mesh group TLV, indicating what group(s) the PE belongs to and what is the address that should be used by the other PEs for setting up the LSPs towards it. Based on the advertisements received and based on the locally configured knowledge of mesh membership, each PE knows to which other PEs to set up LSPs.

When a new PE is added to the network, it is configured with the correct group membership. As soon as the IGP distributes the new PE's membership information in the network, all the other PEs can set up LSPs towards it, automatically (and the other way around). However, what are the properties of the LSPs that are set up this way? This is a matter of configuration of the mesh group properties. This configuration can be minimized if, for example, features like autobandwidth (discussed in the Traffic Engineering chapter, Chapter 2) are used.

To summarize, configuration errors can be avoided by minimizing the amount of configuration required. This can be done at the implementation level, by optimizing the configuration process, and at the protocol level, by building a mechanism to avoid the need for configuration. Examples of the latter are autodiscovery of VPN membership or automesh for RSVP-TE tunnels. However, as long as configuration is necessary, the possibility of errors in the configuration continues to exist. In the next section we will see how to detect and report misconfigurations.

15.4.2 Detecting and reporting misconfigurations

Because configuration always requires human intervention at one level or another, errors will continue to happen. Sometimes, the error causes an easily detectable problem, such as a routing peering not establishing. At other times, there is no immediate feedback on the problem and there is a

need to define new mechanisms for detecting the failure and dealing with it. Let us take a look at a few examples of protocol extensions for detecting and reporting errors caused by misconfigurations.

15.4.2.1 *Interface misconfiguration affecting LDP operation*

Recall from the Foundations chapter (Chapter 1) that LDP label distribution follows the IGP. When a new interface is added to the network and LDP is not enabled over it, a failure will occur (assuming the new interface causes a change to the shortest path). This is shown in Figure 15.1. Assume the interface if5 does not yet exist in the network. The LSP for FEC PE1 (the loopback or router PE1) establishes along the path PE2–C–B–A–PE1. At this point, the operator decides to add the interface if5, and includes it in the IGP, but forgets to enable LDP on it. As a result, the IGP best path for PE1's loopback on router C will be C–A–PE1. Because the label advertisements arrive over a different interface (C–B) than the IGP best path, the forwarding state for the LSP will be removed. If independent control is used, the forwarding entry will be changed to pop the label, but C will continue to advertise its label towards PE2. If ordered control is used, the forwarding entry will be removed and the label advertisement will be withdrawn. (For a detailed discussion of this scenario, refer to the Foundations chapter, Chapter 1). In both cases, no labeled traffic can be forwarded between the two PEs, causing interruption of the service if the two PEs are providing MPLS/VPN services.

Note that the same problem of tearing down the LSP would happen even if LDP were enabled on the new interface. The condition would persist until such time as the LDP session establishes over the new link. To avoid this situation, a mechanism is defined in [RFC5443]. The idea is simple: allow the user to specify the interfaces over which LDP is expected to run and advertise an infinite IGP metric for the link until the LDP session has come up and labels have been exchanged over it. In the example above, the new link A–C would be advertised with the infinite metric, and as a result the IGP best path would continue to go over the path C–B–A. Note that the cost of using this scheme is that the new link is avoided, because of its high metric. This affects not just the LDP traffic but the IP traffic as well.

What we have seen in this example is a mechanism for avoiding the ill effects of a configuration error for the LDP protocol. The undesirable consequences of the misconfiguration are avoided and an error can be reported when the condition is detected, thus allowing the operator to identify and rectify the problem.

15.4.2.2 *Common misconfigurations for VPN scenarios*

Assume a network with two VPN customers, A and B, using the same private address space. Two new sites are added on the same PE: CE1

in VPN A and CE2 in VPN B. Two common misconfigurations are possible:

- Assigning the customer interface to the incorrect VPN. Recall from the introduction to the L3VPN chapter (Chapter 7) that the decision to which VPN CE-originated traffic belongs is based on the interface over which the traffic arrives. If the link from CE2, instead of the one from CE1, is connected to the port configured for VRF A, then CE2 becomes a member of the wrong VPN, as shown in Figure 15.10. Assuming the same address spaces in both VPNs, traffic originating in VPN B is forwarded to destinations in VPN A.
- Configuring the wrong route target (RT). Recall from the introduction to the L3VPN chapter that correct access control between VPNs relies on accurate configuration of the route target. If a site in VPN A starts using the RT that was assigned for VPN B, then destinations in VPN B may become reachable from the site belonging to VPN A and vice versa (the exact outcome depends on whether the import RT, export RT or both are misconfigured).

In both of these cases, the problem is not just one of not providing the required connectivity to the new sites. Perhaps more importantly, the problem is one of violating the security guarantees offered to the two customers by allowing traffic to cross over from one VPN to the other.

For this reason, solutions have been discussed in the l3vpn Working Group in the IETF for handling such misconfigurations. Their goal is

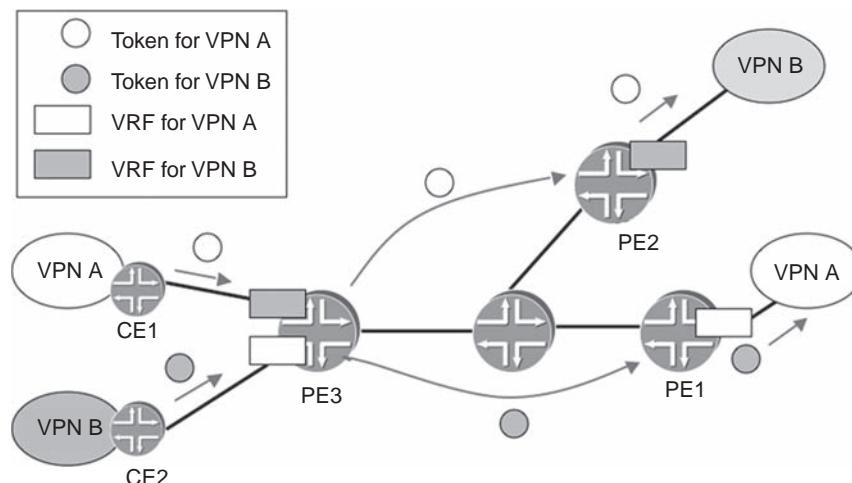


Figure 15.10 Using a token-based mechanism for VPN membership identification

twofold: alert the operator of the problem and prevent misrouting of traffic. Let us take a look at some of the proposals for solving each of the problems mentioned above.

Assigning the customer interface to the incorrect VPN

The most intuitive approach for handling this error is to enable authentication for the PE–CE routing protocol exchanges. Assuming different keys are used in each VPN, the routing protocol session does not establish and routes are not propagated to/from the misconfigured CE to the rest of the VPN. Failure to establish the routing protocol session also triggers error messages that alert the operator to the problem. However, the approach of authenticating PE–CE routing protocol exchanges may not always be feasible, e.g. in setups where no routing protocol is running on the PE–CE link.

For this reason, a new mechanism is proposed in [VPN-CE-CE-AUTH] that provides a CE-based mechanism for VPN membership verification. The idea is to allow the customers to detect security breaches caused by a misconfiguration of the provider network. Here is how it works. To join a VPN, each site sends a token to the PE, which in turn relays it to all the members of the VPN, using similar mechanisms as the ones used for route distribution in a VPN. Customer devices use the token to verify VPN membership. The receipt of an unexpected token indicates that an unauthorized site joined the VPN and, as a result, an alarm is triggered to the operator. In addition, the VPN site receiving the unrecognized token may choose to protect itself from unauthorized access by withdrawing from the VPN, e.g. by discarding VPN traffic sent to it or by withdrawing its routes. Figure 15.10 illustrates this mechanism. Note that at PE3, the two VPN sites are attached to incorrect VRFs. As the tokens propagate to other members of the VPN, the misconfiguration is detected. The actual details of the token implementation can be found in [VPN-CE-CE-AUTH].

Although this mechanism does nothing to prevent the misconfiguration itself, its deployment allows detection of the problem. One of its interesting properties is that it gives the customer control over detecting problems in the provider network and allows him or her to protect against security breaches caused by such problems, e.g. by withdrawing from the VPN.

Configuring the wrong route target

Controlling the access to a VPN is all about controlling access to forwarding information. Correct assignment of the interface to the VPN ensures that forwarding lookups are made in the table associated with the VPN. Correct configuration of the RT limits the information stored in this table. The problem is that these two mechanisms are disjoint.

When discussing incorrect assignment of interfaces to the VPN we saw that the simplest solution is to authenticate the PE–CE routing exchange.

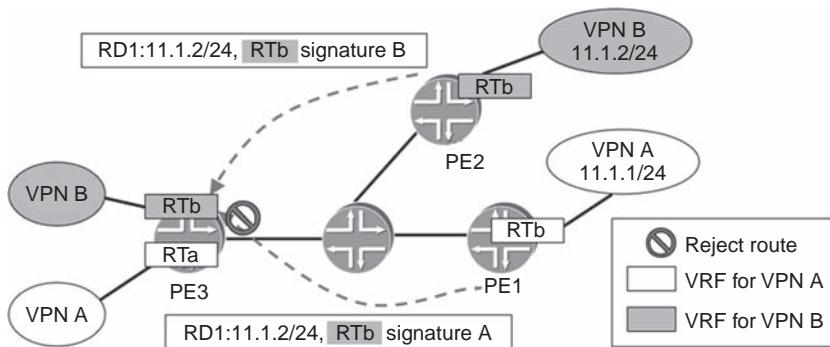


Figure 15.11 Detecting RT misconfigurations using routing update signatures

Could a similar approach be used to ensure that the correct RT is used? [VPN-RT-AUTH] attempts to take exactly this approach by validating PE-PE routing updates using routing update signatures. Note that the solution does not propose to authenticate the routing session but the individual route advertisements. When a VPN route is advertised between PEs, it carries a new BGP attribute, the BGP ‘update authenticator’. This attribute contains a signature generated using an MD5 key. When receiving such an update, the signature is checked against the MD5 key of the remote site, as shown in Figure 15.11. If the check fails, the route is not added to the VRF and the operator is notified. In this way, routes that do not belong to the VPN are not added to the VRF, the information in the VRF is limited to only the routes that are certified as belonging to the VPN and misrouting is prevented. Note that for this approach to work, unique MD5 keys must be assigned to each VPN.

To summarize, preventing configuration errors is possible in many cases. Good operation practices and intuitive, easy-to-use configuration interfaces can prevent many misconfigurations. For others, it is necessary to extend the protocols with built-in machinery for detecting the problem and preventing its ill effects.

15.5 VISIBILITY

A big part of management is about gaining visibility into the network. What exactly is visibility? The answer is different for different people. For example, for a network operations engineer, it may be the ability to see the path that traffic is taking or to determine if a site in a VPN went down, while for an engineer doing capacity planning, it may be the ability to build the traffic matrix between two PEs.

Management information bases (MIBs) play a big role in affording visibility into the network. As explained at the beginning of this chapter, MIBs are available for all the protocols and applications developed in the IETF and can be used to manage the network as described in [MPLS-NM]. MIBs can be used not just for reporting the state of a service or protocol but also for finding network-wide information that is not readily available otherwise. For example, for RSVP, the path of an LSP is known from the RRO Object, but for LDP, this information is not available from the protocol. However, using the label information from the LDP MIB, the path of the LDP LSP can be traced without the need to issue an LSP trace.

SNMP traps can be sent to indicate errors, such as an LSP being torn down. The receipt of the traps at the network management station provides not just an alarm indication to the operator but also valuable information about what failures happened at the same time, allowing correlation of events in the network and identification of the root cause of a problem, such as an interface going down, causing an LSP to go down and causing a VPN site to become unreachable.

Traffic accounting is perhaps the most important visibility feature in the network. It is important not just for billing purposes but also for network planning and debugging. LSPs are interesting in this context because they are treated by many implementations as virtual tunnel interfaces and have the same accounting features as interfaces. For example, some implementations allow the user to see the amount of traffic forwarded over the LSP in real time. This is a useful debugging tool, because it can show if traffic is forwarded along a particular LSP. Another example is the ability of some implementations to apply firewall filters to the LSPs (as explained in the context of policers in the DiffServTE chapter, Chapter 4). Thus, granular accounting features, e.g. taking into account the class of service, can be made.

Because it is easy to count traffic entering an LSP, LSPs have become an increasingly popular way to measure the traffic demands between two points in the network. For example, to find out the traffic matrix, LSPs with zero bandwidth requirements can be set up and the traffic statistics can be monitored over these LSPs.

To summarize, the different MPLS MIBs give visibility into the different components of the MPLS network and LSP traffic statistics give visibility into the traffic patterns in the network. This information can be used for both billing and for capacity planning.

15.6 CONCLUSION

Entire books are dedicated to the subject of managing the MPLS infrastructure and the services running over it; attempting to cover this subject in one chapter cannot do it justice. In fact, the topic is much broader than

just the functionality defined in the standards bodies. An entire industry is built around tools that can aid in tasks such as performance monitoring for a service or an LSP, VPN provisioning, network visualization, offline path computation, event correlation or capacity planning.

Rather than attempting to provide a view of all management functions and tools, this chapter focused on router functionality developed specifically for troubleshooting and managing in an MPLS environment, such as LSPPing, VCCV, ICMP tunneling and membership verification for VPNs. Although these topics constitute only a small part of the MPLS management story, they shed light on some of the unique issues that arise in MPLS environments.

15.7 REFERENCES

- | | |
|-------------------|--|
| [LSPING-INTERAS] | T. Nadeau and G. Swallow, <i>Detecting MPLS Data Plane Failures in Inter-AS and Inter-provider Scenarios</i> , draft-ietf-mpls-interas-lspping-02.txt, expired draft |
| [LSPING-TUNNELED] | N. Bahadur, K. Kompella, <i>Mechanism for Performing LSP-Ping over MPLS Tunnels</i> , draft-ietf-mpls-lsp-ping-enhanced-dsmap.txt (work in progress) |
| [MPLS-BFD] | R. Aggarwal, K. Kompella, T. Nadeau and G. Swallow, <i>Bidirectional Forwarding Detection (BFD) For MPLS Label Switched Paths (LSPs)</i> , in the RFC editors' queue, soon to become RFC5884 |
| [MPLS-NM] | T. Nadeau, <i>MPLS Network Management: MIBs, Tools, and Techniques</i> , Morgan Kaufmann, 2003 |
| [P2MP-LSPING] | S. Yasukawa, A. Farrel, Z. Ali and B. Fenner, <i>Detecting Data Plane Failures in Point-to-Multipoint MPLS Traffic Engineering – Extensions to LSP Ping</i> , draft-ietf-mpls-p2mp-lsp-ping-10txt (work in progress) |
| [PW-OAM-MSG-MAP] | T. Nadeau et al., <i>Pseudo Wire (PW) OAM Message Mapping</i> , draft-ietf-pwe3-oam-msg-map-.txt (work in progress) |
| [RFC3032] | E. Rosen et al., <i>MPLS Label Stack Encoding</i> RFC3032, January 2001 |
| [RFC4379] | K. Kompella and G. Swallow, <i>Detecting MPLS Data Plane Failures</i> , RFC4379, February 2006 |

[RFC4972]	J.P. Vasseur, J.L. Le Roux et al., <i>Routing Extensions for Discovery of Multiprotocol (MPLS) Label Switch Router (LSR) Traffic Engineering (TE) Mesh Membership</i> , RFC4972, July 2007
[RFC5085]	T. Nadeau, C. Pignataro, et al., <i>Pseudowire Virtual Circuit Connectivity Verification (VCCV) A Control Channel for Pseudowires</i> , RFC 5085, December 2007
[RFC5443]	M. Jork, A. Atlas and L. Fang, <i>LDP IGP Synchronization</i> , RFC5443, March 2009
[VPN-CE-CE-AUTH]	R. Bonica et al., <i>CE-to-CE Member verification for Layer 3 VPNs</i> , draft-ietf-l3vpn-l3vpn-auth-01.txt, expired draft
[VPN-RT-AUTH]	M. Behringer, J. Guichard and P. Marques, draft-behringer-mpls-vpn-auth-04.txt, expired draft

15.8 FURTHER READING

[RFC 4176]	Y. El Mghazli et al., <i>Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management</i> , RFC 4176, October 2005
[RFC4377]	T. Nadeau et al., <i>Operations and Management (OAM) Requirements for Multi-Protocol Label Switched (MPLS) Networks</i> , RFC 4377, February 2006
[RFC4378]	D. Allen and T. Nadeau, <i>A Framework for MPLS Operations and Management (OAM)</i> , RFC 4378, February 2006
[RFC4382]	T. Nadeau and B. Van Der Linde, <i>MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base</i> , RFC 4382, February 2006.
[Y1710]	ITU-T Recommendation Y.1710, in <i>Requirements for OAM Functionality for MPLS Networks</i> , 2002

15.9 STUDY QUESTIONS

1. One of the arguments against the LSPing solution (at the time it was originally proposed) was the use of a nonroutable address in the 127/8 range as a destination for the echo request. List some of the advantages of using this type of address.

2. Why are separate extensions necessary to handle LSPing for P2MP LSPs?
3. What type of information would the operator want to know for a P2MP LSPing echo request probe?
4. In the context of automatic reaction to failures, revertive behavior refers to adding back an LSP into the usable pool. What are some of the dangers in aggressively implementing revertive behavior?
5. Assume that an LDP-signaled LSP experiences a failure due to a mis-configuration as described in Section 15.4.2.1. What are the advantages/disadvantages of removing the LSP from resolution as opposed to applying the scheme described in [RFC5443]?
6. An LDP network with high levels of redundancy wants to deploy a scheme to monitor all LSP paths using BFD for MPLS. What would be the model of operation for such a scheme?
7. Imagine an inter-AS LSP spanning three ASs, AS1, AS2 and AS3. Assume at least two hops within each AS, not counting the inter-AS link. A failure is detected at the first node in AS3. Run through the procedures described in [LSPING-INTERAS] and describe the use of the Visited ASBRs Stack. Examine the additional load on the ASBRs as a result of applying this scheme.
8. Revisit the example of LSPtrace in a tunneled environment and apply the procedures described in [LSPING-TUNNELED] on this example.

16

MPLS in Access Networks and Seamless MPLS

16.1 INTRODUCTION

To date, MPLS has typically been used in the core part of service provider networks. However, there is now strong interest in using MPLS in the access part of the network. In this chapter, we describe the drivers behind this interest and discuss two models for employing MPLS for this purpose.

16.2 THE BUSINESS DRIVERS

The driving force for the use of MPLS in the access network is network consolidation. Having one type of access technology allows more efficient use of fiber resources and reduces the quantity and different types of network devices required. Network consolidation has been occurring in the core part of service provider networks, with services from multiple separate networks being consolidated onto one common platform based on MPLS. Consolidation is now also occurring in the access part of the network, in order to avoid the CAPEX and OPEX associated with operating several access networks in parallel. Consolidation can also mean that fewer physical sites are required to house equipment, with traffic being carried to more centralized locations. Ethernet is an important ingredient of the new access network, replacing the various legacy Layer 2

and TDM access technologies that have been used in the past. It was realized that while Ethernet provides a convenient link technology for access networks, Layer 2 Ethernet switching is inherently unsuitable as an aggregation scheme for a carrier-class access network. Such a network can be achieved much more easily using MPLS, with Ethernet as the link technology. Additionally, because MPLS is independent of the link layer, transition from legacy access networks is easier because MPLS nodes can support legacy interfaces as well as Ethernet interfaces. Having MPLS in both the core and the access network is simpler than having different technologies in each part of the network. As we see later in this chapter, MPLS also gives the service provider greater choice about where to locate service delivery functions.

In the following two subsections, we examine these trends in more detail. First we look at how the various different traffic types are handled in traditional access networks. We then discuss how these disparate legacy access methods are being replaced by Ethernet-based access. We show why Layer 2 Ethernet switching is an unsuitable technology for aggregation in the access network and demonstrate how MPLS meets the requirements of carriers wanting to build a reliable, unified access network.

16.2.1 The transition from legacy access to Ethernet access

Let us look at how access has traditionally been provided for the various services offered by a service provider. We then see how these technologies are being replaced by Ethernet. The following are some traditional access methods:

1. Some business customers use low-speed circuits (leased lines), for example $N \times 64$ Kbps, E1 or T1. These are used as point-to-point private circuits between their sites in different locations or as an access tail for Internet or Layer 3 VPN services. These circuits are carried over a PDH or SONET/SDH infrastructure.
2. Some larger business customers use higher-speed TDM circuits such as DS3 or OC3/STM1 SONET/SDH as access tails for Internet or Layer 3 VPN services. These are carried over a SONET/SDH infrastructure, which may be a separate network from case 1 above.
3. Layer 2 services are provided over an ATM or Frame Relay infrastructure. Some service providers use Frame Relay for lower-speed circuits and ATM for higher-speed circuits.
4. ATM and Frame Relay are sometimes also used as access tails for L3VPN or Internet services.

5. ATM is used as an aggregation scheme for broadband DSL traffic. In some cases, the ATM network is separate from the one used in cases 3 and 4 above.
6. Voice traffic ('Plain Old Telephony Service' or POTS) and low-speed dial-up data traffic are carried over a Public Switched Telephone Network (PSTN) infrastructure. Circuits are aggregated by Class 5 switches inside the Local Exchange (Central Office) or closer to the user within remote concentrator units. The aggregated circuits are carried over TDM infrastructure.

As can be seen from the list above, several different access networks are used to support the range of services supplied to residential and business customers. Let us examine how the access technologies in this list are being replaced by Ethernet.

As discussed in Chapter 12, Layer 2 Transport, it is often the case that point-to-point private circuits (leased lines) are used simply for LAN interconnection between a customer's sites. Therefore, it is not a fundamental requirement in those cases for the service to be provided using a TDM circuit.¹ As a result, some service providers are now offering Layer 2 service with Ethernet presentation as an alternative to the private circuits.

Similarly, in the case of Layer 2 services carried over ATM and Frame Relay (item 3 in the list above), the end customer often has no specific need for ATM or Frame Relay encapsulation. As a result, these services are also being replaced by Layer 2 Ethernet services, with VLAN tags providing the equivalence to ATM VCs and Frame Relay DLCIs.

Similar considerations apply to cases in which leased lines or higher-speed TDM circuits, ATM or Frame Relay are being used as access tails for Internet or L3VPN service. Again, there is no particular requirement on the part of the end-customer to have these particular technologies as access circuits – they just happen to be this way because this was the technology offered by the service provider when the customer purchased the connection. These access tail circuits are also being replaced by Ethernet access tails.

The use of Ethernet instead of leased lines, Frame Relay or ATM is more convenient to the end customer because typically Ethernet is more popular within enterprises. Also Ethernet provides more flexibility in terms of access rates compared to leased lines. When using leased lines, bandwidth needs to be provisioned in discrete steps. For example, a customer might start with an E1 or T1 circuit and then progress to several parallel E1 or T1 circuits and finally to an E3/DS3 circuit. Each upgrade in

¹If a TDM circuit is in fact required, for example for Private Branch Exchange (PBX) interconnection, then it can be emulated using a TDM pseudowire. This is discussed further in the Conclusions chapter.

capacity involves effort from both the service provider and the customer, including installing new line cards in the network equipment and adding new configuration. Also, the customer may have to get involved in link bundling technologies, such as multilink PPP (MLPPP), if they are using several parallel E1 or T1 circuits. If Ethernet is used, the customer can be presented with a 100-Mbps tail and the service provider can rate-limit the customer's traffic to an agreed level, which can easily be changed through simple configuration on the service provider's side if the customer needs to upgrade their traffic rate.

For broadband DSL aggregation, DSLAMs with gigabit Ethernet uplinks are now being deployed. These provide greater uplink capacity at a lower cost than the previous generation of ATM-based DSLAMs and so are better suited for new higher bandwidth DSL technologies such as ADSL2+ and VDSL2 and high-bandwidth triple-play services such as IPTV. In some places, fiber access schemes, for example Fiber to the Home (FTTH), are being deployed instead of DSL. The associated access nodes have gigabit Ethernet rather than ATM uplinks.

Network devices called Multi-Service Access Nodes (MSANs) have recently become available. These typically have Ethernet uplinks. They are a key component to allow service providers to migrate from circuit-switched TDM PSTN infrastructure to an IP-based PSTN infrastructure. MSANs combine DSLAM functionality with the ability to convert POTS and ISDN circuits arriving over twisted pairs from customers' premises into voice over IP, from both the bearer and the signaling point of view. From the point of view of the access network, the equipment used to perform TDM grooming of voice circuits using remote concentrator units and switches in the local exchange (central office) can be replaced by MSANs and an Ethernet-based aggregation network. In parallel, DSL service providers are offering VoIP services to residential customers and small businesses, which can be used as replacements for POTS or as a supplementary way of providing additional lines. In the rest of this chapter, we use the term Access Node (AN) when referring to access devices such as DSLAMs and MSANs.

As can be seen from this discussion, just as IP is becoming the technology for service convergence (for example VoIP and IPTV), Ethernet is becoming the access technology for a wide range of network services. The service provider can obtain CAPEX and OPEX savings by using shared Ethernet links and shared aggregation devices within the access part of the network to carry traffic belonging to the various types of service, thus gaining statistical multiplexing advantages. Of course, it is not a trivial task to migrate to such a scheme from multiple legacy networks, because there is impact on provisioning, operations, OSS, billing systems and so on, as well as on the network layer itself.

The use of Ethernet as a common access scheme is often called 'Metro Ethernet' although its applicability is more general than just metropolitan

areas. In the next section, we discuss the following question: what is the appropriate technology to aggregate the traffic as it travels through the access network towards the core?

16.2.2 MPLS as the technology choice for the Ethernet access network

Some early deployments of Ethernet in the access network used Ethernet switches in much the same way as they are used within enterprise networks. That is, they operated at Layer 2 as learning bridges with the forwarding based on MAC addresses. While this scheme is acceptable for enterprise networks within one site, it was never intended for use in service provider networks. The following issues thus arose:

1. *Flooding.* Ethernet uses data-plane based address discovery. If an Ethernet frame has an unknown MAC address, it is flooded on all ports except the port on which it arrived. This flooding makes traffic patterns difficult to predict and results in the wasting of bandwidth.
2. *MAC learning.* For a large deployment, a large number of MAC addresses may need to be learned. The number of MAC addresses may be more than the Ethernet switches can cope with. Furthermore, when the network topology changes, switches have to ‘forget’ learned MAC addresses, then re-flood and relearn them. This can adversely affect convergence times.
3. *Forwarding loops.* These are very disruptive because Ethernet has no time-to-live (TTL) mechanism. This means that frames can loop ‘forever,’ sometimes resulting in saturation of links in the network. The Spanning Tree Protocol (STP) was designed to prevent this problem by automatically creating a loop-free topology, by not using certain ports if necessary. However, STP is not infallible. For example, in certain cases if the link between a pair of Ethernet devices is broken in only one direction, forwarding loops can still occur. In such cases, often the only solution is for the operator to keep disabling ports around the network until the forwarding loops stop, which disrupts legitimate traffic.
4. *Convergence times.* The use of STP results in long convergence times (tens of seconds). Although improvements such as Rapid Spanning Tree Protocol (RSTP) can result in shorter convergence times (a few seconds), these times are still too long for some applications such as video.
5. *Load-balancing.* The use of STP means that certain links in the network are not used. This may be fine in an enterprise environment, but in

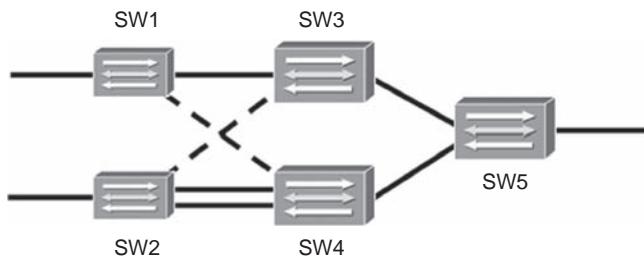


Figure 16.1 Wasting of bandwidth due to STP

in the service provider environment the links may be several kilometers long and hence have significant expense associated with them. This is illustrated in Figure 16.1. The dotted lines show links that have been disabled by STP to prevent loops. This behavior of STP means that in general, traffic cannot be load-balanced over different paths through the network. For example, traffic from SW1 traveling via SW5 cannot be load-balanced over the paths through SW3 and SW4. The only special cases in which load-balancing can be used are as follows:

- (a) Where multiple parallel links between a pair of Ethernet switches and link aggregation are used, traffic can be load-balanced over the links, for example, over the pair of parallel links between SW2 and SW4.
 - (b) Multiple Spanning Tree Protocol (MSTP) allows ports to be blocked on a per-VLAN basis. Thus some degree of load-balancing can be attained by selectively enabling VLANs on the various available paths.
6. *Admission control and traffic engineering.* There is no method to create bandwidth reservations along the traffic path. The only scheme for differentiating between traffic types is using per-hop behaviors based on the values of the IEEE 802.1p bits carried in the header of each Ethernet frame.
 7. *Troubleshooting & Maintenance.* The fact that the address discovery is data-driven causes operational problems. With control-plane based address discovery, the operator can first check the control plane to determine what state there should be in the network. They can then check in a hop-by-hop fashion both the control-plane and data-plane to narrow down issues. In contrast with Ethernet's data-plane initiated forwarding state it is difficult to narrow down connectivity issues as there is no sense of what the state should be. Typically remedies are to unplug cables or to reboot switches, resulting in service downtime even for customers that were not affected by the initial problem.

Not surprisingly, a good solution to the above problems is to use MPLS as the switching mechanism instead of Ethernet switching. This brings many advantages:

1. MPLS is already deployed as the technology for converged networks in the core of service provider networks because it provides a good fit for the requirements. These requirements include the following:
 - (a) Common control plane and data plane scheme for carrying a wide variety of traffic types.
 - (b) For selected types of traffic, low amount of traffic loss following link or node failure.
 - (c) Ability to control the path taken through the network, and to load balance traffic across multiple paths.
 - (d) Ability to reserve bandwidth for certain types of traffic.

In the access part of the network, the requirements must be the same as in the service provider part because it is the same traffic that is being carried! For example, if for certain traffic types, it is important to have fast convergence times, this requirement must hold true for all sections of that traffic's journey through the network, whether the core part or the access part. Hence, it makes sense to also have MPLS in the access part of the network.

2. Having the same scheme in the access and core parts of the network makes operations easier because there is only one type of scheme to set up and maintain. Also, it allows more flexibility in terms of how separate or entwined the core and metro domains are. We look at this in more detail in the next section.

In this model, the MPLS switches would typically have Ethernet ports, because the presentation to the end-customer, or the uplinks on the Access Nodes is Ethernet. In some cases, the Access Nodes themselves may support some MPLS functionality. It is also convenient to have Ethernet links between devices within the access network because they are available in convenient sizes (1, 10, 40 and 100 Gbps). In this way, Ethernet is being used simply as a link-layer encapsulation technology, not as the forwarding scheme within the switches or the Ethernet control plane (see Table 16.1). Note that the use of MPLS switches allows the service provider to migrate the access infrastructure to Ethernet without necessarily having to immediately change the customer-facing tail circuits (which may be ATM or Frame Relay) to Ethernet. These can be migrated later, when the customers are ready.

