MPLS Manager
1.2

DISCOVERY GUIDE SUPPLEMENT

P/N 300-002-531
REV A02

OL-8967-01

EMC Corporation
Corporate Headquarters:
Hopkinton, MA 01748-9103
1-508-435-1000
www.EMC.com

Last Update: 8/19/05

Contents

Preface

This document provides an overview of the EMC Smarts MPLS Manager discovery process and presents procedures for preparing and initiating MPLS Manager for discovery of Multiprotocol Label Switching (MPLS) topology and Virtual Private Network (VPN) topology in an IP environment. This document is intended as a supplement to the *EMC Smarts IP Management Suite Discovery Guide*.

Intended Audience

This document is intended to be read by IT managers seeking to understand how the MPLS discovery process works, and by system administrators responsible for the administration, configuration, or use of EMC Smarts applications that provide MPLS discovery over IP networks.

Prerequisites

Readers of this document should have read the *EMC Smarts IP Management Suite Discovery Guide* first, and be familiar with EMC Smarts discovery concepts.

To perform the procedures in this document, the EMC Smarts MPLS Manager, EMC Smarts IP Availability Manager (Availability Manager), and EMC Smarts Service Assurance Manager (Global Manager) must be installed. The Global Console is also needed to manually invoke discovery.

For information about installing these products, see the *EMC Smarts MPLS Management Suite Installation Guide*, the *EMC Smarts IP Management Suite Installation Guide*, and the *EMC Smarts Service Assurance Management Suite Installation Guide*.

Document Organization

This document consists of the following chapters.

Table 1: Document Organization

1. DISCOVERY OVERVIEW	Describes the concepts of discovering MPLS and VPN topology using the MPLS Manager.
2. DISCOVERY PROCESS	Explains how the MPLS Manager discovers and models MPLS and VPN elements.
3. PREPARING FOR AND INITIATING DISCOVERY	Explains how to prepare and initiate discovery for an MPLS Manager deployment.
4. INVOKING DISCOVERY MANUALLY	Provides information about how to manually invoke discovery updates for the MPLS Manager and how to view and correct discovery errors.

Documentation Conventions

Several conventions may be used in this document as shown in Table 2.

Table 2: Documentation Conventions

CONVENTION	EXPLANATION
`sample code`	Indicates code fragments and examples in Courier font
keyword	Indicates commands, keywords, literals, and operators in bold
%	Indicates C shell prompt
#	Indicates C shell superuser prompt
<parameter>	Indicates a user-supplied value or a list of non-terminal items in angle brackets
[option]	Indicates optional terms in brackets
/InCharge	Indicates directory path names in italics
yourDomain	Indicates a user-specific or user-supplied value in bold, italics
File > Open	Indicates a menu path in italics
▼▲	Indicates a command is wrapped over one or more lines. The command must be typed as one line.

Directory path names are shown with forward slashes (/). Users of the Windows operating systems should substitute back slashes (\) for forward slashes.

Also, if there are figures illustrating consoles in this document, they represent the consoles as they appear in Windows. Under UNIX, the consoles appear with slight differences. For example, in views that display items in a tree hierarchy such as the Topology Browser, a plus sign displays for Windows and an open circle displays for UNIX.

Finally, unless otherwise specified, the term InCharge Manager is used to refer to EMC Smarts programs such as Domain Managers, Global Managers, and adapters.

MPLS Management Suite Installation Directory

In this document, the term ***BASEDIR*** represents the location where EMC Smarts software is installed.

- For UNIX, this location is: */opt/InCharge<n>/<productsuite>*.
- For Windows, this location is: *C:\InCharge<n>\<productsuite>*.

The *<n>* represents the EMC Smarts software platform version number. The *<productsuite>* represents the InCharge product suite to which the product belongs. For example, on UNIX operating systems, MPLS Manager is installed to */opt/InCharge6/MPLS/smarts* by default. On Windows operating systems, this product is installed to *C:\InCharge6\MPLS\smarts* by default. This location is referred to as ***BASEDIR****/smarts*.

Optionally, you can specify the root of ***BASEDIR*** to be something other than */opt/InCharge6* (on UNIX) or *C:\InCharge6* (on Windows), but you cannot change the *<productsuite>* location under the root directory.

For more information about the directory structure of EMC Smarts software, refer to the *EMC Smarts System Administration Guide*.

MPLS Management Suite Products

The MPLS Management Suite offers the following products:

- MPLS Manager
- EMC Smarts Adapter for Cisco ISC
- Perl API

Additional Resources

In addition to this document, EMC Corporation provides the following resources.

Command Line Programs

Descriptions of command line programs are available as HTML pages. The *index.html* file, which provides an index to the various commands, is located in the ***BASEDIR****/smarts/doc/html/usage* directory.

Documentation

Readers of this document may find other documentation (also available in the ***BASEDIR****/smarts/doc/pdf* directory) helpful.

EMC Smarts Documentation

The following documents are product independent and thus relevant to users of all EMC Smarts products:

- *EMC Smarts Documentation Roadmap*
- *EMC Smarts System Administration Guide*
- *EMC Smarts ASL Reference Guide*
- *EMC Smarts Perl Reference Guide*

MPLS Management Suite Documentation

The following documents are relevant to users of the MPLS Management Suite product suite:

- *EMC Smarts MPLS Management Suite Installation Guide*
- *EMC Smarts MPLS Manager User's Guide*
- *EMC Smarts MPLS Manager Configuration Guide*
- *EMC Smarts MPLS Manager Discovery Guide Supplement*
- *EMC Smarts Adapter for Cisco ISC User's Guide*
- *EMC Smarts MPLS Management Suite Release Notes*

Refer to the *EMC Smarts Documentation Roadmap* for documentation resources provided with other EMC Smarts product suites.

Technical Support

For questions about technical support, call your local sales office or service provider. For service, call one of the following numbers:

United States: 800.782.4362 (SVC.4EMC)

Canada: 800.543.4782 (543.4SVC)

Worldwide: 508.497.7901

EMC Powerlink

EMC Powerlink is the EMC Corporation's secure extranet for customers and partners. Powerlink is an essential tool for obtaining web-based support from the EMC Corporation. Powerlink can be used to submit service or information requests (tickets) and monitor their progress, to review the knowledgebase for known problems and solutions, and to download patches and SmartPacks.

From training on EMC products and technologies, to online support, product announcements, software registration, technical white papers, interoperability information, and a range of configuration tools, Powerlink offers resources unavailable elsewhere.

For quickest access when you do not already have a Powerlink account, ask your EMC representative for the access code for your company and register at the Powerlink site. Visit the EMC Powerlink website at:

http://powerlink.emc.com

1

Discovery Overview

The EMC Smarts MPLS Manager works with the EMC Smarts Availability Manager to discover the logical and physical elements in the transport domain, the MPLS domain, and the VPN domain.

VPN Support

The MPLS Manager discovers the following types of provider-provisioned VPNs:

- MPLS Layer 3 VPNs (L3VPNs)

  Defined by IETF RFC-2547bis, MPLS L3VPNs use extensions to the existing Internet routing protocol (BGP-4) to interconnect remote customer sites through an MPLS-enabled network. L3VPN is a virtual private routed network solution for IP data traffic only.

