



## **Cisco Prime Fulfillment Theory of Operations Guide 6.1**

### **Americas Headquarters**

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

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

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

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

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

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

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1005R)

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

*Cisco Prime Fulfillment Theory of Operations Guide 6.1*  
Copyright © 2011 Cisco Systems, Inc. All rights reserved.



# CONTENTS

## About This Guide v

Audience v

Related Documentation v

Document Conventions vi

Formatting vi

Navigation and Screens vii

Callouts vii

Obtaining Documentation and Submitting a Service Request vii

---

## CHAPTER 1

### Layer 2 VPN Concepts 1-1

Layer 2 Terminology Conventions 1-1

MEF Terminology Conventions 1-1

Mapping MEF Terminologies to Network Technologies 1-3

Prime Fulfillment Terminology and Supported Network Types 1-4

L2VPN Service Provisioning 1-5

Point-to-Point Ethernet (EWS and ERS) (EPL and EVPL) 1-5

ATM over MPLS (ATMoMPLS) 1-8

Frame Relay over MPLS (FRoMPLS) 1-9

FlexUNI/EVC Ethernet Service Provisioning 1-10

Overview 1-10

FlexUNI/EVC Features 1-11

Platform Support for FlexUNI/EVC in Cisco Prime Fulfillment 6.1 1-12

IOS Platform Support 1-12

IOS XR Platform Support 1-13

Device Roles with FlexUNI/EVC 1-14

Topology Overview for FlexUNI/EVC 1-14

CE Directly Connected and FlexUNI/EVC 1-15

CE Directly Connected and No FlexUNI/EVC 1-15

CE Not Directly Connected and FlexUNI/EVC 1-15

CE Not Directly Connected and No FlexUNI 1-15

A Note on Checking of Configurations 1-15

FlexUNI/EVC ATM-Ethernet Interworking Service Provisioning 1-16

VPLS Service Provisioning 1-16

Multipoint EWS (EP-LAN) for an MPLS-Based Provider Core 1-17

Multipoint ERS (EVP-LAN) for an MPLS-Based Provider Core	1-17
Topology for MPLS-Based VPLS	1-17
VPLS for an Ethernet-Based (L2) Provider Core	1-18
Multipoint EWS (EP-LAN) for an Ethernet-Based Provider Core	1-18
Multipoint ERS (EVP-LAN) for an Ethernet-Based Provider Core	1-19
Topology for Ethernet-Based VPLS	1-19

**CHAPTER 2**

**MPLS VPN Concepts 2-1**

MPLS VPNs	2-1
Intranets and Extranets	2-2
VPN Routing and Forwarding Tables	2-3
VRF Implementation	2-4
VRF Instance	2-5
Independent VRF Object Management	2-5
Route Distinguishers and Route Targets	2-5
Route Target Communities	2-6
Route Targets	2-6
Hub and Spoke Considerations	2-8
Full Mesh Considerations	2-8
MPLS VPN Security	2-8
Address Space and Routing Separation	2-8
Address Space Separation	2-9
Routing Separation	2-9
Hiding the MPLS Core Structure	2-9
Resistance to Attacks	2-10
Securing the Routing Protocol	2-11
Label Spoofing	2-12
Securing the MPLS Core	2-12
Trusted Devices	2-13
PE-CE Interface	2-13
Routing Authentication	2-13
Separation of CE-PE Links	2-13
LDP Authentication	2-14
Connectivity Between VPNs	2-14
MP-BGP Security Features	2-14
Security Through IP Address Resolution	2-15
Ensuring VPN Isolation	2-16

**CHAPTER 3****Traffic Engineering Management Concepts 3-1**

- Prime Fulfillment TEM Overview 3-1
- Features in Prime Fulfillment 3-2
- Prime Fulfillment TEM Basics 3-2
  - Managed/Unmanaged Primary Tunnels 3-2
  - Conformant/Non-Conformant Tunnels 3-3
    - Defining Conformant/Non-Conformant Tunnels 3-3
    - Managing Non-Conformant Tunnels 3-4
  - Multiple Concurrent Users 3-4
    - Concurrent Use with Managed and Unmanaged Tunnels 3-4
    - Locking Mechanism 3-5
  - Multiple OSPF Areas 3-5
    - Devices Suitable for TE Discovery 3-5
    - TE Discovery and the TE Area Identifier 3-6
    - Example of Multiple OSPF Area Network 3-6
  - Bandwidth Pools 3-6
  - Planning Tools 3-7
  - Connectivity Protection (CSPF) Backup Tunnels 3-8
  - Class-Based Tunnel Selection 3-8
  - Policy-Based Tunnel Selection 3-9

**CHAPTER 4****Prime Diagnostics Overview 4-1**

- IPv6 4-2

**APPENDIX A****MPLS Service Request Transition States A-1****INDEX**





## About This Guide

---

This guide provides an overview to Cisco Prime Fulfillment 6.1. Prime Fulfillment is a carrier-class network and service-management offering for the rapid and cost-effective delivery of IP services.

Prime Fulfillment offers complete MPLS VPN services management with rapid deployment and error-free provisioning capabilities. A service operator can begin with a single machine where Prime Fulfillment is installed and add processing servers that will be used by the Prime Fulfillment's master machine to offload processing and monitoring. Prime Fulfillment's master machine controls and monitors all its processing servers to deliver load-balancing and error-free provisioning.

Prime Fulfillment simplifies and speeds the deployment and management of packet-based services for faster time to revenue while increasing operating efficiencies. It is an end-to-end network management solution that scales as an organization evolves.

## Audience

This guide is designed for network engineers, service operators, and business managers who are responsible for configuring, provisioning, and managing MPLS VPN, L2 VPN, TEM software and Diagnostic services on a network. Network managers and operators should be familiar with the following topics:

- Basic concepts and terminology used in internetworking.
- Layer 2 Virtual Private Network (L2VPN), Virtual Private LAN Service (VPLS), VPN, Multiprotocol Label Switching (MPLS), and terms and technology.
- Network topologies and protocols.

## Related Documentation

The entire documentation set for Cisco Prime Fulfillment, can be accessed at:

[http://www.cisco.com/en/US/products/sw/netmgtsw/ps4748/tsd\\_products\\_support\\_series\\_home.html](http://www.cisco.com/en/US/products/sw/netmgtsw/ps4748/tsd_products_support_series_home.html)

or at:

<http://www.cisco.com/go/isc>

The following documents comprise the Cisco Prime Fulfillment 6.1 documentation set:

**General Documentation (in suggested reading order)**

- *Cisco Prime Fulfillment Getting Started and Documentation Guide 6.1*  
[http://www.cisco.com/en/US/docs/net\\_mgmt/prime/fulfillment/6.1/roadmap/docguide.html](http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.1/roadmap/docguide.html)
- *Release Notes for Cisco Prime Fulfillment 6.1*  
[http://www.cisco.com/en/US/docs/net\\_mgmt/prime/fulfillment/6.1/release/notes/relnotes.html](http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.1/release/notes/relnotes.html)
- *Cisco Prime Fulfillment Installation Guide 6.1*  
[http://www.cisco.com/en/US/docs/net\\_mgmt/prime/fulfillment/6.1/installation/guide/installation.html](http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.1/installation/guide/installation.html)
- *Cisco Prime Fulfillment User Guide 6.1*  
[http://www.cisco.com/en/US/docs/net\\_mgmt/prime/fulfillment/6.1/user/guide/prime\\_fulfill.html](http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.1/user/guide/prime_fulfill.html)
- *Cisco Prime Fulfillment Theory of Operations Guide 6.1*  
[http://www.cisco.com/en/US/docs/net\\_mgmt/prime/fulfillment/6.1/theory/operations/guide/theory.html](http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.1/theory/operations/guide/theory.html)
- *Cisco Prime Fulfillment Third Party and Open Source Copyrights 6.1*  
[http://www.cisco.com/en/US/docs/net\\_mgmt/prime/fulfillment/6.1/third\\_party/open\\_source/copyright/Prime\\_Fulfillment\\_Third\\_Party\\_and\\_Open\\_Source\\_Copyrights61.pdf](http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.1/third_party/open_source/copyright/Prime_Fulfillment_Third_Party_and_Open_Source_Copyrights61.pdf)

**API Documentation**

- *Cisco Prime Fulfillment API Programmer Guide 6.1*  
[http://www.cisco.com/en/US/docs/net\\_mgmt/prime/fulfillment/6.1/developer/guide/apipg.html](http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.1/developer/guide/apipg.html)
- *Cisco Prime Fulfillment API Programmer Reference 6.1*  
[http://www.cisco.com/en/US/docs/net\\_mgmt/prime/fulfillment/6.1/developer/reference/xmlapi.zip](http://www.cisco.com/en/US/docs/net_mgmt/prime/fulfillment/6.1/developer/reference/xmlapi.zip)

**Note**


---

All documentation *might* be upgraded over time. All upgraded documentation will be available at the same URLs specified in this document.

---

## Document Conventions

This guide uses the following documentation conventions.

## Formatting

This guide uses the following formatting conventions:

- User input and controls are indicated in **bold**; for example, “enter **1234**” and “click **Modify Scope**.”
- Object attributes are indicated in *italics*; for example, “the *failover-safe-period* attribute.”
- Cross-references to chapters or sections of chapters are indicated in [blue](#) type. For example, see [Document Conventions, page vi](#).



## Navigation and Screens

This guide uses the following navigation and screen display conventions:

- Windows systems use a two-button mouse. To drag and drop an object, click and hold the left mouse button on the object, drag the object to the target location, then release the button.
- Solaris systems use a three-button mouse. To drag and drop an object, click and hold the middle mouse button on the object, drag the object to the target location, then release the button.
- Screen displays can differ slightly from those included in this guide, depending on the system or browser you use.
- Web UI Navigation bar labels can have IPv4 and IPv6 variants depending on the administrator role privileges assigned. To simplify procedural instructions, this Guide uses the most generic versions of the menu bar labels, unless there is a need to be more specific. For example, the **Address Space** menu label might be rendered as **Address Space v4** and **Address Space v6**. The instructions will have the label simply as **Address Space**.

## Callouts

Callouts in the text have the following meaning:



**Caution**

*Be careful.* The description alerts you to potential data damage or loss.



**Note**

*Take note.* The description is particularly noteworthy.



**Timesaver**

*Save time.* The description can present a timesaver.



**Tip**

*Consider this helpful hint.* The description can present an optimum action to take.

## Obtaining Documentation and Submitting a Service Request

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

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

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





# CHAPTER 1

## Layer 2 VPN Concepts

---

This chapter provides an overview of Prime Fulfillment Layer 2 VPN concepts. It contains the following sections.

- [Layer 2 Terminology Conventions, page 1-1](#)
- [L2VPN Service Provisioning, page 1-5](#)
- [FlexUNI/EVC Ethernet Service Provisioning, page 1-10](#)
- [VPLS Service Provisioning, page 1-16](#)

## Layer 2 Terminology Conventions

Layer 2 service provisioning for the Prime Fulfillment consists of the Layer 2 Virtual Private Network (L2VPN) Service, FlexUNI/EVC Service and the Virtual Private LAN Service (VPLS). The purpose of this section is to clarify the terminologies used in Prime Fulfillment, as well in the industry at large, for these services.

There are three sets of terminologies in use:

- The current Metro Ethernet Forum (MEF) terminology
- The former MEF terminology
- Prime Fulfillment terminology (which is close to the former MEF terminology)

## MEF Terminology Conventions

In general, for L2VPN services, the MEF supports four general Ethernet service type constructs:

- Ethernet Line (E-Line). Provides a point-to-point Ethernet Virtual Circuit (EVC).
- Ethernet LAN (E-LAN). Provides a multipoint-to-multipoint EVC.
- FlexUNI with PW-core. This offers EPL and EVPL.
- FlexUNI with VPLS-core. This offers E-LAN and E-PLAN

Two Ethernet services are available for each type. These are distinguished by the means of service identification used at the user-to-network interface (UNIs), as follows:

- Port based. All-to-one bundling. These are referred to as “private.”
- VLAN-based. These services are multiplexed. The EVC is identified by a VLAN ID. These are referred to as “virtual private.”

Table 1-1 summarizes these relationships.

**Table 1-1 Ethernet Service Definitions**

Service Type	Port-Based	VLAN-Based
E-Line	Ethernet Private Line (EPL)	Ethernet Virtual Private Line (EVPL)
E-LAN	Ethernet Private LAN (EP-LAN)	Ethernet Virtual Private LAN (EVP-LAN)

In addition to E-Line and E-LAN services, two additional service types are available for Layer 2:

- Frame Relay over MPLS (FRoMLS)
- ATM over MPLS (ATMoMPLS)

These service types are not covered in the current MEF documentation, in spite of the fact that the MEF has merged with the Frame Relay forum.

Formerly, another terminology was used by the MEF for Layer 2 services. Table 1-3 maps the older terminology to the current one.

**Table 1-2 MEF Ethernet Service Term Mappings**

Current MEF Term	Former MEF Term
<b>L2VPN over MPLS Core</b>	
Ethernet Private Line (EPL)	Ethernet Wire Service (EWS)
Ethernet Virtual Private Line (EVPL)	Ethernet Relay Service (ERS)
ATM over MPLS (ATMoMPLS)	ATM over MPLS (ATMoMPLS)
Frame Relay over MPLS (FRoMPLS)	Frame Relay over MPLS (FRoMPLS)
<b>VPLS over MPLS Core</b>	
Ethernet Private LAN (EP-LAN)	Ethernet Wire Service (EWS) or Ethernet Multipoint Service (EMS)
Ethernet Virtual Private LAN (EVP-LAN)	Ethernet Relay Service (ERS) or Ethernet Relay Multipoint Service (ERMS)
<b>VPLS over Ethernet Core</b>	
Ethernet Private LAN (EP-LAN)	Ethernet Wire Service (EWS)
Ethernet Virtual Private LAN (EVP-LAN)	Ethernet Relay Service (ERS)

For additional information about MEF conventions and additional useful background information on Metro Ethernet standards, see the MEF website at the following URL:

<http://metroethernetforum.org>

In particular, see the document *Metro Ethernet Services Definitions Phase 2* available under the section Information Center > MEF Technical Specifications on the MEF website for a useful presentation of Metro Ethernet terms and definitions.

## Mapping MEF Terminologies to Network Technologies

The MEF terminology only describes the outside characteristics of a service, that is, what the service looks like from the perspective of a customer looking in towards the user-to-network-interface (UNI) device. It does not describe how the service is implemented.

For details about how these service are implemented, see the following URL:

<http://www.cisco.com/go/ce>

In particular, see the documentation on that site on the subject of Cisco IP Next-Generation Network (NGN) Carrier Ethernet Design. The IP NGN Carrier Ethernet Design represents key elements of the Cisco IP NGN architecture that enable a best-in-class implementation for consistent service delivery optimized to meet the specific demands of each service. It is the end-to-end service transport foundation from the network access to the IP/MPLS core. This design provides integrated linkages with the service and application layer components to offer a converged, intelligent, reliable, and scalable network model to meet current and future network service requirements.

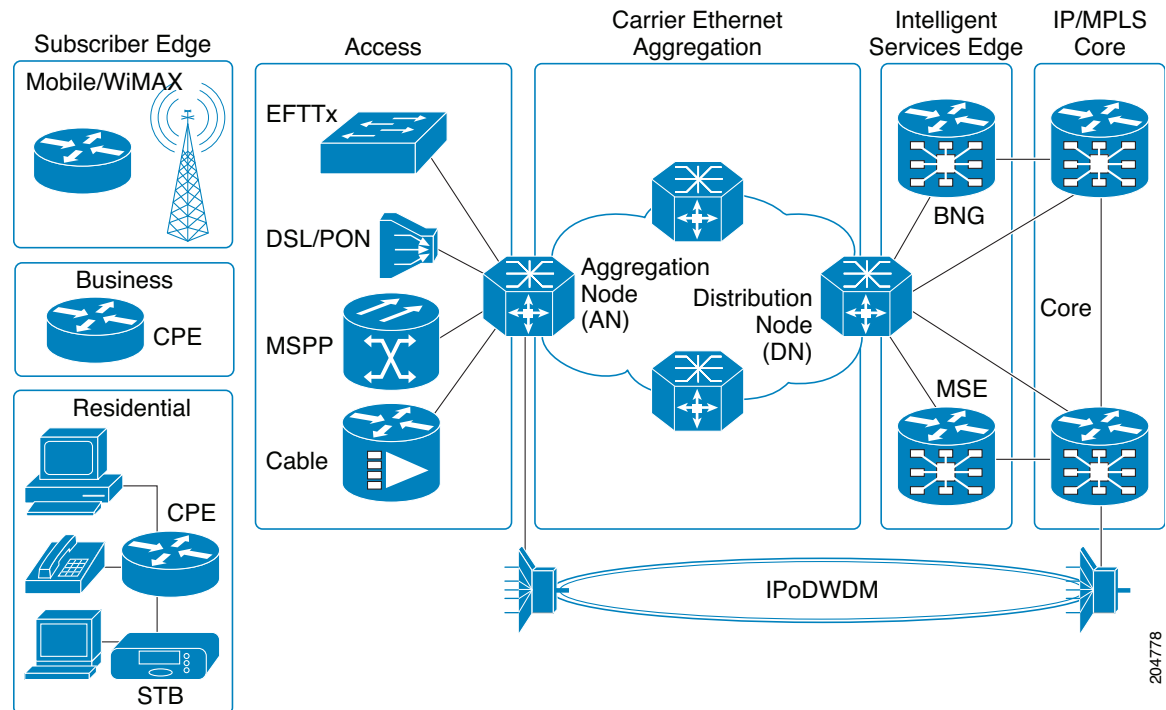
The IP NGN Carrier Ethernet Design (see [Figure 1-1](#)) provides a platform-independent architecture and Ethernet-based services model across all Carrier Ethernet platforms. This allows service providers to optimize service transport with the intelligence of appropriate networking technologies (such as Ethernet, IP, MPLS, multicast, pseudowire, or hierarchical private virtual LAN services) to meet their business and quality-of-experience goals.



**Note**

Real-world network implementations may implement only a subset of this very scalable architecture.