Sometimes people say that the cost of IP/MPLS routers compares unfavorably with the cost of an Ethernet switch. However, such a

Table 16.1 Comparison of Ethernet switching and MPLS as access network technologies

	Ethernet switching	MPLS
Forwarding scheme	MAC address	MPLS label
Control plane	Limited control plane: spanning tree protocol to create loop-free topology. But no control plane for address discovery	IGP and RSVP and/or LDP ²
Link technology	Ethernet	Ethernet

comparison does not compare like with like. A fully fledged IP/MPLS router has the following features that are not present on an Ethernet switch:

1. Supports a wide range of media types.
2. Performs longest match IP lookup at wire rate in a table containing hundreds of thousands of prefixes and supports complex multi-field packet filters.
3. Offers IP services such as stateful firewall and NAT.
4. Has carrier-class reliability, including hardware redundancy.
5. Full-fledged control plane

Note that the MPLS switches required for the access network do not require any of the functions listed above, except for item 4; also, item 1 is required if a deployment requires support for legacy tail circuits. MPLS switches do not need to be capable of high-speed forwarding of IP packets, because all the traffic is encapsulated in MPLS. The only IP packets that the switches are required to handle are the control plane packets. It has been pointed out that there should be no inherent cost difference between an Ethernet switch and an MPLS switch if one strips out the functions that are not necessary for an MPLS node in an access network and adds to the Ethernet switch the cost of providing the missing carrier-class reliability [MPLSWC-2007].

16.3 MODELS FOR MPLS DEPLOYMENT IN ACCESS NETWORKS

We now examine in more detail how to deploy MPLS in the access network. First we need to discuss some terminology.

²Also labeled BGP in the Seamless MPLS scheme discussed later.

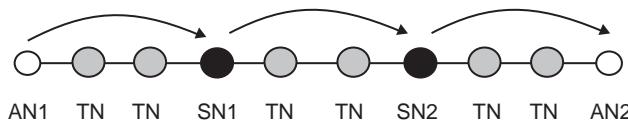


Figure 16.2 High-level end-to-end connectivity blueprint

In section 16.2.1, we introduced the term Access Node (AN) as a generic term for access devices such as DSLAMs and MSANs.

A *Service Node (SN)* is a generic term to describe the various types of node on which specific services to customers are created and delivered. Examples of SNs are as follows:

1. In the case of DSL, the SN can be a Broadband Services Router (BSR). It is on the BSR that the subscriber is identified and authenticated and where policies such as the access rate are applied. In some DSL architectures, an additional SN may be used, for example, a Video Services Router (VSR) for video services.
2. In the case of MPLS VPN services offered to enterprise customers, the SN is the PE router. For example, for L3VPNs, the VRFs containing the customers' routes are on the SN, and inbound policing and outbound shaping and queuing are applied here. As we see later, the SN does not necessarily need to be located at the 'Provider Edge', so the term 'PE' can be a misnomer.

A *Transport Node (TN)* is an MPLS node that is neither an AN nor an SN. It provides the connectivity between the Access Nodes and Service Nodes.

In general, the path taken by a packet through the network is as illustrated schematically in Figure 16.2. The packet is carried from the ingress AN, AN1, to one of its parent SNs³, SN1, typically via one or more TNs. SN1 maps the packet into the appropriate service context, for example, a particular L3VPN or subscriber management instance. It then determines to which of the other SNs in the network the packet should be forwarded. For example, for a L3VPN service, this determination is performed by an IP lookup in the VRF. The packet is then sent to that SN (SN2), typically via one or more TNs. SN2 identifies the appropriate egress AN, AN2, and forwards the packet to it, again typically via one or more TNs.

There are two main methods of deploying MPLS in the core and access network. We refer to these as Option 1 and Option 2.

For Option 1, the core network and each access network are separate MPLS islands, with a non-MPLS hand-off between the islands. The SNs

³In this chapter, we use the term 'parent SN' to denote the SNs that provide the services to the customers attached to a particular AN.

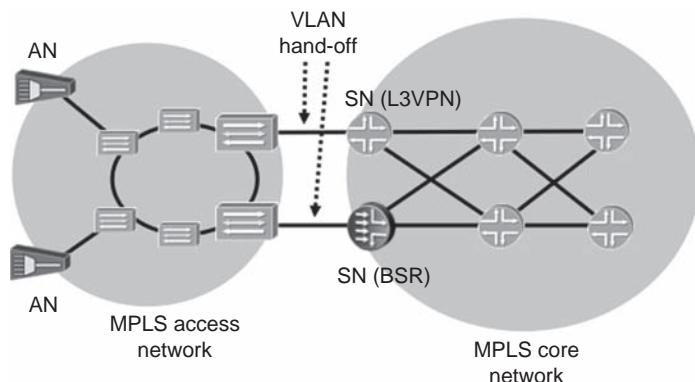


Figure 16.3 Separate MPLS core and access networks

are located in the same place as they were before the MPLS access network was built – at the outer edge of the core network. Hence, the MPLS access network is being used as a backhaul method to transport traffic from the edge to the SNs. That is, it is a direct replacement for the previous access schemes. This is illustrated in Figure 16.3.

In the case of Option 2, known as Seamless MPLS [SEAMLESS] [MPLS2008], the core and access networks form one unified MPLS network. This means that there is no longer a hard boundary between the core and access parts of the network, as illustrated in Figure 16.4. It may be the case that the core and access networks are in different IGP areas or even in different ASs (this may be required for scaling or for administrative reasons; see Section 16.4.2), but from the MPLS point of view, a customer's traffic is carried all the way from one end to the other in MPLS.

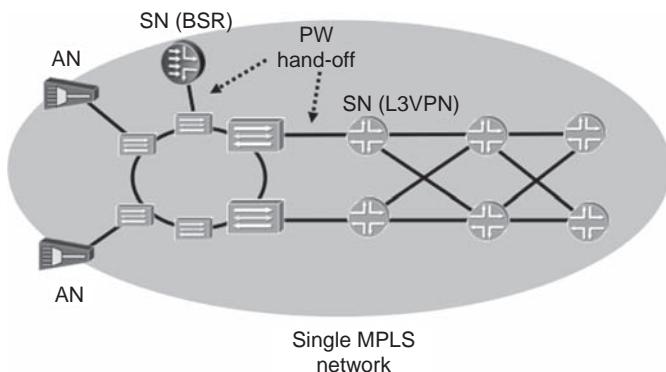


Figure 16.4 Unified MPLS core and access network

The advantage of the Option 1 scheme is that if the access and core networks are operated by different departments, there is still a clear separation between the two. The disadvantage is that there is no flexibility with respect to the location of the SNs. In contrast, with Option 2, the SNs can be located anywhere in the network, and can be distributed differently for different services. For example, for a mainstream service with a high demand volume SNs can be located in each access network. On the other hand, if the service provider wishes to trial a niche service, one or two corresponding SNs can be initially placed in a central location. If the service subsequently proves popular, more SNs can later be deployed in a more distributed way. Indeed, in principle the SN functionality can be placed on the AN, although typically, at the time of writing, ANs tend not to have such capabilities.

Another difference between Option 1 and Option 2 is the nature of the hand-off between a TN and the attached SN. In either case, to transport traffic between the outer edge of the access network and an SN, pseudowires can be used. Note that in this role, the pseudowires are being used as internal transport infrastructure within the service provider, rather than providing an explicit pseudowire service to an end customer. Recall from Chapter 12 that a pseudowire normally has two end-points (the two PEs providing the pseudowire) and an attachment circuit on each PE, such as a VLAN, to provide the hand-off to the client of the pseudowire. In most cases today, for a given pseudowire, the PE at one end would be the TN facing the AN, because most ANs today do not support MPLS or pseudowires. Note however that extending MPLS all the way to the ANs is a topic of interest especially in the context of Option 2 (Seamless MPLS) schemes. We discuss such scenarios in Section 16.4.1.

Let us now compare how traffic from the transport pseudowire is handed off to the SN. Figures 16.5 and 16.6 show how this is done for Option 1 and Option 2, respectively.

In Option 1, there is no MPLS connectivity between the SN and the access network. Hence the PE function for the pseudowires resides on the

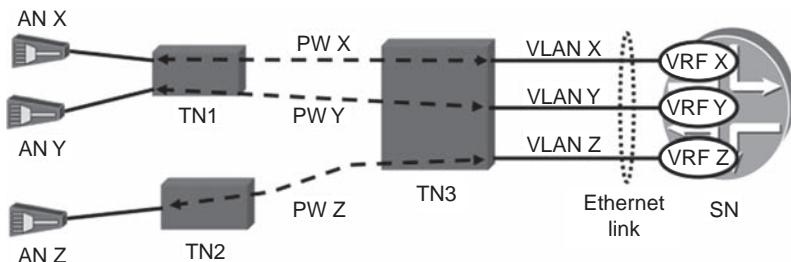


Figure 16.5 Backhaul of VPN traffic over pseudowires terminated on the transport node (TN3)

TN facing the SN, that is TN3. The attachment circuit for each pseudowire is a VLAN between the TN3 and SN, and from the point of view of the pseudowire, the AN is a CE. On the SN, the VLAN is mapped to the required service, for example a L3VPN VRF, a VPLS instance, a L2VPN instance or a subscriber management function. This is illustrated in Figure 16.5. Three customers, X, Y and Z, have L3VPN service. The corresponding ANs are AN X, AN Y and AN Z, respectively. The SN is responsible for providing the L3VPN service, so the customer VRFs are configured on that node. AN X and AN Y are connected to transport node TN1. AN Z is connected to transport node TN2. PW X runs between TN1 and TN3. The corresponding attachment circuit on TN1 is VLAN X, which is terminated in VRF X on the PE. Similarly, PW Y and PW Z are connected to VLAN Y and VLAN Z and then to VRF Y and VRF Z respectively.

In Option 2 (Seamless MPLS), the PE function for the pseudowire resides on the SN itself, so TN3 is simply acting as a P-router. This means that there is no physical attachment circuit for the pseudowire at the SN: the pseudowire is coupled into a L3VPN VRF, a L2VPN instance, or a VPLS instance within the SN. This is illustrated in Figure 16.6. As can be seen, PW X, PW Y, and PW Z go all the way to the SN and are coupled into their respective VRFs, so no VLANs are required on the link between TN3 and SN. As well as the one-to-one mapping between PW and VRF illustrated in Figure 16.6, other mappings can be useful. For example, frames from a given pseudowire can be mapped into different VRFs (or other services) according to their VLAN tag values. Such multiplexing of traffic can be useful to reduce the number of pseudowires present in the network.

Option 1 has the advantage that it is closer to the operating model traditionally used with VPN PE routers, with the customer-facing circuit on the PE being a VLAN (or traditionally an ATM VC or Frame DLCI). If MPLS is being deployed in the access part of the network as a replacement for Ethernet switching, no changes are required on the VPN PE router. This type of deployment also provides a more clear-cut demarcation between

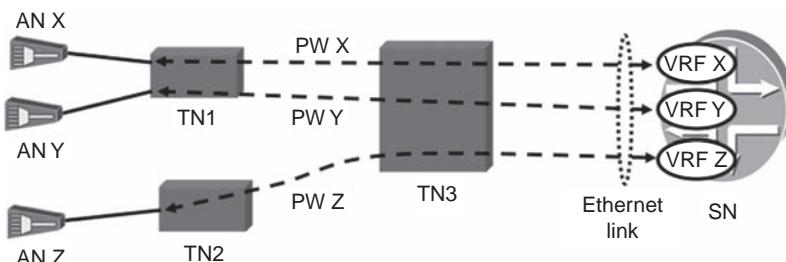


Figure 16.6 Backhaul of VPN traffic over pseudowires terminated on the service node (SN)

the core and the metro parts of the network. On the other hand, Option 2 has the advantage that no customer-specific configuration is required on TN3 because it is simply acting as P-router. In contrast, when Option 1 is used, each time a new customer is provisioned on an SN, a new VLAN needs to be configured between the SN and TN3 and a corresponding pseudowire needs to be configured on TN3.

Note that if Option 2 is being used, the SN needs to be capable of applying the same functions to the pseudowire as it would traditionally apply to a VLAN (or VC or DLCI) PE–CE circuit. For example, for L3VPN service this typically includes segregating traffic into multiple queues and shaping the outbound traffic to an agreed rate.

For some operators, it may be the case that they start using Option 1 because it is more similar to their traditional operating model, and then migrate to Option 2. A great advantage of using MPLS both in the core and the access parts of the network is that it makes migration much easier, compared to a situation in which MPLS is used only in the core part of the network and some other technology is used in the access part.

At the time of writing, the majority of MPLS deployments in the access network were similar to the Option 1 scheme rather than the Option 2 scheme. In Appendix A, we look in more detail at examples of how specific services are deployed using Option 1, focusing especially on how residential broadband models are mapped to this scheme.

In the next section of this chapter, we examine the Option 2 (Seamless MPLS) architecture in more detail.

16.4 SEAMLESS MPLS MECHANISMS

In this section, we look at Seamless MPLS in more detail. We first look at how MPLS can be extended to the Access Node (AN). We then look at how end-to-end connectivity is achieved in the Seamless MPLS architecture. Finally we discuss how Seamless MPLS is achieved for multicast traffic.

16.4.1 Extending MPLS to the Access Node

As mentioned in Section 16.3, one desirable goal of Seamless MPLS is to extend MPLS all the way to the AN. One way to achieve this is for the AN to support full MPLS functionality, including the ability to signal transport LSPs to the SNs and the ability to signal pseudowires using IP service provisioning protocols like LDP or BGP. This option means that the TNs between the ANs and the SNs are pure P-routers for MPLS LSPs between ANs and SNs and do not have any awareness of individual pseudowires. However, in some cases ANs may be incapable of supporting

a full MPLS control and data plane, so schemes in which the ANs need only a limited amount of MPLS functionality are of interest. In such cases, the TN attached to the AN provides some additional functionality to achieve the required connectivity between the AN and the SN. Let us look at two such proposed schemes.

16.4.1.1 Static MPLS labels on the AN

Suppose the ANs in Figure 16.7 have multiple ports (local loops) on the access side, labeled 1 to k, each corresponding to a different end-customer. These could be residential customers or business customers. The ANs do not support any signaling protocols for pseudowires or transport LSPs. Instead, the ANs use a static MPLS label assignment, with each label value corresponding to a different access port. One way to achieve this is to have a label base, from which the label to be assigned to each port is derived. For example, suppose AN1 uses a label base of X. The label assigned to port k could be $X + k$. This is illustrated in Figure 16.7, in which the label base is 1000, so the label assigned to port 2 is 1002. AN1 forwards packets from port 2 with this MPLS label value to TN1. TN1 signals, via BGP or LDP, a pseudowire for each attached AN to the SN. Suppose that pseudowire label X is assigned to the pseudowire corresponding to AN1 and that pseudowire label Y is assigned to the pseudowire corresponding to AN2. TN1 pushes the appropriate pseudowire label onto each MPLS packet arriving from an access node and also pushes the transport label W corresponding to the LSP whose egress is the SN. In this way, a three-label stack is created. As shown in the figure, AN1 and AN2 can use the same value for their label base without causing ambiguity, because the pseudowire label applied by TN1 denotes to which AN traffic belongs. Using the same label base means that the ANs can have the same configuration template which makes rollout operations easier.

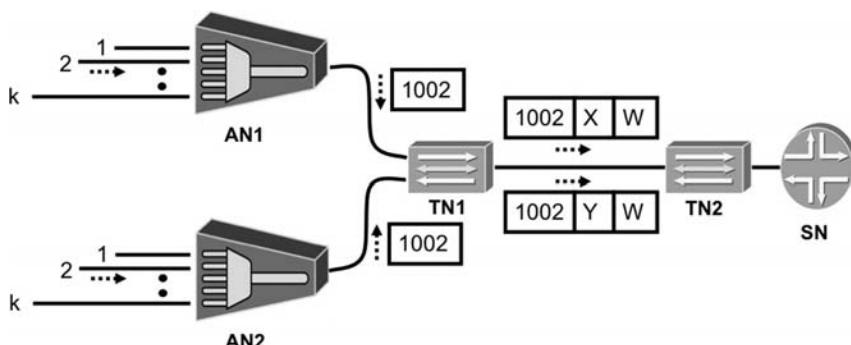


Figure 16.7 Extending MPLS to the access node using static labels

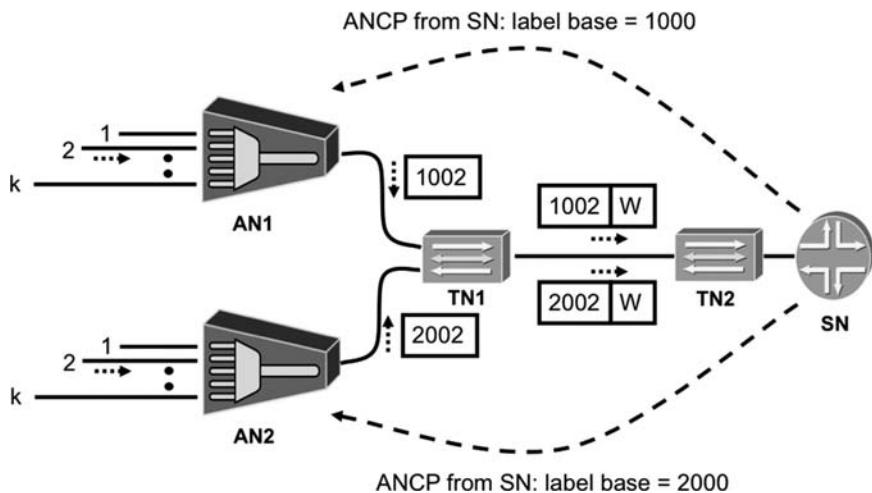


Figure 16.8 Extending MPLS to the access node by ANCP signaling

16.4.1.2 Label assigned to AN via ANCP

An alternative scenario to the static label allocation scheme at the AN described in the previous section is one in which the AN supports the Access Node Control Protocol (ANCP) [ANCP]. ANCP was designed to provide a control plane between SNs and ANs in the context of subscriber management. To extend MPLS to the AN, it has been proposed that the SN use ANCP to distribute pseudowire label values to ANs. Each AN uses ANCP to inform the SN how many access ports it has. This information triggers the SN to signal corresponding label values (or a label block) to the AN, over the ANCP session. This is illustrated in Figure 16.8, in which the SN assigns a label base of 1000 to AN1 and a label base of 2000 to AN2. When a packet arrives on an access port, the AN pushes the corresponding MPLS label assigned by the SN. For example, in Figure 16.8, packets arriving on port 2 will be tagged with label 1002 by AN1 and 2002 by AN2. The AN sends the packet to the local TN, TN1. TN1 in turn pushes the MPLS transport label, W, required to reach the SN.

16.4.2 Seamless MPLS scaling

In the Option 1 scheme discussed in Section 16.3, MPLS scaling is not a particular issue, because the core network and each of the access network islands are separate MPLS networks containing a manageable number of

nodes. In contrast, in the Option 2 (Seamless MPLS) scheme, the core and access networks form one unified MPLS network. Hence, an important topic to discuss is how MPLS scales to a network of this size, bearing in mind that the network might contain on the order of 100,000 nodes, if one includes all the ANs, SNs, and TNs.

One option is to divide the network into multiple OSPF areas or ISIS levels. For example, the core part of the network is OSPF Area 0 or ISIS Level 2, and each access network is a leaf area. Each SN needs an MPLS LSP to each of the other SNs in the network (or at least to the ones involved in the same type of service). Additionally, each AN (or TN facing the AN, if MPLS does not extend all the way to the AN) needs an LSP to its parent SNs. When people first started to think about the MPLS scaling, they worked on the premise that RSVP-signaled or LDP-signaled LSPs would be used throughout to provide the necessary connectivity. When considering the RSVP case, the result was quite an alarming numbers of LSPs. For example, if there are of order 1000 SNs fully meshed with RSVP-signaled LSPs, there would be of order one million RSVP-signaled LSPs in the network for the SN to SN connectivity, because the number of LSPs grows as the square of the number of SNs in the mesh. Such numbers could result in scaling bottlenecks if certain routers in the core of the network carry a sizable proportion of these LSPs. One possible solution to this would be to use LSP hierarchy, as described in Chapter 1, by nesting multiple SN-to-SN LSPs within a smaller number of LSPs that cross the core of the network. In this way, the existence of the SN-to-SN LSPs is ‘hidden’ from the core routers.

In the LDP case, the way LDP is currently specified [RFC5036] and implemented has the following consequences:

1. Typically each node announces itself as the egress for one FEC, which is a /32 loopback address. To install the label associated with a FEC, a node must have an exact match, that is to say the same /32 address (rather than a longest match such as a /24) in its routing table, usually from the IGP. This means that each node needs to have in its routing table a /32 address corresponding to each FEC. As a result, at IGP area boundaries, full summarization cannot be used. Although the link addresses can still be summarized in one prefix, the loopback addresses cannot, because each individual /32 loopback address must be advertised by the IGP throughout all the areas for LDP to operate correctly. This situation leads to a large IGP database if the number of nodes in the network is large.
2. The number of FECs grows in proportion to the number of nodes. Each node in the network needs to process every FEC, exchanging label bindings for each with its LDP neighbors, storing each in its LDP database, and creating corresponding forwarding table entries.

Several schemes have been proposed to address these issues. One [RFC5283] proposed relaxing the LDP match rules to allow an LDP label to be installed if the routing table contains any prefix that encompasses the address advertised in the FEC. In this way, full summarization can be performed at the IGP area boundaries. Other proposals were to add hierarchy and aggregation to LDP to remove the need for each router to have a FEC for every individual node in the network [LDP-SCALE] [LDP-AGG].

However, it was realized that these solutions to RSVP and LDP scaling solve the problem at the wrong layer and that a cleaner approach is to use labeled BGP [RFC3107] between IGP areas. In this way, LDP or RSVP is confined to individual areas and so does not cross area boundaries. As well as avoiding the LDP and RSVP scaling issues, this scheme means that some areas can use LDP and other areas can use RSVP if desired. The labeled BGP scheme for Seamless MPLS is similar to Interprovider VPN Option C or Carriers' Carrier in the sense that labels for the loopback addresses of nodes are distributed using BGP. The difference is that labeled BGP is being used within an AS rather than across AS boundaries.

Let us examine how the BGP-based scheme for Seamless MPLS operates. We need to introduce an additional term, a Border Node (BN), which is a node at the border between different regions in the network. For example, when OSPF areas are used in different regions, each BN is an OSPF Area Border Router (ABR).

Here are the key points about the scheme:

1. Within each region, LDP or RSVP signaled LSPs are used. RSVP can be used in some regions and LDP in other regions if desired. No LDP or RSVP LSPs cross region borders. This is illustrated schematically in Figure 16.9. On the BNs, steps are taken to prevent LDP FECs from being advertised from one region to another.

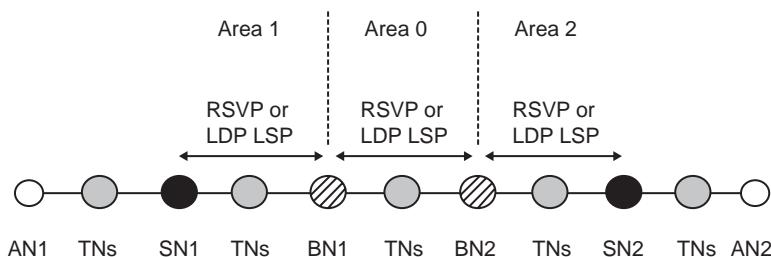


Figure 16.9 High-level end-to-end connectivity blueprint over multiple areas

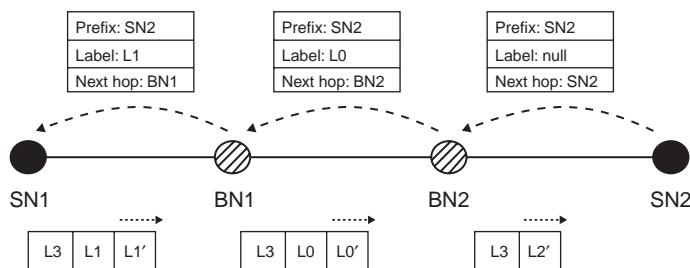


Figure 16.10 Use of labeled BGP for inter-area MPLS connectivity

2. No prefixes are carried in the IGP across region borders.⁴
3. To achieve inter-region MPLS connectivity between SNs, a forwarding hierarchy is created. For example, to send traffic to SN2, SN1 pushes onto each packet a label stack having on top a label corresponding to the local transport LSP to BN1 and a label corresponding to SN2 immediately underneath. The details of how this is achieved are discussed later in this section. This use of forwarding hierarchy is key to the scaling of Seamless MPLS as it avoids the need to fully mesh the SNs with LDP or RSVP LSPs.
4. TNs and ANs are involved only in intra-region LSPs.

The SNs in each region are route-reflector clients of the BNs bordering that region. The BNs are either fully meshed with iBGP sessions, or route-reflectors are used with the BNs as route-reflector clients.⁵ The BNs use BGP to distribute labeled routes corresponding to the loopback addresses of SNs.⁶

Let us now look in more detail at the sequence of steps needed to achieve MPLS connectivity between SN1 and SN2, referring to Figure 16.10.

1. SN2 advertises its loopback address using labeled BGP to its iBGP peers. The label advertised is the implicit-null label.
2. This triggers BN2 to allocate a label L0 corresponding to SN2. BN2 advertises this label using labeled iBGP to its iBGP peers, including BN1. Note that BN2 sets the BGP next-hop to self when it does so.

⁴ In practice in some deployments it may be necessary to carry loopback addresses across region boundaries for in-band management purposes.

⁵ It might seem that deploying BGP on the BNs breaks the concept of a ‘BGP-free core’. However, the use of BGP on the BNs is only in the context of internal infrastructure routes, rather than service-related reachability information.

⁶ In cases where certain ANs have parent SNs in other regions, BNs also need to advertise in BGP labels associated with those ANs. However, for the discussion in this section, we assume that all the ANs have their parent SNs in the same region.

This is key to building the forwarding hierarchy used in Seamless MPLS – in this way, BN1 knows that to forward traffic to SN2, it needs to use an LSP to BN2 for the next stage of the journey.

3. In turn, BN1 allocates a label L1 corresponding to SN2. BN1 advertises this using labeled iBGP to the service nodes in region 1, including SN1 and sets the next-hop to self. Again, this is an important step in building the forwarding hierarchy - SN1 knows that if BN1 receives a packet with label L1, it will forward it towards SN2.
4. When SN1 needs to forward a packet destined to the grey VRF on SN2, it pushes the corresponding VPN label L3 onto the packet. It then pushes the label L1 onto the packet required to reach SN2, and finally pushes the label L1' required to reach BN1. This top label is known to SN1 via LDP or RSVP.
5. Assuming that PHP is used on the LDP or RSVP LSP to BN1, when the packet arrives at BN1, it has label L1 on top and label L3 underneath. BN1 swaps label L1 for label L0, because this is the label advertised by BN2 to reach SN2, and then pushes the label L0' required to reach BN2. Label L0' corresponds to the intra-region LDP or RSVP LSP from BN1 to BN2.
6. Assuming that PHP is used on the LDP or RSVP LSP to BN2, when the packet arrives at BN2, it has label L0 on top and label L3 underneath. BN2 swaps label L0 for the label L2' required to reach SN2. Label L2' corresponds to the intra-region LDP or RSVP LSP from BN2 to SN2.
7. Assuming that PHP is used on the LDP or RSVP LSP to SN2, the packet arrives at SN2 with the VPN label L3, which triggers SN2 to perform a route look-up in the grey VRF.

So far we have discussed the signaling required to set up the MPLS transport connectivity between SNs. The other aspect to consider is the service signaling (for example, L3VPN, L2VPN and VPLS) between SNs. Because there are likely to be several hundred SNs, it is probably not convenient to fully mesh them with BGP sessions for the service signaling. One option is to use the BNs as route-reflectors for the service signaling. An alternative approach is to use a separate route-reflector infrastructure – in this way, the BNs are only involved in the transport infrastructure and do not take part in the service signaling.

16.4.3 Scaling analysis of Seamless MPLS

In this section, we make some rough scaling calculations of the scheme described in the previous section. The network in Figure 16.11 contains 100 access regions and one core region. At the border between each access

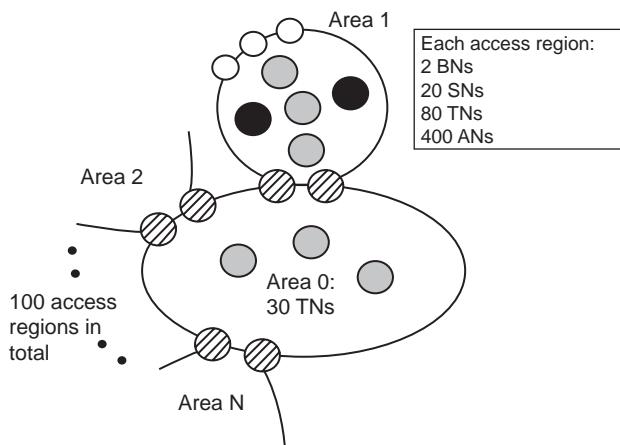


Figure 16.11 Diagram for Seamless MPLS scaling discussion

region and the core region are two BNs, so the network has a total of 200 BNs. Each access region contains 20 SNs and 80 TNs. Each access region also contains 400 ANs, each of which is dual-homed to a pair of SNs. The core region contains 30 TNs. As a result, the total number of each type of node in the network as a whole is as follows:

BN: 200
 SN: 2000
 AN: 40,000
 TN: 8030

Let us now look at how the network scales, first for the case in which LDP is used within each region and then for the case in which RSVP is used within each region.

16.4.3.1 LDP case

Each access region contains 502 nodes, so each node has 501 FECs in its LDP database corresponding to the loopback address of each of the other nodes in the region. The ANs and TNs within an access region need only these FECs.

In addition to the FECs corresponding to intra-region connectivity, each SN has labels that are received via labeled BGP from each of the two BNs at the edge of its region. These correspond to the 1980 SNs in the other access regions. Each BN has labels corresponding to the 2000 SNs in the

network, plus LDP labels corresponding to the approximately ~ 230 nodes in the core region and the approximately ~ 500 nodes in the access region to which it is attached.

As can be seen, the amount of MPLS label state related to inter-node connectivity is quite modest, and is far less than would be the case if LDP were used for both inter-region and intra-region connectivity. In that case, each node would have about 50000 FECs, corresponding to the loopback address of each node in the entire network.

16.4.3.2 RSVP case

If RSVP provides the intra-region connectivity, each AN has four RSVP LSPs, because it has two parent SNs, and an LSP is needed in each direction for connectivity to each SN.

Each SN has 40 pairs of LSPs, so 80 in total, corresponding to the 40 ANs of which it is a parent. In addition, it has a pair of LSPs corresponding to each of the other 19 SNs in the same region and a pair of LSPs corresponding to each of the two BNs at the edge of the region.

In addition to the LSPs corresponding to the 20 SNs in the access region to which it is attached, each BN has a pair of LSPs corresponding to each of the other 199 BNs in the network. In addition to the label-state corresponding to intra-region RSVP connectivity, each SN has labels that are received via labeled BGP from each of the two BNs at the edge of its region. These correspond to the 1980 SNs in the other access regions. Also each BN has label state received via labeled BGP corresponding to the 2000 SNs in the network. As can be seen, the scaling demands on nodes within the access network and on the BNs are quite modest. Another aspect to consider is the number of RSVP LSPs in the core region that result from fully meshing the 200 BNs. This number is about $\sim 200^2$ or about ~ 40000 , although for typical topologies not all these pass through the same TN. The number of RSVP LSPs is far less than would be the case if RSVP rather than labeled BGP provided the inter-region connectivity. In such a case, fully meshing all 2000 SNs with RSVP LSPs would result in about ~ 4 million LSPs, most of which would cross the core region.