- MPLS Layer 2 VPNs (L2VPNs)

  Commonly called Martini VPNs, MPLS L2VPNs extend the customer's Layer 2 connectivity through an MPLS-enabled network by emulating different types of traditional data-link layer protocols, including Ethernet, Frame Relay, ATM, and others.

  The MPLS Manager can discover both Martini-implemented and Kompella-implemented L2VPNs. For Juniper devices running Kompella-implemented (BGP-based) L2VPNs, the MPLS Manager discovers the L2VPNs using a method similar to the method used to discover L3VPNs.

Device Support

The MPLS Manager discovers the following types of routing devices:

- Cisco routers—ESR10xxx, 76xx, 75xx, 72xx, 45xx, and 36xx; GSR12xxx; and 26xx
- Juniper routers—M-Series, T-Series, ERX, and ERX virtual routers

MIB and CLI Discovery Support

As shown in Table 3, Table 4, and Table 5, the MPLS Manager supports both SNMP MIB and Command Line Interface (CLI) discovery.

Table 3: Discovery Sources for Cisco Devices

MPLS Core Elements	MPLS L3VPN Elements	MPLS L2VPN Elements
MPLS-LSR-MIB If MIB discovery fails or is not supported by the Cisco device, the MPLS Manager uses CLI discovery to discover the MPLS elements.	MPLS-VPN-MIB If MIB discovery fails or is not supported by the Cisco device, the MPLS Manager uses CLI discovery to discover the L3VPN elements.	CLI and MPLS-LDP-MIB The MPLS Manager uses: • CLI discovery to discover Cisco L2VPN elements. • MIB discovery to discover Cisco Label Distribution Protocol (LDP) elements associated with the Cisco L2VPN elements.

Table 4: Discovery Sources for Juniper M/T Devices

MPLS Core Elements	MPLS L3VPN Elements	MPLS L2VPN Elements
CLI	jnxVpnMIB If MIB discovery fails or is not supported by the Juniper M/T device, the MPLS Manager uses CLI discovery to discover the L3VPN elements.	Juniper-VPN-MIB

Table 5: Discovery Sources for Juniper ERX Devices

MPLS Core Elements	MPLS L3VPN Elements	MPLS L2VPN Elements
CLI	CLI	Juniper-VPN-MIB

Availability Manager Discovery

In the transport domain, the Availability Manager discovers the Layer 2 and Layer 3 network connectivity within the MPLS network and between the MPLS network and the customer site routers (CE routers, Figure 1). It uses the discovered topology to model the network, and uses SNMP polling and traps to diagnose and pinpoint the root cause of network failures. The Availability Manager sends the analysis results along with topology and event information to the Global Manager, and sends router topology and event information to the MPLS Manager.

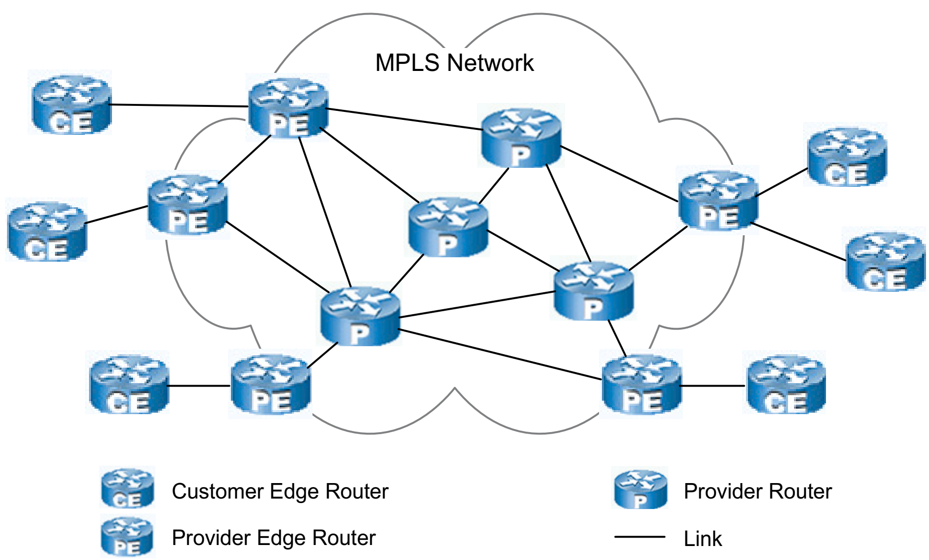

Figure 1: The Transport Domain Discovered by the Availability Manager

MPLS Manager Discovery

The MPLS Manager discovers the MPLS logical topology (Figure 2) and the VPN logical topology (Figure 3 and Figure 4) and models that topology in its repository. It maps the MPLS and VPN topology to the router topology discovered by the Availability Manager.