**Figure 1-1 IP NGN Carrier Ethernet Design**



204778

## Prime Fulfillment Terminology and Supported Network Types

This section discusses the Prime Fulfillment terminology for Layer 2 services and supported network types. Prime Fulfillment can provision the following service types:

- E-Line (EPL/EWS and EVPL/ERS)
- E-LAN (EP-LAN and EVP-LAN/ERMS)
- FRoMPLS
- ATMoMPLS

Prime Fulfillment also supports provisioning Ethernet services on a network that consists only of Ethernet switches (no MPLS), and this is referred to in Prime Fulfillment terminology as VPLS with L2 core.



### Note

For E-Line and E-LAN services, we recommend using the FlexUNI/EVC service policy type (see the appropriate chapters in this guide for how to create FlexUNI/EVC policies and service requests). You might have existing services that have been provisioned using the L2VPN and VPLS service policy types. These are still supported and can be maintained with those service types, but new services should use the FlexUNI /EVC service policy type. For ATM and FRoMPLS services, use the L2VPN service policy, as before.

In the Prime Fulfillment GUI and throughout this user guide, the naming conventions for these Ethernet services appear. These align closely with the earlier MEF conventions. This is expected to be updated in future releases of Prime Fulfillment. The equivalent terms used by the MEF forum are summarized in [Table 1-3](#), for reference.

**Table 1-3 Ethernet Service Term Mappings**

Term Used in Prime Fulfillment 5.2 GUI and This User Guide	Current MEF Equivalent Term
<b>L2VPN over MPLS Core</b>	
Ethernet Wire Service (EWS)	Ethernet Private Line (EPL)
Ethernet Relay Service (ERS)	Ethernet Virtual Private Line (EVPL)
ATM over MPLS (ATMoMPLS)	—
Frame Relay over MPLS (FRoMPLS)	—
<b>VPLS Over MPLS Core</b>	
Ethernet Wire Service (EWS) or Ethernet Multipoint Service (EMS)	Ethernet Private LAN (EP-LAN)
Ethernet Relay Service (ERS) or Ethernet Relay Multipoint Service (ERMS)	Ethernet Virtual Private LAN (EVP-LAN)
<b>VPLS over Ethernet Core</b>	
Ethernet Wire Service (EWS)	Ethernet Private LAN (EP-LAN)
Ethernet Relay Service (ERS)	Ethernet Virtual Private LAN (EVP-LAN)

# L2VPN Service Provisioning

This section provides an overview of Prime Fulfillment provisioning for L2VPN services that provide Layer 2 point-to-point connectivity over an MPLS core. Cisco's Any Transport over MPLS (AToM) enables support for these services. These implementations, in turn, support service types, as follows:

- Ethernet Wire Service (EWS). The MEF term for this service is EPL.
- Ethernet Relay Service (ERS). The MEF term for this service is EVPL.
- ATM over MPLS (ATMoMPLS)
- Frame Relay over MPLS (FRoMPLS)

Instructions on creating policies and service requests for these services are provided in other chapters of the guide. For more information, see the following sections:

- [Point-to-Point Ethernet \(EWS and ERS\) \(EPL and EVPL\)](#), page 1-5
- [ATM over MPLS \(ATMoMPLS\)](#), page 1-8
- [Frame Relay over MPLS \(FRoMPLS\)](#), page 1-9

## Point-to-Point Ethernet (EWS and ERS) (EPL and EVPL)

The EWS and ERS services (also known as EPL and EVPL, respectively, in MEF terminology) are delivered with the Cisco Metro Ethernet offering. The same network architecture can simultaneously provide both ERS (EPL) and EWS (EVPL) connections to diverse customers. Additionally, this Metro Ethernet infrastructure can be used for access to higher-level services, such as IP-based virtual private networking, public internet communications, Voice over IP, or a combination of all applications.

### Ethernet Wire Service (EWS or EPL)

An Ethernet Virtual Circuit (EVC) connects two physical User-to-Network Interfaces (UNI) such that the connection appears like a virtual private line to the customer. VLAN transparency and control protocol tunnelling are supplied by the implementation of 802.1Q-in-Q tag-stacking technology. Packets received on one UNI are transported directly to the other corresponding UNI.

The MEF term for this service is EPL.

### Ethernet Relay Service (ERS or EVPL)

An Ethernet Virtual Circuit (EVC) is used to logically connect endpoints, but multiple EVCs could exist per single UNI. Each EVC is distinguished by 802.1Q VLAN tag identification. The ERS network acts as if the Ethernet frames have crossed a switched network, and certain control traffic is not carried between ends of the EVC. ERS is analogous to Frame Relay where the CE-VLAN tag plays the role of a Data-Link Connection Identifier (DLCI).

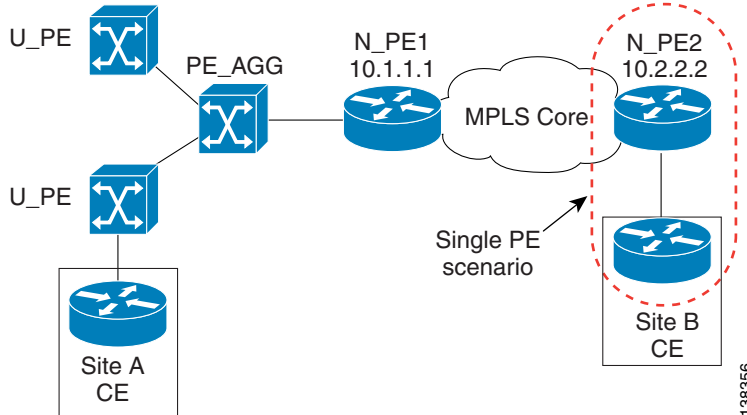
The MEF term for this service is EVPL.

### Topology for L2VPN Ethernet Over MPLS (ERS and EWS) (EPL and (EVPL)

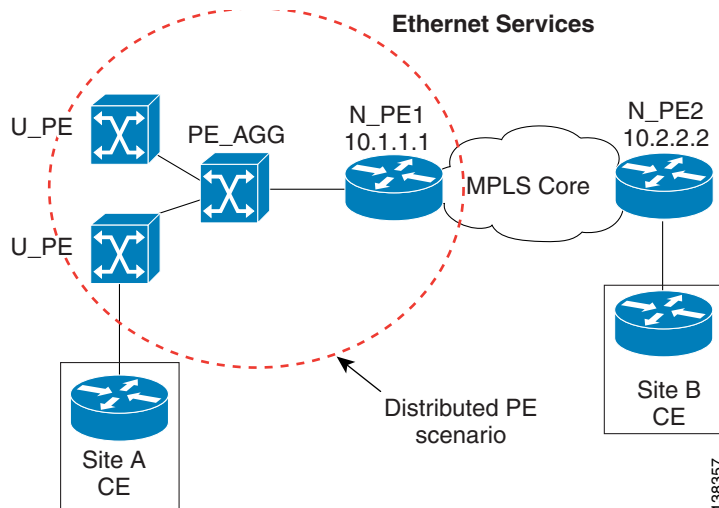
Ethernet Over MPLS (EoMPLS) is a tunnelling mechanism that allows the service provider to tunnel customer Layer 2 traffic through a Layer 3 MPLS network. It is important to remember that EoMPLS is a point-to-point solution only.

The following figures provide a reference for how EoMPLS is utilized. Ethernet Services can be distributed to the end customer in two ways.

- Single PE scenario—The customer is directly connected to an Ethernet port on the N-PE in [Figure 1-2](#).

**Figure 1-2 Single PE scenario**

- Distributed PE scenario—The end customer is connected through an Access Domain to the N-PE in [Figure 1-3](#). That is, there is a Layer 2 switching environment in the middle of CE and N-PE.

**Figure 1-3 Distributed PE Scenario**

In both cases, a VLAN is assigned in one of the following ways:

- Automatically assigned by Prime Fulfillment from the VLAN pool that is predefined by the user.
- Manually assigned by the user through the GUI or the North Bound Interface (NBI).

In EoMPLS, Prime Fulfillment creates a point-to-point tunnel and then targets the EoMPLS tunnel to the peer N-PE router through which the remote site can be reached. The remote N-PE is identified by its loopback address. In [Figure 1-4](#), N-PE1 and N-PE2 have 10.1.1.1 and 10.2.2.2 as loopback addresses. In [Figure 1-4](#), Site A has been allocated a VLAN-100 and Site B a VLAN-200. You can have different VLAN IDs at either end of the circuit because the VLANs have local significance only (that is, within the Ethernet access domain which is delimited by the N-PE).

For the N-PE that is serving Site A, a VLAN interface (Layer 3 interface) is created to terminate all L2 traffic for the customer, and an EoMPLS tunnel is configured on this interface.



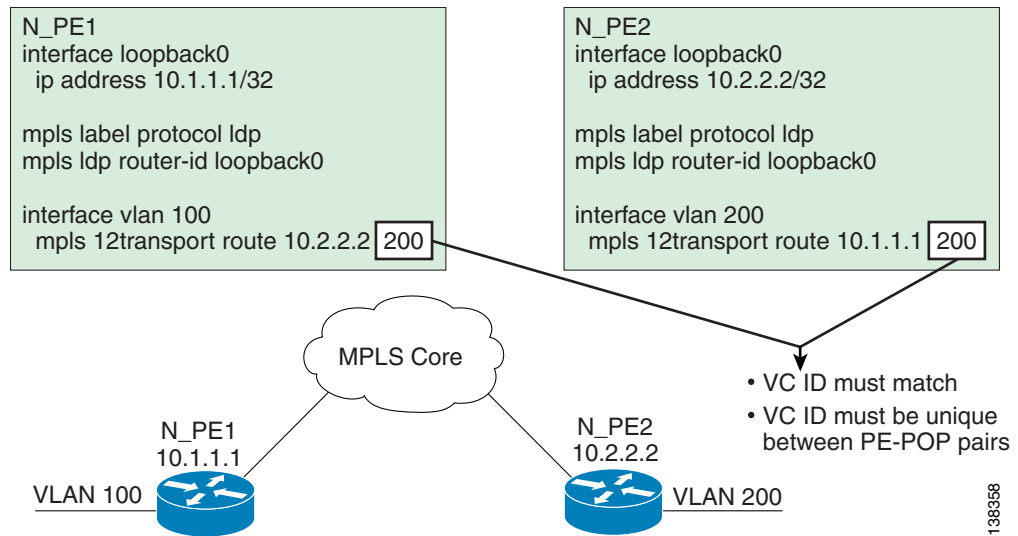


**Note**

This configuration is based on the Cisco 7600 Optical Services Router. Other routers, such as the Cisco 7200, have different configurations.

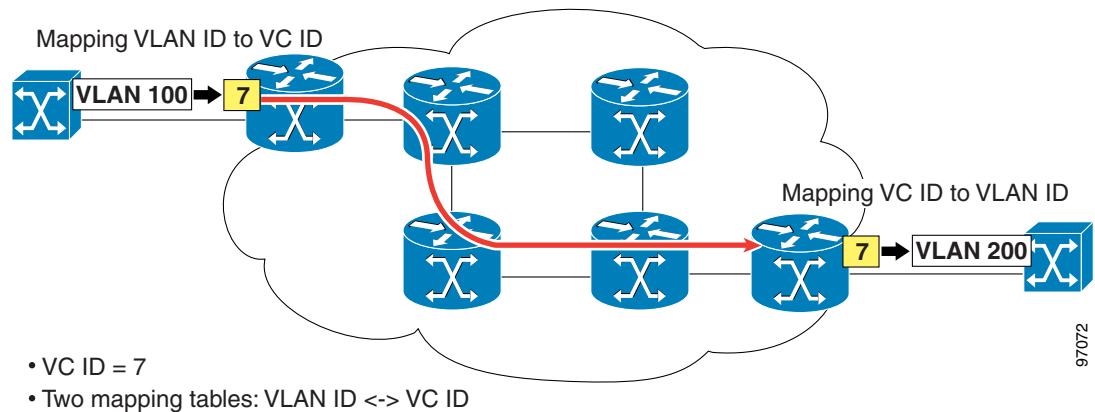
The VC ID that defines the EoMPLS tunnel is 200. (See Figure 1-4.)

**Figure 1-4 Ethernet over MPLS Configuration**



Note that the VC ID has to be the same on both ends of the EoMPLS tunnel. On each N-PE, there is mapping done between the VLANs to the EoMPLS tunnel. (See Figure 1-5.)

**Figure 1-5 EoMPLS Tunnel**

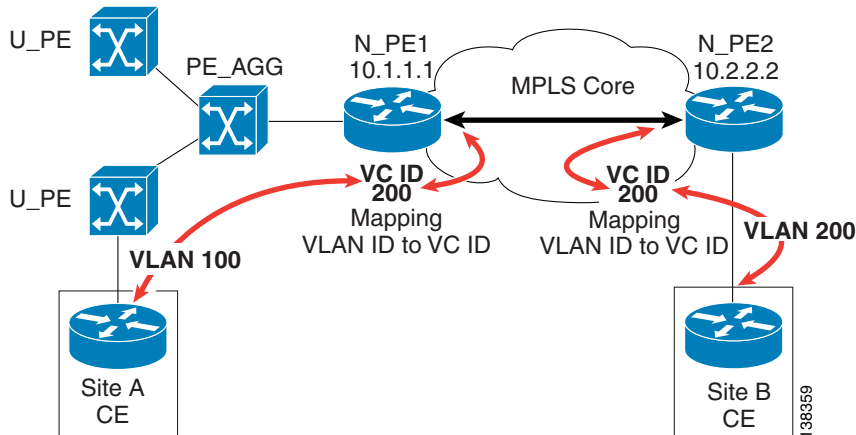


- VC ID = 7
- Two mapping tables: VLAN ID <-> VC ID

For the overall connection, this mapping is: VLAN ID <-> VC ID <-> VLAN ID.

This VLAN-VC ID mapping lets the service provider reuse VLAN IDs in Access Domains. (See Figure 1-6.)

**Figure 1-6 VLAN-VC ID Mapping**



The VLAN IDs allocated and used at each access domain do not have to be the same.

## ATM over MPLS (ATMoMPLS)

With Cisco ATM over MPLS (ATMoMPLS), Cisco supports ATM Adaptation Layer 5 (AAL5) transport and Cell Relay over MPLS.

### AAL5

AAL5 allows you to transport AAL5 PDUs from various customers over an MPLS backbone. ATM AAL5 extends the usability of the MPLS backbone by enabling it to offer Layer 2 services in addition to already existing Layer 3 services. You can enable the MPLS backbone network to accept AAL5 PDUs by configuring the provider edge (PE) routers at both ends of the MPLS backbone.

To transport AAL5 PDUs over MPLS, a virtual circuit is set up from the ingress PE router to the egress PE router. This virtual circuit transports the AAL5 PDUs from one PE router to the other. Each AAL5 PDU is transported as a single packet.

### Cell Relay over MPLS

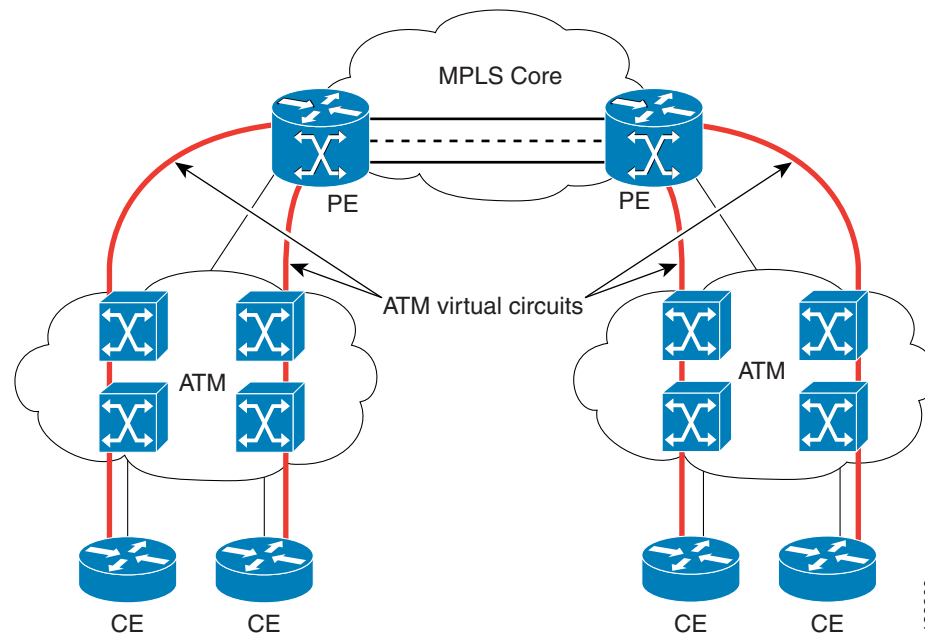
Cell Relay over MPLS allows you to transport ATM cells from various customers over an MPLS backbone. ATM Cell Relay extends the usability of the MPLS backbone by enabling it to offer Layer 2 services in addition to already existing Layer 3 services. You can enable the MPLS backbone network to accept ATM cells by configuring the provider edge (PE) routers at both ends of the MPLS backbone.

To transport ATM cells over MPLS, a virtual circuit is set up from the ingress PE router to the egress PE router. This virtual circuit transports the ATM cells from one PE router to the other. Each MPLS packet can contain one or more ATM cells. The encapsulation type is AAL0.

### Topology for ATMoMPLS

Only the single PE scenario is supported. (See [Figure 1-7](#).)

**Figure 1-7** Configuring AAL5 and Cell Relay over MPLS



136360

## Frame Relay over MPLS (FRoMPLS)

With Cisco AToM for Frame Relay, customer Frame Relay traffic can be encapsulated in MPLS packets and forwarded to destinations required by the customer. Cisco AToM allows service providers to quickly add new sites with less effort than typical Frame Relay provisioning.

Frame Relay over MPLS enables a service provider to transport Frame Relay frames across an MPLS backbone. This extends the reachability of Frame Relay and allows service providers to aggregate frame transport across a common packet backbone. The service provider can integrate an existing Frame Relay environment with the packet backbone to improve operational efficiency and to implement the high-speed packet interfaces to scale the Frame Relay implementations.