16.4.3.3 Comparison of scaling properties

Table 16.2 compares the amount of MPLS forwarding state on each type of node for four different scenarios:

- LDP throughout (no labeled BGP).
- RSVP throughout (no labeled BGP).
- Seamless MPLS scheme: LDP for intra-region connectivity and labeled BGP for inter-region connectivity.

Table 16.2 Comparison of MPLS forwarding state as a function of label distribution scheme

	LDP throughout	RSVP throughout	Seamless MPLS (labeled BGP + LDP)	Seamless MPLS (labeled BGP + RSVP)
Each AN	~50000 (an LDP FEC per node in network)	4 (LSPs to / from two SNs)	~500 (LDP FEC per node in same region)	4 (LSPs to or from two SNs)
Each SN	~50000	4000 (LSPs to / from other SNs) <i>plus</i> 80 (LSPs to / from 40 ANs in the same region)	~500 (LDP FEC per node in same region) <i>plus</i> ~1980 (SNs in other regions, via labeled BGP)	80 (LSPs to / from 40 ANs in the same region) <i>plus</i> 4 (LSPs to / from two BNs) <i>plus</i> ~1980 (SNs in other regions, via labeled BGP)
Each BN	~50000 LDP labels	Transit router for LSPs between SNs in attached region and SNs in other regions ~2000 × 20 = ~40000	~500 LDP (nodes in attached access network) <i>plus</i> ~230 LDP (nodes in core region) <i>plus</i> ~2000 labeled BGP (SNs)	40 LSPs to / from SNs in attached access network <i>plus</i> 2 × 200 RSVP to / from other BNs <i>plus</i> ~2000 labeled BGP (SNs)
Each TN in core region	~50000 LDP labels	Transit for proportion of 2000 ² SN-to-SN LSPs	~230 LDP (nodes in core region)	Transit for proportion of 200 ² BN-BN LSPs
Each TN in access region	~50000 LDP labels	Transit for proportion of (20 × 40 × 2) SN-AN LSPs + (20 × 2000) SN-SN LSPs	~500 LDP (nodes in access region)	Transit for proportion of (20 × 40 × 2) + SN-AN and (20 × 2 × 2) SN-BN LSPs

- Seamless MPLS scheme: RSVP for intra-region connectivity and labeled BGP for inter-region connectivity.

In the RSVP cases, some nodes are transit nodes for a proportion of the RSVP-signaled LSPs. The exact proportion depends on the topology of the network and would vary from node to node. As well as the TN cases shown in the table (and the BNs in case when RSVP is used throughout), in some topologies ANs and SNs may also be transit nodes for RSVP-signaled LSPs.

The table shows that as a consequence of the forwarding hierarchy employed by Seamless MPLS, the amount of forwarding state is considerably reduced compared to the cases in which LDP or RSVP are used throughout.

16.4.4 Seamless MPLS for multicast

So far we have discussed Seamless MPLS for unicast traffic. Let us now examine how Seamless MPLS operation can be achieved for multicast traffic [MCAST] [MPLSWC-2010]. As we saw in Chapter 10 (Multicast in a Layer 3 VPN), and Chapter 13 (VPLS), P2MP LSPs, signaled by RSVP or LDP, can be used as provider tunnels to transport mVPN traffic or VPLS broadcast/multicast/unknown traffic. Given that in the Seamless MPLS model, the network might contain of order thousands of SNs, and that there could be a large number of mVPN and VPLS instances that make use of P2MP LSPs, the total number of P2MP LSPs in the network is likely to be very large. Therefore it is highly desirable to use P2MP LSP hierarchy, discussed in Section 6.8 of Chapter 6 (MPLS Multicast), in order to reduce the amount of control plane state and forwarding plane state on the transit nodes. In Chapter 6, we saw that the following conditions apply to P2MP LSP hierarchy, in order for a P2MP LSP (the ‘inner P2MP LSP’) to be nested inside another P2MP LSP (the ‘outer P2MP LSP’)

1. The inner and outer P2MP LSP must have the same root (ingress) node.
2. Each leaf (egress) node of the inner LSP must also be a leaf node of the outer LSP.⁷

In a network containing a large number of SNs, these conditions would make it difficult to make efficient use of P2MP hierarchy. This is illustrated in Figure 16.12.

The figure shows two P2MP LSPs, one rooted at SN7 and the other rooted at SN1. (Note that the figure does not show the TNs for clarity).

⁷Note that the converse is not true. That is, when aggregating multiple p2mp LSPs, not every leaf node of the outer LSP must also be a leaf node of a particular inner LSP. This means that the leaf nodes of the various inner P2MP LSPs do not need to be fully congruent.

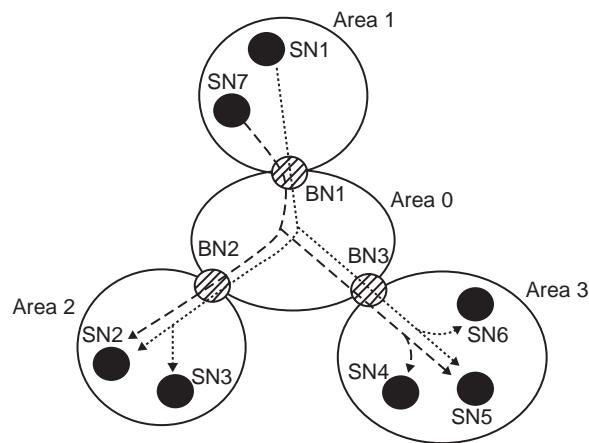


Figure 16.12 Two P2MP LSPs that cannot be aggregated because they have different root nodes

Although the two P2MP LSPs have some overlap, for example in area 0, they are not candidates for aggregation as they do not have the same root node. Another example is shown in Figure 16.13. In this example, the two P2MP LSPs do have the same root node, SN1. Although they could be aggregated, there would be wastage of bandwidth because of the egress nodes of the two inner P2MP LSPs are not fully congruent. This means that, for example, the outer LSP would deliver traffic from LSP X (denoted

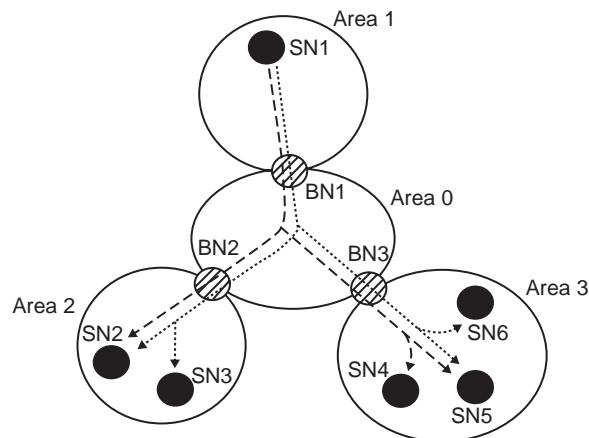


Figure 16.13 Two P2MP LSPs for which aggregation would be inefficient because they have low congruency of leaf nodes

by the dotted line) to SN4 even though SN4 is not an egress node for LSP X. Bear in mind that each P2MP LSP may have of order hundreds of egress nodes, so the chances of a high degree of congruency between P2MP LSPs could be quite low.

In order to address these issues, the multicast solution for Seamless MPLS uses segmented P2MP LSPs. In Chapter 11 (Advanced Topics in BGP/MPLS mVPNs) we discussed segmentation in the context of the inter-AS solution for multicast MPLS, in which intra-AS segments are stitched at the ASBRs. By analogy, in the Seamless MPLS case, each segment is confined to one IGP area, and the segments are stitched at the BNs. Rather than applying P2MP hierarchy to P2MP LSPs as a whole, it is applied to intra-area segments, giving greater scope for aggregation. This is illustrated in Figure 16.14. Although P2MP LSP X and P2MP LSP Y have different root nodes and only partial congruency of egress nodes, within area 0 their respective segments are fully congruent. The segments in area 0 have the same root node, BN1, and the same egress nodes, BN2 and BN3. As a result, BN1 can aggregate the two segments into outer P2MP LSP Z, denoted by the thick grey line in the figure, without wasting bandwidth as the segments are fully congruent.

In general, aggregation can be performed in any area, as long as within that area the intra-area LSP segments being aggregated share the same root node. The root node in each area makes the decision to aggregate independently of root nodes in the other areas. In order to perform aggregation of intra-area segments, a root node needs to know which are the leaf nodes of its intra-area segments, through a process called leaf-tracking. A root

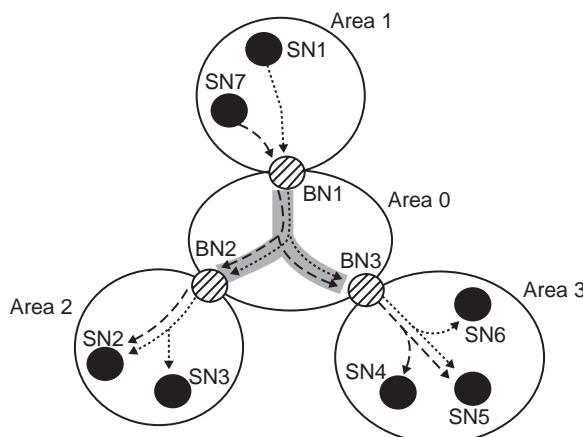


Figure 16.14 P2MP LSP segments aggregated into the same outer LSP (denoted by the thick grey line)

node could be configured to aggregate P2MP LSP segments only if there is full congruity between them. Alternatively it could be configured to perform aggregation if the degree of congruity exceeds a given threshold. Typically an SN or a BN might be the root node of multiple outer P2MP LSPs, each aggregating multiple intra-area segments of inter-area P2MP LSPs, where these segments have the required degree of congruity to be aggregated into the same outer P2MP LSP.

An additional benefit of segmentation is that different P2MP LSP signaling protocols can be used in different areas if desired – some areas could use RSVP and others could use LDP. BGP is employed as the inter-area signaling protocol for P2MP LSPs. In that sense, the multicast Seamless MPLS scheme is very similar to the unicast Seamless MPLS scheme described previously.

Let us examine the procedures required in order to set up a segmented P2MP LSP. Built into the procedures are methods by which the root node in each region can track the leaves of each of its P2MP LSP segments, in order to decide whether to perform aggregation. Figure 16.15 illustrates this scheme. Note that the TNs within each area are not shown to make the diagram clearer. SN1 to SN5 are members of the grey mVPN. Let us suppose that SN1 uses an inclusive tunnel to deliver traffic from sources in its grey VRF to the other SNs, some of which are in different IGP areas. The inclusive tunnel comprises multiple segments, each of which is an LDP or RSVP-sigaled P2MP LSP. Segment T1 within Area 1 extends from SN1 to BN1. Segment T2 extends across Area 0 from BN1 to BN2 and BN3. Segment T3 within Area 2 extends from BN2 to SN2 and SN3. Segment T4 within Area 3 extends from BN3 to SN4 and SN5. It is the responsibility of the BNs to stitch traffic between segments by installing appropriate

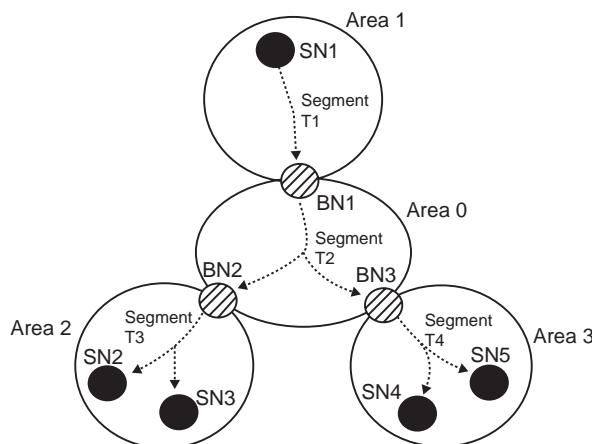


Figure 16.15 P2MP LSP comprising multiple intra-area segments

forwarding state. For example, BN2 forwards traffic arriving on segment T2 onto segment T3.

The delivery to the egress ANs from an egress SN can be either over unicast pseudowires or can be over P2MP pseudowires. Which is better to use depends on the topology of the network between the SN and the ANs and how densely or sparsely ANs needing to receive particular multicast traffic are located.

In Chapter 10 (Multicast in a Layer 3 VPN), we described the tunnel discovery procedures for ‘vanilla’ NG mVPN. We saw that the leaf nodes need to learn the identity of the multicast tunnel that a root node uses to send multicast traffic to them. They learn this identity from the PMI tunnel attribute contained within the autodiscovery route generated by the root node. If the tunnel is signaled by RSVP, the root needs to know the identity of the leaf nodes, because an RSVP P2MP LSP is initiated by the root node. In contrast, if the tunnel is signaled by LDP, the root node does not need to know the identity of the leaf nodes because a LDP P2MP LSP is initiated by the leaf nodes.

In the Seamless MPLS scheme with segmented P2MP LSPs, the procedures are modified such that leaf nodes of a tunnel segment within an IGP area learn about the identity of the segment from the root node of that segment. Unlike the vanilla NG mVPN case, the leaf nodes of a segment need to advertise their existence to the root node regardless of whether the segment is signaled via RSVP or LDP. This is because the root node needs to perform leaf-tracking in order to identify which P2MP LSPs segments rooted at that root node are candidates for aggregation.

A key component of the scheme is that each BN is a route-reflector for I-PMSI and S-PMSI autodiscovery routes for the access region to which it is attached.⁸ This means that when BN1 receives an autodiscovery route from SN1, it can make the required modifications before advertising it to its peers BN2 and BN3. In turn, BN2 and BN3 can make the required modifications before reflecting the route to their respective SN route-reflector clients. Let us see how the scheme operates in more detail.

1. SN1 generates an auto-discovery route containing the identity of segment T1. This includes the identity of the outer P2MP LSP into which it is nested and the upstream-assigned label associated with T1. Note that if this is the first time that SN1 has advertised T1, then SN1 does not yet know which are the leaf nodes of T1, so it does not know which outer LSP into which to nest T1. In order to overcome this, it can either generate the autodiscovery route with the entry ‘no tunnel information

⁸Note that the BNs do not change the BGP next-hop of the autodiscovery route. This is so that if an SN generates a BGP C-multicast route, it is directed at SN1 rather than having to be processed by the BNs.

available' or it can temporarily assign an outer LSP dedicated exclusively to T1. Once it has discovered the leaves of T1, it can decide which outer LSP into which to nest T1, and then resends the auto-discovery route with the identity of the selected outer LSP.⁹

2. When BN1 in turn advertises the auto-discovery route to the other BNs (BN2 and BN3), BN1 modifies the route so that it contains the identity of segment T2 instead of T1.
3. In turn, BN2 modifies the auto-discovery route to contain the identity of segment T3 when it reflects the route to the SNs in Area 2. Similarly, BN3 modifies the auto-discovery route to contain the identity of segment T4 when it reflects the route to the SNs in Area 3.
4. In response to the auto-discovery route, SN2 and SN3 each generate a corresponding leaf autodiscovery route. These routes carry a Route Target community particular to BN2, resulting in only BN2 importing these routes. SN2 and SN3 select BN2 because BN2 is the BGP next-hop of the unicast route to SN1¹⁰. In this way, BN2 learns the identities of the leaf nodes, SN2 and SN3, that are in the same area as BN2 (Area 2). Based on this information, BN2 can decide whether to aggregate segment T3 with other segments rooted at BN2, if they have a sufficient degree of congruency. Similarly, SN4 and SN5 each generate a leaf autodiscovery route with an attached Route Target community particular to BN3 and hence BN3 learns the identities of the leaf nodes SN4 and SN5 that are in the same area as BN3 (Area 3).
5. As a result of receiving the leaf autodiscovery routes, BN2 and BN3 are each triggered to generate a corresponding leaf autodiscovery route. These have attached a Route Target community particular to BN1, because from the point of view of BN2 and BN3, BN1 is the BGP next-hop of the unicast route to SN1. In this way, BN1 knows that BN2 and BN3 are leaf nodes of segment T2, and can make aggregation decisions accordingly.
6. Receiving these leaf autodiscovery routes triggers BN1 to generate a leaf autodiscovery route with an attached Route Target community particular to SN1. In this way, SN1 knows that BN1 is an leaf node of segment T1, and can make aggregation decisions accordingly.

In summary, we have seen that the Seamless MPLS scheme for multi-cast applies the concepts of P2MP hierarchy and segmentation to achieve a solution that can scale to very large networks. Like the Seamless MPLS

⁹ Similarly, initially the BNs do not know the leaf nodes of their intra-area segments so they employ a similar procedure.

¹⁰ Recall that in the Seamless MPLS scheme, a BN changes the next-hop to self when reflecting unicast routes.

scheme for unicast, it employs the concept of confining RSVP or LDP signaling to individual areas, with the use of BGP as the inter-region signaling protocol.

16.5 CONCLUSIONS

In this chapter, we have described the economic drivers behind network consolidation in the access part of the network. We have shown how MPLS is the ideal way to achieve this, by analogy with the way in which it is used in core networks. We have described different models for how the consolidation can be achieved. Using MPLS in both the core and access network gives the operator the option of having a unified core and access network based on common technology, with the flexibility of choosing the optimum location of each service delivery function. In this chapter, we have discussed MPLS in the context of access networks for fixed operators. Another emerging application of MPLS as an access technology is in the mobile Radio Access Network (RAN). This is discussed in more detail in the next chapter.

16.6 REFERENCES

- [ANCP] Access Node Control Protocol Working Group in IETF, <https://datatracker.ietf.org/wg/ancp/>
- [LDP-AGG] G. Swallow, J. Guichard, *Network Scaling with Aggregate LSPs*, draft-swallow-mpls-aggregate-fec-01.txt (expired draft)
- [LDP-SCALE] K. Komppella, *Techniques for Scaling LDP*, MPLS 2007 Conference, October 2007, Washington DC
- [MCAST] Y. Rekhter, R. Aggarwal, T. Morin, I. Grosclaude, *Inter-Area P2MP Segmented LSPs*, draft-raggarwal-mpls-seamless-mcast-00.txt (work in progress)
- [MPLS2008] K. Komppella and E. Peterson, *MPLS in the Access*, MPLS2008 Conference, October 2008, Washington DC
- [MPLSWC-2007] Y. Rekhter, *Towards MPLS/Ethernet-based Transport*, Paper D1-02, MPLS World Congress 2007, February 2007, Paris
- [MPLSWC-2010] Y. Rekhter, *Scaling Multicast MPLS – Seamless MPLS Multicast*, Paper D1-03, MPLS World Congress 2010, February 2010, Paris
- [RFC3107] Y. Rekhter and E. Rosen, *Carrying Label Information in BGP-4*, RFC 3107, May 2001

[RFC5036]	L. Andersson, I. Minei and B. Thomas (eds), <i>LDP Specification</i> , RFC 5036, October 2007
[RFC5283]	B. Decraene, J.L. Le Roux and I. Minei, <i>LDP Extension for Inter-Area Label Switched Paths (LSPs)</i> , RFC 5283, July 2008
[SEAMLESS]	N. Leymann, B. Decraene and D. Steinberg, <i>Seamless MPLS Architecture</i> , draft-leymann-mpls-seamless-mpls-01.txt (work in progress)

16.7 STUDY QUESTIONS

1. Describe some of the issues that arise when using Layer 2 Ethernet switching to build an access network.
2. Describe the relative merits of (i) the Option 1 scheme in which the core and access networks are separate MPLS islands compared to (ii) the Option 2 scheme (Seamless MPLS) in which the core and access networks form one unified MPLS network.
3. Describe the control plane operations used to achieve unicast MPLS connectivity between SNs in the Seamless MPLS model.
4. Describe the control plane operations used to achieve multicast MPLS connectivity between SNs in the Seamless MPLS model.
5. Describe the advantages of seamless MPLS and a forwarding hierarchy in terms of data-plane state.

17

MPLS Transport Profile (MPLS-TP)

17.1 INTRODUCTION

The rise of bandwidth-hungry applications such as Triple Play and IP Video, coupled with the pressure to minimize the cost per bit, is forcing carriers to rethink their transport infrastructure. In light of the rise of packet-based services, the natural transition is to move from a circuit-based transport to a packet-based one, to take advantage of the flexibility and cost benefits of packet-switching technology. In this context, a transport profile of Multi-Protocol Label Switching called MPLS-TP is currently being developed, with the goal of forming the basis for next-generation packet transport networks.

This chapter discusses the business drivers behind this development, the requirements for making MPLS transport-friendly, and the enhancements to the MPLS protocol suite that are in the process of being defined by the IETF. After describing the technology, we look at some deployment scenarios, and finally we examine some of the common misconceptions about MPLS-TP.

17.2 THE BUSINESS DRIVERS

The first communication services were circuit based, for example, the Public Switched Telephone Network (PSTN), leased lines and Frame Relay

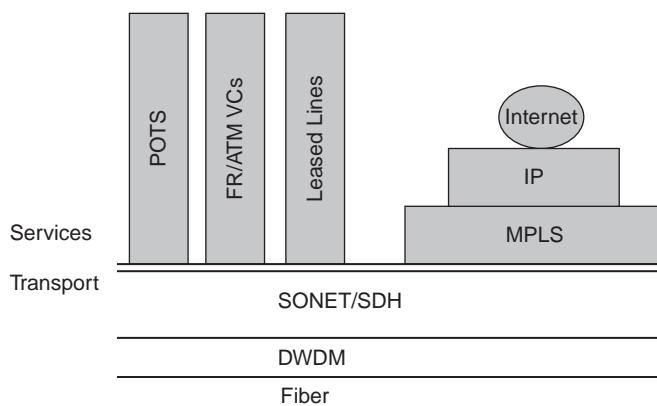


Figure 17.1 The old state of transport and service

and ATM virtual circuits. For this reason, a circuit based digital multiplexing and transport infrastructure was developed to serve them, in the shape of SONET/SDH.¹ As well as serving as the lowest common denominator over which an operator transported all of its own services, it was also the primary way wholesale services were offered. When packet-based services emerged, they represented only a small percentage of the total traffic, and the associated hardware had relatively low speed interfaces. For this reason it was only natural to carry these services over the existing circuit-switched transport. In this context, IP and MPLS were two of many services sitting on top of the existing circuit-based transport infrastructure. This old state of affairs is shown in Figure 17.1, in which the double line represents the demarcation between the transport and the services.

However, three main trends have emerged over the last several years. First, client services have started moving from a circuit based to a packet based one, for example PSTN moving to VOIP and video moving from TDM to IPTV. Second, IP/MPLS packet-based services saw a tremendous increase in volume, driven by applications such as IP Video, Triple Play, and VPN, as well as by growth in the ‘traditional’ IP services. These two together mean that the majority of client traffic is increasingly inherently packet based. Finally, at the same time that the bandwidth requirements driven by these applications went up, there was tremendous pressure to keep the cost per bit down. Carrying packet traffic over circuit-based infrastructure is inefficient because one cannot take advantage of statistical multiplexing – each circuit needs to be sized according to the peak demand on that circuit, which might be substantially higher than the

¹ SONET/SDH was preceded by PDH, but at the time of writing SONET/SDH had been deployed for more than fifteen years by some operators.

average demand [BELOTTI]. Thus, the direction for next-generation networks was to move from SONET/SDH TDM technologies toward packet transport over DWDM, and there was a need to develop a generic packet-based transport infrastructure. This ubiquitous packet transport would cater for all the myriad packet services, in the same way that SONET/SDH catered in the past to all the myriad circuit services.

The challenge developing such a new infrastructure is that it not only has to achieve the same high levels of reliability and operational simplicity as SONET/SDH, but it also must preserve the look and feel and maintain the organizational boundaries of the old architecture. When discussing such a transition, [CE_2008] uses the analogy of a ‘magic layer’ that recaptures the traffic engineering, protection, and OAM capabilities of circuit-switched transport in packet-based transport, including the need for control and deterministic usage of network resources, the ability to support both static and dynamic provisioning of services, and advanced abilities to monitor the network and take action in response to this monitoring.

But what might such a ‘magic layer’ be? To answer this, it is worth noting the trend in moving most services to IP/MPLS. As a result, IP/MPLS must already have the same stability and resilience as the transport infrastructure. If this is the case, it can be made part of the transport infrastructure. This is shown schematically in Figure 17.2, in which the double line representing the demarcation between transport and service is now drawn above the MPLS layer. In this model, OSI layers are collapsed from physical constructs into logical constructs in MPLS.

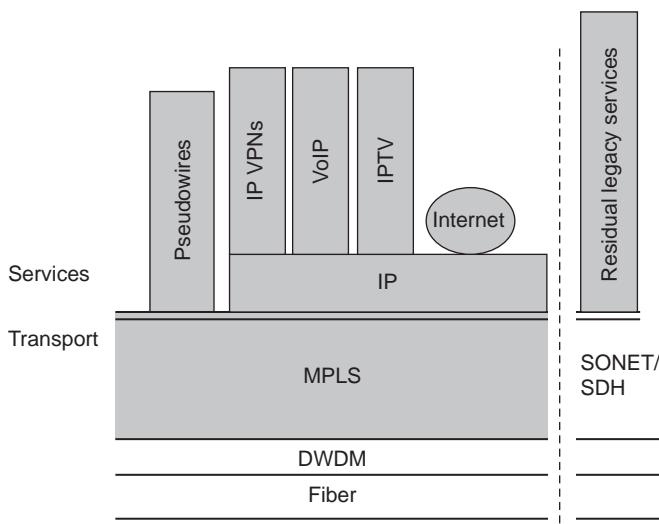


Figure 17.2 The new state of transport and service

The first attempt to define a new packet-based transport infrastructure started in 2006 with the definition of T-MPLS in the ITU-T. Implemented by extensions to the MPLS technology, T-MPLS was intended as a connection-oriented, packet-switched transport based on MPLS. T-MPLS removed MPLS features not relevant to transport and added features important for transport networks, in particular OAM functions. In the course of the implementation of these functions, however, the ITU changed the forwarding paradigm of MPLS with respect to label usage and packet processing in a way not compatible with current MPLS and introduced new OAM functions not compatible with the existing ones.²

Changing existing technology in such a way is not only detrimental for network operators, whose existing hardware all of a sudden may not be able to support the new technology, but also can be harmful to the Internet, as explained in [RFC5704] and debated heatedly in the IETF. As a result of these debates, the ITU-T ceased work on T-MPLS in December 2008, and the IETF and ITU-T formed a joint working team to examine the MPLS architectural considerations for a transport profile [MPLS_TP_22]. This joint team defined a new transport profile for MPLS [RFC5317], called MPLS transport profile, now known widely as MPLS-TP. The rest of this chapter details this technology.

17.3 REQUIREMENTS FOR A TRANSPORT PROFILE FOR MPLS

The goal of a transport profile for MPLS is very simple: provide ‘transport-like’ functionality in MPLS while preserving the existing MPLS architecture. The requirements for meeting this goal are formally documented in [RFC5654]. These requirements are the intersection between MPLS and pseudowire architectures as defined by the IETF and the requirements for packet transport networks as defined by the ITU-T. In a nutshell they can be summarized as three main requirements. The first is to allow static creation of LSPs through an NMS, the second is to provide some of the functionality of SONET/SDH networks such as performance monitoring, fault monitoring and delay measurement and finally the third is to support a layer network.

In the following sections we examine these and other MPLS-TP requirements in more detail. To start the discussion, instead of looking directly at the requirements for MPLS-TP, we first examine the capabilities of current transport networks, because they form the basis for the MPLS-TP requirements.

² Another technology that failed for similar reasons was PBB-TE. It was also an attempt to reuse MPLS functionality, but required a fundamental forwarding change.

17.3.1 Characteristics of transport networks

The purpose of a transport network is to carry any type of customer traffic reliably between end points in the transport network. Let us take a look at traditional transport network characteristics and examine each of them in turn:

- *Client-server relationship.* The transport domain is independent of client networks and, from a functional and operational point of view, the client and transport networks are independent layer networks. As a consequence, the transport network is topologically opaque from the client's point of view. However, there is a well-defined, standardized interface between the client and server network, for example for the exchange of OAM information such as alarms and defects.
- *Connection-oriented services.* The goal of the transport network is to move traffic between end points in the network. As a result, the forwarding paradigm is connection oriented. The connection path computation and its placement are often done statically via a Network Management System (NMS), but they can also be done using a dynamic control plane. Static placement is a widely used technique, because it provides maximum control over the path placement and the network resources. In cases in which the path does not change often (as would be the case with the aggregated paths between entry and exit points in the network), such an approach is, in fact, very attractive. Transport network operators have invested heavily in NMS systems for statically provisioning paths and would like to continue to leverage the static approach in packet-based transport networks.
- *Support for transport Operations, Administration and Maintenance (OAM) capabilities.* ITU-T has extensive specifications for OAM in transport networks. Two of the key requirements, which have their origin in the SONET/SDH technology, are for in-band OAM and for absolute congruency between the OAM and the data plane. Another fundamental requirement is that OAM be independent of the configuration mechanism used for setting up the path. In addition, there is a need for proactive and on-demand connectivity checks, connection verification, performance monitoring, alarm suppression and alarm indication. It is no coincidence that most of the changes for MPLS-TP are in the OAM area, as described later in this chapter.
- *Support for aggregation.* Aggregation is beneficial for achieving scalability and is widely used in transport networks. Running OAM on the aggregated paths (rather than on the individual aggregates) contributes in achieving efficient and reliable operations.

- *Quality of Service.* The transport network must meet the quality of service objectives of its clients. To do so, it implements a wide range of mechanism for bandwidth and QoS guarantees, allowing the network to prioritize critical services, guarantee bandwidth, and control jitter and delay.
- *Resiliency.* Existing transport networks support fast detection and recovery time and in fact, as we have seen in Chapter 3, this was the source of the 50-ms requirement for MPLS FRR. In addition to fast recovery, transport networks have the ability to trigger protection from the OAM mechanisms and to support redundant connections. Finally, there is support for both linear and ring topologies.

The list above, while not comprehensive, gives a feeling for some of the functionality that MPLS-TP must support and sets the context for some of the requirements that were incorporated into MPLS-TP.

17.3.2 Requirements and architectural goals of MPLS-TP

The base requirements for the MPLS transport profile are spelled out in [RFC5654]. This RFC contains a long laundry list of mandatory and non-mandatory requirements, split into various categories such as general, control plane, data plane, QoS, and recovery requirements. In addition to this list, separate documents spell out the requirements for OAM and network management [RFC5860] [TP_NM_REQ], making it impractical to attempt to discuss them all in this chapter. Instead, we look at the key requirements identified in [MPLSTP_HIST], which are essential for making MPLS transport friendly. These requirements allow us to understand the changes made to MPLS to support a transport profile:

- *Static configuration of LSPs and pseudowires and their management via an external NMS.* As the basic building blocks in MPLS, LSPs and pseudowires provide the paths in a packet-based transport. To leverage the existing work practices and tools of transport networks, it must be possible to set up the LSPs and pseudowires statically and to manage them via an external NMS, similar to how paths are handled today in transport networks. This requirement is in addition to support of the dynamic control plane and in fact, some implementations have already supported this model for several years.
- *Transport-style OAM.* Transport networks have extensive OAM capabilities, as mentioned in the previous section. Most changes for MPLS-TP are, in fact, a result of introducing similar OAM capabilities in MPLS-TP, as we show later in this chapter. In addition to implementing new OAM

functions, such as performance monitoring and interlayer fault correlation, there are three additional important OAM requirements. The first is to have congruency between the data path and OAM and to allow for in-band OAM. The second is to allow for transport-like OAM functions for LSPs and pseudowires independently of how they were configured.³ Finally, the third is to have common and consistent OAM capabilities for Layer 2, pseudowires, and LSPs.