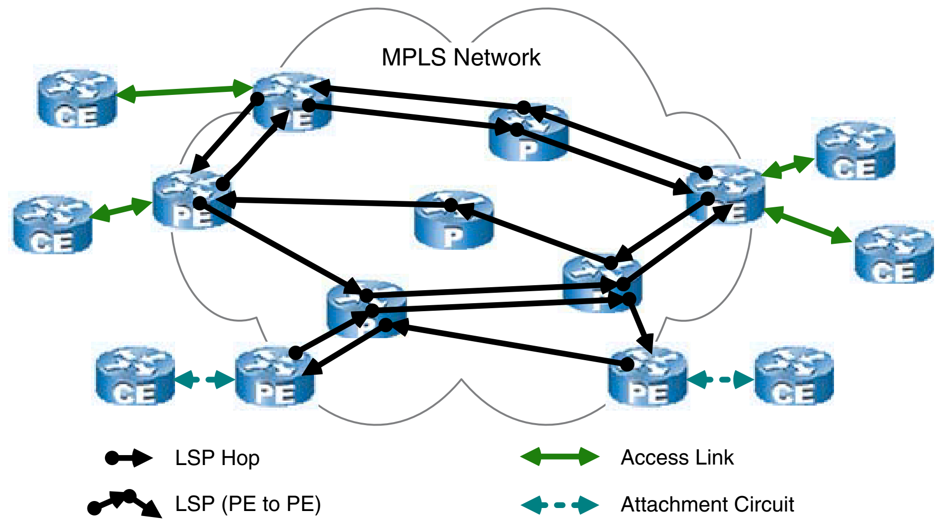

Figure 2: The MPLS Domain Discovered by the MPLS Manager

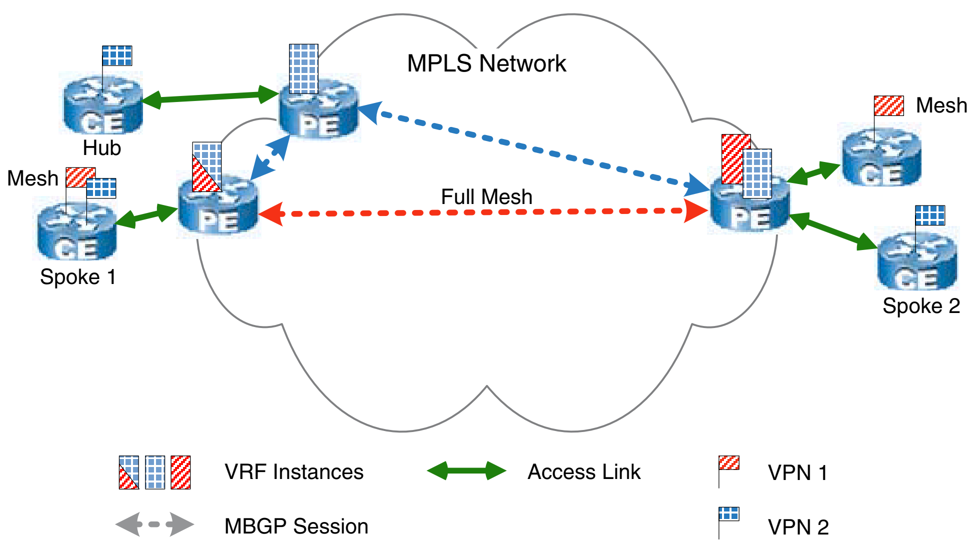

Figure 3: The L3VPN Domain Discovered by the MPLS Manager

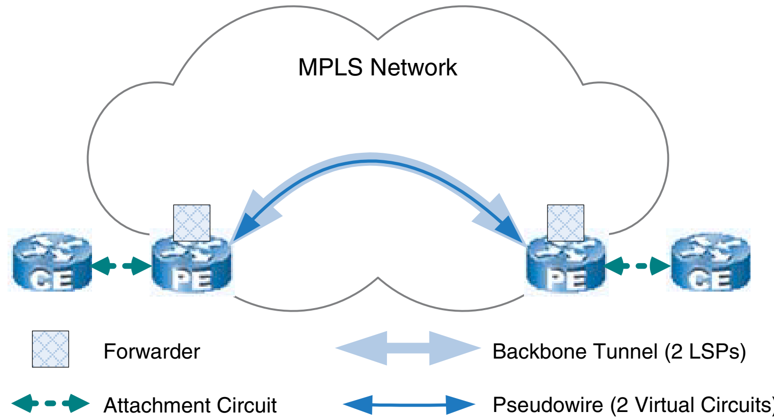

Figure 4: The L2VPN Domain Discovered by the MPLS Manager

Imports Routers from Availability Manager

From the Availability Manager, the MPLS Manager imports the initial router topology, receives router topology updates, and receives events relevant to the imported router topology. MPLS Manager imports the router elements along with the following elements associated with the routers:

- Chassis
- Card
- Port
- Interface
- IP
- DuplicateIP
- IPNetwork
- VPNIP
- VPNIPNetwork
- NetworkConnection
- SNMPAgent
- VLAN
- Partition

VPNIP and VPNIPNetwork contain VPN-IP addresses, each of which consists of a *route distinguisher* and an IPv4 address. The route distinguisher is the means by which the PE router and the MPLS Manager keep track of overlapping customer IP address spaces for L3VPNs.

All elements are described in the *EMC Smarts MPLS Manager User's Guide*.

Initiates MPLS Discovery

After importing the initial router topology from the Availability Manager, the MPLS Manager uses SNMP polling and/or CLI commands to query the routers for MPLS and VPN information. If SNMP polling fails or is not supported by a routing device, and assuming that CLI discovery is enabled, the MPLS Manager logs in to the routing device (via Telnet) and issues CLI commands to query the router for the required information.

For information about configuring CLI login environment variables and enabling CLI discovery, see the *EMC Smarts MPLS Manager Configuration Guide*.

Discovers MPLS and VPN Elements

The MPLS Manager discovers the following MPLS and VPN elements for L3VPNs:

- MPLS services
- Label Switched Paths (LSPs)
- LSP hops
- LSP segments
- Virtual Private Networks (VPNs)
- VPN Routing and Forwarding (VRF) instances
- Route targets

The MPLS Manager discovers the following MPLS and VPN elements for L2VPNs:

- MPLS services
- LSPs
- LSP hops
- LSP segments

- VPNs
- Forwarders
- PseudoWires
- Label Distribution Protocol (LDP) protocol endpoints (Cisco routing devices only)
- LDP adjacencies (Cisco routing devices only)

The MPLS Manager combines these topology elements with those discovered by the Availability Manager to build a complete model of the MPLS network, the VPNs, and the attached customer sites.

Logs Discovery Messages

Whenever a discovery cycle completes, the MPLS Manager prints a report to its log file. The log file is named *<MPLS manager name>.log* (for example, *INCHARGE-MPLS.log*) and is located in the ***BASEDIR****/smarts/local/logs* directory.

Creates CLI Log Files

During a discovery cycle, the MPLS Manager creates a CLI log file for each candidate routing device that either does not support MPLS MIB discovery or failed MIB discovery. Each CLI log file contains a record of the Telnet session with the particular device; it includes the CLI commands issued by the MPLS Manager and the responses returned by the device. MPLS Manager parses the log file to create topology elements in its repository.

A CLI log file is named *EXPECT-MPLS-<device name>-<timestamp>.txt* (for example, *EXPECT-MPLS-lab-gw.emc.com-1119557208.txt*). The timestamp is the number of seconds since January 1, 1970, 00:00:00 GMT. All CLI log files are located in the ***BASEDIR****/smarts/local/logs* directory.

By default, just before an MPLS Manager discovery cycle, the MPLS Manager deletes all CLI log files. If you wish to change this behavior to preserve CLI log files across discovery cycles, you need to set the `RemoveExpectLogs` parameter to `FALSE` in the MPLS Manager's *LOCAL.import* file, as explained in the *EMC Smarts MPLS Manager Configuration Guide*.

When Discovery Occurs

The MPLS Manager initiates a discovery cycle of the MPLS and VPN elements whenever:

- The Availability Manager completes a discovery cycle.

  Each time the Availability Manager completes a discovery cycle (a discovery from a seed file, a full discovery, or a single device discovery), the MPLS Manager performs a topology synchronization to import router topology information from the Availability Manager. The MPLS Manager then decides whether to perform its own discovery by comparing the newly imported router topology with the previously imported router topology.

  If the MPLS Manager finds a new routing device or finds an already known routing device having a newer discovery timestamp, the MPLS Manager sends SNMP polls and/or CLI commands to that device to discover/rediscover the device's MPLS and VPN elements and adds those elements to the MPLS and VPN topology.

- The MPLS Manager receives a manual discovery update request from the network administrator.

  A network administrator can manually invoke a discovery update at any time, as explained in *Invoking Discovery Manually* on page 27.

2

Discovery Process

During the discovery process, the MPLS Manager discovers MPLS and VPN elements in two phases:

- Per-device discovery
- Post-processing discovery

As it discovers the elements, the MPLS Manager uses instances of certain InCharge Common Information Model (ICIM) classes to create data model representations of the discovered elements within its repository.

Discovery Process for L3VPNs

The data model for L3VPN topology, shown in Figure 5, represents the MPLS and L3VPN elements and their relationships. Not all relationships are shown in the figure.

Note: Typically, every relationship has an inverse relationship. For example, the relationship *PartOf* is the inverse relationship of *ComposedOf*.