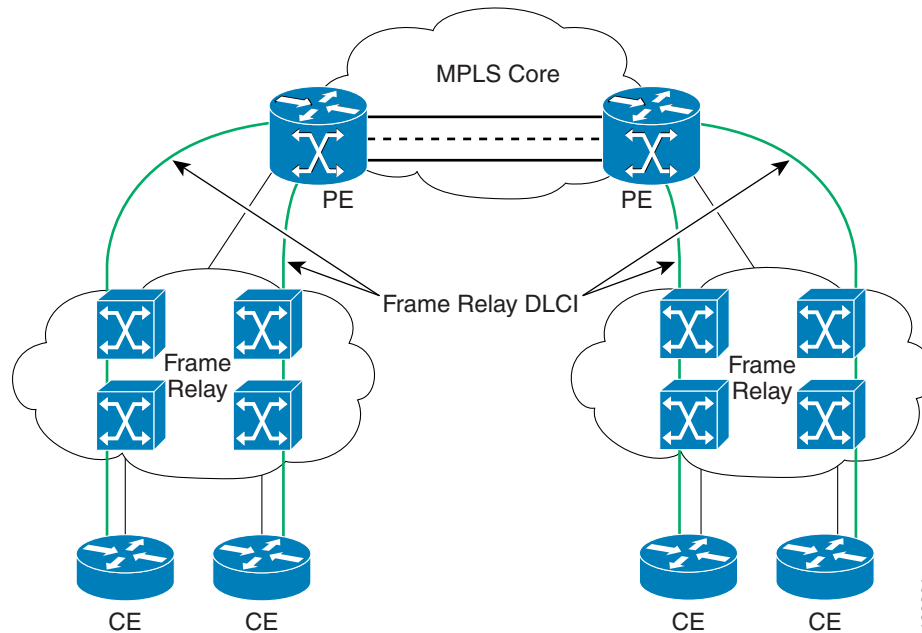
Transporting Frame Relay frames across MPLS networks provides a number of benefits, including:

- Frame Relay extended service.
- Aggregation to a higher speed backbone, such as OC-192, to scale Frame Relay implementations.
- Improved operational efficiency—the MPLS backbone becomes the single network that integrates the various existing networks and services.

### Topology for FRoMPLS

Only the single PE scenario is supported. (See [Figure 1-8](#).)

Figure 1-8 Frame Relay over MPLS



## FlexUNI/EVC Ethernet Service Provisioning

This section describes FlexUNI/EVC Ethernet and FlexUNI/EVC ATM-Interworking support in Cisco Prime Fulfillment 6.1. It contains the following topics:

- [Overview, page 1-10](#)
- [FlexUNI/EVC Features, page 1-11](#)
- [Platform Support for FlexUNI/EVC in Cisco Prime Fulfillment 6.1, page 1-12](#)
- [Device Roles with FlexUNI/EVC, page 1-14](#)
- [Topology Overview for FlexUNI/EVC, page 1-14](#)
- [A Note on Checking of Configurations, page 1-15](#)
- [FlexUNI/EVC ATM-Ethernet Interworking Service Provisioning, page 1-16](#)



### Note

For Ethernet (E-Line and E-LAN) services, use of the FlexUNI/EVC policy and service request is recommended. If you are provisioning services using the FlexUNI/EVC syntax, or plan to do so in the future, use the FlexUNI/EVC service. Existing services that have been provisioned using the L2VPN and VPLS service policy types are still supported and can be maintained with those service types. For ATM and FRoMPLS services, use the L2VPN service policy, as before.

## Overview

Flexible user network interface (FlexUNI) is a generic approach for creating Ethernet services in Prime Fulfillment. It can, if supported by the hardware, be used for all Ethernet provisioning. (For information on what platforms support FlexUNI/EVC see [Platform Support for FlexUNI/EVC in Cisco](#)

[Prime Fulfillment 6.1, page 1-12.](#)) The FlexUNI/EVC policy is flexible and generic and allows for service designers to provide greater service offerings than available through traditional Prime Fulfillment L2VPN and VPLS services.

Certain line cards have interfaces that support the Cisco IOS Ethernet Virtual Circuit (EVC) syntax. These interfaces can be configured with either EVC infrastructure features or with switch-port command-line interface commands (Class). FlexUNI optionally supports the EVC CLI syntax/infrastructure. For this reason, the FlexUNI policies and service requests are referred to by the umbrella term “FlexUNI/EVC.” However, it is important to note that FlexUNI/EVC policies and service request are not tied to the new EVC syntax. Service endpoints can use non-EVC syntax also.

Services leveraging the FlexUNI/EVC infrastructure are varied in nature, and there is not always a clear delineation between different services. This is because FlexUNI/EVC provides great flexibility in the way these services can be delivered. This can make it challenging to define the services. For example, a traditional ERS could be delivered in several ways by variations of the Class on the platform.

The FlexUNI/EVC policy and associated service request offer a generic and flexible service construct to support device capabilities. This policy is flexible enough to cater to different service offerings using the EVC architecture. It allows service designers to utilize most of the EVC features in a flexible manner, to match the hardware and platform capabilities.

The FlexUNI/EVC policy can be used to create only a FlexUNI/EVC service request and not any other existing Prime Fulfillment service request types, such as L2VPN, VPLS, and so on. Likewise, a FlexUNI/EVC service request can be created using only a FlexUNI/EVC policy and not any other existing Prime Fulfillment policies.

The FlexUNI/EVC infrastructure provides several benefits to Carrier Ethernet (CE) deployments, including:

- Flexible frame matching.
- Flexible VLAN tag manipulation and/or translation.
- Multiple services on the same port.
- Flexible service mapping.
- VLAN scaling and locally significant VLANs.

FlexUNI/EVC supports a variety of network configurations, such as the following:

- Provisioning of Ethernet access as a EVC-capable EWS interface on the N-PE.
- Interconnecting Ethernet accesses terminating on a single Cisco 7600 N-PE on one or multiple ports in a bridge domain.
- Interconnecting Ethernet accesses terminating on multiple Cisco 7600 N-PEs in a VPLS service.
- FlexUNI/EVC service support on Cisco ASR 9000 Series Routers running IOS XR.
- Services that combine the existing services with the Ethernet access, including the ERS/EWS interworking service.
- Provisioning of E-Line services, in which one or both N-PE interfaces are FlexUNI.

## FlexUNI/EVC Features

This section summarizes the features supported by the FlexUNI/EVC policy and service request in Prime Fulfillment:

- Choice of topology:
  - Customer edge device (CE) directly connected.

- CE connected through Ethernet access devices.
- Choice of platforms:
  - FlexUNI/EVC on all N-PEs.
  - FlexUNI/EVC on none of the N-PEs.
  - Mix of FlexUNI/EVC and the old infrastructure. This allows both the old and new platforms to co-exist, in order to ensure continued support for deployed platforms.
- Choice of connectivity across the MPLS core (with or without bridge-domain):
  - Pseudo wires
  - VPLS
  - Local (local connects)
- Flexible VLAN handling mechanism that deals with up to two levels of VLAN tags:
  - VLAN matching for service classification. This provides the ability to match both outer and inner VLAN tags, or the ability to match a range of inner VLAN tags.
  - VLAN manipulations, such as pop outer tag, pop inner tag, push outer tag, push inner tag, and VLAN translations (1:1, 2:1, 1:2, 2:2).
- Flexible forwarding options:
  - Configure a pseudowire on the MPLS core directly under a service instance (for E-Line only).
  - Configure a pseudowire on the MPLS core under a switch virtual interface (SVI) by associating it to a bridge domain.

**Note**


---

The appropriate VLAN manipulations are applicable to pseudowire in both cases.

---

- Associate traffic from different interfaces and/or VLANs onto a single bridge domain, with appropriate VLAN manipulations for VPLS.
- Associate traffic from different interfaces and/or VLANs onto a single bridge domain with appropriate VLAN manipulations for local connects.

## Platform Support for FlexUNI/EVC in Cisco Prime Fulfillment 6.1

FlexUNI/EVC services are supported on both IOS and IOS XR platforms, as detailed in the following sections.

### IOS Platform Support

The following IOS platforms are supported for the FlexUNI/EVC service:

- IOS 12.2(33)SRB and SRC
- ES20 line cards (2x10GE and 20x1GE)
- Shared Port Adaptor (SPA) Interface Processor-400 (SIP-400) line cards, version 2.0 (2x1GE and 5x1GE)

**Note**

See the [Cisco Prime Fulfillment Installation Guide 6.1](#) for the most current list of hardware and software platforms supported in this release.

The interfaces on the ES20 and SIP-400 line cards support the IOS EVC syntax.

Two example platform scenarios are covered in the next sections. Note that the UNI characteristics and the FlexUNI capabilities of the N-PE are not inter-dependent.

**Example 1**

FlexUNI/EVC service requests allow operators to add links with either a EVC-capable interface and/or the nonEVC-capable interface on the N-PE. For example, an operator can add three links to a FlexUNI/EVC service request (with VPLS connectivity) with the following configurations:

- Link one has a Cisco 67xx interface and an IOS 12.2 (33)SRB image on a Cisco 7600 N-PE.
- Link two has a Cisco 67xx interface and an IOS 12.2 (33)SRB image on a Cisco 7600 N-PE.
- Link three has an ES20-based interface and an IOS 12.2 (33)SRB image on a Cisco 7600 N-PE.

**Example 2**

As far as Layer 2 access nodes are concerned, configurations on the UNI/NNI of a U-PE and/or PE-AGG are not influenced by the FlexUNI/EVC capability on the N-PE. However, if a selected named physical circuit (NPC) with N-PE interface is configured with FlexUNI/EVC, it cannot be provisioned for traditional configuration. An error will be generated while saving the service request.

On the other hand, if a selected NPC with N-PE interface is configured without FlexUNI, it cannot be provisioned for FlexUNI configuration. An error will be generated while saving the service request.

For example, if for link one of a FlexUNI/EVC service request, if the encapsulation is selected as dot1Q, the interface can share other L2 ERS/VPLS ERMS UNIs on the same U-PE/PE-AGG.

If the N-PE interface that is part of the NPC being picked is already configured with non-FlexUNI/EVC features (using an existing L2VPN or VPLS service request), you cannot configure FlexUNI/EVC on it.

**Note**

If “Dot1Q Tunnel” is selected as the encapsulation type, the port cannot be shared with other services.

## IOS XR Platform Support

FlexUNI/EVC services are supported on Cisco ASR 9000 Series Routers running IOS XR.

**Note**

See the [Cisco Prime Fulfillment Installation Guide 6.1](#) for the most current list of hardware and software platforms supported in this release.

The following FlexUNI/EVC features are supported on IOS XR platforms:

- E-line connections. If an ASR 9000 is added on a direct link, only DOT1Q encapsulation is supported for E-Line services. When using L2 access nodes with NPCs, all supported encapsulations are available.
- E-LAN connections.
- Flexible frame matching.
- Flexible VLAN tag manipulation/translation.

- VLAN scaling and locally significant VLANs.
- The ability to create L2 and L3 services under the same physical interface, restricted only to subinterface.
- All the Layer 2 ports on Cisco ASR 9000 devices are trunk ports, hence only trunk port-based configuration is supported.

The following FlexUNI/EVC services are not supported on IOS XR platforms:

- The N-PE Pseudo-wire on SVI attribute is not supported. SVI interfaces are not available on the devices, which restricts support for Standard UNI and Port Security configuration when the UNI is configured on an N-PE.
- xconnect commands are not directly supported under interface configuration. Support for these commands has been moved to different a hierarchy in IOS XR.
- When the UNI is configured on an N-PE device, EWS service is not supported. Since the Cisco ASR 9000 device is a router, all the Layer 2 ports are default trunk. There is no option for configuring access ports, which restricts support for access port-based services.


**Note**


---

Unless otherwise noted, all of the FlexUNI/EVC policy and service request features documented in this guide are applicable for both IOS and IOS XR platforms.

---

## Device Roles with FlexUNI/EVC

Presently, Prime Fulfillment has U-PE, PE-AGG and N-PE devices. The basic PE device role association of Prime Fulfillment continues for FlexUNI/EVC policy and service requests. In this release of Prime Fulfillment, there are no changes made to the PE role assignment. A device having FlexUNI/EVC capabilities will not call for a change in the existing role assignment in Prime Fulfillment. However, FlexUNI/EVC capabilities in Prime Fulfillment are supported only for interfaces on N-PE and not on PE-AGG or U-PE devices.


**Note**


---

Prime Fulfillment does not support customer edge devices (CEs) for FlexUNI/EVC. If the access port contains any DSLAMS, non-Cisco Ethernet devices and/or other Cisco devices that are not supported by Prime Fulfillment, such nodes and beyond are not in the scope of Prime Fulfillment. In such cases, from the Prime Fulfillment perspective, the interface on the first Prime Fulfillment-managed device is the UNI.

---

## Topology Overview for FlexUNI/EVC

This section provides examples of various topologies supported with FlexUNI/EVC. As mentioned in the note at the end of section [Device Roles with FlexUNI/EVC, page 1-14](#), Prime Fulfillment does not support customer edge devices (CEs) with FlexUNI/EVC. References to the term “CE” in the following topology variations (such as “CE directly connected” and so on) is only to indicate how the customer or third-party devices connect to the N-PE. For all the cases involving FlexUNI/EVC, the CE is not supported in Prime Fulfillment. Also, any provider device that is not supported by Prime Fulfillment, and which is used in the access circuit, marks the boundary for the scope of Prime Fulfillment, beyond which no devices (that is, towards the CE, and including the unsupported node) is managed by Prime Fulfillment.



## CE Directly Connected and FlexUNI/EVC

With this combination, the UNI is the interface on a supported line card, with EVC capability configured. Prime Fulfillment does not configure Prime Fulfillment's standard UNI functionality (for example, port-security, storm control, and Layer 2 Protocol Tunneling). This is because of lack of command support on the FlexUNI/EVC-capable hardware. Operators can use templates to configure relevant platform supported parameters to realize any of these features not provided by Prime Fulfillment. Prime Fulfillment configures only the service instance with VLAN manipulations and pseudowire, VPLS, or local-connect on the UNI. NPCs are not needed while creating such links because NPCs are only required when there are access nodes between the N-PE and CE. Other intermediate Ethernet access nodes are not involved in this topology.

## CE Directly Connected and No FlexUNI/EVC

This is similar to the UNI on N-PE case in Prime Fulfillment. The FlexUNI/EVC service request can be used to create such links with older Cisco 7600 platforms (that is, N-PE interfaces without FlexUNI/EVC capability), but with plans of adding one or more future links with EVC support. If not, one could use the existing ERS/EWS/ERMS/EMS functionality in Prime Fulfillment. NPCs are not needed while creating such links because NPCs are only required when there are access nodes between the N-PE and CE. Other intermediate Ethernet access nodes are not involved in this topology.

## CE Not Directly Connected and FlexUNNEVC

This topology involves the following configurations:

- UNI on a U-PE or PE-AGG to which the CE is connected.
- Ethernet U-PE and/or PE-AGGs.
- N-PE with FlexUNI-capable interface on the CE-facing side.

All service-specific parameters, such as port-security, L2 Protocol Tunneling, storm control, and so on, are applicable to the UNI (Standard UNI) in such links. The U-PE and/or PE-AGG configurations will also have no change in CLIs. However, the EVC commands are applicable only on the N-PE (on the CE-facing interface). NPCs are used while creating such links.

## CE Not Directly Connected and No FlexUNI

This link is identical to an attachment circuit in existing Prime Fulfillment implementations. This has a standard UNI as in existing Prime Fulfillment services. NPCs are used while creating such links.

## A Note on Checking of Configurations

Prime Fulfillment attempts to provision all configurations generated by a FlexUNI/EVC service request. Prime Fulfillment does not perform any prior checks to verify if the CLIs are compatible with the specific devices being provisioned. This is to ensure flexibility of support for device/platform features, which could change over time. Hence, it is important for the service designer or operator to carefully create the FlexUNI/EVC policies and service requests.

## FlexUNI/EVC ATM-Ethernet Interworking Service Provisioning

Prime Fulfillment supports interworking of a service with ATM and Ethernet protocols across the MPLS core or local switching. ATM-Ethernet interworking is supported through the following features:

- Creation of a FlexUNI/EVC policy of type “ATM-Ethernet Interworking.” The ATM-Ethernet Interworking policy type supports a choice of MPLS core options:
  - Pseudowire
  - Local (local connects)
- Provisioning of ATM-Ethernet interworking using a single FlexUNI/EVC service request.
- Combination of EVC and non-EVC syntax, that is, the creation of an L2 circuit consisting of an L2 syntax and EVC syntax.
- Supported platforms:
  - ATM interworking is supported on the Cisco 7600 with ES-20 cards.
  - ASR 9000 device is supported for IOS XR 3.7.3 and IOS XR 3.9. Because there are no ATM interfaces on the Cisco ASR 9000, Prime Fulfillment does not support interworking on the ASR 9000 for ATM interfaces. Only Ethernet interfaces are supported.

## VPLS Service Provisioning

VPLS services are multipoint. They provide multipoint connectivity over an MPLS or an Ethernet core. These implementations, in turn, support service types, as follows:

- VPLS over MPLS core:
  - Ethernet Wire Service (EWS). This is also sometimes referred to as EMS, or Ethernet Multipoint Service. The MEF term for this service is EP-LAN.
  - Ethernet Relay Service (ERS). This is also sometimes referred to ERMS, or Ethernet Relay Multipoint Service. The MEF term for this service is EVP-LAN.
- VPLS over Ethernet core:
  - Ethernet Wire Service (EWS). The MEF term for this service is EP-LAN.
  - Ethernet Relay Service (ERS). The MEF term for this service is EVP-LAN.

Instructions on creating policies and service requests for these services are provided in other chapters of the guide.