- *Transport-style resiliency and protection.* There are stringent requirements for recovery in transport networks, so sub-50-ms protection must be supported, including 1:1, 1+1, and M:N path protection. Because ring topologies are common in transport,⁴ there is a need to efficiently support protection in ring topologies,⁵ as well as in linear or meshed ones.
- *Support for nesting.* Because aggregation is a key requirement in transport networks, support for nesting of LSPs and pseudowires, by analogy with the nesting of circuits in SONET/SDH environments, is an important goal.⁶

The architectural goals for MPLS-TP are derived from these key requirements and from the constraints shaping how these requirements are met. [MPLSTP_HIST] provides a list of the key architectural goals for MPLS-TP, and this list is formalized in the MPLS-TP architectural framework [TP-FMWK]. It is important to discuss these, because they help clarify some of the decisions taken in extending MPLS to support a transport profile. Here are some of the most important architectural goals:

- *Compatibility with the existing MPLS architecture.* In particular, this goal implies making no changes to the forwarding architecture, in contrast to the T-MPLS solution developed earlier. The goal is for MPLS-TP functions to interoperate in a Layer 3 network and coexist with existing pseudowire solutions. For this reason, it is convenient to simply reuse the existing LSP and pseudowire constructs. Existing OAM functions, such as LSPing and BFD, must be able to coexist with the new OAM functions in an MPLS-TP environment, and new functions invented in the context of MPLS-TP should be reusable for MPLS.
- *Availability with both static provisioning systems and dynamic control plane.* The goal is to make it possible to establish and maintain LSPs or pseudowires, or both, both in the absence and presence of a dynamic

³This refers specifically to allowing the OAM functions to run for LSPs or PWES that are statically configured.

⁴This is because of the way the fiber is laid out.

⁵Recall from Chapter 3 that many recovery functions operate non-optimally in ring topologies.

⁶Recall from Chapter 5 that nesting can provide hierarchies of LSPs.

control plane. When static provisioning is used, there must be no dependency on dynamic routing or signaling, especially not for providing OAM functions. Furthermore, to enable building networks spanning both statically provisioned and dynamically signaled LSPs, it must be possible to stitch and nest these two kinds of LSPs, as we will see in the deployment scenarios later in this chapter.

- *Support for, but no dependence on, IP addressing.* The default continues to be IP addressing, but IP routing or forwarding is not required to support OAM or data packets. It must be possible to forward packets solely on the basis of label switching.

To summarize, what is needed is to reuse a large part of the existing MPLS technology and to add a set of additional capabilities required in transport networks such as transport-like OAM and transport-optimized resiliency. MPLS has proven to be adequate for many applications, as we have discussed throughout this book. MPLS-TP adds to MPLS a set of functions to enhance its transport characteristics to make it suitable to serve as the foundation for packet-based transport networks.

17.4 MPLS-TP FUNCTIONALITY

Having described the basic requirements for a transport profile for MPLS-TP, let us now see how they are met either by using a subset of MPLS or by enhancing MPLS. Recall from the previous section that one of the fundamental goals is to reuse the MPLS forwarding architecture and as much of the control plane as possible, and in fact as we will see in the next section, MPLS-TE provides most of the required functionality. Let us start by looking at the functionality shared by MPLS and MPLS-TP.

17.4.1 MPLS-TP as a subset of MPLS

MPLS-TP is based on the fundamental constructs of LSPs and pseudowires, and can be used with the existing dynamic out-of-band control plane. However, not all the control plane functionality is needed. To simplify OAM provisioning and allow identification of the source of the OAM packet from the label, MPLS-TP does not support PHP⁷ and LSP merging.⁸ Thus, there are no MP2P or MP2MP LSPs, and LDP signaling is precluded.⁸

⁷ PHP is disabled by default, but not precluded.

⁸ An example of a MP2P LSP is an LDP LSP for unicast traffic. Figure 1.6 in Chapter 1 (Foundations), shows an LDP LSP rooted at router D. Router D cannot tell which router a packet has arrived from because PHP is used. Even if PHP is not used, the packets would all arrive with the same label and router D would still be unable to tell which router a packet arrived from.

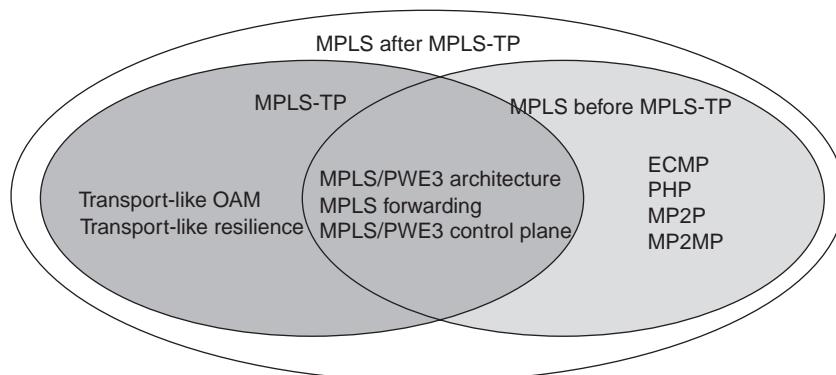


Figure 17.3 MPLS-TP is a subset of MPLS plus a set of new functions

To allow for simpler diagnostics and simplify OAM processing, ECMP is not supported. P2P LSPs can be unidirectional or bidirectional (as defined in GMPLS), but P2MP LSPs are unidirectional.

All MPLS and GMPLS recovery mechanism are reused in MPLS-TP, but as we see in the next section, enhancements are needed for achieving transport-like resilience. Support for nesting, which is one of the requirements of MPLS-TP, is ensured by the existing MPLS hierarchy mechanisms. Although static provisioning of LSPs and pseudowires was not widely used in the past, all major vendors already support it in their products, making it a de facto standard. Thus, the requirement for configuration through NMS, which is central to providing transport-like network operation, is already satisfied.

The connectivity check and connectivity verification OAM functions are based on BFD and LSPing for MPLS LSPs, with enhancements that we discuss later in this section. Figure 17.3 shows the relationship between MPLS and MPLS-TP, in which MPLS-TP is in part a subset of MPLS but has a set of new functions satisfying specific requirements of transport networks, mostly in the areas of resilience and OAM, which we discuss in detail in the next sections.

17.4.2 MPLS-TP resilience functions

Let us start with the resilience functions, because at the time of this writing, less work is in progress in this area than in the area of OAM. MPLS already has a rich set of protection and restoration mechanisms, such as LSP fast reroute, pseudowire redundancy, and path protection. There are two areas of focus in the context of transport-like resilience. The first is the ability to trigger protection from OAM, and the second is to provide

optimized protection in ring topologies. Ring topologies are important because circuit networks are typically built as interconnected rings and it is expected that many initial deployments of MPLS-TP will consist of replacing the circuit switching nodes with MPLS-TP packet switching nodes, thus yielding MPLS-TP networks consisting of interconnected rings. Various optimizations are possible in rings, and many different schemes have been developed to provide protection in rings. While MPLS FRR works in ring topologies, it does so in a very inefficient way, as explained in Chapter 3. Therefore, there is currently work in progress in the IETF to optimize FRR in ring topologies.

17.4.3 MPLS-TP OAM functions

OAM is an important and fundamental functionality in transport networks. Existing transport technologies have extensive OAM suites, and the functionalities required are well understood and documented. The detailed transport OAM mechanisms defined by the ITU serve as a good model for packet-based OAM. In the context of transport networks, OAM helps provide three crucial functions. First, it reduces the operational complexity and cost of running the network by allowing automatic detection, localization, and handling of failures. Second, it ensures network availability by finding and dealing with failures before the customer reports them and with minimal service interruption. Finally, it helps maintain the SLAs both in cases of absolute failure and in cases of service degradation.

To serve this purpose, a transport OAM suite must support rapid detection of faults that cause SLA violation and must be able to localize such faults upon detection. It also must have rapid indication of remote faults so that path protection can be triggered, but must be capable of suppressing some of these alarms to prevent a single event from triggering multiple alarms at different layers. Finally, there needs to be a non-intrusive way to detect the degradation of the service, such as an increase in the packet loss, which in turn would cause an SLA violation. To meet these requirements, the OAM suite must include both continuous (proactive) and on-demand (reactive) functions.

[RFC5860] spells out the MPLS-TP OAM requirements, and [TP_OAM_FM] provides a framework for satisfying these requirements. The key functions identified are:

- Connectivity check (CC), providing rapid, proactive identification of faults.
- Connectivity verification (CV), allowing on-demand localization of the fault after detection by CC.

Table 17.1 OAM function implementation in MPLS-TP

	Continuous (proactive)	On-demand (reactive)
Continuity check	Extensions to BFD	Extensions to LSPing
Fault localization	Not applicable	Extensions to LSPing
Remote integrity	Extensions to BFD	Extensions to LSPing
Alarm suppression	New tools	Not applicable
Performance monitoring	New tools	New tools

- Failure indications, enabling notification of failures. In this area, by analogy with SONET/SDH transport, there are alarms, such as alarm indication signal (AIS), sent in the downstream direction and remote defect indication (RDI) sent in the upstream direction, and alarm suppression. These functions together allow for rapid notification of failure and for efficient handling of such notifications by minimizing the number of alarms sent because of a single failure.
- Performance monitoring, enabling proactive detection of degradations that would lead to SLA violations. In the context of MPLS-TP, these are primarily delay measurement (DM), both 1-way and 2-way, and loss measurement (LM) functions. The goal of performance monitoring is to analyze packet loss in a service path and trigger MPLS protection such as FRR when needed.

These key OAM functions must be provided while satisfying the following three main architectural goals of MPLS-TP. First, as much of the existing BFD and LSPing mechanisms must be reused. Second, there must be congruency between the OAM packets and the data path (the equivalent of an in-band control channel to be used for OAM), and third, the OAM functions must be available even in the absence of a control plane and independent of IP.⁹

As a result, some of the required functions are provided by new tools, while others can be implemented via extensions to the existing toolset. A comprehensive analysis of OAM requirements and existing MPLS OAM tools is available at [TP_OAM_ANALYSIS] and summarized in [MPLS_2009_2] and in Table 17.1. Later in this section, we discuss some of the enhancements that are being defined for LSPing and BFD in the context of MPLS-TP.

Because new tools have to be designed to implement some of the required functions, it is desirable to have one generic, extensible OAM mechanism

⁹Recall that one of the goals of the MPLS-TP architecture was to have no requirement for support of IP routing or forwarding.

that can support them all. For such a mechanism to work, it must have a way to identify each incoming packet as belonging to the relevant function. This concept is not new. Recall from Chapter 15 that LSPing packets were identified as such by arriving on the UDP port allocated to LSPing with a destination address in the 127/8 range, and therefore their payload was interpreted according to the rules for processing LSPing TLVs. A similar mechanism cannot be used in MPLS-TP OAM because of the requirement to operate even in the absence of IP. Thus, the ability to identify a payload as belonging to an OAM function and further demultiplex the function to which it relates has to be encoded in the packet header independently of IP. This is not a new idea. Recall from the discussion on VCCV in Chapter 15, that OAM packets can be tagged with an additional packet header called the pseudowire associated channel (ACH) [RFC4485], which indicates that they must be processed by the appropriate OAM function, rather than being forwarded to the attachment circuit.

The same idea is used in MPLS-TP. The concept of ACH is generalized to a generic ACH (G-ACH) and applies not just to pseudowires, but also to LSPs and segments [RFC5586]. The G-ACH is simply a header in the packet that provides the demultiplexor function for OAM packets, identifying them, for example, as BFD CC messages or as loss measurement messages, and thus enabling appropriate handling of such packets. In the MPLS-TP documents, the term ACH is used for both ACH and G-ACH. To further enhance the flexibility and extensibility of this mechanism, ACH TLVs are also defined [TP_ACHTLV], to encode context information necessary for the processing of the OAM packets, for example, the source address or the LSP ID. The only remaining question is how an LSR would know that a packet has a G-ACH header. Recall that in the pseudowire case, the existence of the ACH was negotiated when the pseudowire was set up, which is not feasible if static provisioning is used. A simple way to solve this problem is to use one of the reserved labels for this purpose. [RFC5586] identifies the reserved value 13 as the generic associated label (GAL), thus providing the necessary tagging. Figure 17.4 shows the tagging of the packet, starting from the LSP label.

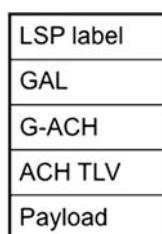


Figure 17.4 Packet headers for OAM packets in MPLS-TP

Use of the GAL for tagging OAM packets also enables easy extraction of the OAM packets at either a midpoint or an endpoint of an LSP or a pseudowire. In transport OAM, endpoints are often referred to as management end points (MEPs) and intermediated nodes are referred to as management intermediate points (MIPs). When an OAM packet arrives at the MEP, the outer (LSP) label is popped and the GAL is exposed. This ensures that OAM packets do not get forwarded beyond the MEP, but instead are processed by the appropriate OAM function. The G-ACH determines which kind of OAM packet this is and thus how it needs to be handled. To cause an OAM packet to be handled by a MIP, TTL expiration is used (similar to how it is applied in the LSPtrace mechanism). The TTL expiration causes the packet to be processed, and the existence of the GAL under the label for which the TTL expired causes the packet to be processed.

At the time of this writing, there is much discussion about extending the existing LSPing and BFD mechanisms and to introduce new mechanisms for the functions that are not available in MPLS, such as loss and delay measurement. A list of all drafts in flight, including working group documents and individual contributions, can be found on the IETF's MPLS TP wikipage at [MPLSTP]. Because this is very much work in progress, we list just some of the extensions that are currently being worked on:

- *LSPing extensions.* LSPing is enhanced for use on statically defined LSPs, on which endpoint information is not available from the control plane. There are also extensions for on-demand continuity check and fault localization, as noted in Table 17.1. These extensions have to work both over IP/UDP and without the IP/UDP encapsulation, so support for two different G-ACH types has to be available and ACH TLVs must be used to carry information that would have otherwise been available (such as the source of the packet).
- *BFD extensions.* Similar to LSPing, BFD packets need to be sent over a G-ACH, with and without IP/UDP, so support for different G-ACH types has to be available and ACH TLVs must be used to carry additional information. Because MPLS-TP supports P2MP LSPs, support for P2MP BFD is also required.
- *Packet loss and delay measurement.* These mechanisms are in the process of being defined, using two different G-ACH values representing loss and delay measurement packets [TP_LM_DM].
- *Alarm indication (fault OAM).* The mechanisms for sending fault indications (among them AIS) are in the process of being defined, using a special G-ACH representing the Fault Management channel.

To summarize, most of the extensions to MPLS in support of a transport profile fall in the area of transport-like OAM. A generic and easily extensible OAM mechanism, able to support the large variety of OAM

functionalities required in transport networks, is defined based on the combination of G-ACH, GAL, and ACH TLVs carrying arbitrary additional data. Although targeted for a transport profile, this solution is equally applicable to a non-transport profile as well. At the time of this writing, most of the extensions are works in progress and are in the early stages of the IETF standardization process. Having seen what the technology is, let us now look at how it can be deployed.

17.5 DEPLOYMENT CONSIDERATIONS

MPLS-TP is a technology for building packet-based transport networks. When looking at deployments of MPLS-TP, two fundamental questions need to be answered:

- Is a packet-based transport even needed? The answer depends largely on the services that the network provides. As shown in Figure 17.2, some legacy services continue to be provided over the old-style transport layer for the foreseeable future, and packet-based transport is not needed in all cases.
- When MPLS-TP is used, should there be static provisioning or should a dynamic control plane be used? The answer depends on the particular deployment. Static provisioning in MPLS has been supported by major vendors for quite some time and allows the transport network operators to continue to use the same tools and work practices from existing transport networks. Static provisioning may have applicability in scenarios in which some equipment, especially cost-reduced equipment used at the edges of the network, does not support a dynamic control plane or in which static configuration is preferred for security reasons, again usually at the edges of the network. A dynamic control plane has its own advantages, in particular with regards to scaling, as discussed in the context of MPLS in access networks. It can also provide advanced protection functions, for example, schemes such as LSP tail-end protection.

Assuming MPLS-TP is used, let us take a look at how the different layered networks can be implemented. To do so, it is necessary to first clarify the concepts and terminology used when discussing transport networks. The ITU-T defines a generic functional architecture for transport networks in [ITU-T.G.805]. This architecture design uses a layered network model with a client/server relationship between layer networks, in which a particular layer network is a server to the client layer network attached to it and a client layer network to the server layer network to which it is attached. In our discussion, when MPLS-TP is deployed as the transport network, it acts as the server layer network,

providing a transport service to the layer networks attached to it. The transport service is typically a circuit between two points of attachment to the MPLS-TP network, and it may be either an L2 service (e.g. SDH/SONET or p2p Ethernet) in which case the L2 frame is transported across the MPLS-TP network from the ingress attachment point to the egress attachment point, or an L3 service (e.g. IP or MPLS) in which case the L3 frame is transported across the MPLS-TP network from the ingress attachment point to the egress attachment point. The former service is provided by pseudowires. The latter service, termed the Network Layer Transport Service (NLTS), has the advantages that the attachment circuits may be heterogeneous (e.g. SDH/SONET to Ethernet) and that the L2 headers are not transported across the MPLS-TP network. Both types of transport services may be offered by a given MPLS-TP network.

This model looks like a clean layer network separation, but can become confusing when the service provided by the MPLS-TP client layer network is itself an MPLS service. In that case, the questions are what transport service is being provided to the client layer network, and what the client layer network really is. To explain this, let us examine the simple L3VPN deployment in Figure 17.5 and Figure 17.6. Both figures show a VPN with two sites (not shown in the figures). The CEs are dual-homed to PEs at the edge of the network, and there is a group of P routers in the core. The difference is in the span of the MPLS-TP network. In Figure 17.5, MPLS-TP is deployed in the core of the network only, spanning among the P routers, while in Figure 17.6 it extends all the way to the PEs. In either case, the

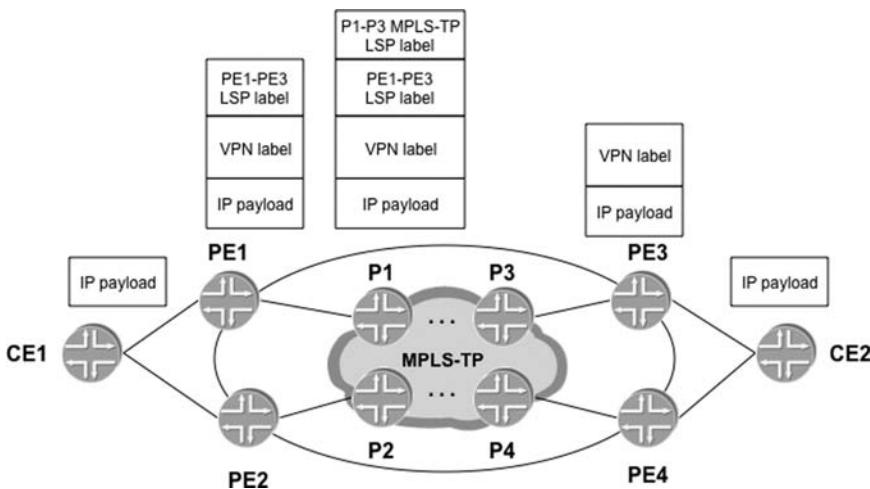


Figure 17.5 L3VPN deployment using MPLS-TP in the core, with NLTS provided by MPLS-TP LSPs

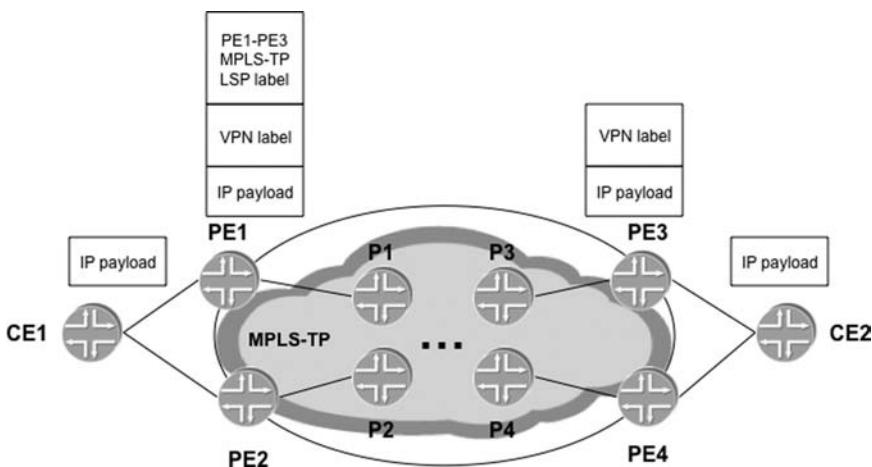


Figure 17.6 L3VPN deployment using MPLS-TP to the edges, with NLTS provided by MPLS-TP LSPs

service being provided to the CEs is an L3VPN service, requiring PE-PE LSPs between the PEs.

Let us start by examining the scenario depicted in Figure 17.5. The span of the transport network is limited to the MPLS-TP core and implements an L2 or L3 transport service through MPLS-TP LSPs that carry the traffic between the P routers. The client layer network for this transport service is represented by the PE-PE LSPs required for the VPN service. This approach yields a model in which MPLS-TP LSPs are deployed in the core and PE-PE LSPs are configured on top of them to satisfy the requirement for an MPLS tunnel to carry the labeled VPN traffic. This approach is very intuitive, because it is similar to nesting LSPs in a normal VPN deployment. There is a clear demarcation here between the client and server layer networks, and they are implemented on separate boxes, the PEs and the P routers, respectively. In practice there could be several client layer networks belonging to the same service provider that owns the MPLS-TP network, for example a business VPN network, an Internet network, a wireless network and possibly client networks belonging to other organizations (wholesale services). In this type of scenario, the model described above could be quite attractive, because it provides a clear split from an organizational/political point of view.

Let us see what happens when the transport network extends all the way to the PE routers, as shown in Figure 17.6. Applying the same reasoning as in the previous case, MPLS-TP LSPs are built between the PE routers. In addition to them, the PE-PE LSPs required to carry the VPN traffic are also set up. This approach yields a model in which two transport

LSPs are effectively built one on top of the other between the same set of routers. Clearly such an approach is not optimal, because it creates an additional layer network with no benefit: both LSPs provide the same function, namely providing transport. In such a case, the VPN service can be provided directly over the MPLS-TP LSPs, as shown in Figure 17.6, in which MPLS-TP LSPs are provisioned between the PEs and the VPN service is delivered over them. By removing one of the redundant layers, this approach reduces the complexity and operational cost of the deployment, but blurs the line between the client and the server layer networks, because now part of the VPN client function is implemented directly by the transport network. In this deployment, the transport service is again represented by the MPLS-TP LSPs, which in this case are providing an NLTS internally within the PEs.

One possibility for reintroducing the clean separation between the client and server functions is to provide an L2 transport service using pseudowires. The transport service is provided by pseudowires, and all parts of the VPN service, including the PE-PE LSPs, are implemented cleanly on top of it as a client layer network. This model is depicted in Figure 17.7, and the MPLS-TP network spans between the PEs. As can be seen in the figure, this approach requires much more signaling and many more layer

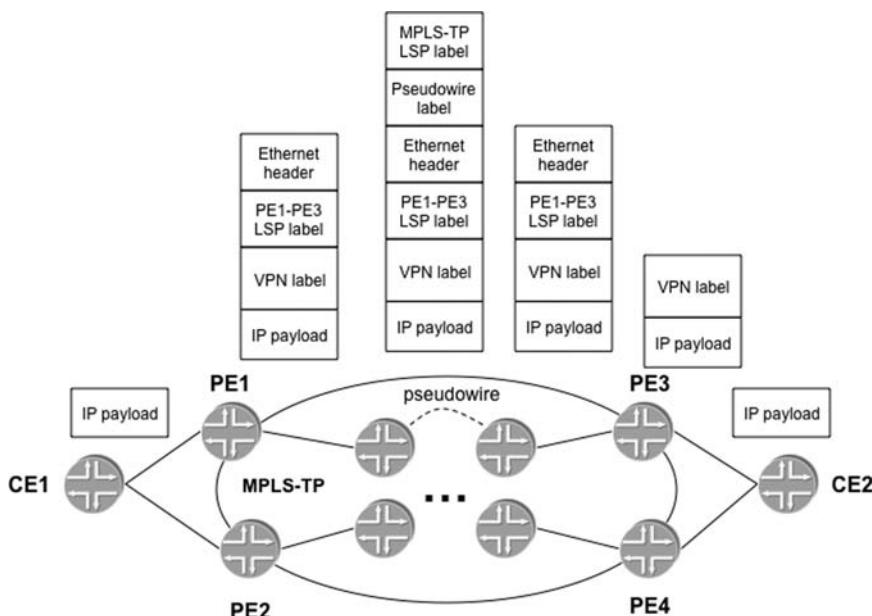


Figure 17.7 L3VPN deployment using MPLS-TP to the edges, with NLTS provided by pseudowires

networks. Although it provides the clean separation that was lacking in the previous solution, it does so at a significantly higher operational cost. Although pseudowires are one of the preferred mechanisms for implementing transport functions in MPLS, they are not required in MPLS-TP deployments. They are just one of the clients of MPLS-TP, along with IP and MPLS. The role of pseudowires in MPLS-TP deployments has been the subject of much debate and many misconceptions. While the use of pseudowires is not precluded, they are not mandated and their function of providing a transport service can be accomplished by other means as well.

17.6 MISCONCEPTIONS ABOUT MPLS-TP

MPLS-TP has been the subject of heated debate and is at the center of much standardization activity. As a result, there is no shortage of opinions on this topic and there is a very large body of drafts and documents, some of which will progress in the standardization process and some which will be abandoned. In this type of environment, it is often hard to see the overall picture, and it is easy to perpetuate misconceptions about the technology and its application. [MPLS2009_1] summarizes well some of the most popular misconceptions about MPLS-TP. Let us examine them here, along with the reasons for their perpetuation.

Because MPLS-TP is a new technology, the first false claim is that MPLS-TP is not part of the MPLS umbrella. In fact, as discussed in Section 17.4 and shown in Figure 17.3, MPLS-TP is based on a subset of MPLS. Related to this claim, it is argued sometimes that the extensions introduced for MPLS-TP do not apply to MPLS. In fact, these extensions, such as the transport-like OAM functions, are meant to apply generally to MPLS LSPs. Continuing on the topic of the relation between MPLS-TP and MPLS, another often-heard claim is that MPLS-TP requires substantial changes to the MPLS technology. In fact, one of the main goals of the MPLS-TP technology is to keep the MPLS architecture intact, and most of the changes for MPLS-TP are in the area of OAM, as discussed in Section 17.4.3. Another important goal is for all the MPLS-TP extensions to be reusable for dynamically signaled MPLS LSPs and pseudowires as well.

Because transport networks heavily rely on static provisioning, another claim often put forth is that MPLS-TP requires static provisioning. In fact, as discussed as early as the requirements stage, MPLS-TP supports both static provisioning and a dynamic control plane (based on GMPLS). Because pseudowires are the current MPLS abstraction for creating virtual circuits, and because MPLS-TP is built on top of LSP and pseudowire constructs, it is sometimes claimed that MPLS-TP requires the use of pseudowires. In fact, as we have seen in Section 17.5, the MPLS-TP server layer can be built using MPLS-TP LSPs, without the need to set up

pseudowires. Pseudowires are yet another client of MPLS-TP LSPs, along with IP and MPLS, but they are not required for providing the server layer functionality.

Finally, because cost was one of the drivers for defining a new packet-based transport, it is sometimes claimed that MPLS-TP will reduce the cost of MPLS transport equipment. In fact, what is required to reduce this cost is a transport-optimized MPLS LSR, which does not support the rich service functionality expected at the edges of the network and thus can be produced more cheaply. While MPLS-TP helps the industry to focus on this vision, MPLS-TP does not in itself solve the cost problem.

With time, as the MPLS-TP technology progresses and matures and the technology starts being adopted, these types of misconception will start disappearing.

17.7 CONCLUSION

Packet-switched transport is becoming increasingly important, driven by the tremendous growth in packet traffic, coupled with the pressure to reduce the cost per bit. The transport profile for MPLS, known as MPLS-TP, enables transport use cases, in particular a transport operational environment, in the MPLS architecture and forms the basis for next-generation converged packet networks. This new transport profile is implemented without the need for radical modifications to the MPLS architecture. MPLS-TP is both a subset of MPLS and a set of extensions, mostly in the area of improved transport-like OAM and resiliency and the ability to operate completely without a control plane. MPLS-TP opens the possibility to consider a transport-optimized MPLS LSR, supporting only the transport features of MPLS and thus cheaper to produce than a full-fledged LER/LSR box, as one way to reduce the cost of transport networks.

At the time of this writing, MPLS-TP is work in progress in the IETF, and many of the extensions have not yet been standardized. However, as a technology, it enjoys a lot of support, both among network operators and equipment vendors and it is already starting to be deployed in some networks.