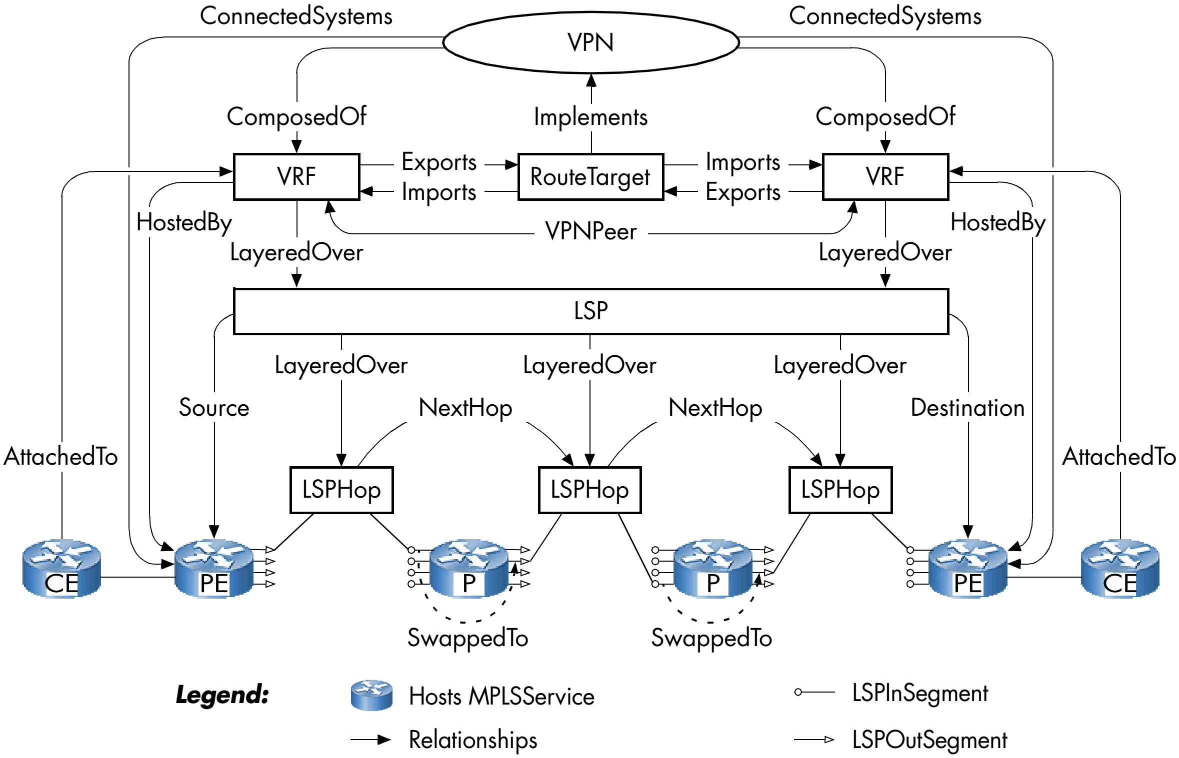

Figure 5: Simplified View of ICIM MPLS and L3VPN Reference Model

For descriptions of the VPN, VRF, RouteTarget, LSP, LSPHop, LSPInSegment, LSPOutSegment, and MPLSService classes, see the *EMC Smarts MPLS Manager User's Guide*.

Per-Device Discovery for L3VPN

Per-device discovery for L3VPN pertains to the MPLS Manager discovery of MPLS and L3VPN elements on a per routing device basis. The discovery may run in multiple concurrent threads (the default value is 10).

During this phase of the L3VPN discovery process, the MPLS Manager creates instances of the following classes for each discovered routing device:

- VRF
- RouteTarget
- LSPInSegment

- LSPOutSegment
- MPLSService

VRF and RouteTarget

The MPLS Manager relates VRFs and RouteTargets through the *Imports/ImportedBy* and *Exports/ExportedBy* relationship sets, and then discovers which interfaces are associated with each VRF.

The *CEs* relationship (not shown in Figure 5) points from the VRF to the CE router attached to the VRF. The MPLS Manager computes this relationship by traversing the *Peer* relationship of the *Underlying* interfaces, and then the *HostedBy* relationship of those peer network adapters.

Other computed relationships are:

- VRF relationship *SendsRoutesTo* (not shown in Figure 5)—Points to all VRFs that import the RouteTargets that this VRF *Exports*.
- VRF relationship *ReceivesRoutesFrom* (not shown in Figure 5)—Points to all VRFs that export the RouteTargets that this VRF *Imports*.
- VRF relationship *VPNPeer*—Points to all VRFs that are related to this VRF by both *SendsRoutesTo* and *ReceivesRoutesFrom*.

LSPInSegment and LSPOutSegment

Each LSPInSegment-LSPOutSegment pair represents an entry in the MPLS routing table of a discovered MPLS-enabled routing device, where the LSPInSegment represents an incoming label, and the LSPOutSegment represents an outgoing label. The two instances are related to each other with the *SwappedTo/SwappedFrom* relationship set (an LSPOutSegment is SwappedFrom an LSPInSegment). Instances of LSPOutSegment that are not *SwappedFrom* an LSPInSegment represent the first label in an ingress LSP.

The MPLS Manager creates an LSPInSegment and LSPOutSegment for an entry in the MPLS routing table *only* if the LSP associated with the entry terminates at a PE router that is already in the MPLS Manager topology.

MPLSService

The MPLS Manager creates an instance of MPLSService for each routing device in the topology. The relationships created for a routing device depend on the type of the device.

For a CE router:

- MPLSService relationship *AttachedTo*—Points to the VRF to which the CE router is attached.

For a PE router:

- MPLSService relationship *VRFs* (not shown in Figure 5)—Points to the VRFs hosted by the PE router.
- MPLSService relationships *LSPOut* and *LSPIn* (not shown in Figure 5)—Point to the LSPs for which the PE router is the source or destination (respectively).
- MPLSService relationship *VRFInterfaces* (not shown in Figure 5)—Points to all the interfaces that are associated with VRFs on the PE router.

For a PE or P router:

- MPLSService relationship *MPLSInterfaces* (not shown in Figure 5)—Points to all interfaces that are MPLS-enabled.

Post-Processing Discovery for L3VPNs

During this phase of the L3VPN discovery process, which starts after per-device discovery completes, the MPLS Manager creates instances of the following classes by combining the information collected during per-device discovery:

- VPN
- LSP
- LSPHop

VPN

The MPLS Manager infers VPNs from VRF and RouteTarget instances and the relationships between the instances. The MPLS Manager creates the following types of VPN:

- Full-mesh
- Hub-and-spoke

The MPLS Manager creates a full-mesh VPN for each RouteTarget (R1, for example) that has the same set of VRFs in its *ImportedBy* and *ExportedBy* relationship sets. The VPN is *ImplementedBy* R1 and is *ComposedOf* the VRFs that are *ExportedBy* (and *ImportedBy*) R1.

The MPLS Manager creates a hub-and-spoke VPN for each pair of RouteTargets (R2 and R3, for example) that has the following characteristics:

- The R2 relationship *ImportedBy* is identical to the R3 relationship *ExportedBy*.
- The set of VRFs that export R2 but do not import R2 is identical to the set of VRFs that import R3 but do not export R3.
- The R2 relationship *ExportedBy* is not identical to the R3 relationship *ImportedBy*, or at least one VRF exists that both *Imports* and *Exports* R2, or both conditions are true.

The VPN is *ImplementedBy* the two RouteTargets. The VPN is *ComposedOf* all the VRFs in the R2 relationship *ImportedBy* and the R2 relationship *ExportedBy*. VPN relationship *Spokes* is the set of VRFs in the R2 relationship *ExportedBy* but not in the R2 relationship *ImportedBy*. VPN relationship *Hubs* includes the rest of the VRFs.