VPLS is a multipoint Layer 2 VPN that connects two or more customer devices using EoMPLS bridging techniques. VPLS EoMPLS is an MPLS-based provider core, that is, the PE routers have to cooperate to forward customer Ethernet traffic for a given VPLS instance in the core. A VPLS essentially emulates an Ethernet switch from a user’s perspective. All connections are peers within the VPLS and have direct communications. The architecture is actually that of a distributed switch. Multiple attachment circuits have to be joined together by the provider core. The provider core has to simulate a virtual bridge that connects these multiple attachment circuits together. To achieve this, all PE routers participating in a VPLS instance form emulated VCs among them.

A Virtual Forwarding Instance (VFI) is created on the PE router for each VPLS instance. PE routers make packet-forwarding decisions by looking up the VFI of a particular VPLS instance. The VFI acts like a virtual bridge for a given VPLS instance. More than one attachment circuit belonging to a given VPLS can be connected to this VFI. The PE router establishes emulated VCs to all the other PE routers in that VPLS instance and attaches these emulated VCs to the VFI. Packet forwarding decisions are

based on the data structures maintained in the VFI. All the PE routers in the VPLS domain use the same VC-ID for establishing the emulated VCs. This VC-ID is also called the VPN-ID in the context of the VPLS VPN.

For more information, see the following sections:

- [Multipoint EWS \(EP-LAN\) for an MPLS-Based Provider Core](#), page 1-17
- [Multipoint ERS \(EVP-LAN\) for an MPLS-Based Provider Core](#), page 1-17
- [Topology for MPLS-Based VPLS](#), page 1-17

## Multipoint EWS (EP-LAN) for an MPLS-Based Provider Core

With multipoint EWS (also known as EP-LAN in MEF terminology), the PE router forwards all Ethernet packets received from an attachment circuit, including tagged, untagged, and Bridge Protocol Data Unit (BPDU) to either:

- Another attachment circuit or an emulated VC if the destination MAC address is found in the L2 forwarding table (VFI).
- All other attachment circuits and emulated VCs belonging to the same VPLS instance if the destination MAC address is a multicast/broadcast address or not found in the L2 forwarding table.

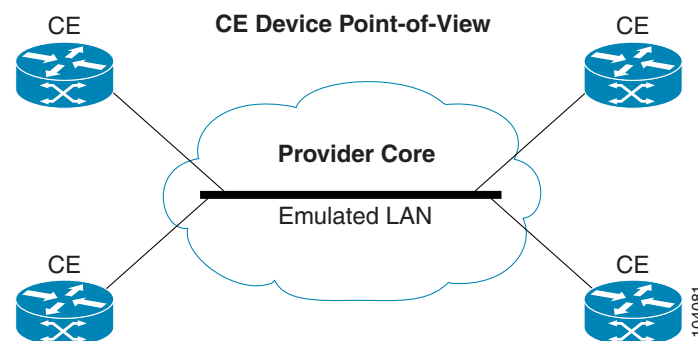
## Multipoint ERS (EVP-LAN) for an MPLS-Based Provider Core

With multipoint ERS (also known as EVP-LAN in MEF terminology), the PE router forwards all Ethernet packets with a particular VLAN tag received from an attachment circuit, excluding BPDU, to another attachment circuit or an emulated VC if the destination MAC address is found in the L2 forwarding table (VFI). If the destination MAC address is not found or if it is a broadcast/multicast packet, then it is sent on all other attachment circuits and emulated VCs belonging to the VPLS instance. The demultiplexing VLAN tag used to identify a VPLS domain is removed prior to forwarding the packet to the outgoing Ethernet interfaces or emulated VCs because it only has local significance.

## Topology for MPLS-Based VPLS

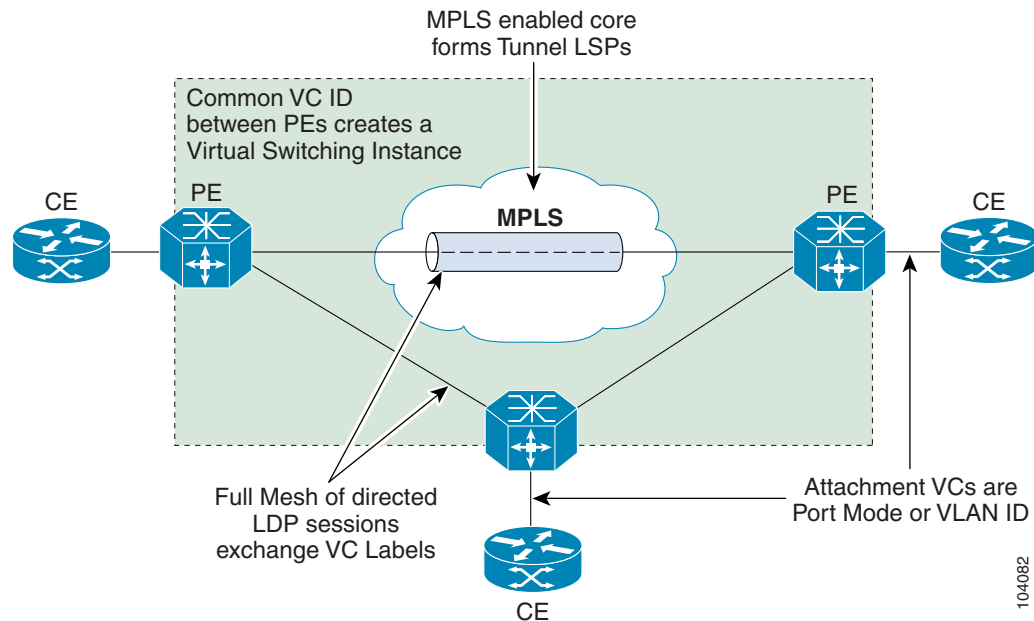
From a customer point of view there is no topology for VPLS. All the CE devices are connected to a logical bridge emulated by the provider core. Therefore, the CE devices see a single emulated LAN. (See [Figure 1-9](#).)

**Figure 1-9** *MPLS-Based VPLS Topology*



The PE routers must create a full-mesh of emulated virtual circuits (VCs) to simulate the emulated LAN seen by the CE devices. Forming a full-mesh of emulated VCs simplifies the task of emulating a LAN in the provider core. One property of a LAN is to maintain a single broadcast domain. That is, if a broadcast, multicast, or unknown unicast packet is received on one of the attachment circuits, it has to be sent to all other CE devices participating in that VPLS instance. The PE device handles this case by sending such a packet on all other attachment circuits and all the emulated circuits originating from that PE. With a full-mesh of emulated VCs, such a packet will reach all other PE devices in that VPLS instance. (See [Figure 1-10](#).)

**Figure 1-10 Full Mesh of Emulated VCs**



## VPLS for an Ethernet-Based (L2) Provider Core

With an Ethernet-based provider core, customer traffic forwarding is trivial in the core. VPLS for an Ethernet-based provider core is a multipoint Layer 2 VPN that connects two or more customer devices using 802.1Q-in-Q tag-stacking technology. A VPLS essentially emulates an Ethernet switch from a users perspective. All connections are peers within the VPLS and have direct communications. The architecture is actually that of a distributed switch.

For more information on VPLS for an Ethernet-based provided core, see the following sections:

- [Multipoint EWS \(EP-LAN\) for an Ethernet-Based Provider Core, page 1-18](#)
- [Multipoint ERS \(EVP-LAN\) for an Ethernet-Based Provider Core, page 1-19](#)
- [Topology for Ethernet-Based VPLS, page 1-19](#)

## Multipoint EWS (EP-LAN) for an Ethernet-Based Provider Core

Multipoint EWS (also known as EP-LAN in MEF terminology) is a service that emulates a point-to-point Ethernet segment. The EWS service encapsulates all frames that are received on a particular User to Network Interface (UNI) and transports these frames to a single egress UNI without reference to the

contents contained within the frame. This service operation means that EWS can be used for untagged or VLAN tagged frames and that the service is transparent to all frames offered. Because the EWS service is unaware that VLAN tags might be present within the customer frames, the service employs a concept of “All to One” bundling.

## Multipoint ERS (EVP-LAN) for an Ethernet-Based Provider Core

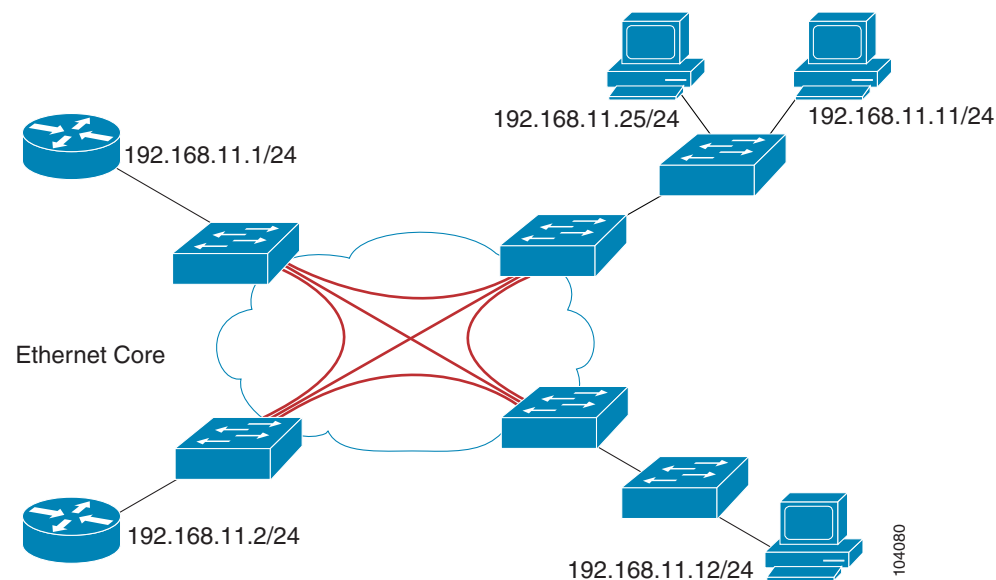
Multipoint ERS (also known as EVP-LAN in MEF terminology) models the connectivity offered by existing Frame Relay networks by using VLAN indices to identify virtual circuits between sites. ERS does, however, offer a far greater degree of QoS functionality depending upon the service provider's implementation and the customer's acceptance of VLAN indices that are administratively controlled by the service provider. Additionally, ERS service multiplexing capability offers lower cost of ownership for the enterprise as a single interface can support many virtual interfaces.

## Topology for Ethernet-Based VPLS

Ethernet-based VPLS differs from the point-to-point L2VPN definitions of EWS (EP-LAN) and ERS (EVP-LAN) by providing a multipoint connectivity model. The VPLS service does not map an interface or VLAN to a specific point-to-point pseudowire, but instead it models the operation of a virtual Ethernet switch. VPLS uses the customer's MAC address to forward frames to the correct egress UNI within the service provider's network for the EWS (EP-LAN).

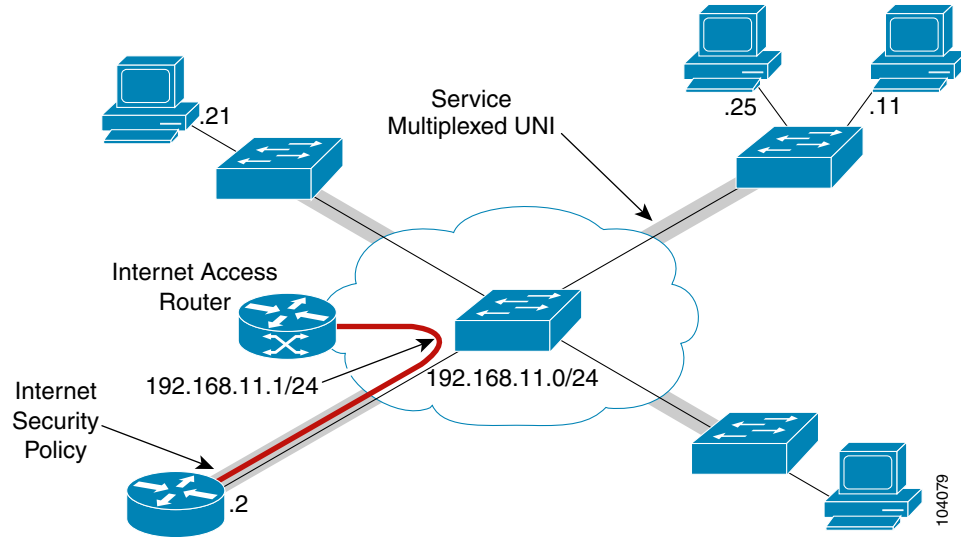
The EWS (EP-LAN) service emulates the service attributes of an Ethernet switch and learns source MAC to interface associations, flooding unknown broadcast and multicast frames. [Figure 1-11](#) illustrates an EWS (EP-LAN) VPLS topology.

**Figure 1-11 VPLS EWS Topology**



The Ethernet Relay Service (ERS or EVP-LAN) offers the any-to-any connectivity characteristics of EWS and the service multiplexing. This combination enables a single UNI to support a customer's intranet connection and one or more additional EVCs for connection to outside networks, ISPs, or content providers. [Figure 1-12](#) illustrates an ERS (EVP-LAN) VPLS multipoint topology.

Figure 1-12 VPLS ERS (EVP-LAN) Multipoint Topology





## CHAPTER 2

# MPLS VPN Concepts

This chapter provides a conceptual information useful for understanding MPLS. It contains the following sections:

- [MPLS VPNs, page 2-1](#)
- [MPLS VPN Security, page 2-8](#)

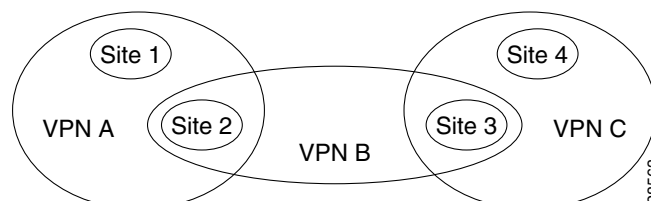
## MPLS VPNs

At its simplest, a virtual private network (VPN) is a collection of sites that share the same routing table. A VPN is also a network in which customer connectivity to multiple sites is deployed on a shared infrastructure with the same administrative policies as a private network. The path between two systems in a VPN, and the characteristics of that path, might also be determined (wholly or partially) by policy. Whether a system in a particular VPN is allowed to communicate with systems not in the same VPN is also a matter of policy.

In an MPLS VPN, the VPN generally consists of a set of sites that are interconnected by means of an MPLS provider core network, but it is also possible to apply different policies to different systems that are located at the same site. Policies can also be applied to systems that dial in; the chosen policies would be based on the dial-in authentication processes.

A given set of systems can be in one or more VPNs. A VPN can consist of sites (or systems) that are all from the same enterprise (intranet), or from different enterprises (extranet); it might consist of sites (or systems) that all attach to the same service provider backbone, or to different service provider backbones.

**Figure 2-1** *VPNs Sharing Sites*



MPLS-based VPNs are created in Layer 3 and are based on the peer model, which makes them more scalable and easier to build and manage than conventional VPNs. In addition, value-added services, such as application and data hosting, network commerce, and telephony services, can easily be targeted and deployed to a particular MPLS VPN because the service provider backbone recognizes each MPLS VPN as a secure, connectionless IP network.

The MPLS VPN model is a true peer VPN model that enforces traffic separations by assigning unique VPN route forwarding tables (VRFs) to each customer's VPN. Thus, users in a specific VPN cannot see traffic outside their VPN. Traffic separation occurs without tunneling or encryption because it is built directly into the network. (For more information on VRFs, see [VPN Routing and Forwarding Tables, page 2-3](#).)

The service provider's backbone is comprised of the PE and its provider routers. MPLS VPN provides the ability that the routing information about a particular VPN be present *only* in those PE routers that attach to that VPN.

### Characteristics of MPLS VPNs

MPLS VPNs have the following characteristics:

- Multiprotocol Border Gateway Protocol (MP-BGP) extensions are used to encode customer IPv4 address prefixes into unique VPN-IPv4 Network Layer Reachability Information (NLRI) values. NLRI refers to a destination address in MP-BGP, so NLRI is considered “one routing unit.” In the context of IPv4 MP-BGP, NLRI refers to a network prefix/prefix length pair that is carried in the BGP4 routing updates.
- Extended MP-BGP community attributes are used to control the distribution of customer routes.
- Each customer route is associated with an MPLS label, which is assigned by the provider edge router that originates the route. The label is then employed to direct data packets to the correct egress customer edge router. When a data packet is forwarded across the provider backbone, two labels are used. The first label directs the packet to the appropriate egress PE; the second label indicates how that egress PE should forward the packet.
- Cisco MPLS CoS and QoS mechanisms provide service differentiation among customer data packets.
- The link between the PE and CE routers uses standard IP forwarding.  
The PE associates each CE with a per-site forwarding table that contains only the set of routes available to that CE.

### Principal Technologies

There are four principal technologies that make it possible to build MPLS-based VPNs:

- Multiprotocol Border Gateway Protocol (MP-BGP) between PEs carries CE routing information.
- Route filtering based on the VPN route target extended MP-BGP community attribute.
- MPLS forwarding carries packets between PEs (across the service provider backbone).
- Each PE has multiple VPN routing and forwarding instances (VRFs).

## Intranets and Extranets

If all the sites in a VPN are owned by the same enterprise, the VPN is a corporate *intranet*. If the various sites in a VPN are owned by different enterprises, the VPN is an *extranet*. A site can be in more than one VPN. Both intranets and extranets are regarded as VPNs.

While the basic unit of connection is the site, the MPLS VPN architecture allows a finer degree of granularity in the control of connectivity. For example, at a given site, it might be desirable to allow only certain specified systems to connect to certain other sites. That is, certain systems at a site might be members of an intranet and members of one or more extranets, while other systems at the same site might be restricted to being members of the intranet only.



A CE router can be in multiple VPNs, although it can only be in a single site. When a CE router is in multiple VPNs, one of these VPNs is considered its primary VPN. In general, a CE router's primary VPN is the intranet that includes the CE router's site. A PE router might attach to CE routers in any number of different sites, whether those CE routers are in the same or in different VPNs. A CE router might, for robustness, attach to multiple PE routers. A PE router attaches to a particular VPN if it is a router adjacent to a CE router that is in that VPN.