17.8 REFERENCES

[BELOTTI]

P. Belotti, A. Capone, G. Carello, F. Malucci, *Multi-layer MPLS Network Design: the Impact of Statistical Multiplexing*, Computer Networks: The International Journal

- [CE_2008] of Computer and Telecommunications Networking, Vol 52, Issue 6 (April 2008) (available from http://antlab.elet.polimi.it/PUB/ComNet00_network_design.pdf)
- [ITU-T_G.805] K. Komppella, *Crisis in Transport (Moving the Purple Line)*, Carrier Ethernet World Congress 2008, Berlin
- [MPLS-WG] ITU-T Recommendation G.805 (03/2000), *Generic Functional Architecture of Transport Networks*
- [MPLSTP] <http://ietf.org/html.charters/mpls-charter.html>
- [MPLSTP_HIST] <http://trac.tools.ietf.org/misic/mpls-tp/>
- [MPLS2009_1] D. Ward, *MPLS-TP History and Future*, MPLS 2008, Washington DC
- [MPLS2009_2] R. Aggarwal, *MPLS in Transport Networks*, MPLS 2009, Washington DC
- [RFC4385] M. Bocci and L. Tancevsky, *Enabling Efficient Packet Transport with MPLS-TP*, MPLS 2009, Washington DC
- [RFC5317] S. Bryant, G. Swallow, L. Martini, and D. McPherson, *Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*, RFC4385, February 2006
- [RFC5586] S. Bryant and L. Andersson, *Joint Working Team (JWT) Report on MPLS Architectural Considerations for a Transport Profile*, RFC5317, February 2009
- [RFC5654] M. Bocci, M. Vigoureux, and S. Bryant, *MPLS Generic Associated Channel*, RFC5586, June 2009
- [RFC5704] B. Niven-Jenkins, D. Brungard, M. Betts, N. Sprecher, and S. Ueno, *Requirements of an MPLS Transport Profile*, RFC5654, September 2009
- [RFC5860] S. Bryant and M. Morrow, *Uncoordinated Protocol Development Considered Harmful*, RFC5704, November 2009
- [MPLS_TP_22] M. Vigoureux, D. Ward, and M. Betts, *Requirements for OAM in MPLS Transport Networks*, RFC5860, May 2010
- IETF - ITU-T Joint Working Team, *MPLS Architectural Considerations for a Transport Profile*, April 2008, http://www.ietf.org/MPLS-TP_overview-22.pdf

[TP_ACHTLV]	S. Boutros, S. Bryant, S. Sivabalan, G. Swallow, D. Ward, and V. Manral, <i>Definition of ACH TLV Structure</i> , draft-ietf-mpls-tp-ach-tlv-02.txt (work in progress)
[TP_OAM_FM]	I. Busi, B. Niven-Jenkins, and D. Allan, <i>MPLS-TP OAM Framework</i> , draft-ietf-mpls-tp-oam-framework-06.txt (work in progress)
[TP_FMWK]	M. Bocci, S. Bryant, D. Frost, L. Levrau, and L. Berger, <i>A Framework for MPLS in Transport Network</i> , draft-ietf-mpls-tp-framework-12.txt (work in progress)
[TP_LM_DM]	D. Frost and S. Bryant, <i>Packet Loss and Delay Measurement for the MPLS Transport Profile</i> , draft-frost-mpls-tp-loss-delay-01.txt (work in progress)
[TP_OAM_ANALYSIS]	N. Sprecher, H. van Helvoort, E. Bellagamba and Y. Weingarten, <i>MPLS-TP OAM Analysis</i> , draft-ietf-mpls-tp-oam-analysis-01.txt (work in progress)
[TP_NM_REQ]	H.K. Lam, S. Mansfield, and E. Gray, <i>MPLS TP Network Management Requirements</i> , draft-ietf-mpls-tp-nm-req-06.txt, in the RFC editors queue, soon to become an RFC

17.9 STUDY QUESTIONS

1. Why is interoperability with existing MPLS a top goal for MPLS-TP?
2. List three OAM functions added for MPLS-TP.
3. List three MPLS features that were dropped in MPLS-TP.
4. List two enhancements needed for LSPing in the context of MPLS-TP.
5. Explain the role of the GAL in MPLS-TP. Is the GAL always used?

18

Conclusions

18.1 INTRODUCTION

At the beginning of this book, we observed that in only a few years MPLS has become a mainstream technology used in a large proportion of service provider networks worldwide. One of the most successful MPLS-based services so far is Layer 3 VPN, now a lucrative revenue earner for many service providers. Also MPLS-based Layer 2 services are becoming available in many regions, either in point-to-point or multi-point form.

One of the reasons why MPLS has developed relatively rapidly is because of the pragmatic way that existing protocols have been adapted and new ones developed to support MPLS. In this book, we saw that an existing protocol, RSVP, was used as the basis for MPLS traffic engineering because of its properties of resource reservation and admission control. Additional properties were added to cater for the requirements of MPLS, such as the ability to distribute labels and specify the path to be followed by the traffic. On the other hand, LDP was developed specifically for MPLS, because no existing protocol had the required properties. An existing protocol, BGP, was adapted to carry the routes (and associated labels) of L3VPN customers, and the same scheme was then carried through to L2VPN and VPLS. A lesson learnt from the development of these VPN services was that no single VPN service type suits all customers so, for example, L3VPN is not inherently ‘better’ than L2VPN or vice versa. Thus in order to address the widest possible range of customers,

a service provider should consider offering the full range of MPLS-based VPN services. As more experience of running MPLS-based services is gained, additional features continue to be added to the underlying protocols, e.g. the addition of automated Route Target Filtering (RTF) to BGP to ensure that PE routers and route reflectors are not inundated with VPN routes that they are not interested in. Another example of the learning process was the realization that early schemes for supporting multicast over L3VPN did not have good scaling properties, leading to the recent work in the IETF to develop new schemes [BGP-ENCOD] [VPN-MCAST].

Sometimes MPLS has a reputation for being complex. We think this reputation is undeserved and probably arises because MPLS has a large variety of features. Note that these features were introduced for practical reasons – because service providers needed them in order to deploy services and to migrate their traffic from legacy networks. However, no one service provider needs all of the features, each chooses the subset they need. Recently, there have been moves to simplify the configuration of IP/MPLS routers [MPLS-PnP]. Here are some examples:

1. Traditionally, one important but tedious configuration task is assigning an IP subnet to each link in the network and making sure that the router at either end is configured correctly using a different address from that subnet. However, over time the protocols have been updated to cater for un-numbered interfaces. This means that it is only necessary to assign a loopback address to each node.
2. As described in previous chapters, the protocols can be leveraged to give automation, thus avoiding manual configuration. RSVP automesh and RSVP autobandwidth are examples of this.
3. Modern router operating systems have built-in scripting functions to allow router configurations to be automatically built according to user-defined requirements. For example, on each internal link, an IGP and an MPLS protocol can be turned on automatically and a queue scheduler automatically configured with the required weight for each queue. Typically, when configuring VPN services for multiple customers, many of the parameters are common to all customers and so in-built scripting functions can be used to configure those parameters automatically. Other parameters that differ between VPNs, such as route targets and route distinguishers, can be algorithmically derived by the script from a customer-specific ID number that the operator enters into the configuration.

In the rest of this concluding chapter, we take a look at some emerging trends in the field of MPLS, starting with an examination of converged networks.

18.2 NETWORK CONVERGENCE

A driving force for MPLS becoming the prevalent network technology in the future is network convergence, with many service providers consolidating disparate ‘stovepipe’ networks on to a single one based on MPLS. The endpoint for this convergence is for an MPLS network to carry all of a service provider’s traffic, including PSTN and mobile voice traffic, Layer 2 data, Layer 3 VPN, Internet and broadcast television.

Let us review the reasons why MPLS has made possible the deployment of critical services that IP-based networks were previously regarded as not capable of supporting:

1. *Flexibility with respect to connectivity.* An issue with native IP networks is that it is not possible to achieve end-to-end (PE-to-PE) bandwidth guarantees. MPLS achieves end-to-end bandwidth guarantees through traffic engineering and admission control, on a per-class basis if required. This ‘connection-oriented’ approach is highly desirable to meet the QoS requirements of traffic such as PSTN voice, broadcast video and some Layer 2 services, e.g. those that emulate or replace ATM CBR services. On the other hand, other classes of traffic can be handled without any bandwidth guarantees (or can be allowed to oversubscribe their bandwidth reservation) in order to make use of statistical multiplexing. This ‘mix and match’ approach helps the service provider make good use of bandwidth resources without being too rigid (all traffic having to have an associated bandwidth reservation) or too loose (no bandwidth guarantees for any traffic).
2. *Aggregation properties.* An LSP can carry all of the traffic of a particular class, or of multiple classes, between a pair of PEs. In general, many end-to-end microflows would be aggregated on to that LSP. As a consequence, the core of the network does not contain the state related to those individual flows. If the number of LSPs in the core of the network becomes an issue, because of the control-plane overhead associated with maintaining them, the hierarchical properties of MPLS allow LSPs to nest inside other LSPs. In this way, the growth of the network is not constrained by the number of microflows or the number of LSPs that the network is required to carry.
3. *Single forwarding mechanism.* This is based on label swapping, regardless of the traffic type being carried. This makes it easier to carry new types of traffic, as only the PE routers need to understand the semantics of the native encapsulation of the traffic being carried. The core routers are shielded from this detail.
4. *Failover mechanisms.* IP-based networks have a reputation for a slow response to events such as the failure of a transmission link. In contrast,

MPLS fast reroute gives failover times comparable to SONET/SDH transmission networks.

5. *Ability to support multiple services on common equipment.* An MPLS PE router can offer multiple services and can encapsulate multiple protocol types into MPLS, including IP packets, ATM cells and Ethernet frames. The variety of media types supported gives the service provider flexibility in the way customers are connected to that service provider, including over an ATM or Frame Relay access network, over Ethernet, over DSL or over SDH/SONET.

MPLS has helped change the reputation of IP-based networks from being regarded as only suited for 'best-effort' service to being considered capable of carrying critical traffic. Other factors unrelated to MPLS have also contributed to this improved reputation. These include hardware-based forwarding, which allows better control over latency and jitter, as well as increasing the forwarding rate that the equipment is capable of handling. Also, vendors are introducing high availability features, including the following:

1. The separation of control and forwarding planes on modern routers allows schemes such as graceful restart, where control processes can restart without interruption to traffic.
2. Component redundancy, such as control processor modules and switch fabrics.
3. In-service software upgrades, allowing software upgrades while the equipment is still running, thus reducing the reliance on maintenance windows.
4. Bidirectional forwarding detection (BFD), allowing forwarding plane failures to be detected in a timely manner.

A key advantage of building a converged network is capital expenditure (CAPEX) savings, as fewer pieces of equipment are required to support the range of services in the service provider's portfolio. As well as CAPEX savings, operational expenditure (OPEX) savings can be made because there are fewer networks to maintain and manage. However, even when having a single network to run, in order to fully realize OPEX savings, it is important to make the network simple to run, e.g. by having common signaling infrastructure and common operational procedures for each type of service being offered. Otherwise the convergence is incomplete, only taking place at the physical equipment level and packet-forwarding level rather than also encompassing the control plane.

In this book, we have discussed the reasons why BGP was chosen as the signaling and autodiscovery mechanism for Layer 3 VPNs. We also showed how the BGP-based mechanisms and associated operational procedures

Table 18.1 Reachability information carried by BGP

Service type	Reachability information carried by BGP
Unicast L3VPN	VPN-IP prefixes
Multicast L3VPN	Multicast receivers
L2VPN	CE IDs
VPLS	VE IDs

have been carried through to a BGP-based signaling and autodiscovery scheme for Layer 2 and VPLS services. Furthermore, we have discussed current proposals in the IETF to allow BGP to be used as a signaling mechanism within the core of the network for Layer 3 VPN multicast traffic. Thus BGP can be used to carry reachability information for all the MPLS-based services that exist today. Table 18.1 shows the services and the reachability information conveyed by BGP for each.

From the operational point of view, having a single protocol and a shared signaling infrastructure (comprising the BGP sessions and the route reflectors) to carry reachability information for all these services is a key advantage. The flexibility of the BGP protocol means that it can be used to support future services, as well as those listed in the table, without having to compromise the fundamental protocol semantics. This is achieved by simply defining a new address family to carry the requisite reachability information. One example is a proposal to use BGP to carry ATM addresses, e.g. Network Service Access Point (NSAP) addresses [ATM-BGP]. Another example is the use of BGP to carry CLNS reachability information, which we discuss further in Section 18.5.

The hierarchical routing support available in BGP through route reflectors helps in the control plane scaling of networks as they grow, as a full mesh of sessions is not required between PEs in order to convey reachability information. The autodiscovery properties of BGP mean that adding a new customer site to an existing service involves having only to configure the PE(s) directly attached to that site, rather than having to configure every PE in the network that serves the customer in question. Network operators take this property for granted when it comes to Layer 3 VPN and would find it unacceptable if they had to configure every PE in the network simply to accommodate a new customer site. BGP gives the same ease of configuration to L2VPN and VPLS services. This property will become increasingly necessary as networks grow and services begin to span multiple service providers.

The model for converged networks using the ingredients discussed in this chapter and throughout this book is summarized in Figure 18.1. The

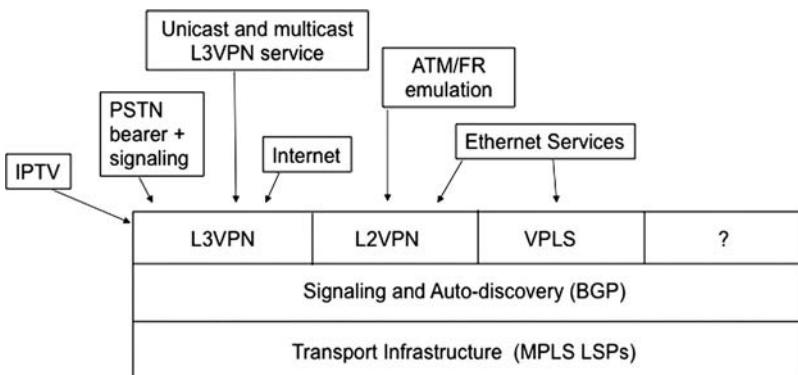


Figure 18.1 Model for converged network

packet transport layer at the bottom of the diagram is based on a mixture of point-to-point and point-to-multipoint LSPs, to cater for different traffic types. If needed, DiffServ Aware TE can be used to perform per-class admission control and give bandwidth guarantees to those classes of traffic that need it. The signaling and autodiscovery function for all the MPLS-based applications is provided by BGP. The MPLS-based applications shown are Layer 3 VPN, Layer 2 VPN and VPLS. The box with the '?' symbol represents future MPLS-based applications, to convey the fact that these are also likely to be underpinned by the same BGP signaling layer and MPLS LSP transport infrastructure.

Shown in the diagram are various traffic types that map onto the MPLS-based applications. Layer 3 VPNs, as well as being used for explicit L3VPN service to end-customers, can be used as an infrastructure tool to carry PSTN traffic and Internet traffic. In a similar way, the Next-Generation Multicast L3VPN scheme described in Chapter 10 is now being deployed as an infrastructure tool for transporting IPTV. L2VPNs can be used to carry emulated ATM and Frame Relay services and point-to-point Ethernet services, while multipoint Ethernet services are provided by VPLS.

18.3 INTERACTION WITH CLIENT EDGE EQUIPMENT

In earlier chapters, we discussed the admission control of LSPs into the network. It is anticipated that a further level of admission control will be required for some traffic types: admission control of client connections into LSPs by the ingress PE. As discussed in the DiffServ TE chapter of this book (Chapter 4), some implementations already have the ability to manually configure parameters such as the bandwidth requirement of a pseudowire on the ingress PE. The ingress PE then performs admission

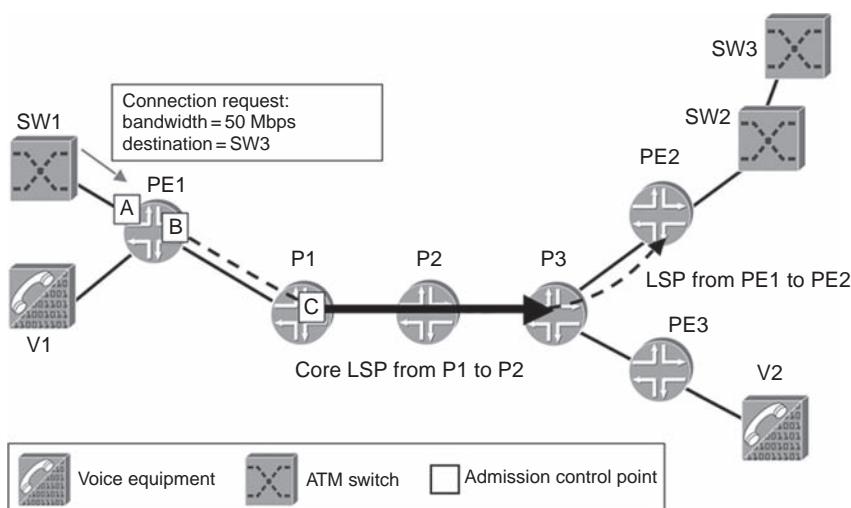


Figure 18.2 Resource reservation by client equipment

control of the pseudowire on to an LSP that goes to the appropriate egress PE. The natural next step is for the client edge equipment to signal its bandwidth and other connection requirements to the ingress PE, to avoid manual configuration on the PE. Schemes have been discussed in the MPLS and Frame Relay Alliance,¹ in the context of signaling interworking between ATM switches and MPLS PE routers [MPLS-ALL]. In this scheme, the ATM switch aggregates multiple VCs and VPs that need to pass to the same remote ATM switch into a single connection. The ATM switch signals to its local PE router the bandwidth requirements for that connection, which the PE translates into bandwidth requirements for a corresponding pseudowire to the egress PE. This is illustrated in Figure 18.2, in which an ATM switch, SW1, requires a trunk to SW3. It signals the bandwidth requirements of the connection to the attached PE router, PE1. PE1 performs admission control of a pseudowire on to the LSP PE2, and the connection is admitted if sufficient resources exist.

In this way, in the network there is a hierarchy of admission control. The admission control points are shown in the diagram. At point A, admission control is performed of client circuits on to LSPs. At point B, admission control of edge-to-edge LSPs from PE1 into the network is performed. In turn, LSPs from the edge may be nested inside other LSPs, in which case additional admission control occurs, e.g. at point C. This hierarchy

¹The MPLS and Frame Relay Alliance became the IP and MPLS Forum and subsequently merged with the Broadband Forum.

of admission control improves scalability. PE1 is aware of the connection from SW1, but is not aware of the individual VCs and VPs contained within. P1 is aware of the LSP from PE1 to PE2, but is not aware of the pseudowires or other services carried within. P2 is only aware of the core LSP from P1 to P3, and is not aware of any PE-PE LSPs carried within.

An interesting question is how does PE1 know that SW3 is in the ATM island 'behind' PE2? One proposal, already mentioned in passing in this chapter, is for BGP to carry ATM address reachability information between the PE routers [ATM-BGP].

The principle of client equipment signaling bandwidth requirements to the MPLS network could be applied to cases other than ATM. For example, a possibility is for voice equipment (e.g. V1 and V2 in the figure) to signal bandwidth requirements for voice bearer traffic to their local PE.

18.4 INTERPROVIDER CAPABILITY

Today the MPLS-based services described in this book are predominantly used in a single provider, single customer mode. That is to say, typically when data from a customer arrives on the service provider's network, it is carried across that network and is delivered to another site of that same customer. Usually the only common denominator that allows traffic to pass between different end customers and across multiple service providers is the Internet. This is a problem because the Internet does not have any guarantees with respect to bandwidth or treatment of packets at each hop. Also roaming corporate workers usually only have access to their corporate network via a VPN solution that runs over the Internet, which can result in poor performance for certain applications. Because many large corporations have presence in all corners of the world and increasingly need connectivity between sites to run their business applications, seamless global connectivity with quality guarantees is an important requirement.

However, no single service provider has the combination of global coverage and high penetration within each country required to offer such customers seamless global MPLS-based services. In some cases, certain service providers do have interconnection arrangements for MPLS-based services. To date, such arrangements are very fragmented, typically involving isolated pairs of service providers. As yet, there is no general interprovider connectivity where traffic can pass through a chain of service providers with a similar SLA to that experienced in the single provider case. Ideally, one would wish to arrive at a position analogous to the PSTN, where it is taken for granted that one can dial any number in the world regardless of which service provider the dialed number is attached to, or which other service providers the call needs to pass through.

At the time of writing, work is in progress towards creating such a generic interprovider connection scheme [IPSPHERE]. The scope of the work required extends beyond the MPLS layer to aspects such as:

- Negotiation of session parameters such as QoS, bandwidth and security requirements between the end customer and the carrier, or between carriers.
- Billing to the end customer and intercarrier settlement payments.

The work is still in progress, but aims to solve significant constraints in the way data networks are used today. As discussed earlier in this book, work has also been carried out in the IETF to enable better interprovider connectivity at the MPLS layer, for example the work on interdomain traffic engineering.

18.5 MPLS IN THE DATA COMMUNICATIONS NETWORK (DCN)

Typically, service providers have large internal networks called Data Communications Networks (DCNs). These carry OSS and management traffic for various types of network equipment, including SONET/SDH cross-connects, DWDM equipment and routers. These networks can have a large number of sites – for example in the case of a DWDM transmission system, each amplifier site needs to be hooked into the DCN. In many service providers, several separate DCNs have been built over the course of time, each being used to manage different types of equipment. There is strong interest in merging the separate DCN networks into one network having an MPLS backbone. A popular option is to use VPNs to maintain separation between the different DCNs, as sometimes they are managed by different departments and may have overlapping addresses.

Some of the network equipment being managed via the DCN uses the ISO Connectionless Network Service (CLNS) rather than IP. In order to carry the CLNS traffic across the MPLS backbone, a scheme has been devised which is very similar in concept to a BGP/MPLS IP VPN [BGP-CLNS]. Like the IP VPN scheme, a model in which the PE is a routing peer with the CE is used. Native CLNS routing is used between a PE and locally attached CEs, and the CLNS routes from different VPNs are stored in separate VRFs. CLNS reachability information is communicated between PEs via multiprotocol BGP in the form of labeled routes, using an address family called VPN-ISO.

The CLNS VPN is another proof-point of the versatility of multiprotocol BGP, in that it can be used to carry reachability information for a wide variety of VPN schemes.

18.6 MPLS IN MOBILE NETWORKS

At the time of writing, the cores of some mobile networks had been migrated to an MPLS infrastructure, providing a common platform for transport of both mobile voice and mobile data services [MPLSWC 2006]. As with the migration of fixed network core infrastructure to MPLS, the driver was to reduce CAPEX and OPEX by having one common network platform capable of carrying all the services.

In the previous chapter, we discussed how service providers are now turning their attention to the migration of fixed access networks to MPLS, again as a way to have one common network to replace multiple legacy access networks. In parallel, the same process is happening for the Radio Access Network (RAN) within mobile networks. In this way, legacy ATM and TDM equipment and expensive leased lines can be replaced by an MPLS-based access network. An additional driver is that future releases of third-generation (3G) mobile network architectures will be IP based rather than ATM or TDM based, so MPLS provides a natural fit to transport such traffic.

Many of the techniques described in the previous chapter, such as the use of pseudowires, are also applicable to mobile access networks.

In some cases, the same MPLS-based access network would be used for both mobile and fixed traffic, for example if a service provider offers both fixed and mobile services. Another example is where a fixed operator sells transport services across an MPLS access network to a mobile operator that does not have its own transport infrastructure.

An MPLS-based RAN needs to be able to cater for the backhaul of both the second-generation (2G) and the 3G traffic as many mobile networks have 2G-based services but are migrating over time to 3G. Figure 18.3 shows a schematic diagram of the existing mobile access network.

The role of the RAN is to interconnect each mobile base station to its parent regional controller device.² The network has a star topology as there is no interconnection between base stations in today's architectures. Typically, around 50 or 100 mobile base stations are homed to each regional controller. The traffic is carried from the base stations using E1 or T1 leased lines or microwave links. A problem is that mobile data traffic volume is currently growing rapidly, resulting in the need for more E1 or T1 circuits to serve each base station, but the revenue is not growing at the same rate because of the use of flat rate tariffs. These circuits are expensive to rent, so currently the access network comprises a significant proportion of a network operator's expenditure. Furthermore, if a base station

² In the 2G case, the base station is known as a Base Transceiver Station (BTS) and the controller is known as a Base Station Controller (BSC). In the 3G case, the base station is known as a Node B and the controller is known as a Radio Network Controller (RNC).

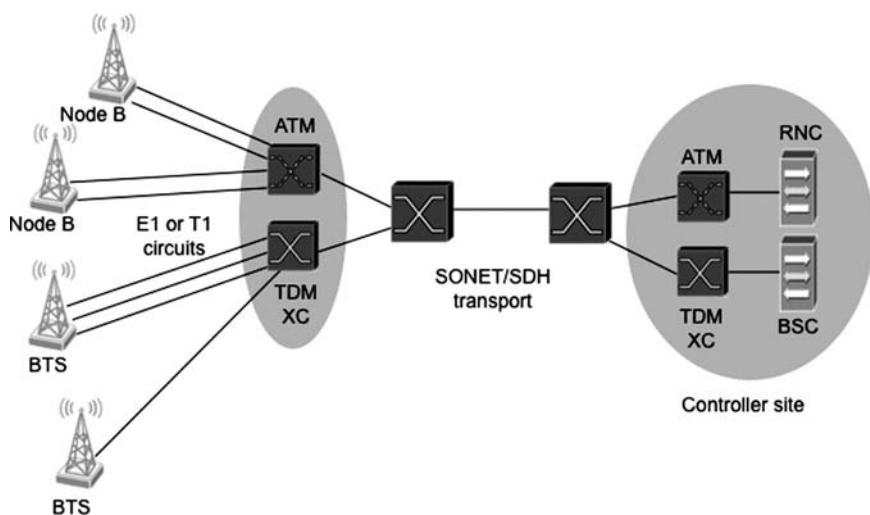


Figure 18.3 Schematic diagram of existing RAN

site supports both 2G and 3G services, then separate circuits are needed for each.

In some cases, traffic from multiple base stations is aggregated within the RAN. In the 2G case, the traffic is in TDM format. At the aggregation site, a TDM cross-connect multiplexes the E1/T1 circuits into SONET/SDH circuits. In the 3G case, the traffic is in ATM format, so an ATM switch is used to aggregate the traffic.

It is more bandwidth efficient and cost effective to have an MPLS packet-based network to aggregate traffic from multiple base stations and transport it to the regional controller site. This is illustrated schematically in Figure 18.4. This allows the elimination of expensive E1 or T1 circuits and TDM and ATM aggregation equipment. It also means that the back-haul is not tied to SDH/SONET transport, so an Ethernet-based access network can be used instead.

If the transport LSPs within the MPLS RAN are signaled by RSVP, fast reroute can be deployed to achieve rapid recovery from link failures. Also, admission control can be used on the LSPs in order to give bandwidth guarantees to the transported traffic.

In order to migrate 2G TDM traffic to an MPLS RAN network, TDM pseudowires can be used to emulate the E1 or T1 circuits, as described in [RFC4553] or [RFC5086]. In this way, the contents of an entire E1 or T1 circuit can be transported across the MPLS network without having to modify the client equipment in the base stations or the regional controller sites.

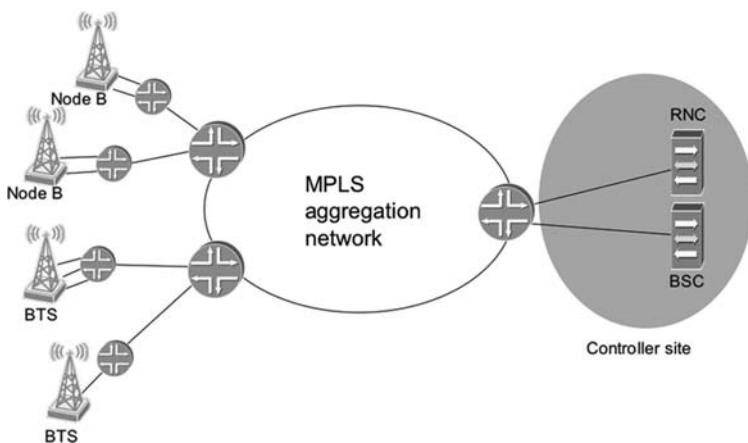


Figure 18.4 Schematic diagram showing MPLS RAN

In the 3G case, up to and including Release 4, traffic is carried in ATM format. In order to carry this traffic over an MPLS-based RAN, ATM cell transport over pseudowires can be used, as described in Chapter 11.

As mentioned earlier, in later releases of 3G, traffic will be carried in IP format and Node B equipment will have IP/Ethernet interfaces rather than the current E1/T1 ATM interfaces. The 3G longterm evolution (LTE) changes the RAN connectivity model from the existing star-based scheme to a meshed scheme to allow direct base-station to base-station connectivity at the IP layer, for handover purposes. In an MPLS-based RAN, L3VPN can provide the infrastructure for carrying the IP traffic and can also be used to support WiMAX traffic, which is also IP-based.

In conclusion, MPLS provides a good fit to the requirements of mobile RANs. The ability of MPLS to carry a wide range of traffic types and its independence of the underlying link technology make it well suited to support the TDM, ATM and IP technologies associated with successive generations of mobile architecture.

18.7 MPLS IN THE ENTERPRISE

The focus of this book has been the use of MPLS by service providers to offer services to enterprise customers, as this is the main way in which MPLS is used today. However, an emerging trend is for larger companies to use MPLS as part of their own internal network infrastructure. Such an enterprise treats its network as a mini version of a service provider network, providing services to the various departments within the company. In some cases, there is a requirement for data separation between

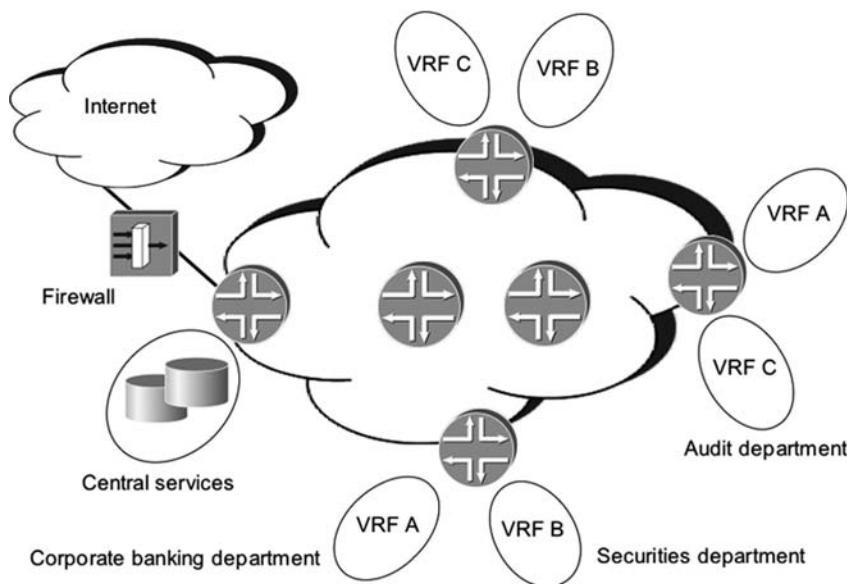


Figure 18.5 Use of Layer 3 VPN in an enterprise network

certain areas of the company. For example, some financial institutions have sensitive areas that should not be accessible from the company in general, or may need to maintain internal walls between different business units or subsidiaries for reasons of client confidentiality or avoiding conflicts of interest. A typical model is to use Layer 3 VPNs, to constrain the connectivity between different departments, while still having the ability to have shared resources accessible from all the departments that require it. This model is illustrated schematically in Figure 18.5.

Another emerging enterprise application at the time of writing is the use of VPLS to provide layer 2 connectivity between data centers. Several distributed functions associated with data centers require layer 2 connectivity between the devices involved. Using VPLS is a convenient way to allow these functions to span multiple data centers, thus allowing more efficient use of processing and storage resources regardless of location. Examples of such functions include:

- *Server virtualization.* Server virtualization is becoming increasingly prevalent. A virtual server residing on one host can be moved seamlessly to another host in the case of hardware failure, for maintenance purposes or in order to optimize resources. Because a virtual server retains the same IP address regardless of its physical location, all of the possible physical locations for the server must be on the same LAN.

- *Data replication and backup.*
- *Disaster recovery.* If one data center should become unavailable due to fire or flood, a backup data center can be brought into action in a seamless way.