LSP

The MPLS Manager scans all VRFs in the topology and examines the VRF *AccessedVia* relationship. If this relationship includes an LSPOutSegment, MPLS Manager creates an LSP instance *between* the PE router that hosts the LSPOutSegment *and* the PE router to which the LSP of this LSPOutSegment refers. The *Source* and *Destination* relationships of the LSP are set appropriately.

LSPHop

For each LSP instance that the MPLS Manager creates, it also creates the LSPHop instances along the path of the LSP. An LSPHop is *ConnectedTo* an LSPOutSegment and an LSPInSegment. If the LSPHop is the last one for an LSP, it is only *ConnectedTo* an LSPOutSegment (which has a label value of 3).

The MPLS Manager computes the *NextHop* and *PreviousHop* relationships of LSPHop instances. To compute the *NextHop* relationship, the MPLS Manager traces the *ConnectedTo* relationship to the LSPInSegment, then the *SwappedTo* relationship to an LSPOutSegment, and then the *ConnectedVia* relationship to an LSPHop.

Discovery Process for L2VPNs

The data models for L2VPN topology, shown in Figure 6 for Martini-implemented L2VPNs and Figure 7 for Kompella-implemented L2VPNs, represent the MPLS and L2VPN elements and their relationships. Not all relationships are shown in the figures.

Note: Typically, every relationship has an inverse relationship. For example, the relationship *PartOf* is the inverse relationship of *ComposedOf*.

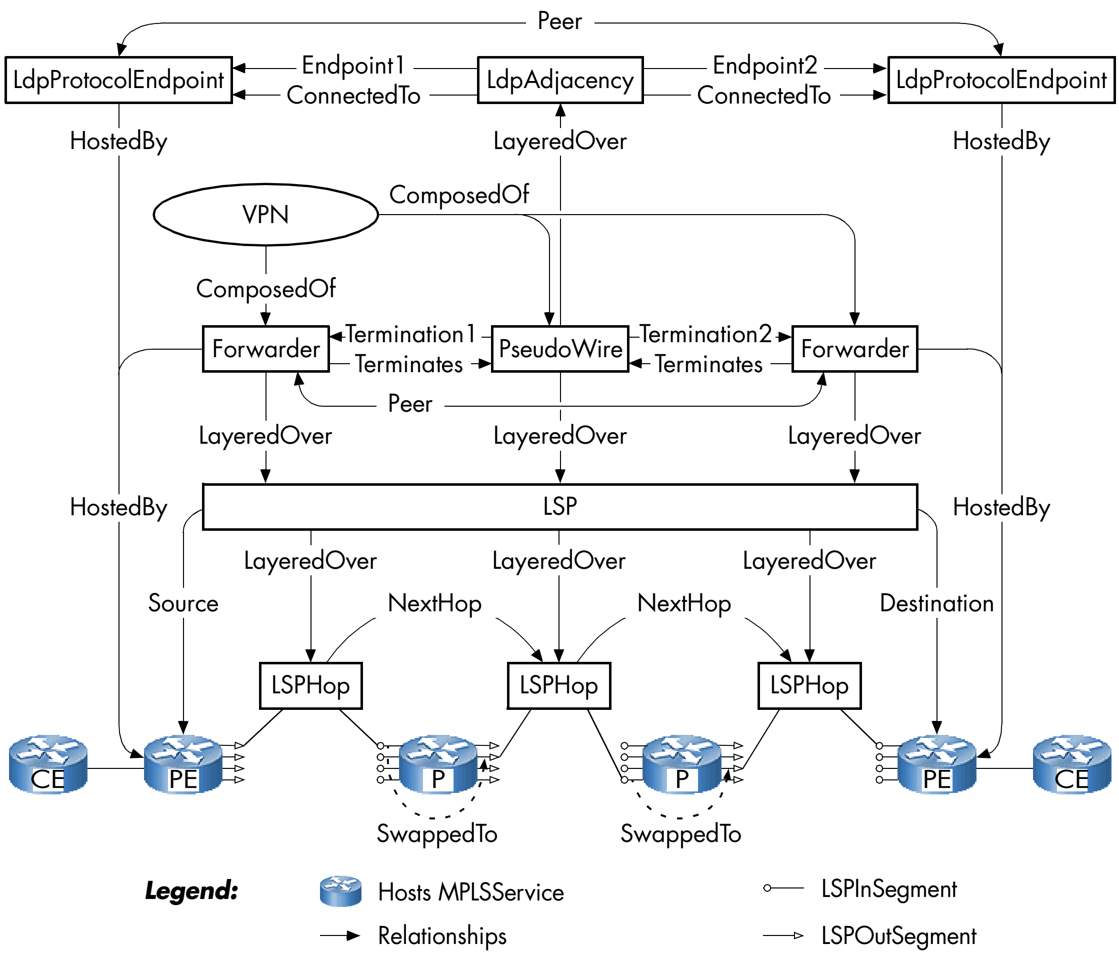

Figure 6: Simplified View of ICIM MPLS and Martini-Implemented L2VPN Reference Model