## VPN Routing and Forwarding Tables

The VPN routing and forwarding table (VRF) is a key element in the MPLS VPN technology. VRFs exist on PEs only (except in the case of a Multi-VRF CE). A VRF is a routing table instance, and more than one VRF can exist on a PE. A VPN can contain one or more VRFs on a PE. The VRF contains routes that should be available to a particular set of sites. VRFs use Cisco Express Forwarding (CEF) technology, therefore the VPN must be CEF-enabled.

A VRF is associated with the following elements:

- IP routing table
- Derived forwarding table, based on the Cisco Express Forwarding (CEF) technology
- A set of interfaces that use the derived forwarding table
- A set of routing protocols and routing peers that inject information into the VRF

Each PE maintains one or more VRFs. Prime Fulfillment software looks up a particular packet's IP destination address in the appropriate VRF only if that packet arrived directly through an interface that is associated with that VRF. The so-called "color" MPLS label tells the destination PE to check the VRF for the appropriate VPN so that it can deliver the packet to the correct CE and finally to the local host machine.

A VRF is named based on the VPN or VPNs it services, and on the role of the CE in the topology. The schemes for the VRF names are as follows:

- The VRF name for a hub: `ip vrf vx:[VPN_name]`
- The `x` parameter is a number assigned to make the VRF name unique.

For example, if we consider a VPN called Blue, then a VRF for a hub CE would be called:

```
ip vrf v1:blue
```

A VRF for a spoke CE in the Blue VPN would be called:

```
ip vrf v1:blue-s
```

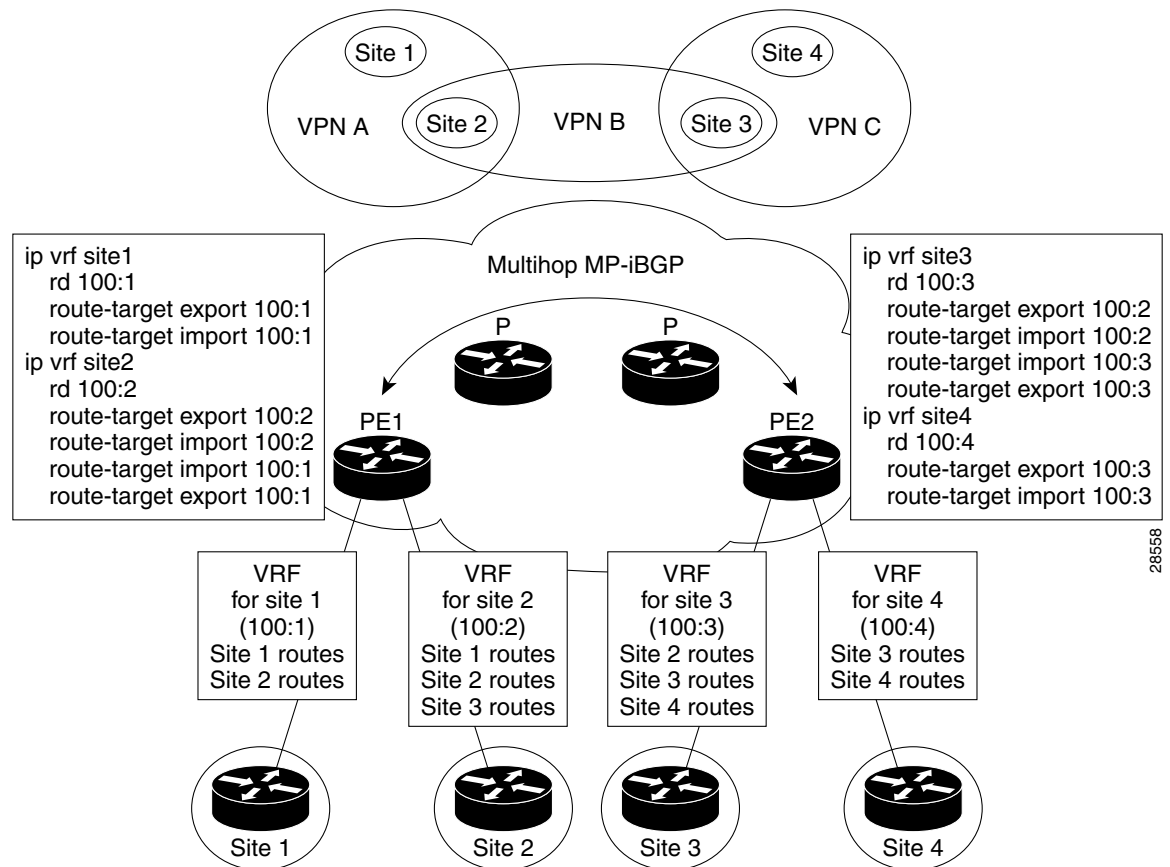
A VRF for an extranet VPN topology in the Green VPN would be called:

```
ip vrf v1:green-etc
```

Thus, you can read the VPN name and the topology type directly from the name of the VRF.

[Figure 2-2](#) shows a network in which two of the four sites are members of two VPNs, and illustrates which routes are included in the VRFs for each site.

Figure 2-2 VRFs for Sites in Multiple VPNs



## VRF Implementation

When implementing VPNs and VRFs, we recommend you keep the following considerations in mind:

- A local VRF interface on a PE is not considered a directly-connected interface in a traditional sense. When you configure, for example, a Fast Ethernet interface on a PE to participate in a particular VRF/VPN, the interface no longer shows up as a directly-connected interface when you issue a **show ip route** command. To see that interface in a routing table, you must issue a **show ip route vrf vrf\_name** command.
- The global routing table and the per-VRF routing table are independent entities. Cisco IOS commands apply to IP routing in a global routing table context. For example, `show ip route`, and other EXEC-level show commands—and utilities such as **ping**, **traceroute**, and **telnet**—all invoke the services of the Cisco IOS routines that deal with the global IP routing table.

- You can issue a standard Telnet command from a CE router to connect to a PE router. However, from that PE, you must issue the following command to connect from the PE to the CE:

```
telnet CE_RouterName /vrf vrf_name
```

Similarly, you can utilize the **Traceroute** and **Ping** commands in a VRF context.

- The MPLS VPN backbone relies on the appropriate Interior Gateway Protocol (IGP) that is configured for MPLS, for example, EIGRP, or OSPF. When you issue a **show ip route** command on a PE, you see the IGP-derived routes connecting the PEs together. Contrast that with the **show ip route vrf VRF\_name** command, which displays routes connecting customer sites in a particular VPN.

## VRF Instance

The configuration commands to create a VRF instance are as follows:

	Command	Description
Step 1	Router# <b>configure terminal</b> Router(config)#	Enter global configuration mode.
Step 2	Router(config)# <b>ip vrf vrf_name</b>	For example, <b>ip vrf CustomerA</b> initiates a VPN routing table and an associated CEF table named CustomerA. The command enters VRF configuration submode to configure the variables associated with the VRF.
Step 3	Router(config-vrf)# <b>rd RD_value</b>	Enter the eight-byte route descriptor (RD) or IP address. The PE prepends the RD to the IPv4 routes prior to redistributing the route into the MPLS VPN backbone.
Step 4	Router(config-vrf)# <b>route-target import   export   both community</b>	Enter the route-target information for the VRF.

## Independent VRF Object Management

Prime Fulfillment allows you to specify VPN and VRF information in an independent VRF object, which is subsequently deployed to a PE device and then associated with an MPLS VPN link via an MPLS VPN service request. For details on using this feature, see to the *Cisco Prime Fulfillment User Guide 6.1*.

## Route Distinguishers and Route Targets

MPLS-based VPNs employ BGP to communicate between PEs to facilitate customer routes. This is made possible through extensions to BGP that carry addresses other than IPv4 addresses. A notable extension is called the *route distinguisher* (RD).

The purpose of the route distinguisher (RD) is to make the prefix value unique across the backbone. Prefixes should use the same RD if they are associated with the same set of route targets (RTs) and anything else that is used to choose routing policy. The community of interest association is based on the route target (RT) extended community attributes distributed with the Network Layer Reachability Information (NLRI). The RD value must be a globally unique value to avoid conflict with other prefixes.

The MPLS label is part of a BGP routing update. The routing update also carries the addressing and reachability information. When the RD is unique across the MPLS VPN network, proper connectivity is established even if different customers use non-unique IP addresses.

For the RD, every CE that has the same overall role should use a VRF with the same name, same RD, and same RT values. The RDs and RTs are *only* for route exchange between the PEs running BGP. That is, for the PEs to do MPLS VPN work, they have to exchange routing information with more fields than usual for IPv4 routes; that extra information includes (but is not limited to) the RDs and RTs.

The route distinguisher values are chosen by the Prime Fulfillment software.

- CEs with hub connectivity use `bgp_AS:value`.
- CEs with spoke connectivity use `bgp_AS:value + 1`

Each spoke uses its own RD value for proper hub and spoke connectivity between CEs; therefore, the Prime Fulfillment software implements a new RD for each spoke that is provisioned.

Prime Fulfillment chooses route target values by default, but you can override the automatically assigned RT values if necessary when you first define a Route Target in the Prime Fulfillment software.

## Route Target Communities

The mechanism by which MPLS VPN controls distribution of VPN routing information is through the VPN route-target extended MP-BGP communities. An extended MP-BGP community is an eight octet structure value. MPLS VPN uses route-target communities as follows:

- When a VPN route is injected into MP-BGP, the route is associated with a list of VPN route-target communities. Typically, this is set through an export list of community values associated with the VRF from which the route was learned.
- An import list of route-target communities is associated with each VRF. This list defines the values that should be matched against to decide whether a route is eligible to be imported into this VRF.

For example, if the import list for a particular VRF is {A, B, C}, then any VPN route that carries community value A, B, or C is imported into the VRF.

## Route Targets

A VPN can be organized into subsets called Route Targets. A Route Target describes how the CEs in a VPN communicate with each other. Thus, Route Targets describe the logical topology of the VPN. Prime Fulfillment can be employed to form a variety of VPN topologies between CEs by building hub and spoke or full mesh Route Targets. Route Targets are building blocks that allow you to form complex VPN topologies and CE connectivity.

The most common types of VPNs are *hub-and-spoke* and *full mesh*.

- A hub-and-spoke Route Target is one in which one or a few CEs act as hubs, and all spoke CEs talk only to or through the hubs, never directly to each other.
- A full mesh Route Target is one in which every CE connects to every other CE.

These two basic types of VPNs—full mesh and hub and spoke—can be represented with a single Route Target.

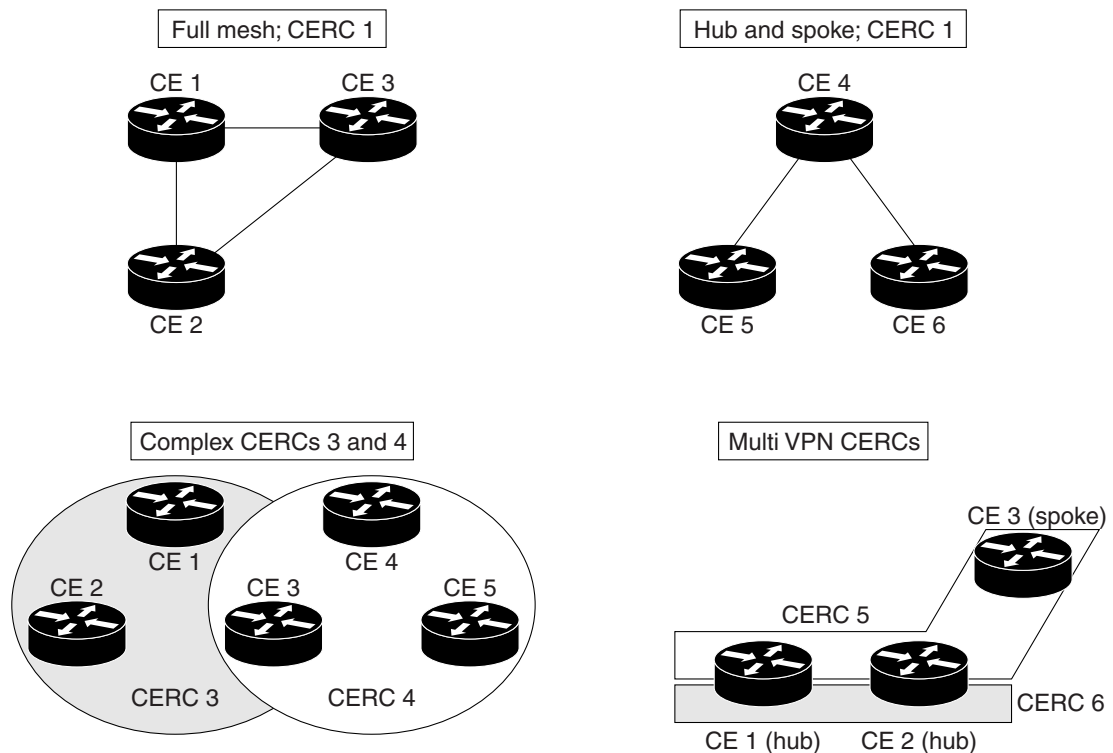
Whenever you create a VPN, the Prime Fulfillment software creates one default Route Target for you. This means that until you need advanced customer layout methods, you will not need to define new Route Targets. Up to that point, you can think of a Route Target as standing for the VPN itself—they are one and the same. If, for any reason, you need to override the software's choice of route target values, you can do so only at the time you create a Route Target in the Prime Fulfillment software.

To build very complex topologies, it is necessary to break down the required connectivity between CEs into groups, where each group is either fully meshed, or has a hub and spoke pattern. (Note that a CE can be in more than one group at a time, so long as each group has one of the two basic patterns.) Each subgroup in the VPN needs its own Route Target. Any CE that is only in one group just joins the corresponding Route Target (as a spoke if necessary). If a CE is in more than one group, then you can use the Advanced Setup choice during provisioning to add the CE to all the relevant groups in one service request. Given this information, the provisioning software does the rest, assigning route target values and VRF tables to arrange exactly the connectivity the customer requires. You can use the Topology tool to double-check the Route Target memberships and resultant VPN connectedness.

Prime Fulfillment supports multiple CEs per site and multiple sites connected to the same PE. Each Route Target has unique route targets (RT), route distinguisher (RD) and VRF naming. After provisioning a Route Target, it is a good idea to run the audit reports to verify the Route Target deployment and view the topologies created by the service requests. The product supports linking two or more Route Targets in the same VPN.

Figure 2-3 shows several examples of the topologies that Prime Fulfillment Route Targets can employ.

**Figure 2-3** Examples of Route Target Topologies



28902

## Hub and Spoke Considerations

In hub-and-spoke MPLS VPN environments, the spoke routers have to have unique Route Distinguishers (RDs). In order to use the hub site as a transit point for connectivity in such an environment, the spoke sites export their routes to the hub. Spokes can talk to hubs, but spokes never have routes to other spokes.

Due to the current MPLS VPN implementation, you must apply a different RD for each spoke VRF. The MP-BGP selection process applies to all the routes that have to be imported into the same VRF plus all routes that have the same RD of such a VRF. Once the selection process is done, only the best routes are imported. In this case this can result in a best route which is not imported. Thus, customers must have different RDs per spoke-VRF.

## Full Mesh Considerations

Each Route Target has two distinct RTs, a hub RT and a spoke RT. When building a full mesh topology, always use the hub RT. Thus, when a need arises to add a spoke site for the current full mesh topology, you can easily add the spoke site without reconfiguring any of the hub sites. The existing spoke RT can be used for this purpose. This is a strategy to prevent having to do significant reprovisioning of a full mesh topology to a hub-and-spoke topology.

## MPLS VPN Security

This section discusses the security requirements for MPLS VPN architectures. This section concentrates on protecting the core network against attacks from the “outside,” that is, the Internet and connected VPNs.



### Note

---

Protection against attacks from the “inside,” that is, when an attacker has logical or physical access to the core network is not discussed here, since any network can be attacked with access from the inside.

---

## Address Space and Routing Separation

Between two non-intersecting VPNs of an MPLS VPN service, it is assumed that the address space between different VPNs is entirely independent. This means, for example, that two non-intersecting VPNs must be able to both use the 10/8 network without any interference. From a routing perspective, this means that each end system in a VPN has a unique address, and all routes to this address point to the same end system. Specifically:

- Any VPN must be able to use the same address space as any other VPN.
- Any VPN must be able to use the same address space as the MPLS core.
- Routing between any two VPNs must be independent.
- Routing between any VPN and the core must be independent.

## Address Space Separation

From a security point of view, the basic requirement is to avoid that packets destined to a host a.b.c.d within a given VPN reach a host with the same address in another VPN or the core.

MPLS allows distinct VPNs to use the same address space, which can also be private address space. This is achieved by adding a 64-bit route distinguisher (RD) to each IPv4 route, making VPN-unique addresses also unique in the MPLS core. This “extended” address is also called a *VPN-IPv4 address*. Thus customers of an MPLS service do not need to change current addressing in their networks.

In the case of using routing protocols between CE and PE routers (for static routing this is not an issue), there is one exception—the IP addresses of the PE routers the CE routers are peering with. To be able to communicate to the PE router, routing protocols on the CE routers must configure the address of the peer router in the core. This address must be unique from the CE router’s perspective. In an environment where the service provider manages also the CE routers as CPE (customer premises equipment), this can be made invisible to the customer.

## Routing Separation

Routing separation between the VPNs can also be achieved. Every PE router maintains a separate Virtual Routing and Forwarding instance (VRF) for each connected VPN. Each VRF on the PE router is populated with routes from one VPN, through statically configured routes or through routing protocols that run between the PE and the CE router. Since every VPN results in a separate VRF, there are no interferences between the VPNs on the PE router.

Across the MPLS core to the other PE routers, this routing separation is maintained by adding unique VPN identifiers in multi-protocol BGP, such as the route distinguisher (RD). VPN routes are exclusively exchanged by MP-BGP across the core, and this BGP information is not redistributed to the core network, but only to the other PE routers, where the information is kept again in VPN-specific VRFs. Thus routing across an MPLS network is separate per VPN.

Given addressing and routing separation across an MPLS core network, MPLS offers in this respect the same security as comparable Layer 2 VPNs, such as ATM or Frame Relay. It is not possible to intrude into other VPNs through the MPLS core, unless this has been configured specifically.