Using VPLS rather than native layer 2 switching to provide the extended layer 2 domain has the advantage that the spanning tree protocol can be confined to individual data centers. As discussed in more detail in Chapter 16 (MPLS in Access Networks and Seamless MPLS), operators dislike having large spanning tree domains because of risks of instability and operational difficulties. Using BGP for the VPLS control plane also brings the advantages of easy provisioning, due to BGP's autodiscovery properties, and the VPLS multihoming scheme for resilience. The use of multihoming is shown in Figure 18.6. Suppose in Data Center A, normally PE1 is the active VPLS PE. However, if PE1 itself or PE1's attachment circuit into the data center should fail, PE2 becomes the active PE, through the mechanisms described in the VPLS chapter (Chapter 13). As described in that chapter, the BGP multihoming scheme has in-built loop-prevention

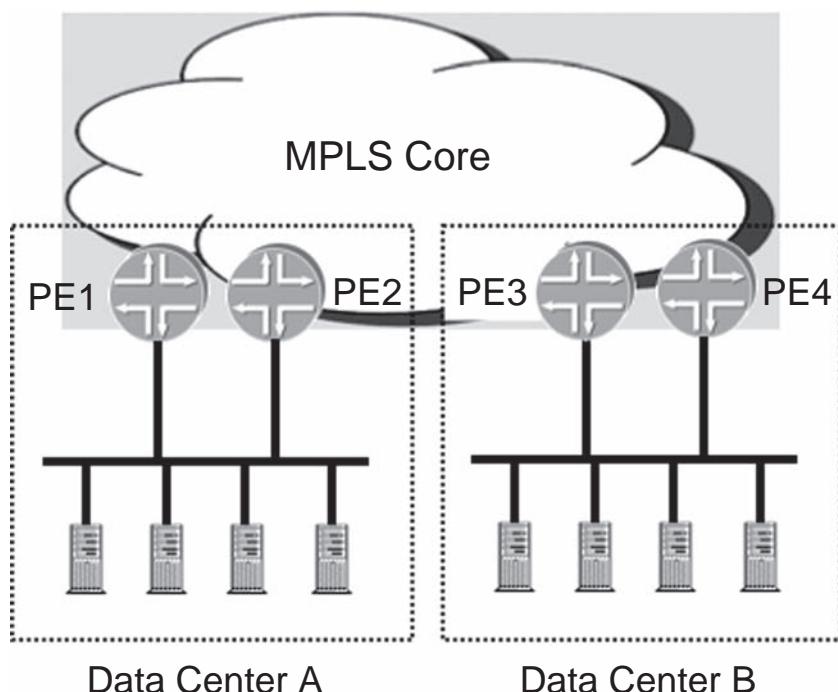


Figure 18.6 Use of VPLS to interconnect data centers

mechanisms. For example, while PE1 is the active forwarder for Data Center A, PE2 blocks its attachment circuit to traffic, in order to prevent forwarding loops occurring.

While the actual routers in enterprise deployments may be smaller in capacity than those used in service provider networks, the principles remain the same, with P routers that perform MPLS forwarding without awareness of the routes carried within each VPN, and PE routers with VRFs corresponding to the various departments that the network serves. For multinational companies having a presence in all continents, the geographical span of such networks can sometimes be greater than that of some service provider networks. In such cases, attention is often paid to optimizing the path of intercontinental traffic to avoid excessive latency, either by juggling IGP metrics or through MPLS traffic engineering.

18.8 MPLS IN THE TRANSPORT

Chapter 17 described the MPLS Transport Profile (MPLS-TP), currently under development in the IETF. The ability to add a new profile to MPLS without any changes to the existing architecture is proof of the flexibility of this technology. While MPLS-TP is not an application per se, it is an important future direction for MPLS, and a natural evolution for the packet-transport functionality of MPLS. MPLS-TP opens up a whole new world of possibilities, among them the ability to build cost-optimized MPLS LSRs that can help reduce the cost for transport networks.

18.9 FINAL REMARKS

As can be seen from the examples given in this book, the scope of MPLS and the way it is being used has extended beyond what even its more optimistic proponents might have predicted when the work started only a few years ago. Indeed, the use of MPLS in the enterprise discussed in the previous section is an example of MPLS technology being used in an unexpected way. This shows that the question is not whether the technology was originally intended for a particular purpose but whether it can fulfill a particular purpose efficiently. The scaling properties and extensibility of a solution is what determines its ultimate success and deployment. In this final chapter we have attempted to give a flavor of the directions MPLS may go in the future as it takes center-stage as a technology that spans multiple providers and underpins the delivery of all voice and data services.

18.10 REFERENCES

- [ATM-BGP] C. Kodeboyina, C. Metz and P. Busschbach, *Carrying ATM Reachability Information in BGP*, draft-ck-bgp-atm-nlri-01.txt (expired draft)
- [BGP-CLNS] Q. Vohra, D. Steinberg, A. De Carolis, *BGP-MPLS IP VPN Extensions for ISO/CLNS VPN*, draft-vohra-l3vpn-bgp-clns-00.txt (expired draft)
- [BGP-ENCOD] R. Aggarwal, E. Rosen, T. Morin, Y. Rekhter, C. Kodeboniya, *BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs*, draft-ietf-l3vpn-2547bis-mcast-bgp-08.txt (work in progress)
<http://www.tmforum.org/ipsphere>
- [IPSPHERE]
- [MPLS-ALL] T. Walsh and R. Cherukuri, *Two Reference Models for MPLS Control Plane Interworking*, MPLS/FR Alliance Technical Committee document mpls2005.050.00, March 2005
- [MPLS-PnP] K. Kompella, *MPLS Plug-and-Play for Metro Ethernet*, Paper D1-04, MPLS World Congress, Paris, February 2007
- [MPLSWC 2006] J. Gomes and P. Bridge, *Orange: Experience on Multiservice MPLS Core Networks*, Paper D3-06, MPLS World Congress, Paris, February 2006
- [RFC4553] A. Vainshtein and Y.J. Stein (eds), *Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)*, RFC 4553, June 2006
- [RFC5086] A. Vainshtein (ed.), I. Sasson, E. Metz, T. Frost, P. Pate, *Structure-aware TDM Circuit Emulation Service over Packet Switched Network (CESoPSN)*, RFC 5086, December 2007
- [VPN-MCAST] E. Rosen and R. Aggarwal (eds), *Multicast in MPLS/BGP IP VPNs*, draft-ietf-l3vpn-2547bis-mcast-10.txt (work in progress)

Appendix A: Selected Backhaul Scenarios in MPLS-Based Access Networks

A.1 INTRODUCTION

In this Appendix, we look in detail at how traffic can be back-hauled across an MPLS-based access network. The scenarios we look at are for the case where the access networks and core networks are separate MPLS islands, with a non-MPLS hand-off between access and core. First, we will examine the case of dedicated business VPN services and then we will examine some DSL scenarios. This Appendix assumes familiarity with the connectivity blueprints shown in Figure 16.2 of Chapter 16.

A.2 DATA SERVICES FOR BUSINESS CUSTOMERS

Let us assume that the service provider is offering the following services to business customers:

- L3VPN. As discussed in the L3VPN chapter, at the time of writing this was the predominant MPLS-based service offered to business customers.
- L2VPN service, as a replacement for leased lines and ATM and Frame Relay services.
- VPLS service for LAN interconnect.

- Internet access service. In the case of some service providers, Internet routes are stored in the main routing instance of the service node (SN). In other cases, an L3VPN is used as the internal infrastructure for Internet service, with the routes being stored in a particular ‘Internet VRF’. Either way, the traffic needs to be carried across the access network between the customer sites and the SNs.

Note that in some cases, the Ethernet access network also carries PSTN bearer and signaling traffic. Often an L3VPN is used as the internal infrastructure to carry this traffic over the core part of the network, as the signaling plane is a closed application with strict security requirements. In this case, similar considerations apply for backhauling the traffic across the access part of the network as for explicit L3VPN services supplied to business customers. From the point of view of the L3VPN carrying the PSTN traffic, the CE devices are the MSANs, voice gateways and other voice equipment.

In order to transport the traffic associated with the services listed above from the customer sites across the access network to the SNs (e.g. the PEs of the VPN services), pseudowires can be used. In principle, the pseudowires can be signaled using BGP or LDP. In practice, it is useful to have as much automation as possible especially in large deployments, so the BGP signaling and autodiscovery scheme is likely to be the most convenient.

A.3 RESILIENCE CONSIDERATIONS

In some cases, a customer requires fully redundant access to their VPN service. This implies that the service provider needs to provide access to redundant VPN SNs and alternative paths through the MPLS access network that uses different transport nodes (TNs). This is illustrated in Figure A.1. Similar considerations apply when L3VPN is being used as infrastructure to carry PSTN traffic.

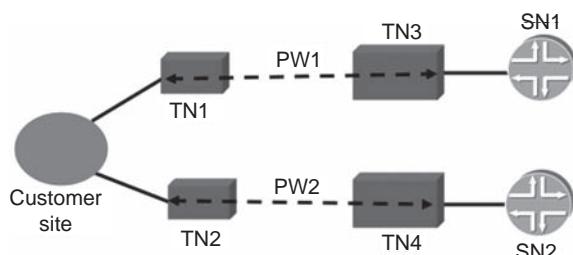


Figure A.1 Providing resilient access to VPN services

In the case where an L3VPN service or Internet service is provided, both access ‘circuits’ between the customer site and the two VPN SNs can be permanently up, and routing on the SNs and CEs controls whether only one circuit or both are used to carry traffic under normal circumstances.

In the case where VPLS service is provided, the service provider may enforce that only one circuit is used at any one time to avoid the danger of forwarding loops occurring. This can be achieved by applying the VPLS PE multi-homing scheme discussed in Chapter 13 to SN1 and SN2.

In the case where an L2VPN service is provided, the service provider could allow both circuits to be used simultaneously. From the point of view of the customer, they then have two complete parallel and independent pseudowires (assuming the mirror image of the arrangement in Figure A.1 is provided in the metro areas serving the customer’s other sites). Alternatively, by analogy with the VPLS case, the service provider could use the L2VPN PE multi-homing scheme discussed in Chapter 12 on SN1 and SN2 such that only one of the access ‘circuits’ carries traffic at any one time.

In all of the cases discussed above, the SNs and the CEs need to be able to detect failures in the circuit between them so that traffic is not blackholed. In the L3VPN case, if a routing protocol is being used between the CE and the SN, such as eBGP or OSPF, the time-out on the routing protocol can be used to detect failure. If this is not fast enough, or if static routing is used, then BFD can be used to detect failure. An alternative to BFD is to use Ethernet OAM, for example in the form of 802.1ag [802.1ag]. In the case of L2VPN and VPLS, Ethernet OAM can also be used to detect failure.

In this section, we discussed how business VPN services can be backhauled to the VPN SNs located at the border between the access and the core network. In the next section, we look at how xDSL traffic can be backhauled to xDSL SNs, again located at the border between the access and the core network.

A.4 xDSL BACKHAUL SCHEMES

In this section, we turn our attention to the backhaul of xDSL traffic to SNs across the MPLS access network. In order to do this, we will discuss two alternative schemes that exist for xDSL backhaul when using an access network comprising Layer 2 Ethernet switches. These schemes have been created in recent years to allow service providers to migrate from ATM-based to Ethernet-based xDSL access networks. This migration has been driven by the increasing bandwidth requirements of triple-play services and the need to carry multicast traffic efficiently, which is easier to achieve with Ethernet than ATM. We will show how each of the xDSL backhaul schemes can be migrated to an MPLS-based access network. It is beyond the scope of this book to discuss xDSL architectures in detail

and the pros and cons of each model. However, from the point of view of people designing MPLS-based access networks, it is important to know that multiple xDSL models exist and how to map them to an MPLS-based access network. A very good reference for readers wanting to know more about the details of xDSL architectures is [BBARCH].

A.4.1 Single edge, customer VLAN model

In the pure Customer VLAN model (sometimes known as the 1:1 model), each DSL customer has its own VLAN which carries all of the traffic belonging to that customer: Internet, VoD, VoIP and Broadcast TV. However, in practice, it is advantageous to put the Broadcast TV traffic into one common multicast VLAN which serves all of the DSL subscribers. This saves bandwidth compared to sending multiple copies of the same content, one copy in each customer VLAN. This model is sometimes called the hybrid 1:1 model, which is the model that we will examine in this section. Typically the 1:1 model, or hybrid 1:1 model, is used in situations where the service provider is using a single type of SN to deliver all the DSL services. Customer VLANs (C-VLANs) that are associated with the same DSLAM are grouped into Stacked-VLANs (S-VLANs). It is the role of the access network to carry the S-VLANs between the DSLAMs and the SNs – the access network is not aware of the individual C-VLANs. Figure A.2 illustrates how the S-VLANs are implemented, with the individual C-VLANs nested inside. This is analogous to previous ATM architectures with a VP/VC hierarchy.

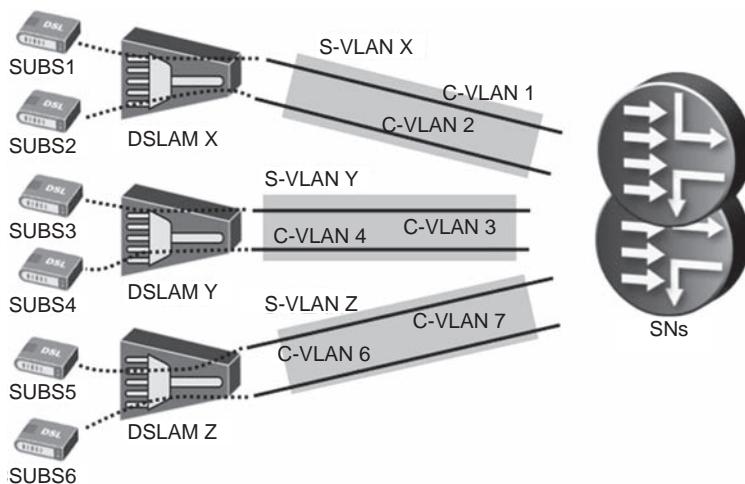


Figure A.2 Connectivity of customer VLANs in the single-edge model

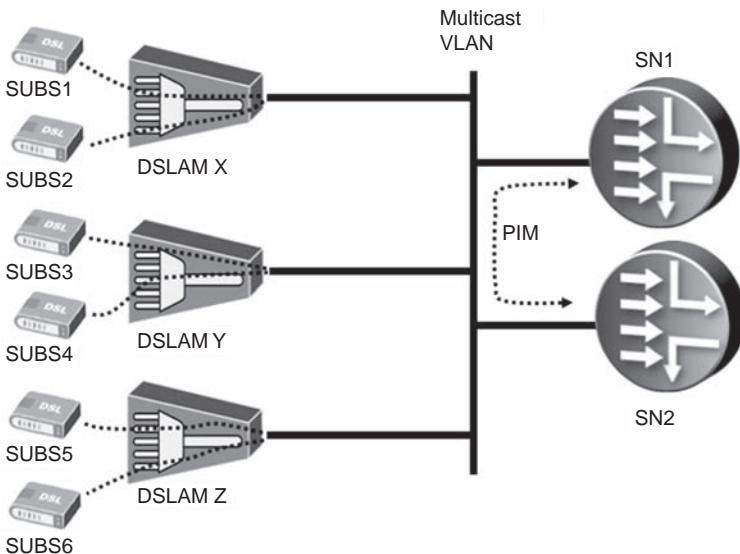


Figure A.3 Connectivity of multicast VLAN

Figure A.3 illustrates how the multicast VLAN is implemented. Let us see how this hybrid 1:1 scheme can be implemented using an MPLS-based access network to transport the VLANs between the SNs and the DSLAMs.

The most natural way to carry the S-VLANs is to map each onto a separate pseudowire. This is illustrated in Figure A.4. From the point of view of each pseudowire, the two PEs are the TN facing the SN and the TN facing the DSLAM. For example, for pseudowire X, the PEs are TN1 and TN3. The two CEs from the pseudowire point of view are the SN and the DSLAM, for example SN1 and DSLAM X from the point of view of pseudowire X. Unlike the case where Ethernet switches are used to carry the S-VLANs, an advantage when using pseudowires is that no MAC learning is required on the MPLS nodes in order to carry the S-VLANs, as each pseudowire is simply providing a point-to-point connection.

Typically in DSL deployments in the past, a redundant SN was not used. However, more recently, as availability requirements increase for residential voice and video services, there is increasing interest in providing redundancy. In that case, a redundant SN can be used (SN2 in Figure A.4). The switch-over to the redundant SN can be achieved using the BGP multi-homing scheme for L2VPNs described in Chapter 12. This requires TN3 and TN4 to be configured with the same CE ID, with TN3 as the preferred exit point. In this way, if TN3 goes down or the link to the SN1 or the SN itself goes down, then TN4 takes over and the traffic can pass to SN2.

Note that this scheme differs from the situation where the access network is based on Layer 2 Ethernet switching as both SN1 and SN2 would

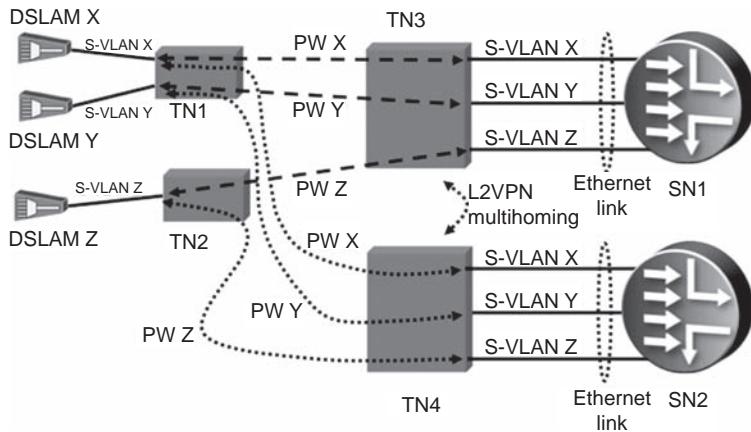


Figure A.4 Mapping S-VLANs to separate pseudowires

be reachable under normal circumstances in the latter case. However, this ‘active–active’ configuration results in non-deterministic behavior as it is not predictable which SN a particular subscriber will become attached to, in terms of their PPP or DHCP session. In contrast, with the ‘active-standby’ scheme that the L2VPN multi-homing allows, it is known that all the sessions are on the primary SN under normal circumstances. Whichever scheme is used, typically the main contributor to the failover time is the time taken for the PPP or DHCP session associated with each subscriber to come up on the remaining SN when the other one fails. If the operator in fact prefers an ‘active-active’ scheme, VPLS would be used instead of L2VPN in order to give the required multipoint behavior.

Note that the L2VPN scheme can provide multiple levels of redundancy, for example 1:1 failover between nodes within the same POP, and a further level of emergency fall back to a more central site.

A.4.2 Multicast VLAN in the 1:1 hybrid model

Let us now look at the requirements for multicast traffic. The multicast VLAN needs to serve all of the DSLAMs, as illustrated in Figure A.3. Suppose a set-top box sends an IGMP request for the required TV channel. The DSLAM snoops the IGMP messages so that it knows which subscriber needs to receive which multicast group. In some deployments, the DSLAM sends the IGMP request upstream in the C-VLAN, in other deployments in the multicast VLAN and in other deployments both. Regardless of this, the actual multicast data traffic flows downstream from the SN in the multicast VLAN. In order to achieve redundancy, two SNs can be deployed, each being attached to the common multicast VLAN. They form

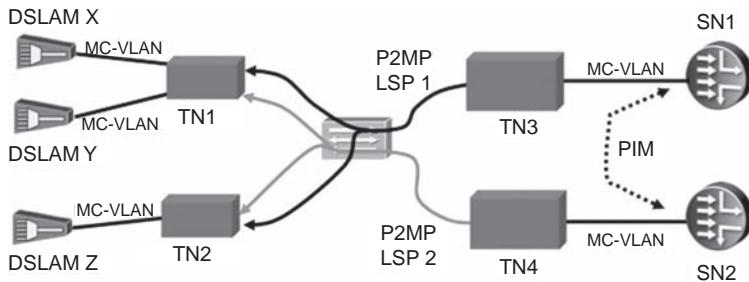


Figure A.5 Using VPLS to emulate the multicast VLAN

a PIM adjacency on the multicast VLAN, PIM procedures determining which of the two should forward multicast traffic onto the multicast VLAN.

In order to emulate the multipoint operation of the multicast VLAN when MPLS is being used on the access network, VPLS provides a natural fit.¹ This is illustrated in Figure A.5. From the point of view of the VPLS, the PEs are the TN1, TN2, TN3 and TN4. The CEs are the SNs and the DSLAMs. As described in Chapter 13, the most efficient way to carry multicast traffic in VPLS is to use P2MP LSPs. In this scenario, TN3 and TN4 in Figure A.5 is each the root of a P2MP LSP. Each autodiscovers the other PEs in the network and so knows which other PEs (i.e. TN1 and TN2) should be the leaves of the P2MP LSP. Note that TN1 and TN2 do not need to be roots of P2MP LSPs because multicast data traffic does not travel upstream, only control traffic.

An additional optimization is for TN1 and TN2 to perform IGMP snooping, so that they only send traffic downstream to a particular DSLAM if there is an interested subscriber attached to that DSLAM. For example, in the figure if TN1 knows through IGMP snooping that a particular multicast group is only required by DSLAM X, it does not send it to DSLAM Y, thus saving bandwidth on the link to DSLAM Y.

For many deployments, this level of optimization is sufficient as typically each TN with DSLAMs attached needs to receive all of the multicast groups since there is almost certainly an interested subscriber somewhere downstream, due to the large number of subscribers downstream from each node. If this is not the case, an additional level of optimization is for each such TN to communicate the identities of the snooped groups using BGP, as described in Chapter 13, thus enabling the SN-facing TNs to build selective P2MP LSPs according to which DSLAM-facing TNs are interested in a particular multicast group.

When VPLS is being used as infrastructure in this way, it is important to prevent direct communication between subscribers to ensure security and

¹ Note that it is fine to use L2VPN for the unicast traffic while using VPLS for the multicast traffic.

quality of service. This can be achieved by disallowing packets received on one access circuit to be sent out on another access circuit on the same node – in effect local switching is prohibited. For example, TN1 should not allow traffic received on the multicast VLAN from DSLAM X to be sent out to DSLAM Y. Also packets should not be allowed to pass into one transport node facing a DSLAM, then to another transport node and out towards another DSLAM. This is easily achieved using a hub-and-spoke topology for the VPLS, through manipulation of BGP extended communities, by analogy with hub-and-spoke L3VPNs.

A.4.3 Multi-edge, VLAN per service model

In contrast with the single-edge model discussed in the previous section, in the multi-edge case, more than one type of SN is used, each delivering a subset of the services. Internet service is delivered using a Broadband Services Router (BSR). Video services, whether VoD or broadcast TV, are delivered by a VSR. In some deployments, VoIP is delivered by the BSR and in other deployments by another SN.

The multi-edge scheme goes hand-in-hand with the VLAN per service model. In this model, there is a separate VLAN for each service which carries traffic to all the subscribers. In contrast, the scheme in the previous section required a separate VLAN for each individual subscriber. The VLAN per service model is illustrated in Figure A.6. Note that only the VoD and the Internet VLAN are shown in the figure for clarity.

In addition, a multicast VLAN would be present, as shown in Figure A.3. In the case of the multi-edge scheme, the SNs shown in the figure are VSRs. Let us look at each service in turn and how it can be mapped to an MPLS-based access network.

A.4.3.1 *Internet VLAN*

There are two options for providing the Internet VLAN functionality when migrating to MPLS:

1. Remain with the model of having one Internet VLAN serving all of the DSLAMs. Because the VLAN has more than two nodes attached, a point-to-point pseudowire cannot be used but VPLS can be used instead.
2. Change to a scheme where a separate Internet VLAN is used to serve each DSLAM. In this case, for each Internet VLAN, a point-to-point pseudowire is provisioned. Option (b) has the advantage over (a) that no MAC learning is required on the MPLS nodes but has the disadvantage that extra VLAN configuration is required on the BSR and the TN that faces it, because one internet VLAN is required per DSLAM. As already

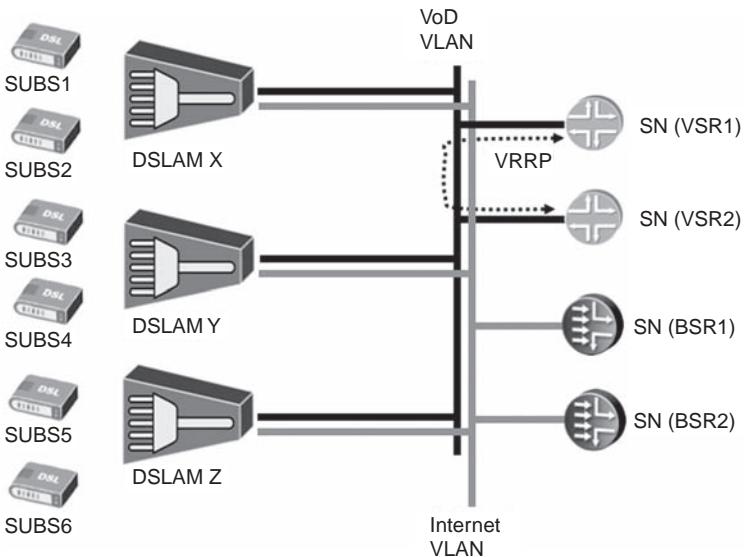


Figure A.6 Multi-edge VLAN per service model

described in Section A.3.1, if BSR redundancy is used for the Internet service, it is desirable to have an active-standby approach. This can be achieved for option (b) by using BGP L2VPN multi-homing and for option (a) by using BGP VPLS multi-homing.

A.4.3.2 VoD VLAN

This case is handled slightly differently to the Internet VLAN case. There is typically no session state maintained on the VSR, at most a customer address allocated by DHCP. This means that if VSR redundancy for the VoD VLAN is required, an active-active approach can be used. This is achieved by using VRRP, so in the upstream direction traffic passes through the VSR that has been elected the default gateway by the VRRP protocol. In the downstream direction, traffic can pass through both VSRs. Because the two VSRs and the DSLAMs need to be on the same LAN for VRRP to operate, VPLS is used to emulate the LAN when using an MPLS-based access network. From the point of view of the VPLS, the CEs are the DSLAMs and the VSRs, and the PEs are the TNs to which they are connected.

A.4.3.3 VoIP VLAN

In deployments where a separate SN is used for the VoIP service, the same scheme as for VoD can be used, that is to say the SNs are used in active-active mode with VRRP, and so VPLS is used to emulate the LAN.

In deployments where the SN for the VoIP service is the BSR, then the option chosen for the Internet VLAN is also used for the VoIP VLAN.

A.4.3.4 Multicast VLAN

The multicast VLAN has a similar functional model to the multicast VLAN in the 1:1 hybrid model already described in Section A.3.1. A single multicast VLAN carries multicast data in the downstream direction and serves all of the DSLAMs. Only control traffic such as IGMP requests travel upstream. Redundancy is provided by having a pair of VSRs running PIM in order to elect a forwarder for the multicast VLAN. Therefore, as in the 1:1 hybrid case, VPLS with P2MP LSPs can be used to emulate the multicast VLAN.

A.4.4 Summary of options for xDSL backhaul

In the previous sections, we have discussed two models for xDSL deployment, each having various options for some of the triple-play services being carried. Table A.1 summarizes the available options.

Table A.1 Deployment options for xDSL backhaul

xDSL architecture	Component	Associated MPLS infrastructure	Redundancy mechanism
Single edge, 1:1 hybrid	Customer VLAN	L2VPN	Active-standby, using L2VPN multi-homing
	Multicast VLAN	VPLS with P2MP LSPs	Active-active, PIM adjacency between SNs
Multi-edge, VLAN per service	Internet VLAN	L2VPN or VPLS	Active-standby, using L2VPN multi-homing or VPLS multi-homing
	VoD VLAN	VPLS	Active-active, using VRRP
	VoIP VLAN	Same scheme as Internet VLAN if SN is the BSR Same scheme as VoD VLAN if SN is VSR	
Multicast VLAN	VPLS with P2MP LSPs	Active-active, PIM adjacency between VSRs	

A.5 REFERENCES

- [802.1ag] *Connectivity Fault Management*, IEEE document 802.1ag,
<http://www.ieee802.org/1/pages/802.1ag.html>
- [BBARCH] C. Hellberg, D. Greene, T. Boyes, *Broadband Network Architectures*, Prentice Hall, 2007

Appendix B: MPLS Resources

MPLS is a rapidly developing field that has seen lots of activity in recent years. To help track these, we have compiled a list of standards bodies, conferences and venues for interoperability testing.

B.1 STANDARDS BODIES

1. IETF – Internet Engineering Task Force. Some of the working groups relevant to MPLS are mpls, ccamp, l3vpn, l2vpn, pwe3 and pce. Each working group has its own mailing list which anyone can join. See the working group homepages for details on how to join (<http://ietf.org/>). The IETF meets three times a year, with separate sessions for each working group. Note that the sessions are not tutorial in nature, rather they discuss the specifics of Internet drafts that are progressing through the working groups. Much of the technical work of the IETF is carried out through the mailing lists rather than at the actual meetings.
2. ITU. ITU is the United Nations agency for information and communication technologies. Standardization efforts are carried out through ITU-T recommendations (<http://www.itu.int/net/home/index.aspx>).
3. The Broadband Forum works on broadband wireline solutions. Some of these involve MPLS, for example the use of MPLS in the access network (<http://www.broadband-forum.org/>).
4. The IPsphere initiative within the TM Forum is a forum of service providers and equipment vendors working on a framework for the rapid creation and automated deployment of IP services, in particular

in the context of services spanning multiple providers. It defines architecture and implementation specifications, as well as being a venue for interoperability testing (<http://www.tmforum.org/ipsphere>).

5. Metro Ethernet Forum (MEF) is a forum of service providers, equipment vendors and testing organizations. It develops technical specifications and implementation agreements to promote the deployment of Carrier Ethernet worldwide (<http://metroethernetforum.org/>).

B.2 CONFERENCES

1. MPLS<year number> is an annual conference with focus on MPLS, held in Washington, DC every October. The name of the conference contains the current year (<http://mpls2010.com/>).
2. MPLS and Ethernet World Congress (formerly MPLS World Congress) is an annual conference with focus on MPLS, held in Paris, France, every February (<http://www.upperside.fr/mplsworld2011/mpls2011intro.htm>).
3. FutureNet (formerly MPLScon) is an annual conference with focus on MPLS, held in various locations in the US (<http://www.futurenetexpo.com/>).
4. Carrier Ethernet World Congress is an annual conference with focus on carrier Ethernet, held in various locations in Europe. MPLS topics are discussed in the context of carrier Ethernet (<http://www.iir-telecoms.com/event/cewc>).
5. IPoP – yearly conference with focus on IP and optical networks, held in Tokyo, Japan. MPLS topics are discussed in this context (<http://www.pilab.jp/ipop2010/index.html>).
6. Nanog – The North American Network Operators' Group holds meetings three times a year at various locations in the US. The conference focus is on IP deployments, MPLS topics are sometimes discussed. The website contains recorded tutorials on various MPLS topics, all conference proceedings are also available online (<http://nanog.org/>).

B.3 INTEROPERABILITY TEST LABS

1. Isocore Interworking Lab – <http://www.isocore.com/>.
2. EANTC AG (European Advanced Networking Test Center) – <http://www.eantc.de/en>.
3. IPoP – <http://www.pilab.jp/ipop2010/index.html>.
4. UNH (University of New Hampshire) Interoperability Lab – <http://www.iol.unh.edu/>.

Appendix C: Solutions to Selected Study Questions

CHAPTER 1

1. See Section 1.3.1.
2. The two schemes are Label-Inferred LSPs (L-LSPs) and EXP-inferred LSPs (E-LSPs). See Section 1.3.1.1 for more details.
3. See Section 1.3.2.1.
4. See Section 1.3.2.3.
5. Each LER builds one LSP to each of the other LERs. So each LER builds 99 LSPs. As there are 100 LERs, the total number of LSPs in network is 100×99 , or 9900.
6. See Section 1.3.3.
7. See Figure 1.11 and the associated text.