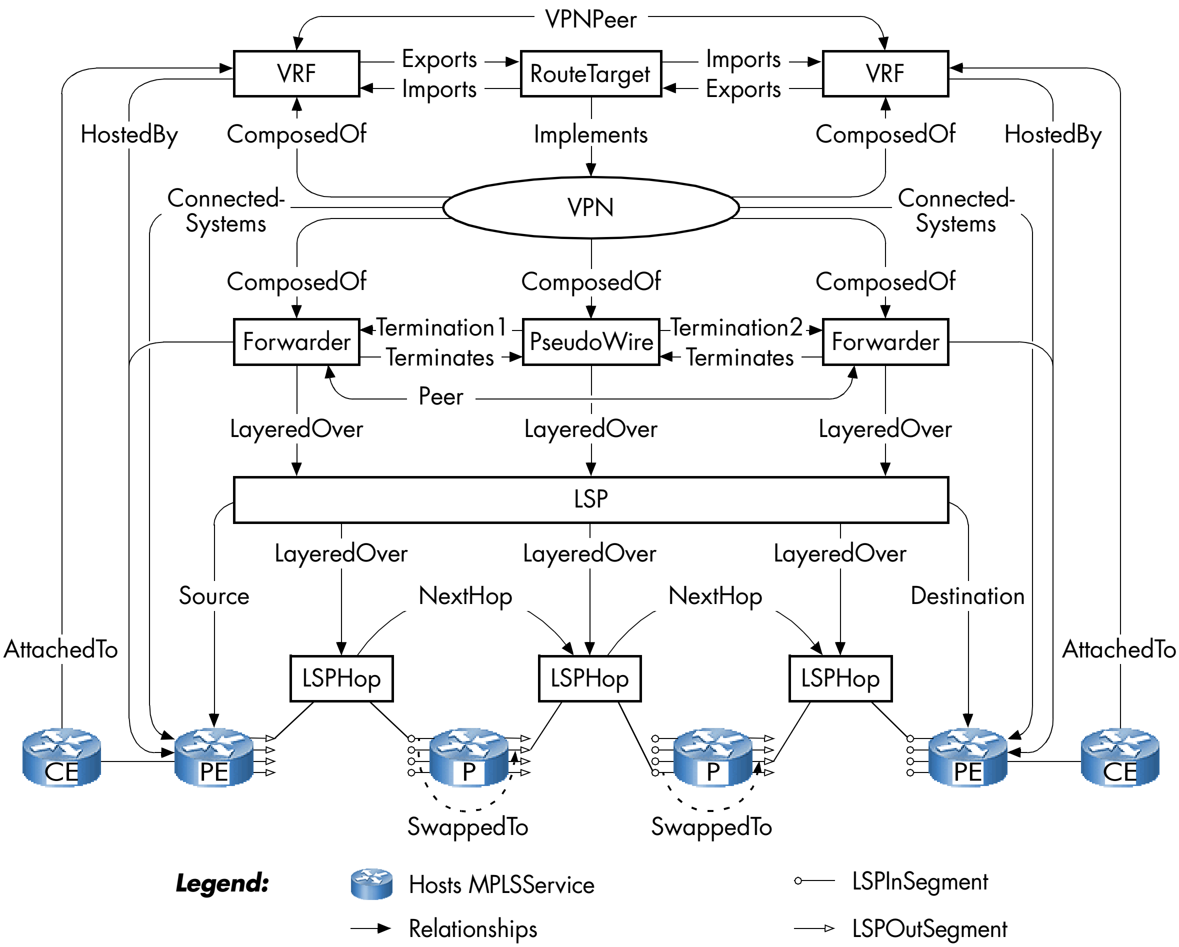

Figure 7: Simplified View of ICIM MPLS and Kompella-Implemented L2VPN Reference Model

For descriptions of the VPN, Forwarder, PseudoWire, LdpProtocolEndpoint, LdpAdjacency, VRF, RouteTarget, LSP, LSPHop, LSPInSegment, LSPOutSegment, and MPLSService classes, see the *EMC Smarts MPLS Manager User's Guide*.

The discovery and modeling of the LdpProtocolEndpoint and LdpAdjacency elements is specific to Martini-implemented L2VPNs, and the discovery and modeling of the VRF and RouteTarget elements is specific to Kompella-implemented L2VPNs (and to L3VPNs). Martini-implemented L2VPNs use

Label Distribution Protocol (LDP) as their signaling protocol, and Kompella-implemented L2VPNs use Multiprotocol Border Gateway Protocol (MBGP) as their signaling protocol.

Note: The model in Figure 7 is specific to Juniper's implementation of L2VPNs based on the Kompella methodology. The L2VPN specifications do not require the construction and maintenance of VRFs for L2VPNs.

Per-Device Discovery for L2VPN

Per-device discovery for L2VPN pertains to the MPLS Manager discovery of MPLS and L2VPN elements on a per routing device basis. The discovery may run in multiple concurrent threads (the default value is 10).

During this phase of the L2VPN discovery process, the MPLS Manager creates instances of the following classes for each discovered routing device:

- Forwarder
- LdpProtocolEndpoint (Martini only)
- VRF (Kompella only)
- RouteTarget (Kompella only)
- LSPInSegment
- LSPOutSegment
- MPLSService

Forwarder

For each Forwarder discovered on a PE device, the MPLS Manager discovers the one-to-one mapping between the customer-side attachment circuit—a VLAN or an Ethernet port, for example—and the MPLS-side pseudowire—a Martini Tunnel, for example. The MPLS Manager creates a Forwarder instance for each discovered Virtual Connection Identifier (VC ID).

Note that the VC ID identifies a particular pseudowire and represents the two virtual circuits—one carried over an outgoing LSP and the other carried over an incoming LSP—comprising a pseudowire. Together, the two virtual circuits provide bidirectional communication for a subscriber.

The MPLS Manager relates a Forwarder with other discovered L2VPN elements through the following computed relationships:

- Forwarder relationship *Terminates*—Relates the Forwarder to the pseudowire associated with this Forwarder.
- Forwarder relationship *Peer*—Points to the Forwarder that *Terminates* the other end of the pseudowire associated with this Forwarder.
- Forwarder relationship *LayeredOver*—Points to the *Underlying* interfaces and LSPs associated with this Forwarder. For a Forwarder associated with a VLAN attachment circuit, also points to the *Underlying* VLAN attachment circuit.

LdpProtocolEndpoint (Martini Only)

For an L2VPN using LDP to distribute Layer 2 connectivity and reachability information between PE devices, the MPLS Manager creates an LdpProtocolEndpoint instance for each discovered LDP speaker associated with the L2VPN. Martini-implemented L2VPNs use LDP signaling to construct, advertise, maintain, and delete pseudowires—also known as Martini Tunnels.

The *Peer* relationship for an LdpProtocolEndpoint points to the LdpProtocolEndpoint at the other end of the LDP session. Because an LDP session is TCP-based, *Peer* LdpProtocolEndpoints can terminate on PE devices that are multiple Layer 3 hops apart.

VRF and RouteTarget (Kompella Only)

For an L2VPN using MBGP to distribute Layer 2 connectivity and reachability information between PE devices, the MPLS Manager creates a VRF instance and a RouteTarget instance for each discovered VRF and route target associated with the L2VPN. Kompella-implemented L2VPNs not only use MBGP signaling to construct, advertise, maintain, and delete pseudowires, but also use MBGP and route targets to discover all the VRFs comprising a VPN.

As with the discovery and modeling of L3VPNs, the MPLS Manager relates VRFs and RouteTargets through the *Imports/ImportedBy* and *Exports/ExportedBy* relationship sets, and then discovers which interfaces are associated with each VRF. By matching the VRF interfaces to the Forwarder interfaces, the MPLS Manager discovers which Forwarders and pseudowires are part of a VPN.

For information about additional relationships created for VRFs and RouteTargets, see *VRF and RouteTarget* on page 11.

LSPInSegment and LSPOutSegment

Each LSPInSegment-LSPOutSegment pair represents an entry in the MPLS routing table of a discovered MPLS-enabled routing device, where the LSPInSegment represents an incoming label, and the LSPOutSegment represents an outgoing label. The two instances are related to each other with the *SwappedTo/SwappedFrom* relationship set (an LSPOutSegment is SwappedFrom an LSPInSegment). Instances of LSPOutSegment that are not *SwappedFrom* an LSPInSegment represent the first label in an ingress LSP.

The MPLS Manager creates an LSPInSegment and LSPOutSegment for an entry in the MPLS routing table *only* if the LSP associated with the entry terminates at a PE router that is already in the MPLS Manager topology.

MPLSService

The MPLS Manager creates an instance of MPLSService for each PE, P, and CE device in the topology. A PE or P device is a router, and a CE device may be a router, a switch, a host, or just about anything that the L2VPN customer wants to connect to the L2VPN. The only restriction is that the CE be a device that can be discovered by the Availability Manager.

The relationships created for an MPLSService instance depend on whether the host device is a CE, PE, or P.

For a CE device (Kompella only):

- MPLSService relationship *AttachedTo*—Points to the VRF to which the CE device is attached.

For a PE router:

- MPLSService relationship *Forwarders* (not shown in Figure 6 or Figure 7)—Points to the Forwarders hosted by the PE router.
- MPLSService relationships *LSPOut* and *LSPIn* (not shown in Figure 6 or Figure 7)—Point to the LSPs for which the PE router is the source or destination (respectively).

For a PE or P router:

- MPLSService relationship *MPLSInterfaces* (not shown in Figure 6 or Figure 7)—Points to all interfaces that are MPLS-enabled.