## Hiding the MPLS Core Structure

The internal structure of the MPLS core network (PE and Provider router devices) should not be visible to outside networks (either the Internet or any connected VPN). While a breach of this requirement does not lead to a security problem itself, it is generally advantageous when the internal addressing and network structure remains hidden to the outside world. The ideal is to not reveal any information of the internal network to the outside. This applies equally to the customer networks as to the MPLS core.

Denial-of-service attacks against a core router, for example, are much easier to carry out if an attacker knows the IP address. Where addresses are not known, they can be guessed, but when the MPLS core structure is hidden, attacks are more difficult to make. Ideally, the MPLS core should be as invisible to the outside world as a comparable Layer 2 infrastructure (for example, Frame Relay or ATM).

In practice, a number of additional security measures have to be taken, most of all *extensive packet filtering*. MPLS does not reveal unnecessary information to the outside, not even to customer VPNs. The addressing in the core can be done with either private addresses or public addresses. Since the interface to the VPNs, and potentially to the Internet, is BGP, there is no need to reveal any internal information. The only information required in the case of a routing protocol between a PE and CE is the address of the PE router. If this is not desired, you can configure static routing between the PE and CE. With this measure, the MPLS core can be kept completely hidden.

To ensure reachability across the MPLS cloud, customer VPNs will have to advertise their routes as a minimum to the MPLS core. While this could be seen as too open, the information known to the MPLS core is not about specific hosts, but networks (routes); this offers some degree of abstraction. Also, in a VPN-only MPLS network (that is, no shared Internet access), this is equal to existing Layer 2 models, where the customer has to trust the service provider to some degree. Also in a Frame Relay or ATM network, routing information about the VPNs can be seen on the core network.

In a VPN service with shared Internet access, the service provider typically announces the routes of customers that want to use the Internet to his upstream or peer providers.

In summary, in a pure MPLS VPN service, where no Internet access is provided, the level of information hiding is as good as on a comparable Frame Relay or ATM network—no addressing information is revealed to third parties or the Internet. If a customer chooses to access the Internet by way of the MPLS core, he will have to reveal the same addressing structure as for a normal Internet service.

If an MPLS network has no interconnections to the Internet, this is equal to Frame Relay or ATM networks. With Internet access from the MPLS cloud, the service provider has to reveal at least one IP address (of the peering PE router) to the next provider, and thus the outside world.

## Resistance to Attacks

It is not possible to directly intrude into other VPNs. However, it is possible to attack the MPLS core, and try to attack other VPNs from there. There are two basic ways the MPLS core can be attacked:

- Attacking the PE routers directly.
- Attacking the signaling mechanisms of MPLS (mostly routing)

There are two basic types of attacks: *denial-of-service (DoS) attacks*, where resources become unavailable to authorized users, and *intrusion attacks*, where the goal is to gain unauthorized access to resources.

For intrusion attacks, give unauthorized access to resources, there are two basic ways to protect the network:

- Harden protocols that could be abused (for example, Telnet to a router)
- Make the network as inaccessible as possible. This is achieved by a combination of filtering packets and hiding the IP addresses in the MPLS core.

Denial-of service attacks are easier to execute, since in the simplest case, a known IP address might be enough to attack a machine. The only way to be certain that you are not be vulnerable to this kind of attack is to make sure that machines are not reachable, again by packet filtering and pinging IP addresses.

MPLS networks must provide at least the same level of protection against both forms of attack as current Layer 2 networks provide.

To attack an element of an MPLS network it is first necessary to know this element, that is, its IP address. It is possible to hide the addressing structure of the MPLS core to the outside world, as discussed in the previous section. Thus, an attacker does not know the IP address of any router in the core that he wants to attack. The attacker could guess addresses and send packets to these addresses. However, due to the address separation of MPLS, each incoming packet is treated as belonging to the address space of the customer. It is therefore impossible to reach an internal router, even through guessing the IP addresses. There is only one exception to this rule—the peer interface of the PE router.



## Securing the Routing Protocol

The routing between the VPN and the MPLS core can be configured two ways:

1. **Static.** In this case, the PE routers are configured with static routes to the networks behind each CE, and the CEs are configured to statically point to the PE router for any network in other parts of the VPN (usually a default route).

The static route can point to the IP address of the PE router, or to an interface of the CE router (for example, serial0).

Although in the static case the CE router does not know any IP addresses of the PE router, it is still attached to the PE router by way of some method, and could guess the address of the PE router and try to attack it with this address.

In the case of a static route from the CE router to the PE router, which points to an interface, the CE router does not need to know any IP address of the core network, not even of the PE router. This has the disadvantage of a more extensive (static) configuration, but from a security point of view, it is preferable to the other cases.

2. **Dynamic.** A routing protocol (for example, RIP, OSPF, or BGP) is used to exchange the routing information between the CE and the PE at each peering point.

In all other cases, each CE router needs to know at least the router ID (RID; peer IP address) of the PE router in the MPLS core, and thus has a potential destination for an attack.

In practice, access to the PE router over the CE-PE interface can be limited to the required routing protocol by using access control lists (ACLs). This limits the point of attack to one routing protocol, for example BGP. A potential attack could send an extensive number of routes, or flood the PE router with routing updates. Both of these attacks could lead to a denial-of-service attack, however, not to an intrusion attack.

To restrict this risk it is necessary to configure the routing protocol on the PE router as securely as possible. This can be done in various ways:

- Use VRFs. There are mechanisms within the context of a VRF for a service provider to monitor and control the number of routes that a customer can have in the VPN. When such thresholds are breached, for example 80 percent of the allowed number of routes syslog messages can be generated indicating to the service provider that the VRF is reaching the allowed limit.
- Use ACLs. Allow the routing protocol only from the CE router, not from anywhere else. Furthermore, no access other than that should be allowed to the PE router in the inbound ACL on each PE interface.

ACLs must be configured to limit access only to the port(s) of the routing protocol, and only from the CE router.

- Where available, configure MD-5 authentication for routing protocols.

This is available for BGP, OSPF, and RIP2. It avoids the possibility that packets could be spoofed from other parts of the customer network than the CE router. This requires that the service provider and customer agree on a shared secret between all CE and PE routers. The problem here is that it is necessary to do this for all VPN customers; it is not sufficient to do this only for the customer with the highest security requirements.

**Note**

Prime Fulfillment does not provide for the provisioning of MD5 authentication on PE-CE links using routing protocols. The VPN customer and the service provider must manually configure this.

MD5 authentication in routing protocols should be used on all PE-CE peers. It is easy to track the source of such a potential denial-of-service attack.

- Configure, where available, the parameters of the routing protocol to further secure this communication.

In BGP, for example, it is possible to configure *dampening*, which limits the number of routing interactions. Also, a maximum number of routes accepted per VRF should be configured where possible.

In summary, it is not possible to intrude from one VPN into other VPNs or the core. However, it is theoretically possible to exploit the routing protocol to execute a denial-of-service attack against the PE router. This in turn might have negative impact on other VPNs. For this reason, PE routers must be extremely well secured, especially on their interfaces to the CE routers.

## Label Spoofing

Assuming the address and routing separation as discussed above, a potential attacker might try to gain access to other VPNs by inserting packets with a label that he does not own. This is called *label spoofing*. This kind of attack can be done from the outside, that is, another CE router or from the Internet, or from within the MPLS core. The latter case (from within the core) is not discussed since the assumption is that the core network is provided in a secure manner.

Within the MPLS network, packets are not forwarded based on the IP destination address, but based on the labels that are prepended by the PE routers. Similar to IP spoofing attacks, where an attacker replaces the source or destination IP address of a packet, it is also possible to spoof the label of an MPLS packet.

The interface between any CE router and its peering PE router is an IP interface, that is, without labels. The CE router is unaware of the MPLS core, and is only aware of the destination router. The intelligence exits in the PE device, where based on the configuration, the PE chooses a label and prepends it to the packet. This is the case for all PE routers, toward CE routers, and to the upstream service provider. All interfaces into the MPLS cloud require IP packets without labels.

For security reasons, a PE router should never accept a packet with a label from a CE router. Cisco routers implementation is such that packets that arrive on a CE interface with a label are dropped. Thus, it is not possible to insert fake labels because no labels are accepted. Additional security can be implemented by using MD5 authentication between peer routers in the core if the service provider is using LDP to distribute labels.

There remains the possibility to spoof the IP address of a packet that is being sent to the MPLS core. However, since there is strict addressing separation within the PE router, and each VPN has its own VRF, this can only do harm to the VPN the spoofed packet originated from, in other words, a VPN customer can attack himself. MPLS does not add any security risk here.

## Securing the MPLS Core

The following is a list of recommendations and considerations on configuring an MPLS network securely.



### Note

The security of the overall solution depends on the security of its weakest link. This could be the weakest single interconnection between a PE and a CE, an insecure access server, or an insecure TFTP server.

## Trusted Devices

The PE and P devices, and remote access servers and AAA servers must be treated as trusted systems. This requires strong security management, starting with physical building security and including issues such as access control, secure configuration management, and storage. There is ample literature available on how to secure network elements, so these topics are not discussed here in more detail.

CE routers are typically not under full control of the service provider and must be treated as “untrusted.”

## PE-CE Interface

The interface between PE and CE routers is crucial for a secure MPLS network. The PE router should be configured as close as possible. From a security point of view, the best option is to configure the interface to the CE router unnumbered and route statically.

Packet filters (Access Control Lists) should be configured to permit only one specific routing protocol to the peering interface of the PE router, and only from the CE router. All other traffic to the router and the internal service provider network should be denied. This avoids the possibility that the PE and P routers can be attacked, since all packets to the corresponding address range are dropped by the PE router. The only exception is the peer interface on the PE router for routing purposes. This PE peer interface must be secured separately.

If private address space is used for the PE and P routers, the same rules with regard to packet filtering apply—it is required to filter all packets to this range. However, since addresses of this range should not be routed over the Internet, it limits attacks to adjacent networks.

## Routing Authentication

All routing protocols should be configured with the corresponding authentication option toward the CEs and toward any Internet connection. Specifically: BGP, OSPF, and RIP2. All peering relationships in the network need to be secured this way:

- CE-PE link: use BGP MD-5 authentication
- PE-P link: use LDP MD5 authentication
- P-P

This prevents attackers from spoofing a peer router and introducing bogus routing information. Secure management is particularly important regarding configuration files, which often contain shared secrets in clear text (for example for routing protocol authentication).

## Separation of CE-PE Links

If several CEs share a common Layer 2 infrastructure to access the same PE router (for example, an ethernet VLAN), a CE router can spoof packets as belonging to another VPN that also has a connection to this PE router. Securing the routing protocol is not sufficient, since this does not affect normal packets.

To avoid this problem, we recommend that you implement separate physical connections between CEs and PEs. The use of a switch between various CE routers and a PE router is also possible, but it is strongly recommended to put each CE-PE pair into a separate VLAN to provide traffic separation. Although switches with VLANs increase security, they are not unbreakable. A switch in this environment must thus be treated as a trusted device and configured with maximum security.

## LDP Authentication

The Label Distribution Protocol (LDP) can also be secured with MD-5 authentication across the MPLS cloud. This prevents hackers from introducing bogus routers, which would participate in the LDP.

## Connectivity Between VPNs

MPLS provides VPN services with address and routing separation between VPNs. In many environments, however, the devices in the VPN must be able to reach destinations outside the VPN. This could be for Internet access or for merging two VPNs, for example, in the case of two companies merging. MPLS not only provides full VPN separation, but also allows merging VPNs and accessing the Internet.

To achieve this, the PE routers maintain various tables: A *routing context table* is specific to a CE router, and contains only routes from this particular VPN. From there, routes are propagated into the *VRF* (virtual routing and forwarding instance) *routing table*, from which a *VRF forwarding table* is calculated.

For separated VPNs, the VRF routing table contains only routes from one routing context. To merge VPNs, different routing contexts (from different VPNs) are put into one single VRF routing table. In this way, two or several VPNs can be merged to a single VPN. In this case, it is necessary that all merged VPNs have mutually exclusive addressing spaces; in other words, the overall address space must be unique for all included VPNs.

For a VPN to have Internet connectivity, the same procedure is used: Routes from the Internet VRF routing table (the default routing table) are propagated into the VRF routing table of the VPN that requires Internet access. Alternatively to propagating all Internet routes, a default route can be propagated. In this case, the address space between the VPN and the Internet must be distinct. The VPN must use private address space since all other addresses can occur in the Internet.

From a security point of view, the merged VPNs behave like one logical VPN, and the security mechanisms described above apply now between the merged VPN and other VPNs. The merged VPN must have unique address space internally, but further VPNs can use the same address space without interference. Packets from and to the merged VPNs cannot be routed to other VPNs. All the separation functions of MPLS apply also for merged VPNs with respect to other VPNs.

If two VPNs are merged in this way, hosts from either part can reach the other part as if the two VPNs were a common VPN. With the standard MPLS features, there is no separation or firewalling or packet filtering between the merged VPNs. Also, if a VPN receives Internet routes through MPLS/BGP VPN mechanisms, firewalling or packet filtering has to be engineered in addition to the MPLS features.

## MP-BGP Security Features

Security in Prime Fulfillment MPLS-based networks is delivered through a combination of MP-BGP and IP address resolution. In addition, service providers can ensure that VPNs are isolated from each other.

Multiprotocol BGP is a routing information distribution protocol that, through employing multiprotocol extensions and community attributes, defines who can talk to whom. VPN membership depends upon logical ports entering the VPN, where MP-BGP assigns a unique Route Distinguisher (RD) value (see [Route Distinguishers and Route Targets, page 2-5](#)).

RDs are unknown to end users, making it impossible to enter the network on another access port and spoof a flow. Only preassigned ports are allowed to participate in the VPN. In an MPLS VPN, MP-BGP distributes forwarding information base (FIB) tables about VPNs to members of the same VPN only, providing native security by way of logical VPN traffic separation. Furthermore, iBGP PE routing peers

can perform TCP segment protection using the MD5 Signature Option when establishing iBGP peering relationships, further reducing the likelihood of introducing spoofed TCP segments into the iBGP connection stream among PE routers (for information on the MD5 Signature Option, see RFC 2385).

The service provider, not the customer, associates a specific VPN with each interface when provisioning the VPN. Users can only participate in an intranet or extranet if they reside on the correct physical or logical port and have the proper RD. This setup makes a Cisco MPLS VPN virtually impossible to enter.

Within the core, a standard Interior Gateway Protocol (IGP) such as OSPF or IS-IS distributes routing information. Provider edge routers set up paths among one another using LDP to communicate label-binding information. Label binding information for external (customer) routes is distributed among PE routers using MP-BGP multiprotocol extensions instead of LDP, because they easily attach to VPN IP information already being distributed.

The MP-BGP community attribute constrains the scope of reachability information. MP-BGP maps FIB tables to provider edge routers belonging to only a particular VPN, instead of updating all edge routers in the service provider network.

## Security Through IP Address Resolution

MPLS VPN networks are easier to integrate with IP-based customer networks. Subscribers can seamlessly interconnect with a provider service without changing their intranet applications because MPLS-based networks have built-in application awareness. Customers can even transparently use their existing IP address space because each VPN has a unique identifier.

MPLS VPNs remain unaware of one another. Traffic is separated among VPNs using a logically distinct forwarding table and RD for each VPN. Based on the incoming interface, the PE selects a specific forwarding table, which lists only valid destinations in the VPN. To create extranets, a provider explicitly configures reachability among VPNs.

The forwarding table for a PE contains only address entries for members of the same VPN. The PE rejects requests for addresses not listed in its forwarding table. By implementing a logically separate forwarding table for each VPN, each VPN itself becomes a private, connectionless network built on a shared infrastructure.

IP limits the size of an address to 32 bits in the packet header. The VPN IP address adds 64 bits in front of the header, creating an extended address in routing tables that classical IP cannot forward. The extra 64 bits are defined by the Route Distinguisher and the resultant route becomes a unique 96-bit prefix. MPLS solves this problem by forwarding traffic based on labels, so one can use MPLS to bind VPN IP routes to label-switched paths. PEs are concerned with reading labels, not packet headers. MPLS manages forwarding through the provider's MPLS core. Since labels only exist for valid destinations, this is how MPLS delivers both security and scalability.

When a virtual circuit is provided using the overlay model, the egress interface for any particular data packet is a function solely of the packet's ingress interface; the IP destination address of the packet does not determine its path in the backbone network. Thus, unauthorized communication into or out of a VPN is prevented.

In MPLS VPNs, a packet received by the backbone is first associated with a particular VPN by stipulating that all packets received on a certain interface (or subinterface) belong to a certain VPN. Then its IP address is looked up in the forwarding table associated with that VPN. The routes in that forwarding table are specific to the VPN of the received packet.

In this way, the ingress interface determines a set of possible egress interfaces, and the packet's IP destination address is used to choose from among that set. This prevents unauthorized communication into and out of a VPN.

## Ensuring VPN Isolation

To maintain proper isolation of one VPN from another, it is important that the provider routers not accept a labeled packet from any adjacent PE unless the following conditions are met:

- The label at the top of the label stack was actually distributed by the provider router to the PE device.
- The provider router can determine that use of that label will cause the packet to exit the backbone before any labels lower in the stack and the IP header will be inspected.

These restrictions are necessary to prevent packets from entering a VPN where they do not belong.

The VRF tables in a PE are used only for packets arriving from a CE that is directly attached to the PE device. They are not used for routing packets arriving from other routers that belong to the service provider backbone. As a result, there might be multiple different routes to the same system, where the route followed by a given packet is determined by the site from which the packet enters the backbone. So one might have one route to a given IP network for packets from the extranet (where the route leads to a firewall), and a different route to the same network for packets from the intranet.