CHAPTER 2

1. The Tunnel Id field in the Session Object, uniquely identifying the LSP at the head end is encoded as a 16-bit field. Therefore, the head end can be ingress for up to 16k LSPs only.
2. Many times it is desirable to keep the path within one geographic location in the normal case, but it is acceptable to let it take an out-of-area path following a failure. In such cases, link coloring may place too strict a constraint on the computation. The problem is that if no path is available within the geographic area (e.g. following a failure), link colors will not allow a path that crosses the domain (see also the

discussion on fate sharing in Chapter 3). In contrast, IGP metrics could be set up such that the cross-geography links would be given very high metric and used as a last resort.

3. See RFC 5712. The two main ideas behind soft preemption are a) There should be minimum impact to customer traffic. Therefore, the LSPs being preempted should be given a chance to switch to an alternate path in a make-before-break fashion. b) Traffic will not be switched immediately onto a newly established LSP. This is because to actually use the LSP, routing must be involved. Therefore, the temporary link overbooking that is the result of soft preemption should not cause congestion.
4. When a new traffic stream is mapped onto the LSP, policing will react by limiting the amount of data sent, by dropping some of the traffic, potentially impacting existing streams. In contrast, admission control will prevent the new stream from being mapped to the LSP, thus protecting existing streams. Policing is always possible, while admission control may only be feasible in some cases (e.g. when using the LSP as a transport tunnel for a service, but not when using it in the IGP).
5. A fixed metric shields the IGP from seeing reoptimizations of the LSP path. On the other hand, this can lead to surprising behavior as shown in Section 2.5.
6. (a) Total number of LSPs is $10 \times 9 = 90$. Maximum number of LSPs that can traverse a node is 90 minus the LSPs that the router is head end or tail end for, 72. On average, this yields eight LSPs. (b) The London and DC routers are choke points in the network. All intercontinental LSPs must cross them. There are 4×5 LSPs in each direction, total $20 \times 2 = 40$. If any of the LSPs reoptimizes within the continent, make-before-break behavior will be applied and the number of LSPs at the choke points will increase even more. Although this increase is temporary, it must be accounted for, as it represents a worst-case scenario.

CHAPTER 3

1. (a) Because secondary paths are not reserving bandwidth, they do not use resources. However, because of the same reason, if traffic does need to be switched to the secondary paths, there are no guarantees in terms of delivery, not just for the traffic taking the secondary path but also for the traffic of other LSPs traversing the same links. A common solution to this problem is to rely on DiffServ to drop less important traffic. (b) Secondary LSPs may be preempted by new primary LSPs. This is not desirable behavior, because it disrupts existing flows in favor of setting

up new flows. In many cases, it is preferred to refuse service setup rather than drop an existing connection.

2. In the facility protection case, a bypass tunnel must end either at the node immediately downstream of the PLR (in the link protection case) or at the next next-hop node (in the node protection case), so the overall path from the PLR to the egress node of the main LSP may be quite long. In the 1:1 protection case, the detour can go directly to the egress node of the main LSP and so may be shorter in some topologies.
3. Imagine a topology where routers R1, R2, R3, R4 are daisy-chained and a high-metric link exists between R1 and R3. LSP1 is set up from R1 to R4 along the path R1–R2–R3–R4, and the protection path for link R2–R3 will be R2–R1–R3.
4. The assumption is that link coloring is used to mark the two types of links and that the LSPs are set up according to these constraints. What the operator requires in this case is to use the same link-coloring constraints for the bypass paths.
5. One option is to require manipulation of the metric assigned to the LSP, when the LSP is on the protection path for longer than time X. By assigning a less attractive metric, traffic will be forwarded over other LSPs, if they are available.
6. Having a presignaled secondary path results in less outage time following a failure of the primary path, as time is not wasted signaling for the secondary path to be set up. However, a presignaled secondary path results in extra control plane overhead in the nodes on the secondary path. Also if bandwidth is reserved on the secondary path, resources have been reserved which are only occasionally used.
7. In the path protection case, the ingress router is responsible for moving traffic to the protection path. The ingress router may be several hops away from the point of failure, so it may be some time before the ingress realizes that the failure has occurred (e.g. via an RSVP message). In the local protection case, the router immediately upstream from the PLR is responsible for moving the traffic on the protection path. In the case of link failure, it typically detects the failure rapidly (e.g. as a result of loss of light on the physical interface).

CHAPTER 4

1. Because the TE deployment uses the priorities 2 and 3, the combination of (CT0, 2) and (CT0, 3) must continue to exist at the same positions as before. Thus, TE2 and TE3 must be defined such that they correspond to these two combinations, respectively.

2. Given that only eight TE-classes are supported and the customer wants to use eight CTs, a single priority is available for setup and hold for all LSPs in the network. For this reason, the MAM bandwidth allocation model is better suited in this case, since it does not require the use of priorities to determine how to share resources.
3. The problem can be addressed by coloring links to noncompliant routers and avoid them in path computation. As routers are upgraded, the link coloring is updated as well. This approach can be used in cases where it is important to make the service available before the entire network has been upgraded.
4. CSPF computations rely on the available bandwidth. During CSPF, preemption priorities are taken into account to evaluate the possibility of placing an LSP over a link that already carries a different LSP. When different bandwidth models are used, the computation of the available bandwidth becomes increasingly complex if different models are used for different links.
5. (a) LSP-size overbooking could be used for the data LSPs by simply setting them up with three times less bandwidth than the bandwidth they would actually use. (b) Local overbooking multipliers can be set up for each link such that the bandwidth available to the CT corresponding to data traffic is tripled while the bandwidth available to the CT for voice traffic is left unchanged. The two approaches differ in the following aspects: (1) local overbooking multipliers allow overbooking of just some of the links in the network. (2) LSP-size overbooking relies on the correct configuration of the bandwidth on all routers which can be head ends for an LSP. In contrast, local overbooking multipliers affect the local configuration only and are thus more appropriate for a gradual deployment. (3) Autobandwidth cannot be used with LSP-size overbooking.
6. Consider a network where traffic is sent from the CE to the local PE and from there to two remote PEs over two LSPs, LSP1 and LSP2. If both these LSPs have the same first-hop, then per-interface accounting could not provide the granular billing required. By using per-LSP policing, the LSPs are treated like virtual interfaces in this case.

CHAPTER 5

1. As seen in Chapter 2, reoptimization is typically achieved in a make-before-break fashion by first setting up the optimized path, then switching to it and only afterwards tearing down the old path. Therefore, reoptimization typically doubles the amount of state maintained in the network. Let us examine the impact on the state created based on the signaling method. (a) For a contiguous LSP, the state is doubled in

all three areas. (b) When stitching is used, the amount of state is doubled in area 1 and at ABR2. (c) When nesting is used as shown in Figure 5.4, state is doubled in areas 1 and 2 and at ABR2 and ABR4.

2. In all cases, a bypass is set up for link ABR2-R1 following the path ABR2–ABR1–R1. However, this bypass is used differently depending on the LSP setup method used: (a) in the contiguous LSP setup method the end-to-end LSP is mapped into it, (b) in the stitching method the segment starting at ABR2 is mapped into it and (c) in the nesting method the FA LSP is mapped into it.
3. In the end-to-end case, the EXP bits must be changed at the border routers at each domain. This could be accomplished, for example, by setting up a firewall filter to perform this translation (assuming that the providers share this information). In the stitching case, the segment in each AS must be set up with the correct EXP bits. In the nesting case, several FA LSPs must be set up, one for each class of traffic, and admission control must be performed into the correct FA LSP.
4. (1) The number of border routers that must be specified as loose hops in the path is small (as they must be manually listed) and (2) to protect against a failure of a border node, a secondary path must be configured that uses a different border router as a loose hop.
5. Advantages: Global visibility for all LSPs in the network, the ability to run more sophisticated algorithms than CSPF and the ability to cross AS and area boundaries. Disadvantages: The PCE introduces an extra layer in the network, which must be maintained and debugged, the computation is only as accurate as the information maintained in the PCE for the bandwidth availability and the reservations and the routers must support the PCEP protocol.
6. The fundamental difference is that the PCE must service computation requests dynamically. Therefore, (1) it is more limited in the algorithms it can apply than an offline tool, (2) its view of the network must be continuously kept in sync with the state of the network to ensure correct calculations, (3) it only has knowledge about past and current reservations, not about future ones, and (4) its deployment relies on the routers implementing the PCEP protocol and being able to perform PCE discovery.
7. See Section 5.4. In addition, the constraints used in one domain (e.g. the CT used for voice traffic in a DSTE deployment) may not be the same as in the neighboring domain and a translation function may be required.

CHAPTER 6

1. See Section 6.2.

2. A shortest-path tree aims to have the shortest possible path from the root to each egress node. It is therefore useful in case where minimum delay is important. In contrast, a minimum cost tree aims to minimize the sum of the metrics of the links occupied by the tree in order to minimize the overall bandwidth consumption in the network.
3. See the Foundations chapter for the unicast case. See Section 6.3.2 for the multicast case.
4. In this case, the upstream router to X would need to send two copies of each packet to X. The upstream router would pop the label on one copy – this is the copy that exits the P2MP LSP at X. The other copy would have an MPLS label and would be the copy that transits X on its way towards other egress router(s) located downstream from X. The way to avoid this situation is not to use PHP in this scenario.
5. In the case of P2MP LSP hierarchy, PHP must not be used on the outer P2MP LSP, as each egress router of the P2MP LSP must perform context-specific lookup of the inner label pertaining to the inner P2MP LSP. Similarly, in the case where multiple VPNs share the same P2MP LSP, each egress router must perform context-specific lookup of the inner (VPN) label, so PHP must not be used.
6. In the unicast case, an LDP speaker advertises a FEC to all of its neighbors. In the multicast case, an LDP speaker advertises the FEC only to the neighbor on the shortest path to the root of the P2MP LSP.

CHAPTER 7

1. A customer would opt for the overlay model if he prefers to run his own network, either for security reasons or because he is himself a service provider. In Chapter 11, we will discuss the use of pseudowires for providing the inter-CE connectivity to such a customer.
2. 100000 sessions total and on every PE (1000×100). This approach would prevent the growth of the VPN service because the BGP scaling properties of the PE would become the scaling bottleneck.
3. This approach could work for any-to-any connectivity but would not be able to provide the complex VPN access topologies enabled through RTs, such as hub-and-spoke or overlapping VPNs.
4. The RD plays no role in providing the access control in this scenario. Therefore, the RD is allocated using any of the schemes of Section 7.5.3. The RT is what constrains the distribution of routing information. For VPNA, all VRFs have import and export RT RTA and similarly in VPNB all VRFs have import and export RT RTB. In addition, on the VRF of site 1 of VPN B, there is an import RT for RTC. The route 10.1.1.1 is tagged with two RTs, RTA and RTC, through an export policy configured in

the VRF for site 1 of VPNA. Thus, it will get imported into all VRFs for VPNA and in the VRF for site 1 of VPN B. Note that using the additional import of RTA in this VRF would have given access to customers in site 1 of VPNB to all routes of VPNA, which was not the intention. So far, we have reachability from site 1 of VPN B to the server. To allow for traffic to flow in the opposite direction (from the server back to site 1 in VPN B), a similar policy must be configured for all the routes in site 1 of VPN B.

5. The VPN route becomes unresolved. The corresponding forwarding state is removed. If the prefix was redistributed into the PE–CE routing protocol, it must be withdrawn.

CHAPTER 8

1. Compared with OSPF or RIP, BGP is easier to troubleshoot and has incremental (rather than periodic) updates. Other useful BGP features are the availability of import and export policies, the ability to tag routes with additional communities as part of the policy processing. The customer may tag all voice destinations with a community that is agreed upon between the customer and the provider. The provider will translate this to the well-known community indicating ‘voice-destinations’ in its core. Route resolution for VPN routes marked this way can happen over a restricted set of LSPs, such that the latency properties are met.
2. See discussion on self-healing in the MPLS Management chapter (Chapter 15).
3. The next-hop solution requires (1) adding more loopbacks into the IGP and into LDP (if LDP is used) and (2) consistent addressing throughout the network. In some implementation, the BGP next-hop must be modified through policy, making the deployment more cumbersome.
4. See Section 8.4.
5. Once routes are partitioned between the two RRs, it is difficult to change the partition (since it requires changing the BGP sessions on the PEs). What may happen is that the set of routes on one RR may grow much faster than on the other.
6. The scenario assumes an average of 100 updates sent to an average of 24 PEs every minute (when RT filtering is enabled). This assumes the CPU on the PEs is not highly loaded, and the same rate of processing is available on all PEs. This latter assumption may not always hold true, especially in real-world scenarios, where equipment from different generations is used in the same network. BGP update generation is often optimized by calculating the update once and replicating it to

all clients. The efficiency of this strategy depends on how similar the clients are in terms of how they flow-control the sender. In the worst case scenario, if all receivers have very different rates at which they process updates, the efficiency can drop down to computing updates on a per-peer basis. The particular strategy used by the sender to deal with clients operating at different rates is very implementation dependant, but the basic fact that the load increases is true across the board.

7. For example, in a network where all PEs have sites from all VPNs and a single reflector is used, route target filtering does not provide a benefit. On the other hand, in networks which expect large growth (and as a consequence deployment of several reflectors) and where PEs service only subsets of VPNs, setting up route target filtering at an early stage can provide a significant advantage.
8. For example, transport LSPs can be set up for voice traffic such that they establish only over low-latency links. Because the same low-latency links may be used for other transport LSPs as well, the voice traffic is additionally marked such that it receives preferential diffserv treatment at the time it is forwarded onto the transport LSP.
9. See the introductory paragraph to Section 8.9.
10. At minimum, the VPN-IPv6 address family needs to be invoked on the existing BGP sessions and an IPv6 protocol and IPv6 addressing invoked on the PE-CE links. The service provider may choose to have separate BGP sessions, potentially involving different route reflectors for the VPN-IPv4 and VPN-IPv6 routes, so extra BGP sessions would be required accordingly.

CHAPTER 9

1. The VRF at the PE does not contain entries for the external (Internet) prefixes and therefore cannot forward traffic for such destinations. The inter-CE tunnel allows this traffic to be forwarded across the service provider network without an IP destination lookup at the PE.
2. Using the notation of Figure 9.5, the segments are PE4-CE2 (set up with either BGP or LDP), CE2-PE2 (set up with BGP), PE2-PE1 (set up with BGP), PE1-CE1 (set up with BGP) and CE1-PE3 (set up with either BGP or LDP). BGP refers here to labeled BGP.
3. The VPN label for the VPN routes exchanged between the PEs in the different ASs (the label assigned by the PEs) (bottom label), (2) the BGP label assigned by the ASBR (middle label) and (3) the label for the transport LSP (LDP or RSVP) within the AS (top label).

4. An inter-AS LSP means that the labeled BGP sessions at the ASBRs are not required. On the other hand, CSPF cannot be performed across multiple domains. Therefore, setting up inter-AS LSPs with RSVP requires knowledge of the hops, for example through the use of a path computation element with cross-AS visibility, as described in Chapter 5.
5. By using options A or B. The ASBR must have knowledge of the routes in order to enable such a mapping. For further discussion on how to map the voice and data traffic to different LSPs, refer to Section 8.3 (differentiated VPN treatment in the core).

CHAPTER 10

1. The following options exist for the trees in the provider network:
(1) separate trees per VPN (PIM-SM or bidirectional PIM), (2) separate trees per VPN per PE (PIM-SSM), (3) separate trees per VPN (PIM-SM or bidirectional PIM), plus distinct trees for an (S, G) or subset of (S, G) of a given VPN (PIM-SSM), (4) separate trees per VPN per PE (PIM-SSM), plus distinct trees for an (S, G) or subset of (S, G) of a given VPN (PIM-SSM). State in the provider network
(1) $\text{num_trees} = \text{num_VPNs}$, (2) $\text{num_trees} = \text{num_VPNs} \times \text{num_PEs}$,
(3) $\text{num_trees} = \text{num_VPNs} + \text{num_of_data_MDT}$, (4) $\text{num_trees} = \text{num_VPNs} \times \text{num_PEs} + \text{num_of_data_MDT}$. For LSPs, (a) when LDP is used, $O(\text{num_PE})$, b) when RSVP is used, $O(\text{num_PE}^2)$.
2. There are two main issues when discussing the bandwidth consumption in the context of multicast: (1) the requirement that a packet cross any link from source to receiver at most once and (2) the need to deliver traffic along an ‘arbitrarily optimal’ path (e.g. a min-cost tree vs. a shortest path tree). These two goals are far easier to accomplish using traffic-engineered LSPs rather than using tunnels set up by PIM.
3. For LDP, one of the main considerations can be an existing unicast VPN deployment using LDP, for more details on P2MP RSVP advantages, see Chapter 6. For PIM, some of the considerations can be existing multicast deployment using PIM, lack of MPLS support in the provider network, familiarity of the operations staff with PIM but not MPLS.
4. The assumption is that the tunnel can uniquely identify the ingress. Thus, MP2MP tunnels are precluded, as are hybrid unicast-multicast solutions where the shared tree is rooted in the middle of the network and traffic is unicast from the PE to the root of the shared tree.
5. The prune semantics can be implemented as a withdrawal of a C-multicast join route, so there is no need for a separate type of route in BGP.

6. The C-multicast Import RT is used to import C-multicast routes only in the VRF on the PE which is connected to the source. This is because the C-multicast route carries join information which should not be propagated in customer sites that do not contain the source. The C-multicast Import RT is locally generated and therefore must be advertised to the remote PEs. Rather than having a separate advertisement, the information is piggybacked onto the unicast VPN route for the source in the VRF Route Import extended community.
7. Based on the autodiscovery route from PE1, PE2 and PE3 would have learned the identity of the tree (included in the PMI tunnel attribute) and its root (which is the originator of the autodiscovery route, PE1) and would have joined the tree by sending LDP label map messages with the corresponding P2MP LDP FEC.
8. Aggregation is supported by using the same tunnel for multiple VPNs. To distinguish the traffic arriving at a remote PE, a separate MPLS label is used. This label is an upstream-allocated label which is disambiguated in the context of the tunnel. See Section 6.9.2 for more details.
9. The following will happen: PE3 receives a PIM Register (S_2, G). As a result, PE3 originates a Source Active autodiscovery route and propagates it to PE1 and PE2. PE2 had previously received a previous Join $(*, G)$ from CE2. When the Source Active autodiscovery route arrives at PE2, PE2 generates a C-multicast route for (S_2, G) .

CHAPTER 11

2. See Section 11.6.
3. If a route-reflector receives a C-multicast route from multiple PEs for the same multicast flow, it only needs to reflect one instance of the C-multicast route.
4. One application is for distributing information feeds (for example, real-time pricing information) from an information provider to subscribers to the service. Another application is for IPTV wholesale.
5. Because: a) it is a proprietary protocol and b) the information sent by the protocol can be easily carried in an mVPN through support of PIM DM.
6. The Live-Live scheme typically has lower interruption times than the Live-Standby scheme following a failure in the network. On the other hand, the Live-Live scheme uses more network bandwidth than the Live-Standby scheme because two copies of the data are sent through the network.

CHAPTER 12

1. If non-IP traffic needs to be carried, then L2VPN is appropriate but L3VPN is not. If the enterprise wishes to have control over its own routing, then L2VPN is a better choice than L3VPN.
2. This has the benefit of saving bandwidth. ATM cells are relatively small (52 bytes without the HEC byte) so having only one cell per packet would incur a large encapsulation overhead, once the MPLS encapsulation and the link-layer encapsulation are taken into account.
3. The use of autodiscovery avoids the need to explicitly configure on each PE the identity of the remote PE involved with a particular pseudowire. This avoids a N^2 provisioning overhead when deploying a fully meshed L2VPN. Also, when adding a new CE to an existing L2VPN, only the PE attached to that CE needs new configuration, assuming that corresponding ACs have been preprovisioned on the other PEs.
4. Both schemes use route distinguishers to disambiguate reachability information belonging to different VPNs. Both schemes use route targets to constrain route distribution. Both schemes have similar options for interprovider connectivity. In both schemes, route reflectors can be used to avoid having a full mesh of signaling sessions between the PE routers in the network.
5. See Section 12.7.

CHAPTER 13

1. Point-to-multipoint LSPs can be used, rather than ingress replication, to avoid sending multiple copies of same data frame on a link. IGMP or PIM snooping can be used to determine which PEs have interested receivers for a multicast group, thus avoiding sending the traffic to PEs that do not have interested receivers in their attached sites.
2. MAC of CE6 -> local interface PE3-CE6
MAC of CE8 -> pseudowire to PE4
MAC of CE9 -> pseudowire to PE4
MAC of F -> pseudowire to PE1.
3. MAC J1 -> pseudowire to PE1
MAC J2 -> pseudowire to PE2
MAC J3 -> interface if1
MAC J4 -> pseudowire to B
MAC J5 -> pseudowire to B
MAC J6 -> pseudowire to D

MAC J7 -> pseudowire to PE4
 MAC J8 -> pseudowire to PE3.

4. As in the L2VPN case described in the previous chapter, the use of autodiscovery avoids the need to explicitly configure on each PE the identity of the remote PE involved with a particular pseudowire. This avoids a N-squared provisioning overhead when deploying a fully meshed L2VPN. Also, when adding a new CE to an existing L2VPN, only the PE attached to that CE needs new configuration, assuming that corresponding attachment circuits have been preprovisioned on the other PEs. An additional advantage in the VPLS case is that autodiscovery of the PEs to which P2MP trees are required to be built saves having to configure the P2MP trees manually.
5. The label values are as follows:
 PE with VE ID 1: 200 000
 PE with VE ID 2: 200 001
 PE with VE ID 4: 200 003
 PE with VE ID 5: 200 004
 PE with VE ID 6: 200 005.

CHAPTER 14

1. As good as convergence speeds may be, any control-plane solution cannot guarantee deterministic recovery times.
2. NSR does not provide protection for the case of a catastrophic failure of the hardware at the PE (for example, power failure), nor does it solve the problem of PE-CE link failure.
3. The VPN label provides a way to identify traffic arriving from the core as belonging to a particular VPN. The context label tags traffic in a similar way, allowing the lookup to happen in a particular table.
4. The context id, pseudowire id and neighbor address are required in addition to the pseudowire label. These allow identification of the correct attachment circuit and insertion in the right context table.
5. Because any traffic arriving on this LSP will be forwarded in the context of the mirrored forwarding state of the primary.

CHAPTER 15

1. (1) A packet exiting an LSP at the wrong endpoint will never be forwarded as IP, avoiding the situation described in Section 15.3.2.1.
 (2) Because the address need not be configured on the router, various destination addresses can be picked such that different paths in the core

are exercised, as explained in Section 15.3.3.2. (3) The endpoint of an LSP is not always known (e.g. for LDP-signaled LSPs), so using one of the router addresses would not always be possible, even if it was desired.

2. The main difference is in the number of replies that are bound to arrive in response to a single echo request. Most of the extensions deal with minimizing the impact of a large number of replies. In addition to these, new TLVs identifying the P2MP LSPs must be supported.
3. (1) The set of endpoints which the router is expecting replies from (note that this information is not always provided by the user during echo request), (2) the endpoints which failed to reply, (3) the endpoints which replied and the times when they received the request and (4) the status of the individual replies.
4. If the network is undergoing churn, caused for example by a flap of an interface or a routing peering, aggressively implementing revertive behavior can increase the churn. For example, if an LDP LSP is coming and going periodically because of an interface flap, then adding it periodically to the usable pool will trigger a lot of re-resolution of routes. Instead, the LSP must be up and stable for a while before acting on a change. This is in line with the principle ‘react quickly to bad news, react slowly to good news’ implemented elsewhere in network protocols.
5. IP traffic will not be affected. However, alternate paths will not be used, even if they are available.
6. For the discovery of all paths, LSPtrace is employed. Once the paths are identified, LSPing is used to bootstrap the BFD sessions. LSPtrace must continue to run periodically to ensure new paths are discovered (if they become available).

CHAPTER 16

1. (i) MAC learning scheme requires flooding of traffic on all ports, causing wastage of bandwidth. (ii) Spanning Tree Protocol results in certain ports not being used to avoid loops, which wastes bandwidth. (iii) Potential for persistent forwarding loops to form. (iv) No traffic engineering capability. (v) Relatively slow recovery from link failure compared to MPLS fast reroute. (vi) In xDSL backhaul schemes, potentially a large number of MAC addresses need to be learnt. (vii) Multiple customers share the same LAN, resulting in need for MAC security measures.
2. Option 1 is closer to the operating model used in most existing networks and so is likely to be more closely aligned with organizational structure, OSSs and so on. Option 2 requires less per-customer configuration effort

because only the SN itself requires customer-specific configuration. In contrast, Option 1 also requires customer-specific configuration on the TN attached to the SN as often a VLAN per customer forms the hand-off between the two. Option 2 gives more flexibility with respect to the placement of SNs within the network.

3. See Section 16.4.2.
4. See Section 16.4.4.
5. See Section 16.4.2.

CHAPTER 17

1. Because of the large existing deployed base of MPLS, it is important that existing deployments continue to work and interoperate.
2. Loss and delay measurements, alarm suppression, remote defect indication.
3. LDP signaling, MP2P and MP2MP LSPs, ECMP, PHP.
4. (1) Because certain MPLS-TP deployment might not use IP addressing, LSPing must be enhanced to run over a pure MPLS network using the G-ACH construct. (2) If IP is not used, the source address cannot be gleaned from the LSPing packet and must be specifically carried in a separate TLV. See more details on further enhancements in draft-ietf-mpls-tp-lsp-ping-bfd-procedures-00.txt.
5. The Generic Alert Label (GAL) is a fundamental construct in MPLS-TP but its use is limited to OAM functions. A router receiving a packet tagged with the special GAL label knows that it must process it using the appropriate OAM function encoded by the GACH header.

Appendix D: Acronyms

2G	Second-generation mobile network
3G	Third-generation mobile network
6PE	A scheme to allow IPv6 traffic to be carried over an IPv4 MPLS network
AAL	ATM Adaptation Layer
ABR	Area Border Router. A router used to connect two OSPF areas
AC	Attachment Circuit. In the context of Layer 2 VPNs, the physical or logical circuit used to connect a CE to a PE
ACH	Associated Channel. In the context of VCCV, the ACH is the header that is added to the packet to trigger its processing by the control plane
AF	Assured Forwarding DiffServ class
AFI	Address Family Identifier. In BGP, the identity of the network layer protocol associated with the network layer reachability information being advertised
AIS	Alarm Indication Signal. In SONET/SDH networks and ATM networks, a means of signaling in the downstream direction the existence of a fault
AN	Access Node. Used especially in the context of Seamless MPLS, an AN is a generic term for a customer-facing device at the outer edge of the network, for example a DSLAM
APS	Automatic Protection Switching. A method for providing protection at the SONET/SDH layer by moving the traffic to a standby link
ARP	Address Resolution Protocol

AS	Autonomous System. A collection of routers belonging to the same administrative entity and having a common external routing policy
ASBR	Autonomous System Border Router. A router used to connect two ASs
ATM	Asynchronous Transfer Mode
ATM PVC	ATM Permanent Virtual Channel
BC	Bandwidth Constraint. In DiffServ Aware Traffic Engineering, BCs determine the bandwidth availability on a link for a Class Type or group of Class Types
BE	Best Effort DiffServ class
BECN	Backward Explicit Congestion Notification. In Frame Relay networks, a message sent towards the transmission source indicating the existence of congestion in the network
BFD	Bidirectional Forwarding Detection. A protocol to detect faults in the bidirectional path between two forwarding engines
BGP	Border Gateway Protocol. An interautonomous system routing protocol. The current version of BGP is BGP-4, described in RFC 4271
BGP-MP	BGP Multiprotocol extensions, documented in RFC 4760
BN	Border Node. Used especially in the context of Seamless MPLS, a BN resides at the boundary between different regions of the network, for example an ABR
BSC	Base Station Controller, used in 2G mobile networks
BSR	Broadband Services Router
BTS	Base Transceiver Station, used in 2G mobile networks
CAC	Call Admission Control
CAPEX	CAPital EXpenditure
CBR	Constant Bit Rate. An ATM service category having a constant maximum bandwidth allocation. Often used for real-time applications
CC	Connectivity Check
CCC	Circuit Cross Connect. A scheme for the transport of Layer 2 frames over an MPLS network
CE	Customer Edge (usually designates equipment at the edge of the customer's network)
CIR	Committed Information Rate. In Frame Relay networks, the bandwidth associated with a logical connection
CLI	Command Line Interface
CLNS	ConnectionLess Network Service. A service defined by the Open Systems Interconnect (OSI) that does not

	require the existence of a connection in order to send data
CLP	Cell Loss Priority. A bit in the ATM cell header that indicates whether the cell is a candidate for being dropped in the presence of congestion
CoC	Carrier of Carriers. In the context of BGP/MPLS L3VPN, a carrier providing VPN transit to a customer who is himself a carrier
CoS	Class of Service
CPE	Customer Premise Equipment
CPU	Central Processing Unit
CR-LDP	Constrained-based Routing LDP
CsC	Carrier's Carrier – see CoC
CSPF	Constrained Shortest Path First. In traffic engineering, the algorithm used to compute the paths of MPLS LSPs
CSV	Circuit Status Vector. In BGP-signaled L2VPNs, a means for a PE to communicate to remote PEs the state of its connectivity
CT	Class Type. In Differentiated Services Aware Traffic Engineering, a set of classes that have a common aggregate bandwidth requirement of the network
CV	Connection Verification
DCN	Data Communications Network
DE	Discard Eligible. A bit in the Frame Relay header that indicates whether the cell is a candidate for being dropped in the presence of congestion
DiffServ	Differentiated Services
DiffServ-TE	Differentiated Services Aware Traffic Engineering
DLCI	Data Link Connection Identifier. In Frame Relay networks, the means by which a logical circuit is identified
DM	Delay Measurement
Dos	Denial of Service
DSCP	DiffServ Code Point. A 6-bit field in the IP packet header that determines the class-of-service treatment received by the packet
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
EBGP or eBGP	External Border Gateway Protocol
ECMP	Equal Cost Multi-Path
EF	Expedited Forwarding DiffServ class
EIGRP	Enhanced Interior Gateway Routing Protocol
E-LSP	EXP-inferred LSP (LSP for which the DiffServ behavior is inferred from the EXP bits in the MPLS header)