Post-Processing Discovery for L2VPNs

During this phase of the L2VPN discovery process, which starts after per-device discovery completes, the MPLS Manager creates instances of the following classes by combining the information collected during per-device discovery:

- PseudoWire
- LdpAdjacency (Martini only)
- VPN
- LSP
- LSPHop

PseudoWire

The MPLS Manager matches the VC IDs associated with the Forwarders discovered on different PE devices and creates a PseudoWire instance for each VC ID match.

Thus, a PseudoWire is *TerminatedBy* two Forwarders, where one Forwarder is *HostedBy* one PE device, and the second Forwarder is *HostedBy* another PE device. The PseudoWire provides a bidirectional data path between the two PE devices.

LdpAdjacency (Martini Only)

The MPLS Manager matches various attributes (LocalAddress, PeerAddress) associated with the LdpProtocolEndpoints discovered on different PE devices and creates an LdpAdjacency instance for each complimentary match.

Thus, an LdpAdjacency is *ConnectedTo* two LdpProtocolEndpoints, where one LdpProtocolEndpoint is *HostedBy* one PE device, and the second LdpProtocolEndpoint is *HostedBy* another PE device. The LdpAdjacency provides a bidirectional signaling path through which the two PE devices exchange VC ID and other types of information for the purpose of constructing, advertising, maintaining, or deleting pseudowires between the two PE devices.

The LdpAdjacency *Underlying* relationship points to the *LayeredOver* PseudoWires associated with this LdpAdjacency.

VPN

For discovered Martini-implemented L2VPNs, the MPLS Manager infers VPNs from Forwarder and PseudoWire instances and the relationships between the instances. A Martini-implemented VPN is *ComposedOf* all the Forwarders and PseudoWires comprising the VPN.

The MPLS Manager discovers and creates the following type of VPN for Martini-implemented L2VPNs: point-to-point full mesh. Because the signaling protocol (LDP) used by Martini-implemented L2VPNs does *not* advertise L2VPN membership information among the PE routers, the MPLS Manager cannot determine to which L2VPN a discovered Pseudowire belongs. Accordingly, the MPLS Manager considers each discovered Pseudowire (Martini Tunnel) a point-to-point full-mesh L2VPN.

For discovered Kompella-implemented L2VPNs, the MPLS Manager infers VPNs from Forwarder, PseudoWire, VRF, and RouteTarget instances and the relationships between the instances. The MPLS Manager matches the *Underlying* interfaces of the Forwarders with the *Underlying* interfaces of the VRFs to determine which Forwarders and PseudoWires belong to a VPN. A Kompella-implemented VPN is *ComposedOf* all the Forwarders, PseudoWires, VRFs, and RouteTargets comprising the VPN.

The MPLS Manager discovers and creates the following types of VPN for Kompella-implemented L2VPNs: full mesh and hub and spoke. Because the signaling protocol (MBGP) used by Kompella-implemented L2VPNs *does* advertise L2VPN membership information among the PE routers, the MPLS Manager can determine to which L2VPN a discovered Pseudowire belongs.

LSP

Once the PseudoWires are discovered, the MPLS Manager has all the LSPs associated with the PseudoWires and creates an LSP instance for each of the LSPs that terminate on the PseudoWire termination points (that is, the Forwarders). The *Source* and *Destination* relationships of the LSPs are set appropriately.

Note: For L2VPNs, an LSP acts as a tunnel carrying multiple virtual circuits, and a virtual circuit acts like the actual circuit carrying subscriber Layer 2 frames.

LSPHop

For each LSP instance that the MPLS Manager creates, it also creates the LSPHop instances along the path of the LSP. An LSPHop is *ConnectedTo* an LSPOutSegment and an LSPInSegment. If the LSPHop is the last one for an LSP, it is only *ConnectedTo* an LSPOutSegment (which has a label value of 3).

The MPLS Manager computes the *NextHop* and *PreviousHop* relationships of LSPHop instances. To compute the *NextHop* relationship, the MPLS Manager traces the *ConnectedTo* relationship to the LSPInSegment, then the *SwappedTo* relationship to an LSPOutSegment, and then the *ConnectedVia* relationship to an LSPHop.

3

Preparing for and Initiating Discovery

Configuring an MPLS Manager deployment for discovery requires configuring the MPLS Manager, Availability Manager, and Global Manager applications in the deployment. Initiating discovery for the deployment requires initiating discovery for each Availability Manager application that is to serve as a source to MPLS Manager.

Preparing for Discovery

To prepare your MPLS Manager deployment for discovery, follow these steps:

1. Configure each Availability Manager for discovery as instructed in the *EMC Smarts IP Management Suite Discovery Guide*.
2. Configure the MPLS Manager for discovery as instructed in the *EMC Smarts MPLS Manager Configuration Guide*. The configuration consists of three basic tasks:
   - Configuring CLI login environment variables for CLI discovery
   - Configuring CLI discovery for specific routing devices
   - Adding one or more Availability Managers as sources to the MPLS Manager

3 Configure the Global Manager for discovery as instructed in the *EMC Smarts MPLS Manager Configuration Guide*. For general instructions on configuring the Global Manager, see the *EMC Smarts Service Assurance Manager Configuration Guide*.

Initiating Discovery

After configuring your MPLS Manager deployment for discovery, initiate discovery for each Availability Manager as instructed in the *EMC Smarts IP Management Suite Discovery Guide*.

Then, initiate discovery for the MPLS Manager by starting the MPLS Manager. When the MPLS Manager is registered as a service during installation, you can manually start and stop the MPLS Manager service from the command line. To start the MPLS Manager service named `ic-mpls-server`:

```
# BASEDIR/smarts/bin/sm_service start ic-mpls-server
```

The MPLS Manager starts and registers with the EMC Smarts Broker, reads the *LOCAL.import* file in the ***BASEDIR****/smarts/local/conf/mpls-vpn* directory, and imports the router topology discovered by each Availability Managers listed as a source in the *LOCAL.import* file. In addition, the MPLS Manager instance creates a log file named for the MPLS Manager in the ***BASEDIR****/smarts/local/logs* directory. For example, the MPLS Manager is named INCHARGE-MPLS by default, so the log file would be named *INCHARGE-MPLS.log*.

Note: If the MPLS Manager is *not* registered as a service during installation, you can still manually start and stop the MPLS Manager from the command line. For more information, see the *EMC Smarts MPLS Management Suite Installation Guide*.

After the initial import of router topology from the Availability Manager, the MPLS Manager sends its own SNMP polls and/or CLI commands to the routers to discover the MPLS and VPN information that it needs to build the MPLS and VPN topology. Thereafter, the MPLS Manager imports the router topology from the Availability Manager whenever the Availability Manager completes a discovery cycle.

Each time that the MPLS Manager imports the router topology, it compares the router topology with the previously imported router topology. Whenever the MPLS Manager finds a new routing device or finds an already known routing device having a newer discovery timestamp, the MPLS Manager sends SNMP polls and/or CLI commands to the device to discover/rediscover the device's MPLS and VPN elements and adds those elements to the MPLS and VPN topology.

4

Invoking Discovery Manually

After initial discovery, you can manually invoke a discovery update for the MPLS Manager at any time. You can perform this function through a Domain Manager Administration Console attached to the MPLS Manager, or from the command line.