## CHAPTER 3

# Traffic Engineering Management Concepts

---

This chapter includes an overview of Cisco Prime Fulfillment and of some of the concepts used in this guide. This chapter includes the following sections:

- [Prime Fulfillment TEM Overview, page 3-1](#)
- [Features in Prime Fulfillment, page 3-2](#)
- [Prime Fulfillment TEM Basics, page 3-2](#)
  - [Managed/Unmanaged Primary Tunnels, page 3-2](#)
  - [Conformant/Non-Conformant Tunnels, page 3-3](#)
  - [Multiple Concurrent Users, page 3-4](#)
  - [Multiple OSPF Areas, page 3-5](#)
  - [Bandwidth Pools, page 3-6](#)
  - [Planning Tools, page 3-7](#)
  - [Connectivity Protection \(CSPF\) Backup Tunnels, page 3-8](#)
  - [Class-Based Tunnel Selection, page 3-8](#)
  - [Policy-Based Tunnel Selection, page 3-9.](#)

## Prime Fulfillment TEM Overview

TEM is the Traffic Engineering Management module of Prime Fulfillment. It is a tool for managing Multiprotocol Label Switching Traffic Engineering (MPLS TE) primary tunnels and backup tunnels for the purpose of offering traffic Service Level Agreement (SLA) guarantees. It provides bandwidth protection management, network discovery, and support for configuring MPLS TE. It includes a number of powerful planning tools, including a sophisticated primary path calculation tool and backup tunnel calculation for element protection.

MPLS TE mechanisms are provided to support requirements for predictability, traffic flow matched to QoS requirements, and Fast Restoration with Guaranteed Bandwidth, ensuring that strict SLA performance criteria (availability, delay, jitter) are met.

## Features in Prime Fulfillment

Prime Fulfillment adds a range of MPLS TE primary tunnel management features:

- Tunnel Audit—finding inconsistencies after making tunnel modifications
- Tunnel Admission—admitting new tunnels onto the network
- Tunnel Repair—fixing tunnel inconsistencies after network and service changes
- Network Grooming—optimizing global network utilization.

In addition, Prime Fulfillment offers interaction and integration with Prime Fulfillment features:

- Service activation focus
- Integration with other Prime Fulfillment modules
- Data Persistence
- Logging of user intent
- Service state management
- Service auditing
- Web-based GUI
- Role-Based Access Control (RBAC).

## Prime Fulfillment TEM Basics

To understand how Prime Fulfillment works, certain key concepts must be explained.

### Managed/Unmanaged Primary Tunnels

In Prime Fulfillment, the concept of managed tunnels is at the center of TE planning activities.

It is important to understand the differences:

- Managed TE tunnels:
  - (setup/hold) priority zero
  - non-zero RSVP bandwidth
  - explicit first path option
  - auto bandwidth must have a max value
- Unmanaged tunnels: All other tunnels.

In the Prime Fulfillment Graphical User Interface (GUI), there is a separate entry point for dealing with managed and unmanaged tunnels.



## Conformant/Non-Conformant Tunnels

Understanding the concepts of conformant and non-conformant tunnels is key to making the most efficient use of Prime Fulfillment.

Prime Fulfillment only allows the creation of conformant tunnels. Non-conformant tunnels can be introduced through the TE Discovery process (see [Chapter 36, “TE Network Discovery”](#) of the User Guide).

### Defining Conformant/Non-Conformant Tunnels

In the Prime Fulfillment design, a sharp distinction has been made between conformant and non-conformant tunnels:

- Conformant tunnel—A well-behaved tunnel that meets Prime Fulfillment’s TE management paradigm (described below). A managed tunnel can only be a conformant tunnel. A non-zero priority unmanaged tunnel would also be a conformant tunnel. However, a conformant tunnel is not necessarily a managed tunnel.

A connectivity protection tunnel is marked Conformant = true if it has zero tunnel bandwidth, unlimited backup bandwidth, and an 'exclude address' first path option. For the BW Protected setting, a tunnel should have a defined non-zero backup bandwidth, and a strict path option 1.

- Non-conformant tunnel—A TE tunnel, which might impact Prime Fulfillment’s ability to meet bandwidth guarantees. This could be due to unknown bandwidth requirements such as no max bandwidth configured for auto-bandwidth, potential for pre-emption, dynamic paths, etc. A zero priority unmanaged tunnel would also be a non-conformant tunnel.

The following are examples of non-conformant tunnels:

- a tunnel with zero setup and hold priority, an explicit first path option, but with zero bandwidth;
- a tunnel with zero setup and hold priority, a non zero bandwidth, but with a dynamic first path option;
- a tunnel with zero setup and hold priority, an explicit path option of 1 and an auto bandwidth without a maximum defined.;
- a connectivity protection tunnel marked Conformant = false is reserved for backup tunnels, which have neither zero tunnel bandwidth, unlimited backup bandwidth, or an 'exclude address' first path option.

Why are the above tunnels non-conformant? Because Prime Fulfillment attempts to manage all tunnels with zero setup and hold priority, to ensure the links they pass through all have sufficient bandwidth, are affinity consistent, and do not break delay or FRR constraints defined in the TE policy.

But if the tunnel’s path is dynamic or the amount of bandwidth it requires is undefined, Prime Fulfillment does not have the information with which to manage the tunnel, so it marks it as non-conformant. All the non-conformant tunnels are displayed in the TE Unmanaged Primary Tunnels SR window.

## Managing Non-Conformant Tunnels

It is important to understand that non-conformant tunnels not only might cause the SLAs to be violated, they might also have an adverse effect on the managed tunnels (taking away bandwidth from them, for example).

However, when a non-conformant tunnel is discovered, a warning is logged. Prime Fulfillment tracks non-conformant tunnels so that they can be decommissioned.

So conformant tunnels are preferred. They allow the system to offer bandwidth guarantees for managed tunnels. Unmanaged non-conformant tunnels might or might not provide the needed bandwidth and no bandwidth guarantees are given.

The action to take when you have non-conformant tunnels is either to change the setup and hold priorities to non-zero values (so they cannot preempt the managed tunnels) or migrate them to managed tunnels, allowing the tool to find a suitable explicit path.

## Multiple Concurrent Users

In previous releases TEM only supported a single GUI user. This release introduces support for multiple concurrent users, for all browsing, updating, and provisioning operations.

### Concurrent Use with Managed and Unmanaged Tunnels

To understand how the multiple user feature is implemented in TEM, it is important to understand the difference between a managed and an unmanaged tunnel. This is described in the Managed/Unmanaged Primary Tunnels section on page F-2.

There are important differences between how managed and unmanaged tunnels are handled when it comes to multiple user support:

- For managed tunnels, an SR encapsulates all managed tunnels. A SR operation might optimize all the objects within the snapshot following path computations performed by the Router Generator server.
- For unmanaged tunnels, an SR is defined as a tunnel-head end router. Thus, with unmanaged tunnels there are certain restrictions. For example, two users cannot concurrently provision on the same device.
- TEM prevents Unmanaged Tunnel SRs from provisioning concurrently on the same device but supports Unmanaged Tunnel SRs provisioning concurrently on different devices.
- All managed tunnels are contained within a shared Managed TE Tunnel SR for each TE Provider. For unmanaged tunnels, a distinct Unmanaged TE Tunnel Service Request is created per head device. TEM supports multiple SRs per TE Provider.

Multiple TEM users can browse and provision in TEM. Up to 20 concurrent users are supported, of which up to seven can perform provisioning tasks.

Previously all primary tunnels, managed and unmanaged were in a single TE tunnel SR per TE provider. Now, to facilitate multiple simultaneous changes to managed tunnels, the TE Tunnel SR has been split into one managed tunnel SR per TE provider and one unmanaged tunnel SR per head TE router.

Parallel provisioning is not possible on the same SR, but because SRs exist at router level for unmanaged tunnels, unmanaged tunnels can be provisioned on separate routers at the same time.

## Locking Mechanism

When an unmanaged tunnel is provisioned, the head TE router of the tunnel is locked. This can be seen on the TE Nodes window in the System Lock Status column. The locking prevents any other user from deploying any kind of tunnel to that router until the provisioning task completes and the TE router is unlocked.

The locking mechanism also applies to other Prime Fulfillment features, such as backup tunnels, resource SRs, link deletion, and TE traffic admission. Resource SRs include deleting/editing explicit paths, deleting protected elements, deleting/editing SRLG's, etc.

In the case of link deletion, a level of intelligence is built in. When there are no more tunnels to be rerouted or deleted by the user or Prime Fulfillment and left with TE associated objects alone, a user intervention will be required to carry out link deletion. As part of this deletion, if there are any backup tunnels protecting any interfaces that have been selected for deletion, the locking mechanism will be in place during deployment of backup tunnels. For further information about deleting TE links, see [Deleting TE Links, page 37-5](#) of the User Guide.

Some of the potential errors you might encounter are described in [Locking Operation Errors, page 42-10](#) of the User Guide.

When a managed primary tunnel or a backup tunnel is provisioned, the TE provider it is associated with is locked. This can be seen on the TE Provider window in the System Lock Status column. A lock at the TE provider level prevents another user from making any tunnel change on this TE provider, irrespective of which TE router the tunnel starts at.

The reason why the locking mechanism of managed tunnels and backup tunnels is different from that of unmanaged tunnel is that the managed tunnels and backup tunnels use a path generation algorithm to find an optimal route for the tunnel that fulfills all constraints, and this algorithm needs a stable global view of the TE topology and all the tunnels in it on which to base its routing decisions. This can only be achieved by allowing only one user to make changes at one time.

For more information about how to manage Prime Fulfillment locking mechanism, see [Managing the Locking Mechanism, page 42-9](#) of the User Guide.

## Multiple OSPF Areas

Prime Fulfillment supports the discovery, management, and provisioning of TE Tunnels within multiple Open Shortest Path First (OSPF) areas.

Prime Fulfillment only manages primary and backup TE tunnels within the scope of an OSPF area. There is no support for the discovery and creation of inter-OSPF areas.

In Prime Fulfillment, an OSPF area is represented by a TE provider. After an area is assigned to a TE provider, it might not be changed. Multiple TE providers can be associated with one Prime Fulfillment provider.

## Devices Suitable for TE Discovery

In a network with multiple OSPF areas, where each OSPF area is represented by a TE provider, any router in an OSPF area can be used for TE Discovery. Using multiple TE providers (multiple OSPF areas) under one provider allows the provisioning of inter-area L3VPN.

**Note**

Prime Fulfillment will not discover or provision inter-area TE tunnels (those with a head router in one area and a tail router in a different area).

To discover a multi area network, you have to discover each area in turn using TE Discovery (see [Chapter 36, “TE Network Discovery”](#) of the User Guide). The seed node can be any device within that area, including an Area Border Router (ABR).

## TE Discovery and the TE Area Identifier

TE Discovery is associated with a TE provider and each TE provider is assigned an area. The area is assigned during the process of creating the TE Provider (see [Creating a TE Provider, page 35-8](#) of the User Guide) and can be a simple integer value or dotted decimal notation, Area 0.6.0.0 for example.

TE provider objects are aware of which area they are responsible for, either specified on creation or automatically populated during discovery, and will accommodate conversion between Dot notation and Decimal notation, defaulting to the notation used in the network.

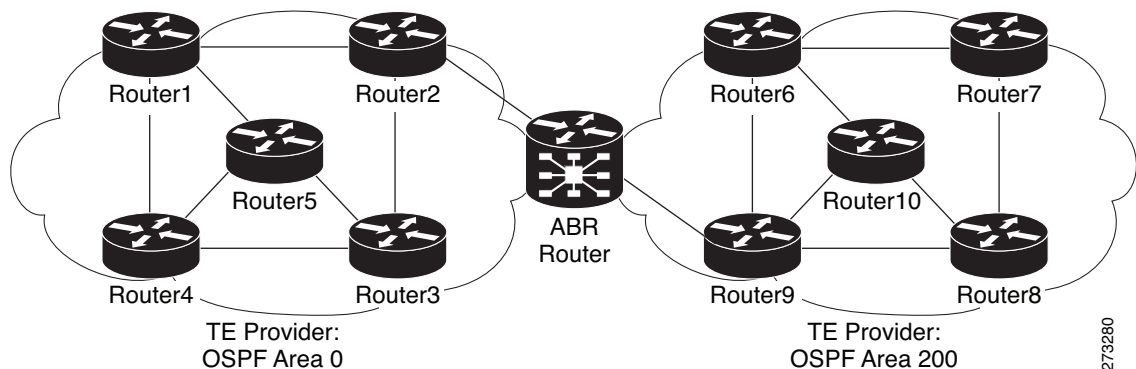
When discovery is run against an area with a selected TE provider, all tunnels and explicit paths associated with that area will be imported into the Prime Fulfillment database. The steps for performing a per area discovery are documented in the [Managing Per Area Discovery, page 36-5](#) of the User Guide.

## Example of Multiple OSPF Area Network

TE routers within a TE provider can be assigned to different regions, for example on a geographical basis, so that devices are grouped in regions in a logical way. Also, Prime Fulfillment allows you to filter by region. Assigning objects to specific regions is a manual task that is carried out after discovery from **Inventory > Provider Devices**. Here the region of any PE device can be changed via the Select Region pop-up window.

In the following example [Figure 3-1](#), two TE providers are each responsible for one OSPF area that is created and visualized under one Prime Fulfillment provider.

**Figure 3-1 Multiple OSPF Areas Network Diagram**



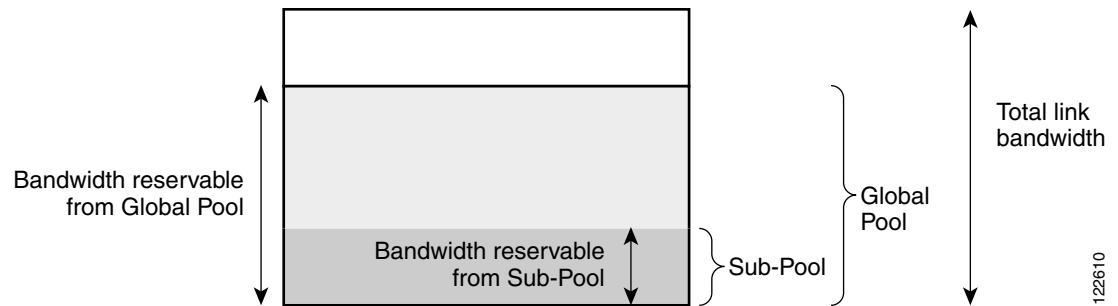
For instructions on how to manage TE providers, see [Creating a TE Provider, page 35-8](#) of the User Guide.

## Bandwidth Pools

The bandwidth of each TE enabled interface is assigned a number of nested bandwidth pools. Currently, IOS supports two, namely Global Pool and Sub Pool.

For a better understanding of bandwidth pools, see [Figure 3-2](#).

**Figure 3-2** Bandwidth Pools



As [Figure 3-2](#) illustrates, Sub Pool is nested inside Global Pool. Thus, if a primary tunnel reserves bandwidth from the Sub Pool, it will also reserve the same bandwidth from the Global Pool.

Bandwidth reservations (primary tunnels) from the Sub Pool must not exceed, in total, the Sub Pool size. Likewise, bandwidth reservations from the Global Pool must not exceed, in total, the Global Pool size.

## Planning Tools

They are intended for evaluating planned improvements to a traffic-engineered network based on What-If scenarios.

The planning tools include the following features:

- Primary planning tools:
  - Tunnel Audit—Audits for inconsistencies in primary placement on the existing network with or without proposed tunnel or resource changes.
  - Tunnel Placement—Usually for new tunnels. Tunnel Placement can generate a new route. It can be used for a tunnel that did not have a path before and needs to be placed.
  - Tunnel Repair—Logically performed after Tunnel Audit (if something is wrong). Tunnel Repair has rerouting capabilities and can be used to move tunnels.
  - Grooming—An optimization tool that works on the whole network. It is only available when no tunnel attributes have been changed.
- Protection planning tools:
  - Audit SR—Audits protection for manually added, modified, and deleted backup tunnels before they are deployed.
  - Compute Backup—Automatically calculates the optimal backup tunnel for selected network elements.
  - Audit Protection—Audits protection of the selected elements against the existing backup tunnels.

The planning tools are fully integrated within Prime Fulfillment and are available from various locations within the GUI:

- TE Protected Elements (Compute Backup and Audit Protection)
- Create Managed TE Tunnel (Tunnel Audit, Tunnel Placement, Tunnel Repair, Grooming)

- Create TE Backup Tunnel (Audit SR).

## Connectivity Protection (CSPF) Backup Tunnels

In addition to the bandwidth-protected backup tunnels created by TEM, you can create a set of CSPF-routed backup tunnels within Prime Fulfillment. These CSPF-routed backup tunnels are managed from the TE Protection SR window.

A connectivity protection backup tunnel uses an “exclude-address” explicit path. This explicit path is created in the TE Explicit Path List window. An exclude address path is different from a strict path in that instead of listing the hops the path should use, it lists the hops the path should avoid. The CSPF algorithm on the router will make the decision as to which precise path to use, but it will be constrained to not be able to use the hops in the exclude address path configuration. This sort of path is particularly useful for backup tunnels, as the interfaces the exclude address path should avoid can be the interfaces that the backup tunnel is protecting.

In Prime Fulfillment, these backup tunnels are configured with unlimited backup bandwidth. Unlimited means no bandwidth is guaranteed, but as much as is available at the time of the failure will be used. So in effect the bandwidth protection is best effort but the connectivity is guaranteed. Connectivity protection backup tunnels can be used in addition to or instead of bandwidth protection backup tunnels.

Differences between bandwidth protection and connectivity protection backup tunnels:

- A bandwidth protection backup tunnel has a strict explicit path as its first path option, whilst a connectivity protection tunnel has an exclude address explicit path as its first path option.
- A bandwidth protection backup tunnel has a defined backup bandwidth whilst a connectivity tunnel has unlimited backup bandwidth on a best effort basis.
- A bandwidth protection backup tunnel is passed to the Route Generator algorithm which generates optimal backup tunnels and verifies existing tunnels fully protect the elements, whereas connectivity protection tunnels are not passed to the algorithm and it is up to you to ensure they are fulfilling their purpose.

## Class-Based Tunnel Selection

Multi-Protocol Label Switching Traffic Engineering Class-Based Tunnel Selection (CBTS) enables you to dynamically route and forward traffic with different class of service (CoS) values onto different TE tunnels between the same tunnel head end and the same tail end. The packet’s CoS values are located in the EXP bits. There are 8 EXP bits, numbered 0 to 7.