ERO	Explicit Route Object (used in RSVP-TE to encode path information)
Ethernet OAM	An Operations and Maintenance scheme for Ethernet
EXP	Experimental bits in the MPLS header
FA	Forwarding Adjacency
FA LSP	Forwarding Adjacency LSP, used in LSP hierarchy as a container for other LSPs
FCS	Frame Check Sequence. A set of bits added to a frame in order to detect errors in the frame
FEC	Forwarding Equivalence Class. Packets that are to be forwarded to the same egress point in the network along the same path and with the same forwarding treatment along that path are said to belong to the same FEC
FEC 128	The FEC originally used for signaling pseudowires with LDP. This has now been superseded by FEC129
FEC 129	The FEC now specified for signaling pseudowires with LDP. It supersedes FEC 128
FECN	Forward Explicit Congestion Notification. In Frame Relay networks, a message sent towards the receiver indicating the existence of congestion in the network
FR	Frame Relay
FRR	Fast ReRoute. The process of quickly routing traffic around the point of failure
FTP	File Transfer Protocol
G-ACH	Generic Associated Channel. In the context of MPLS-TP, a header applied to OAM packets identifying the type of OAM processing that must be applied to it
GAL	Generic Alert Label. Label 13 used in the context of MPLS-TP to trigger processing of the packet by an OAM function
GFP	Generic Framing Procedure. A mechanism to encapsulate packets into SONET/SDH frames
GRE	Generic Routing Encapsulation. A protocol for encapsulation of an arbitrary network layer protocol over another arbitrary network layer protocol
HDLC	High-level Data Link Control
H-VPLS	Hierarchical VPLS
IBGP or iBGP	Internal Border Gateway Protocol
ICMP	Internet Control Message Protocol
IETF	Internet Engineering Task Force: www.ietf.org
IGMP	Internet Group Management Protocol. A protocol to enable the host to join or leave a multicast group. Described in RFC 3376

IGMP snooping	A scheme by which routers can inspect the contents of IGMP packets in order to determine the location of receivers and optimize traffic forwarding to only those destinations
IGP	Interior Gateway Protocol
IP	Internet Protocol
IPsec	IP security
IPX	Internetwork Packet eXchange. The network layer protocol in the NetWare operating system
IS-IS	Intermediate System-to-Intermediate System. A link-state IGP described in RFC 1195
ISO	International Organization for Standardization
ISP	Internet Service Provider
ITU-T	International Telecommunications Union – Telecommunications
LAN	Local Area Network
LDP	Label Distribution Protocol. LDP is documented in RFC 5036
LER	Label Edge Router
L-LSP	Label-inferred LSP. An LSP for which the DiffServ behavior is inferred from the label in the MPLS header
LM	Loss Measurement
LMI	Local Management Interface. A set of enhancements to the basic Frame Relay specification
LOM	Local Overbooking Multiplier. In the context of DiffServ-TE, it is a factor by which the bandwidth for one particular CT is overbooked
LSA	Link State Advertisement. The advertisement sent by a link-state IGP such as OSPF or IS-IS, containing information about the state of the links
LSP	Label Switched Path
LSPing	LSP ping. A mechanism for detecting MPLS data plane failures, based on similar concepts as ping
LSPtrace	A mechanism based on LSPing, which provides failure localization through a traceroute-like capability
LSR	Label-Switching Router. A router that can forward packets based on the value of a label attached to the packet
MAC address	Media Access Control address. A unique 48-bit identifier that represents the physical address of a device
MAM	Maximum Allocation Model. A bandwidth constraint model for DiffServ-TE. The model enforces strict

	separation between the bandwidth allocated to the different CTs
Mbps	Mega bits per second
MD5	Message digest 5. The MD5 algorithm is documented in RFC 1321. Its purpose is to take as input a message of arbitrary length and produce as output a 128-bit fingerprint (signature)
MDT	Multicast Distribution Tree. In the context of VPN multicast, these are the multicast trees in the provider network that provide connectivity to all the PE servicing sites of a multicast-enabled VPN. Conceptually, the MDT creates the abstraction of a LAN to which all the PEs belonging to a particular VPN are attached. This property is very important for the C-instance PIM sessions between the PEs, which can consider each other as directly connected neighbors over this LAN
MEP	Maintenance End Point. In the context of OAM, MEPs are the endpoints of a maintenance entity and are responsible for activating and controlling the OAM functionality
MIB	Management Information Base. A formal description of a set of objects that can be managed using SNMP
MIP	Maintenance Intermediate Point. In the context of OAM, a MIP is a point between two MEPs and is capable of responding to some OAM packets
MP	Merge Point. In the context of MPLS FRR, it is the tail end of the backup tunnel and the point where traffic from the backup merges back into the protected LSP
MP2MP	MultiPoint to MultiPoint. An LSP is MP2MP if it has multiple ingress and egress points
MP2P	MultiPoint to Point. An LSP is MP2P if it has multiple ingress points and one egress point
MP-BGP	BGP with multi-protocol extensions, as described in RFC 2858, that allow BGP to carry routing information for multiple network layer protocols
MPLS	MultiProtocol Label Switching. A set of IETF standards to allow traffic to be forwarded based on labels rather than destination addresses
MPLS-TE	MPLS Traffic Engineering. The traffic engineering capabilities of MPLS, implemented through a combination of source-based routing and constrained-based routing
MPLS-TP	MPLS Transport Profile. The transport profile is designed for use as a network layer technology

	in transport networks and is currently under standardization in the IETF
MSAN	Multi-Service Access Node
MSDP	Multicast Source Discovery Protocol, described in RFC 3618
MTU	Maximum Transmission Unit. The largest physical packet size (measured in octets) that can be sent in a packet or frame-based network
mVPN	Multicast VPN. A VPN that carries multicast traffic
NAT	Network Address Translation
NG mVPN	Next-generation mVPN. The BGP/MPLS-based scheme for supporting multicast in a VPN. Also referred to as next generation to differentiate it from the PIM/GRE-based scheme, which was originally proposed for solving this problem
NLRI	Network Layer Reachability Information. In BGP terminology, route prefix is referred to as NLRI. Different AFI/SAFI pairs are considered to be different NLRI types
NMS	Network Management System
Node B	Base station for 3G mobile networks
NSAP	Network Service Access Point. Type of addressing used by ISO network layer protocols
NSR	Non Stop Routing. A functionality allowing the router to continue to operate after a failure of the control plane. Care must be taken with this term, as different equipment vendors use the term in different ways, some meaning that the forwarding plane will continue its operation based on stale control plane information, others meaning that both control and forwarding planes can seamlessly continue after a failure
OAM	Operations And Management, or Operations, Administration and Management. A set of network management functions covering fault detection, performance data and diagnosis capabilities
OPEX	OPerational EXPenditure
ORF	Outbound Route Filtering. A method for minimizing the number of BGP advertisements between two peers. The main difference between ORF and RTF is in the scope of the filtering: ORF operates between two peers while RTF can propagate filtering information across multiple hops
OSPF	Open Shortest Path First link-state IGP. OSPFv2 (version 2) is documented in RFC 2328

P device	Provider device. Designates a router in the core of a provider's network
P2P	Point to Point. An LSP is P2P if it has exactly one ingress and one egress point
P2MP	Point to MultiPoint. An LSP is P2MP if it has one ingress and multiple egress points
PABX	Private Automatic Branch eXchange. A telephone switch used inside a corporation. It connects internal extensions with each other and provides access (by dialing an access number) to the public telephone network
PBX	Private Branch eXchange. Same as PABX
PCC	Path Computation Client. A client of a PCE. The PCC may be either a router or another PCE
PCE	Path Computation Element. A network element that can compute TE LSPs for which it is not the head end. For example, an ABR or ASBR can play the role of a PCE. The PCE may also be an independent device in the network
PCEP	PCE Protocol. Protocol for PCE-PCC and PCE-PCE communication, developed in the PCE working group in the IETF
PDH	Plesiochronous Digital Hierarchy
PDU	Protocol Data Unit
PE device	Provider Edge Device. Designates equipment at the edge of the provider's network, providing aggregation of the different CE devices
PHB	Per-Hop Behavior. In the context of DiffServ, defines the packet scheduling, queuing, policing or shaping behavior on a particular node
PHP	Penultimate Hop Popping. The act of removing the MPLS label one hop before the LSP egress
PIM	Protocol Independent Multicast. Defined in RFC 2362, RFC 3973 and in several documents in the pim Working Group in the IETF
PIM-SM	PIM Sparse Mode, documented in RFC 4601
PIM-SM ASM	See PIM-SM. In the any-source multicast (ASM) mode of operation, PIM-SM provides a service model where there are multiple sources and multiple receivers for the same group
PIM-SM SSM	See PIM-SM. In the SSM mode of operation, PIM-SM provides a service model where there is a single multicast source and multiple receivers for each group

PLR	Point of Local Repair. In the context of MPLS FRR, it is the head end of the backup tunnel and the point at which traffic from the protected LSP is locally rerouted around the failed resource using the backup tunnel
PMSI	Provider Multicast Service Interface. An abstraction used in the mVPN architecture document in the IETF. A packet sent over a PMSI by a PE router servicing a certain mVPN will arrive to all or some of the other PEs in the mVPN, and the receiving PEs will know which mVPN the packet belongs to
PoP	Point of Presence. Physical location at which a carrier establishes itself for obtaining local access and transport
POTS	Plain Old Telephony Service
PPP	Point-to-Point Protocol
PP VPN	Provider-provisioned VPN. VPNs for which the service provider (SP) participates in the management and provisioning of the VPNs
PSTN	Public Switched Telephone Network
PVC	Permanent Virtual Channel
PWE	Pseudowire. A method of emulating a Layer 2 service (such as FR or ATM) over an MPLS backbone by encapsulating the Layer 2 information and then transmitting it over the MPLS backbone
QoS	Quality of Service. A measure of performance that reflects both the quality of the service and its availability
RAN	In the context of mobile networks, the Radio Access Network
RD	Route distinguisher. In the context of BGP/MPLS L3VPNs, an 8-byte string that is concatenated to the VPN-IP prefixes, for the purpose of making them unique before advertising them over the common provider core
RDI	Remote Defect Indication. Failure indication sent in the upstream direction
RDM	Russian Dolls Model. A bandwidth constraint model for DiffServ-TE. The model allows sharing of a bandwidth across different CTs
RFC	Request For Comments. A type of IETF document. An overview of the IETF process can be found in RFC 1718
RIPv2	Routing information protocol version 2, described in RFC 2453
RP	Rendezvous point. In the context of PIM-SM, a meeting point for multicast sources and receivers

RPT	RP tree. Tree rooted at the RP
RR	Route reflector. In the context of BGP, a route reflector acts as a focal point for iBGP sessions, eliminating the need for a full mesh of sessions. Instead of peering with each other in a full mesh, routers peer with just the reflector
RRO	Record Route Object. Object used in RSVP-TE to track the path along which traffic is forwarded
RSTP	Rapid Spanning Tree Protocol
RSVP	Resource reSerVation Protocol. The base specification of the protocol is in RFC 2205
RSVP-TE	RSVP with traffic engineering extensions. The RSVP extensions for setting up LSPs are defined in RFC 3209
RT	Route Target. In the context of BGP/MPLS L3VPN, the route target is an extended BGP community, which is attached to a VPN route. The RT is what accomplishes the constrained route distribution between PEs that ends up defining the connectivity available between the VPN sites
RTF	Route Target Filtering. A method for constraining VPN route distribution to only those PEs interested in the RT with which the route is tagged. The method relies on each PE advertising the RTs for which it is interested in receiving updates and can achieve significant savings in the number of advertisements sent and received
(S, G)	In the context of multicast, a combination of source S and group G
S-VLAN	Service VLAN
SAFI	Subsequent Address Family Identifier. In combination with an AFI it defines an NLRI type
SDH	Synchronous Digital Hierarchy
SDP	Service Delivery Point
SE	Shared Explicit. A reservation style used by RSVP that allows an LSP to share resources with itself
SLA	Service-Level Agreement
SN	Service Node. Used especially in the context of Seamless MPLS, an SN is the node at which the service is applied. For example, in the context of residential broadband, it is the Broadband Services Router
SNA	Systems Network Architecture. A set of network protocols originally designed to support main-frame computers
SNMP	Simple Network Management Protocol

SONET	Synchronous Optical NETwork
SP	Service Provider
SPF	Shortest Path First. The shortest path computation performed by the IGP
SPT	Shortest path tree
SRLG	Shared Risk Link Group. A group of links that is affected by the same single event
STP	Spanning Tree Protocol
T-MPLS	Transport MPLS is a transport network layer technology designed by the ITU-T for application in transport networks. It was later abandoned in favor of MPLS-TP
TCP	Transmission Control Protocol. Reliable transport protocol used in IP
TDM	Time Division Multiplexing
TE	Traffic Engineering. The ability to steer traffic on to desired paths in the network
TED	Traffic Engineering Database. Database created from the traffic engineering information distributed by the IGP
TE LSP segment	In the context of setting up an interdomain TE LSP using the stitching method, these are the smaller LSPs that get stitched together
TLV	Type-Length-Value. Type of encoding of information in protocol messages
TN	Transport Node. Used especially in the context of Seamless MPLS, a TN is a P-router residing entirely within the core region or an access region
ToS	Type of Service. A field in the IP header designed to carry information that would allow deployment of QoS
TTL	Time To Live
UDP	User Datagram Protocol. Unreliable transport protocol used in IP
UHP	Ultimate Hop Popping. The act of removing the MPLS label at the LSP egress
VC	Virtual Circuit
VCI	Virtual Channel Identifier
VCCV	Virtual Circuit Connection Verification. The connection verification protocol for pseudowires set up using LDP
VE ID	VPLS Edge Identifier. In BGP-signaled VPLS, a means of uniquely identifying a site within a VPLS
VLAN	Virtual LAN
VoD	Video on Demand
VoIP	Voice over IP
VP	Virtual Path
VPI	Virtual Path Identifier

VPLS	Virtual Private LAN Service. A scheme in which a service provider's customer site appear to be attached to the same LAN
VPN	Virtual Private Network. A private network realized over a shared infrastructure
VRF	VPN Routing and Forwarding. The per-VPN routing and forwarding tables that ensure isolation between different VPNs
VSR	Video Services Router
WAN	Wide Area Network
WiMAX	Worldwide Interoperability for Microwave Access
xDSL	DSL stands for Digital Subscriber Line. There are several variants of DSL, such as Asymmetric DSL (ADSL), High-speed DSL (HSDL) and Very high-speed DSL (VDSL). xDSL is used as a generic term to cover all the DSL variants

Index

- 6PE 32–34
- AC 346, 348, 355–6, 361
- ACH 459, 520
- Access networks and MPLS 547–59
- Administrative attributes 45–7
- Admission control
 - of client flows at edge of network during RSVP signaling 49, 66, 72, 168, 179
 - handling an admission control failure 50
 - hierarchy of admission control 537
 - for layer 2 circuits 131–2, 457
 - and reoptimization 48, 50, 154–5
- Advertise-LSP 52
- Alternates, *see* IP FRR
- APS, *see* SONET APS
- Auto-RP 317–18
- Autobandwidth 26, 58–9
- Autodiscovery 236, 293, 295–6
 - in Layer 2 VPNs 337, 344–5, 353
- Automesh 55–6, 152, 175, 469, 532
- Autoroute 52
- Backup
 - configuration 98
 - and DSTE 565
 - facility 83–5
 - one-to-one 75, 85–6
 - tunnel 81–9, 110, 156, 580, 583
- Bandwidth constraint 122–3
- Bandwidth constraint model 122–7
- BC, *see* Bandwidth constraint
- BFD 67, 69, 72, 101, 111, 183–4, 436
- BGP
 - carrying ATM addresses 377
 - as a CE-PE protocol 226–8
 - and L3VPN multicast 191–2, 284, 307, 402, 535
 - labeled unicast 260, 264
 - labeled VPN-IP 268–9
 - MP extensions 318
 - route-refresh capability 212
 - signaling for VPLS 392
 - in VPN setups 138, 233, 280, 446, 461
- BGP autodiscovery routes 295–6, 307, 312
- BGP source active autodiscovery routes 300–1
- BGP/MPLS mVPN 286–300, 311–38
- Broadcast TV and P2MP LSPs 169
- BSC 576
- BSR
 - PIM bootstrap 317–19
 - broadband services router 487–8, 554–6
- BTS 576
- Bypass 75–6, 78, 95
- C-VLAN 550, 552
- Capacity planning and traffic engineering 71

- Carriers' carrier
 VPN setup 256–7
- CCC, *see* Circuit Cross Connect
- Circuit Cross Connect
 and point-to-multipoint LSPs 159, 187–9
- Class-type 134
- CLNS 535, 539, 576
- C-multicast routes 287–8, 303, 307, 313, 318, 322, 327, 337
- C-multicast import RT 288–9, 309, 313–4, 570
- CoC, *see* Carrier's carrier
- Colors, *see* Administrative attributes
- Connectivity check 513, 517–8, 576
- Connectivity verification (CV) 517–8
- Constrained SPF 48
 CSPF tie-breaking rules 48
- Context Id 435, 437
- Context label 178, 435–6, 439
- Constraint
 for LSP computation 142, 157
 translation at domain boundaries 151
- Control Word 348–9, 350
- Crankback 147–9
- CsC, *see* Carriers' carrier
- CSPF, *see* Constrained SPF
- CT, *see* Class-type
- CT object 121
- DCN 539, 577
- Detour 77, 79, 86, 434–5, 439
- Differentiated Services, *see* DiffServ
- DiffServ
 in DSTE 6, 64, 110, 159, 343, 536
 MPLS support for 9
 using Diffserv in a mixed RSVP/LDP network 59
 using Diffserv in a VPN setup 138, 233, 280, 446
 using Diffserv when doing bandwidth protection 93–7
- Diffserv Aware TE 6, 64, 110, 159, 343, 536
- Downstream mapping TLV 463, 465–6
- DSLAM 482, 487, 550–1, 553
- DSTE, *see* Diffserv Aware TE
- ECMP, *see* Equal Cost Multi-Path
- Edge protection virtual circuit 435, 437
- E-LSP 10–1, 53, 129
- Enterprise networks and MPLS 483–6
- Equal Cost Multi-Path 20, 577
- ERO 23–4
- ERO expansion 145–6, 151, 162
- Ethernet OAM 362, 426, 428, 436, 446, 549
- EXP bits 7, 10
- FA LSP 30–1, 142–3, 146, 156–7
- Failure detection
 link failure 27, 40, 44, 62, 68, 75, 88–9, 105, 107, 169
 for non silent failures 445–6, 460
 for silent failures 446–8
- Fast-reroute
 interdomain 155–7
 in IP networks 10, 68, 74, 533
 in LDP networks 60
 in MPLS networks using RSVP 88–9
 object 5, 13, 27, 80, 106–7, 110–11
 for P2MP LSPs 172, 175–6, 178–9, 180, 182
- Fate sharing, *see* Shared risk link group
- FEC, *see* Forwarding Equivalence Class
- FEC 128 352, 358, 385
- FEC 129 358, 385–6
- Forwarding-adjacency 30, 53, 142, 578
- Forwarding Equivalence Class 6, 174, 449
- FRR, *see* Fast-reroute
- G-ACH 520–1, 522, 574, 578
- GAL 520–2
- Hierarchical VPLS (H-VPLS) 386–9
- Hose model 247
- Hub-and-spoke 212, 291, 342, 345
- Hub PE (in VPLS) 213, 387–8, 397

- ICMP tunneling 461–2
IGMP snooping 383, 400, 553, 579
Independent control, *see LDP*
Inter-AS auto discovery route 311–5
Interdomain pseudowires 345, 350, 352, 357–8
Interprovider
 L3VPN 266–72
 TE 139, 140–1
IP fast reroute
 loop-free alternates 108–9, 111
 u-turn alternates 109, 111
IPFRR, *see IP fast-reroute*, *see*
 Fast-reroute, in IP networks
IPv6VPN 32
ISP
 as VPN customer, *see Layer 3 Virtual Private Network*
L2VPN, *see Layer 2 VPN*
L3VPN, *see Layer 3 Virtual Private Network; VPN*
L3VPN multicast 191–2
L3VPN QoS 246
Label block
 in L2VPN 3, 205, 345, 351
 in VPLS 357, 359
Label Distribution Protocol 15
 comparison with RSVP 21, 25
 independent control 17–9, 244, 448, 470
 label distribution in hierarchical VPNs 60, 436
 label distribution modes 15, 17
 label retention modes 15
 ordered control 17
 and point-to-multipoint LSPs 187–8
 scaling to large networks 137–8
 synchronization with IGP 13–4, 446
 and upstream label allocation 177–8
 tunneling over RSVP 205, 219
 in VPLS 5, 8, 12, 175
Label Edge Router 6, 579
Label spoofing 262, 271
Label switched path, *see LSP*
Label Switching Router 6, 579
Layer 2 interworking 365
Layer 2 transport over MPLS 341
 of ATM 343, 349
 autodiscovery 353–4
 BGP signaling for 392, 402
 of Ethernet 350
 failure notification mechanisms 361–2
 of Frame Relay 113, 201, 341–2
 label block 355–6
 LDP signaling for 351–3
Layer 2 VPN 353–7
 comparison with Layer 3 VPN 344, 358
Layer 2.5 VPNs 365
Layer 3 Virtual Private Network 7, 57, 163, 175, 188, 191, 199–338
 comparison with Layer 2 VPN 344, 358
convergence time 243
external routes 256–7
hierarchical 255
interdomain LSP 138
internal routes 257
interprovider, *see Interprovider*, L3VPN
ISP as VPN customer 257–62
membership verification 472, 475
multi-AS 266–71
multicast 275–335
PE–CE routing 226–30
PE–PE LSP 244, 247
QoS 246, 271
route authentication 14, 158
scalability 98, 235–8
security 244–6
VPN provider as VPN customer 262–6
LDP, *see Label Distribution Protocol*
LDP-BGP VPLS interworking 406–13
LDP FRR, *see IPFRR*
LDP session protection 15, 106
Least-fill CSPF tie-breaking rule 48
LER, *see Label Edge Router*
Link colors, *see Administrative attributes*

- Live-live multicast delivery 183, 332–5
 Live-standby multicast delivery 244, 329–32, 451
 Liveness detection 244, 451
 L-LSP 10–1
 Local overbooking multiplier 128
LOM, *see* Local overbooking multiplier
 Loop-free alternate (LFA), *see* IPFRR
 LSP
 hierarchy 30, 54
 multiclass 133
 point-to-multipoint 167, 179
 primary 70, 72
 priorities 43–4
 secondary 72
 setup, *see* Path setup
 LSP contiguous 446
 LSPing
 basic operation 448–9, 451–2
 and BFD for MPLS LSPs 452–3
 use for self-healing networks 452–3
 extensions for point-to-multipoint
 LSPs 187–8
 echo request 451–2
 echo reply 451, 457
 use in VCCV 457–8
 LSP ping, *see* LSPing
 LSP nesting 142–3
 LSP priorities 43–4
 LSP stitching 141–2
 LSP traceroute 462–3
 LSPtrace, *see* LSP traceroute
 LSP usage
 by BGP 51, 57
 by IGP 19, 21–2
 using policy 15, 41
 LSR, *see* Label Switching Router
- MAC learning in VPLS 375–82, 396
 MAC limiting in VPLS 376–9, 390
 Make-before-break 50–1, 55, 59, 88, 100–1, 154, 172, 562
 MAM, *see* Maximum allocation model
 Management information base 474, 476
 Management VPN 215
 Maximum allocation model 122
- MDT
 Aggregate tree 403–4
 data MDT 283, 285, 293–5, 298
 group address 282, 285
 Inclusive tree 331–3, 404–5
 Selective tree 298, 320–1, 401, 404–5
 Merge point 74, 176, 580
 Mesh group 408–9, 415
MIB, *see* Management information base
 Microloops 108
 Minimum cost tree (Steiner tree) 170
 mLDP *see* “LDP and
 point-to-multipoint LSPs”
 Most-fill CSPF tie-breaking rule 48
 MP, *see* Merge point
 MPLS
 in the access network 479–501
 in the enterprise 255, 276, 342, 370, 542–5
 header stacking 8
 header structure 7
 original problem statement 5
 overview of mechanisms 6–7
 MPLS echo reply, *see* LSPing; LSP
 traceroute
 MPLS echo request, *see* LSP traceroute
 MPLS management 443–5
 MPLS OAM 447–8
 MSAN 482, 487
 MPLS-TP 509–27
 MPLS transport profile, *see* MPLS-TP
 MSTP 484
 Multicast 163–4
 in L3VPN, *see* L3VPN multicast
 Multicast distribution tree *see* MDT
 Multicast VLAN 552–3
 Multihoming (in VPLS) 394–6
 mVPN
 NG mVPN *see* BGP/MPLS mVPN
 draft-rosen *see* PIM/GRE mVPN
 comparison of BGP/MPLS and
 PIM/GRE mVPN 303–7
 and P2MP LSPs 171, 175, 295–9
 inter-AS operations 311–6
 inter-PE tunnels *see* multicast
 distribution tree
 see also L3VPN multicast

- Network convergence 533–6
model for 536
- Network layer transport service, *see* NLTS
- Node B 81, 89, 90, 99, 103, 542
- Node-ID 160
- NLTS 523–5
- One-hop LSP
and protection 105–6, 269
- Option A 266–7, 270–1, 392
- Option B 268–9, 393
- Option C 269–72
- Ordered control 17–8, 470
- ORF, *see* Outbound route filtering
- OSPF
as a CE–PE protocol 227–8
- Outbound route filtering 234
- Overbooking 127–9
- P2MP LSPs, *see* Point-to-Multipoint LSPs
- Path computation
for Diffserv-TE LSPs 118, 121
interdomain 47, 64, 137–8
offline 48
per-domain 142–3
for TE LSPs 139
- Path computation client 152, 582
- Path computation element 150, 152–3
- Path setup
interdomain 49
for traffic engineered LSPs 49, 140–1 95, 220
- PCC *see* path computation client
- PCE, *see* Path computation element
- PCEP 152–3
- PE
device scalability 235
- Penultimate Hop Popping 8, 84–5
- PHP, *see* Penultimate Hop Popping
- PIM
C-instance 280
in L3VPN multicast 191–2
P-instance 282–3
- PIM/GRE mVPN 279–86
- PIM-SM ASM 300
- Pim-SM SSM 286–7
- Ping 246, 460
- Pipe model 246
- PLR, *see* Point of local repair
- P-multicast Service Interface tunnel
attribute 297, 583
- PMSI tunnel attribute *see* P-multicast Service Interface tunnel attribute
- Point of local repair 73, 156
- Point-to-multipoint LSPPing *see* LSPPing extensions for point-to-multipoint LSPs
- Point-to-Multipoint LSPs 165–77
comparison of LDP-signaled and RSVP-signaled LSPs 175
fast reroute 181–3
forwarding plane mechanisms 165–7
- IP multicast traffic 163–4, 180
- IP unicast traffic 179, 180
- and L3VPN multicast 191, 193, 286–300
- LAN procedures 176, 178
- Layer 2 traffic 178–9
- LDP signaling for 173–4
minimum cost tree (Steiner tree) 165, 170, 172
- path computation 167–70
- RSPV signaling for 167–76
- shortest path tree 165, 170
and VPLS 179, 193
- Policing
and bandwidth reservations 60, 96
and Diffserv-TE 113, 118
- Preemption
and LSP priorities 43
and RDM 123–4
- Priority
Hold priority, *see* LSP priorities
Setup priority, *see* LSP priorities
- Protection
bandwidth 93–4
and Diffserv-TE 133
egress 433–7
end-to-end 70–1, 97, 155
at layer 68
link 81–9
local, *see* Fast-reroute

- Protection (*Continued*)
 node 89–91
 path, *see* End-to-end
 signaling 87
 service 424
 traffic forwarding 84
 using the protection path 85
- Pseudowire
 connection verification using VCCV 457–9
 interdomain 427
 multihop 427–8
 PWE redundancy 362, 364, 425, 427, 583
 PWE ACH 448, 457–8
- QoS, *see* Quality-of-Service
 Quality-of-Service
 in L3VPNs, *see* L3VPN QoS
 and overprovisioning 114–6, 125
- Radio Access Network (RAN) 369, 507, 540, 583
- RD, *see* Route distinguisher
 RDM, *see* Russian dolls model
 Record route object
 and bandwidth protection 23–4, 93–4, 96, 99, 103
 and fast-reroute 13, 22, 27, 80
 Reoptimization 154–5
 Reservation granularity 56–7
 RESV messages
 and LSP setup 24, 49, 88, 167–9
 RNC 540
 Route distinguisher 208–9, 248, 321–2, 354
 Route reflectors
 in L3VPN setups 3, 7, 27, 57, 231–5, 240–1
 Route resolution 230, 567
 Route target 209–15
 export RT 212, 214, 471, 586
 filtering 207, 210, 226–7, 237
 import RT 211–2, 288–9
 RRO, *see* Record route object
 RSTP 483, 584
- RSVP 21–5
 comparison with LDP 25–31, 175
 path message 22–4, 83, 167–9
 and point-to-multipoint LSPs 159, 187–91
 refresh reduction 25
 Resv message 23–4, 49, 88, 167–9
 RSVP error message 72
 and crankback 147–9
 and fast-reroute 13, 107
 RT, *see* Route target
 RTF, *see* Route target filtering
 Russian dolls model 122–3, 583
- S-bit (in MPLS header) 7
- S-VLAN 550–1
- SDP 299, 308
- Secondary path 64, 71–2
- Segmented tunnels 316
- Self-healing networks 452–3
- Shared explicit 51, 88, 169
- Shared risk link group 91–2
- Shortest path tree (of P2MP LSP) 165, 170–1, 191, 299
- SONET APS 68
- source AS extended community 311–6
- Split horizon 380, 382, 408
- Spoke PE (in VPLS) 213, 387–88, 397
- SRLG, *see* Shared risk link group
- STP 396, 416, 483–4
- Sub-LSP 167–8
- TE LSP segment 141–2, 146, 156–7, 159
- TE-class 119, 120
- TE-class matrix 119, 120
- TED, *see* Traffic engineering, database
- Traceroute
 in a VPN setup 138, 233, 280, 446, 450
 LSP traceroute, *see* LSP traceroute
- Traffic engineering
 by manipulating IGP metrics 61, 334–5
 constraints 292, 333
 database 45

- extensions to link-state protocols
 - 45, 119
- in LDP networks 60
- inter-area, *see* Interdomain traffic engineering
- inter-AS, *see* traffic engineering
- interdomain 13
- interprovider 138–9
- metric 45
- with MPLS 3
- and offline path computation 48, 61, 151
- scalability 3, 26, 54–56, 154
- shortcuts 53

- Upstream label allocation 176–7, 187
 - and penultimate hop popping 84–5, 192, 220, 366

- VC ID 352–3, 358, 385
- VCCV 457–60, 520
- VE ID 389–90, 391–2
- Virtual Private LAN Service
 - autodiscovery 236, 389, 391–2
 - BGP signaling for 392, 402
 - comparison of LDP and BGP schemes 365, 396–7
 - forwarding of multicast and broadcast frames 382–3
 - forwarding of unicast frames 379, 381–2
 - hierarchical VPLS (H-VPLS) 384, 386–7

- hub PE 387–8
- interprovider mechanisms 392–4, 413–5
- label block 356, 391
- LDP/BGP interworking 406–9
- LDP signaling for 9, 29, 173, 175, 341, 345, 351, 357–8
- MAC learning 390, 393, 407, 483
- MAC limiting 483
- mechanism overview 375–9
- multihoming (in VPLS) 363, 394, 413
- point-to-multipoint LSPs in 187–8
- policing 66, 115, 131, 136
- spoke PE 213, 387–8, 397
- VE ID 389
- Virtual private network
 - choice of BGP for route distribution 202–4, 206
 - connectivity models 212
 - goal of the VPN solution 203
 - label 216–8
 - overlapping 213–5
 - overlay model 201–2
 - peer model 202–4
- VPN-IP address family 208, 215
- VPLS, *see* Virtual Private LAN Service
- VPN, *see* L2VPN; Virtual private network; L3VPN
- VRF 205–7
- VRF Route Import extended community 314, 330–2, 570

- xDSL 549