Opening the Domain Manager Administration Console

Attaching the Global Console to a Domain Manager such as the MPLS Manager requires an InCharge user account with the following privileges and permissions:

- *All* privileges, specified in the *serverConnect.conf* file (or its equivalent) read by the Domain Manager.
- Permission to use the console operation named *Configure Domain Manager Admin Console*. Through the Global Manager Administration Console, this permission is specified in the *Console Operations* section of the user profile.

For information about configuring access privileges, see the *EMC Smarts System Administration Guide*. For information about configuring permissions to perform specific console operations, see the *EMC Smarts Service Assurance Manager Configuration Guide*.

To open the Domain Manager Administration Console, follow these steps:

1. Start the Global Console.
   - On a Windows system, choose *Start > Programs > InCharge 6 > InCharge Global Console*.
   - On a UNIX system, go to the ***BASEDIR****/smarts/bin* directory in the Global Manager installation area and invoke `./sm_gui`.

   The InCharge Manager attach dialog box opens.
2. In the dialog box,
   - Ensure that the Broker for your deployment appears in the Broker text box.
   - Select the MPLS Manager application in your deployment as the EMC Smarts Manager to which you want to connect.
   - Enter your login user name and password.
   - Click **OK**.

   The Topology Browser Console opens.
3. In the Console, choose *Configure > Domain Manager Administration Console*. The Domain Manager Administration Console opens.

Figure 8 is an example of a Domain Manager Administration Console. In the display, the Domain Manager Administration Console is attached to an MPLS Manager named `INCHARGE-MPLS`.

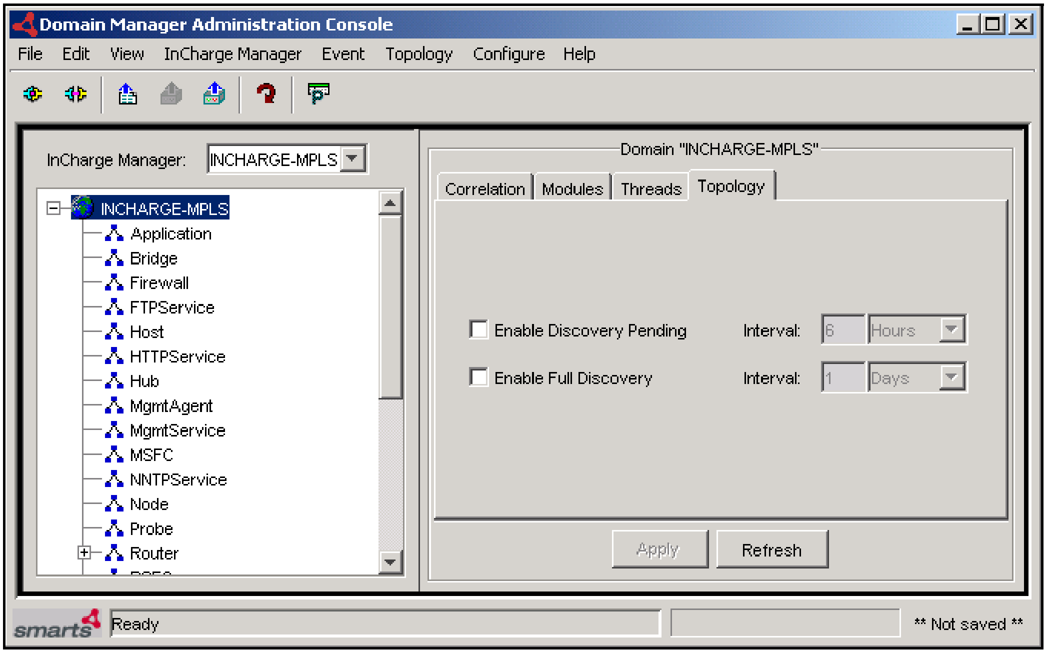

Figure 8: Domain Manager Administration Console Attached to an MPLS Manager—Example

Using the Domain Manager Administration Console

Most administration capabilities normally available through the Domain Manager Administration Console are available to administrators of the MPLS Manager. However, because the MPLS Manager imports its topology and topology updates from the source Availability Manager, the following features are either not available through the Domain Manager Administration Console (Figure 8) or should not be performed for the MPLS Manager through the Domain Manager Administration Console:

- Autodiscovery

  The *Enable Auto Discovery* checkbox, normally available through the *Topology* tab on the Domain Manager Administration Console, and the *Import from seed file* and *Add Agent* menu options, normally available through the *Topology* menu on the Domain Manager Administration Console, are not available to an administrator of the MPLS Manager.

Without this checkbox and these menu options, an administrator cannot enable or initiate autodiscovery for the MPLS Manager. In addition, without the menu options, an administrator cannot initiate manual discovery to add new routing devices to the MPLS Manager's repository.

- Enabling and scheduling automatic discovery

  Because the MPLS Manager performs discovery whenever the source Availability Manager performs discovery, automatic discovery (if used) should be enabled and scheduled for the Availability Manager, through the *Topology* tab of a Domain Manager Administration Console attached to the Availability Manager.

For detailed descriptions of the administration capabilities available through the Domain Manager Administration Console, see the *EMC Smarts Service Assurance Manager Operator's Guide*.

Invoking a Manual Discovery Update

There are several methods for manually invoking a discovery update from the Domain Manager Administration Console, including:

- Choose an element, expand its HostsServices relationship, select the SNMPAgent, and then select *Topology > Rediscover* to initiate a rediscovery of the selected element. You can also rediscover an element by right-clicking its SNMPAgent and selecting **Rediscover** from the drop-down menu.
- Select *Topology > Discover All* to initiate a rediscovery of all elements in a Domain Manager's repository.

During an MPLS Manager discovery update, initiated through a Domain Manager Administration Console attached to the MPLS Manager, a Discovery Progress window opens and displays progress messages. When you see *Last discovery completed* near the end of the progress report in the Discovery Status section, the probing of topology information is complete. If any errors occur during the discovery process, see *Resolving Discovery Errors* on page 31. Click **Close** to close the Discovery Progress window.

You can also invoke a full discovery update using the *dmctl* utility. Invoke this command from the ***BASEDIR***/*smarts/bin* directory in the MPLS Manager installation area:

```
# ▼ ./dmctl -s <MPLS_Manager_name> invoke
     ICF_TopologyManager::ICF-TopologyManager discoverAll ▲
```

▼▲ Indicates that the command must be typed as one line.

Resolving Discovery Errors

When the MPLS Manager attempts to discover the MPLS and VPN elements for a routing device, one of the following results occurs:

- Discovery successfully completes, and the MPLS and VPN elements associated with the routing device are added to the managed topology.
- Discovery starts but communication is lost between the MPLS Manager and the routing device.
- Discovery starts but the routing device does not respond to SNMP polls and/or CLI commands.
- Discovery starts but one or more of the MPLS Manager probes encounters an error during the discovery process.
- Discovery is unable to start or successfully complete.

When a discovery error occurs, follow these basic steps:

1. On a Domain Manager Administration Console attached to the MPLS Manager, click *Configure > Show Discovery Progress* to launch the Discovery Progress window.
2. Check the messages in the Discovery Status section of the window.
3. Check the log file named *<MPLS_Manager_name>.log* in the ***BASEDIR***/*smarts/local/logs* directory. The discovery process writes an error to the log when it encounters an error and discovery is unsuccessful.

Note: MPLS Manager does not notify discovery errors.

For descriptions and resolutions of common discovery errors, see the *EMC Smarts IP Management Suite Discovery Guide*.

Index