The set of TE (or DS-TE) tunnels from the same head end to the same tail end can be configured to carry different CoS values. After configuration, CBTS dynamically routes and forwards each packet into the tunnel that:

- is selected for traffic admission using the standard autoroute or static route mechanisms, and
- has EXP bits matching that of the packet.

Thus CBTS is not a form of traffic admission to TE tunnels directly, is it rather an additional criteria that traffic must satisfy before being admitted to tunnels via the autoroute or static route mechanisms that TEM supports.

Because CBTS offers dynamic routing over DS-TE tunnels and requires minimum configuration, it greatly eases deployment of DS-TE in large-scale networks. CBTS can distribute all CoS values onto many different tunnels.

The CBTS feature has the following restrictions:

- For a given destination, all CoS values are carried in tunnels terminating at the same tail end. Either all CoS values are carried in tunnels or no values are carried in tunnels. In other words, for a given destination, you cannot map some CoS values in a DS-TE tunnel and other CoS values in a Shortest Path First (SPF) Label Distribution Protocol (LDP) or SPF IP path.
- CBTS does not allow load-balancing of a given EXP value in multiple tunnels. If two or more tunnels are configured to carry a given experimental (EXP) value, CBTS picks one of these tunnels to carry this EXP value.
- The operation of CBTS is not supported with Any Transport over MPLS (AToM), MPLS TE Automesh, or label-controlled (LC)-ATM.

When traffic admission to tunnels is achieved using global static routes, and when there is more than one tunnel to a given destination with the same administrative weight, the CBTS attribute acts as a tiebreaker in selecting the right tunnel. (See above discussion of load-balancing with CBTS.)

## Policy-Based Tunnel Selection

Multi-Protocol Label Switching Traffic Engineering Policy-Based Tunnel Selection (PBTS) enables you to dynamically route and forward traffic based on a policy onto different TE tunnels between the same tunnel head end and the same tail end. The routing algorithm is performed on the headend router's ingress interface prior to forwarding lookup.

In the Prime Fulfillment implementation of PBTS, traffic is directed into specific TE tunnels using the interface command `policy-class`. Whereas CBTS is aimed at IOS devices, PBTS is strictly designed for IOS XR devices.

Like CBTS, PBTS is not a form of traffic admission to TE tunnels directly, but rather an addition criteria that traffic must satisfy before being admitted to tunnels via the autoroute or static route mechanisms that TEM supports.



### Note

---

Prime Fulfillment itself does not provision the policy class, it merely associates a tunnel with an existing policy class. This is done by specifying the `policy-class` attribute in the range 1 to 7.

---

For more information on CBTS, see [Class-Based Tunnel Selection, page 3-8](#).

For general information on PBTS and IOS XR, see

[http://www.cisco.com/en/US/docs/ios\\_xr\\_sw/iosxr\\_r3.7/mpls/configuration/guide/gc37te.html#wp1325561](http://www.cisco.com/en/US/docs/ios_xr_sw/iosxr_r3.7/mpls/configuration/guide/gc37te.html#wp1325561).







# CHAPTER 4

## Prime Diagnostics Overview

Prime Diagnostics is an automated, workflow-based network management application that troubleshoots and diagnoses problems in Multiprotocol Label Switching (MPLS) VPNs. Prime Diagnostics offers users the capability to reduce the amount of time required to diagnose MPLS-related network outages—in many cases from hours to minutes. It performs diagnostics based on analysis of network failure scenarios, across MPLS access, edge, and core networks. It is equally applicable to both service provider and enterprise “self-deployed” MPLS VPN networks. Network operations center (NOC) support technicians as well as second-line and third-line support can benefit from this product. Prime Diagnostics optionally integrates with the Prime Fulfillment MPLS VPN provisioning component. To diagnose MPLS VPN core problems, Cisco IOS and IOS XR software releases supporting MPLS operations and maintenance (OAM) features including label-switched path (LSP) ping and LSP traceroute are required.

In effective fault finding and troubleshooting, there are five steps:

1. Detection
2. Isolation
3. Diagnosis
4. Repair
5. Verification

Prime Diagnostics is designed to support reactive situations in which an end customer reports a problem with their VPN service. This is essentially the Detection step in [Figure 4-1](#). The Repair function is not supported because many providers prefer to be in complete control of any changes made to router devices and might have specific in-house procedures for doing so.

**Figure 4-1** The Reactive Fault Lifecycle



137563

**Note**


---

Steps 2, 3, and 5 are performed by Prime Diagnostics. Steps 1 and 4 must be performed manually.

---

Prime Diagnostics focuses on the Isolation, Diagnosis, and Verification steps. It provides invaluable functionality for isolating and diagnosing failures in the network, determining the device(s) at fault, and checking appropriate device status and configuration to determine the likely reason for the failure. Prime Diagnostics also provides the ability to rerun tests to verify that changes made to the device configuration have resolved the issue.

The functionality can be used on its own, without any dependency on any other modules in Prime Fulfillment (for example, VPN provisioning or Traffic Engineering Management). It can also be used in Prime Fulfillment installations where some or all of the other Prime Fulfillment modules are used. If the MPLS VPN Provisioning functionality is used, then Customer and VPN data can be used as a starting point for troubleshooting, to locate the endpoints (for example, Customer Edge devices) between which connectivity is tested.

In addition to troubleshooting, Prime Diagnostics can also be used for VPN post-provisioning checks. After deploying a VPN, either manually or using Prime Fulfillment VPN provisioning, a connectivity test can be run to verify that the VPN has been provisioned successfully.

**Note**


---

Prime Diagnostics does not have any support for underlying configuration or routing changes during troubleshooting. During the execution of Prime Diagnostics, any changes made either by the operator or through the control plane of the routers, will not be reflected in the actual troubleshooting performed. Prime Diagnostics does not guarantee that the correct Failure Scenario or observation will be found in cases where such changes are made.

---

## IPv6

The IPv4 address free pool held by the Internet Assigned Numbers Authority (IANA) is running out. Cisco is addressing this shortage by adopting IPv6 addressing.

Prime Diagnostics supports configuration and selection of devices with both IPv4 and IPv6 addresses. Prime Diagnostics can troubleshoot MPLS VPN services where the attachment circuits:

- use IPv6 addressing
- use dual stack IPv4/IPv6 addressing.

Dual stack is a technique that allows both IPv4 and IPv6 to coexist on the same interfaces. For many years, if not forever, there will be a mix of IPv6 and IPv4 nodes on the Internet. Thus compatibility with the large installed base of IPv4 nodes is crucial for the success of the transition from IPv4 to IPv6. For example, a single interface can be configured with an IPv4 address and an IPv6 address. All the elements referenced as dual-stacked, such as provider edge and customer edge routers, run IPv4 as well as IPv6 addressing and routing protocols.

**Note**


---

Prime Diagnostics supports only global unicast IPv6 addresses. A global unicast address is very similar in function to an IPv4 unicast address such as 131.107.1.100. In other words, these addresses are conventional and publicly routable addresses. A global unicast address includes a global routing prefix, a subnet ID, and an interface ID.

---

**Table 4-1** General Unicast Address Structure

Fields	Network prefix	Subnet	Interface Identifier
Bits	48	16	64

**Note**

Prime Diagnostics permits to launch a test where both attachment circuit endpoints are either IPv6 and IPv6 or IPv4 and IPv4. No mixed addressing formats can be specified

For more details about when a test is initiated on an IPv6 address, see Cisco Prime Fulfillment User Guide 6.1.





# APPENDIX A

## MPLS Service Request Transition States

This chapter documents service request transition states for MPLS.

[Table A-1](#) and [Table A-2 on page A-2](#) show the state transition paths for Prime Fulfillment service requests. The beginning state of a service request is listed in the first column; the states that service requests transition to are displayed in the heading row.

For example, to use [Table A-1](#) to trace the state of a Pending service request to Functional, find **Pending** in the first column and move to your right until you find **Functional** in the heading. You can see that for a service request to move from Pending to Functional, a successful routing audit must take place.

[Table A-1](#) shows the service request transitions from *Requested* to *Lost*.

**Table A-1 State Transition Paths for Prime Fulfillment Service Requests (Part 1)**

Service Request States	Requested	Pending	Failed Audit	Deployed	Functional	Lost
Requested	No transition to Requested	Successful service request deployment	No transition to Failed Audit	No transition to Deployed	No transition to Functional	No transition to Lost
Pending	No transition to Requested	Successful service request deployment	Audit is not successful	Audit is successful	Routing audit is successful	No transition to Lost
Failed Audit	No transition to Requested	Successful service request redeployment	No transition to Failed Audit	Audit is successful	Routing audit is successful	No transition to Lost
Deployed	No transition to Requested	Successful service request redeployment	No transition to Failed Audit	Audit is successful	Routing audit is successful	Audit found error
Functional	No transition to Requested	Successful service request redeployment	No transition to Failed Audit	No transition to Deployed	Routing audit is successful	Audit found error
Lost	No transition to Requested	Successful service request redeployment	No transition to Failed Audit	Audit is successful	Routing audit is successful	Audit found error
Broken	No transition to Requested	Successful service request redeployment	No transition to Failed Audit	No transition to Deployed	Routing audit is successful	Audit found error

Table A-1 State Transition Paths for Prime Fulfillment Service Requests (Part 1) (continued)

Service Request States	Requested	Pending	Failed Audit	Deployed	Functional	Lost
<b>Invalid</b>	No transition to Requested	Successful service request redeployment	Redeployment caused service request error	No transition to Deployed	No transition to Functional	No transition to Lost
<b>Failed Deploy</b>	No transition to Requested	Successful service request redeployment	Redeployment service request failed. Configlet cannot be downloaded.	No transition to Deployed	No transition to Functional	No transition to Lost
<b>Closed</b>	No transition to Requested	No transition to Pending	No transition to Failed Audit	No transition to Deployed	No transition to Functional	No transition to Lost

Table A-2 shows the service request transitions from *Broken* to *Closed*.

Table A-2 State Transition Paths for Prime Fulfillment Service Requests (Part 2)

Service Request States	Broken	Invalid	Failed Deploy	Closed
<b>Requested</b>	No transition to Broken	Deploy service request error	Deployment failed	No transition to Closed
<b>Pending</b>	Route audit is not successful. Configlet is correct.	Redeployment caused service request error	Redeployment service request failed. Configlet cannot be downloaded.	Removal of the service request is successful
<b>Failed Audit</b>	Route audit is not successful. Configlet is correct.	Redeployment caused service request error	Redeployment service request failed. Configlet cannot be downloaded.	No transition to Closed
<b>Deployed</b>	Route audit is not successful. Configlet is correct.	Redeployment caused service request error	Redeployment service request failed. Configlet cannot be downloaded.	No transition to Closed
<b>Functional</b>	Route audit is not successful. Configlet is correct.	Redeployment caused service request error	Redeployment service request failed. Configlet cannot be downloaded.	No transition to Closed
<b>Lost</b>	Route audit is not successful. Configlet is correct.	Redeployment caused service request error	Redeployment service request failed. Configlet cannot be downloaded.	No transition to Closed
<b>Broken</b>	Route audit is not successful. Configlet is correct.	Redeployment caused service request error	Redeployment service request failed. Configlet cannot be downloaded.	No transition to Closed
<b>Invalid</b>	No transition to Broken	Redeployment caused service request error	Redeployment service request failed. Configlet cannot be downloaded.	No transition to Closed

Table A-2 State Transition Paths for Prime Fulfillment Service Requests (Part 2) (continued)

Service Request States	Broken	Invalid	Failed Deploy	Closed
Failed Deploy	No transition to Broken	Redeploy service request error	Redeployment service request failed. Configlet cannot be downloaded.	No transition to Closed
Closed	No transition to Broken	No transition to Invalid	No transition to Failed Deploy	No transition to Closed







## INDEX

---

### A

- AAL5 [1-8](#)
- address space separation [2-8](#)
- ATMoMPLS [1-8](#)
- ATM over MPLS (ATMoMPLS) [1-8](#)
- audience [v](#)
- authenticating
  - LDP [2-13](#)
  - routes [2-13](#)

---

### B

- bandwidth pools [4-6](#)

---

### C

- CBTS
  - Class-Based Tunnel Selection [4-8](#)
- cell relay
  - over MPLS [1-8](#)
- CEs
  - security of the PE-CE interface [2-13](#)
- Class-Based Tunnel Selection (CBTS) [4-8](#)
- concurrent use
  - overview [4-4](#)
  - with managed and unmanaged tunnels [4-4](#)
- conformant/non-conformant tunnels
  - defining [4-3](#)
  - managing [4-3](#)
  - overview [4-2](#)
- Connectivity Protection (CSPF) Backup Tunnels [4-8](#)
- CSPF

- Connectivity Protection Backup Tunnels [4-8](#)

---

### D

- devices
  - suitable for TE Discovery [4-5](#)
  - trusted devices [2-12](#)

---

### E

- ERS
  - multipoint ERS (EVP-LAN) for an Ethernet-based provider core [1-19](#)
  - multipoint ERS (EVP-LAN) for an MPLS-based provider core [1-17](#)
- Ethernet relay service (ERS or EVPL) [1-5](#)
- Ethernet wire service (EWS or EPL) [1-5](#)
- EWS
  - multipoint EWS (EP-LAN) for an Ethernet-based provider core [1-18](#)
  - multipoint EWS (EP-LAN) for an MPLS-based provider core [1-17](#)
- extranets [2-2](#)

---

### F

- frame relay over MPLS (FRoMPLS) [1-9](#)
- FRoMPLS [1-9](#)
- full mesh topologies [2-8](#)

---

### H

- hub and spoke topologies [2-8](#)

**I**

implementing, VRFs [2-4](#)  
 intranets [2-2](#)

**L**

## L2VPN

service provisioning [1-5](#)  
 terminology conventions [1-1](#)

L2VPN Ethernet over MPLS (ERS and EWS) (EPL and EVPL) [1-5](#)

label spoofing [2-12](#)

LDP authentication [2-13](#)

## links

provisioning regular PE-CE links [3-1, 5-3](#)

locking mechanism [4-4](#)

**M**

managed/unmanaged primary tunnels [4-2](#)

## managing

independent VRF objects [2-5](#)

## MEF

mapping MEF terminologies to network technologies [1-3](#)

terminology conventions [1-1](#)

Metro Ethernet Forum (see MEF) [1-1](#)

## MPLS VPNs

concepts [2-1](#)  
 security [2-8](#)

multiple concurrent users [4-4](#)

multiple OSPF areas [4-5, 4-6](#)

## multipoint

ERS (EVP-LAN) for an Ethernet-based provider core [1-19](#)

ERS (EVP-LAN) for an MPLS-based provider core [1-17](#)

EWS (EP-LAN) for an Ethernet-based provider core [1-18](#)

EWS (EP-LAN) for an MPLS-based provider core [1-17](#)

**O**

objective [v](#)

organization [v](#)

## OSPF areas

example of network [4-6](#)

multiple [4-5](#)

## overview

Diagnostics [5-1](#)

**P**

## PBTS

Policy-Based Tunnel Selection [4-9](#)

## PEs

security of the PE-CE interface [2-13](#)

planning tools [4-7](#)

## point-to-point

Ethernet (EWS and ERS) (EPL and EVPL) [1-5](#)

Policy-Based Tunnel Selection (PBTS) [4-9](#)

prerequisite knowledge [5-2](#)

## providers

multipoint ERS (EVP-LAN) for an Ethernet-based provider core [1-19](#)

multipoint ERS (EVP-LAN) for an MPLS-based provider core [1-17](#)

multipoint EWS (EP-LAN) for an Ethernet-based provider core [1-18](#)

multipoint EWS (EP-LAN) for an MPLS-based provider core [1-17](#)

## provisioning

regular PE-CE links [3-1, 5-3](#)

**R**

reactive fault lifecycle [5-1](#)

relay service, Ethernet [1-5](#)

- route distinguishers [2-5](#)
- route targets [2-5](#)
  - communities [2-6](#)
- routing
  - authentication [2-13](#)
  - separation [2-8, 2-9](#)
- routing and forwarding tables [2-3](#)
- routing protocols
  - securing [2-10](#)

---

## S

- security
  - ensuring VPN isolation [2-15](#)
  - hiding the MPLS core structure [2-9](#)
  - label spoofing [2-12](#)
  - LDP authentication [2-13](#)
  - MP-BGP security features [2-14](#)
  - MPLS VPNs [2-8](#)
  - of the PE-CE interface [2-13](#)
  - resistance to attacks [2-10](#)
  - securing the MPLS core [2-12](#)
  - securing the routing protocol [2-10](#)
  - security through IP address resolution [2-15](#)
  - separation of CE-PE links [2-13](#)
  - trusted devices [2-12](#)
- service provisioning, for L2VPN [1-5](#)

---

## T

- TE area identifier
  - TE Discovery [4-5](#)
- TE Discovery
  - devices suitable for [4-5](#)
  - TE area identifier [4-5](#)
- terminology conventions
  - L2VPN [1-1](#)
  - MEF [1-1, 1-3](#)

- TE tunnels
  - concurrent use with managed and unmanaged tunnels [4-4](#)
- topologies
  - full mesh [2-8](#)
  - hub and spoke [2-8](#)
- topology
  - for ATMoMPLS [1-8](#)
  - for Ethernet-based VPLS [1-19](#)
  - for FRoMPLS [1-9](#)
  - for L2VPN Ethernet over MPLS (ERS and EWS) (EPL and (EVPL) [1-5](#)
  - for MPLS-based VPLS [1-17](#)

---

## V

- VPLS
  - for an Ethernet-based (L2) provider core [1-18](#)
  - service provisioning [1-16](#)
  - topology for Ethernet-based VPLS [1-19](#)
  - topology for MPLS-based VPLS [1-17](#)
- VPNs [2-1](#)
  - connectivity between VPNs [2-14](#)
  - ensuring VPN isolation [2-15](#)
  - VPN routing and forwarding tables [2-3](#)
- VRF objects
  - independent VRF object management [2-5](#)
- VRFs
  - implementation of [2-4](#)
  - VRF instance [2-5](#)

