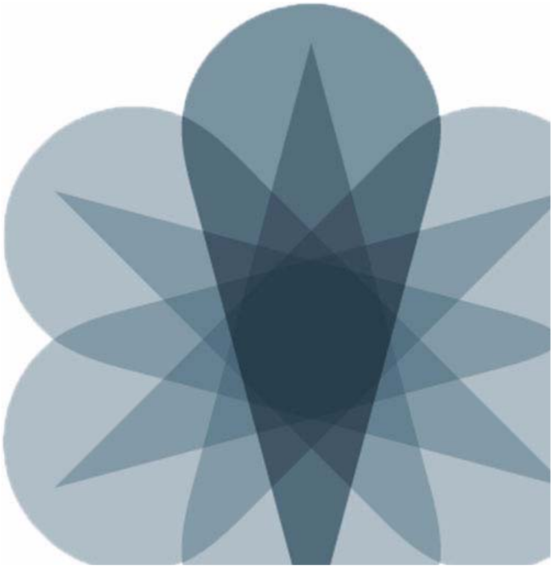
JNCIS-SP Study Guide—Part 3

Worldwide Education Services

1194 North Mathilda Avenue
Sunnyvale, CA 94089
USA
408-745-2000
www.juniper.net



This document is produced by Juniper Networks, Inc.

JNCIS-SP Study Guide—Part 3

Printed in USA.

The information in this document is current as of the date listed above.

The information in this document has been carefully verified and is believed to be accurate for software Release 10.3R1.9. Juniper Networks assumes no responsibilities for any inaccuracies that may appear in this document. In no event will Juniper Networks be liable for direct, indirect, special, exemplary, incidental, or consequential damages resulting from any defect or omission in this document, even if advised of the possibility of such damages.

Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

YEAR 2000 NOTICE

Juniper Networks hardware and software products do not suffer from Year 2000 problems and hence are Year 2000 compliant. The Junos operating system has no known time-related limitations through the year 2038. However, the NTP application is known to have some difficulty in the year 2036.

SOFTWARE LICENSE

The terms and conditions for using Juniper Networks software are described in the software license provided with the software, or to the extent applicable, in an agreement executed between you and Juniper Networks, or Juniper Networks agent. By using Juniper Networks software, you indicate that you understand and agree to be bound by its license terms and conditions. Generally speaking, the software license restricts the manner in which you are permitted to use the Juniper Networks software, may contain prohibitions against certain uses, and may state conditions under which the license is automatically terminated. You should consult the software license for further details.

Contents

Overview

Welcome to the JNCIS-SP Study Guide—Part 3. The purpose of this guide is to help you prepare for your JN0-360 exam and achieve your JNCIS-SP credential. The contents of this document are based on the *Junos MPLS and VPNs* (JMV) course. This study guide is designed to provide MPLS-based virtual private network (VPN) knowledge and configuration examples. The content includes an overview of MPLS concepts such as control and forwarding plane, RSVP Traffic Engineering, LDP, Layer 3 VPNs, next-generation multicast virtual private networks (MVPNs), BGP Layer 2 VPNs, LDP Layer 2 Circuits, and virtual private LAN service (VPLS). This study guide also covers Junos operating system-specific implementations of Layer 2 control instances and active interface for VPLS. This guide is based on the Junos OS Release 10.3R1.9.

Document Conventions

CLI and GUI Text

Frequently throughout this study guide, we refer to text that appears in a command-line interface (CLI) or a graphical user interface (GUI). To make the language of these documents easier to read, we distinguish GUI and CLI text from chapter text according to the following table.

Style	Description	Usage Example
Franklin Gothic	Normal text.	Most of what you read in the Study Guide.

`Courier New`	Console text:
	• Screen captures
	• Noncommand-related syntax
	GUI text elements:
	• Menu names
	• Text field entry
		`commit complete`
		`Exiting configuration mode`
		Select `File > Open`, and then click `Configuration.conf` in the `Filename` text box.

Input Text Versus Output Text

You will also frequently see cases where you must enter input text yourself. Often these instances will be shown in the context of where you must enter them. We use bold style to distinguish text that is input versus text that is simply displayed.

Style	Description	Usage Example

`Normal CLI`
`Normal GUI`
	No distinguishing variant.	`Physical interface:fxp0, Enabled`
		View configuration history by clicking `Configuration > History`.

`CLI Input`
`GUI Input`
	Text that you must enter.	`lab@San_Jose>` **`show route`**
		Select `File > Save`, and type **`config.ini`** in the `Filename` field.

Defined and Undefined Syntax Variables

Finally, this study guide distinguishes between regular text and syntax variables, and it also distinguishes between syntax variables where the value is already assigned (defined variables) and syntax variables where you must assign the value (undefined variables). Note that these styles can be combined with the input style as well.

Style	Description	Usage Example

`CLI Variable`
`GUI Variable`
	Text where variable value is already assigned.	`policy` *`my-peers`*
		Click *`my-peers`* in the dialog.

`CLI Undefined`
`GUI Undefined`
	Text where the variable's value is the user's discretion and text where the variable's value as shown in the lab guide might differ from the value the user must input.	Type **`set policy`** ***`policy-name`***.
		`ping 10.0.`***`x.y`***
		Select `File > Save`, and type ***`filename`*** in the `Filename` field.

Additional Information

Education Services Offerings

You can obtain information on the latest Education Services offerings, course dates, and class locations from the World Wide Web by pointing your Web browser to: http://www.juniper.net/training/education/.

About This Publication

The *JNCIS-SP Study Guide—Part 3* was developed and tested using software Release 10.3R1.9. Previous and later versions of software might behave differently so you should always consult the documentation and release notes for the version of code you are running before reporting errors.

This document is written and maintained by the Juniper Networks Education Services development team. Please send questions and suggestions for improvement to training@juniper.net.

Technical Publications

You can print technical manuals and release notes directly from the Internet in a variety of formats:

- Go to http://www.juniper.net/techpubs/.
- Locate the specific software or hardware release and title you need, and choose the format in which you want to view or print the document.

Documentation sets and CDs are available through your local Juniper Networks sales office or account representative.

Juniper Networks Support

For technical support, contact Juniper Networks at http://www.juniper.net/customers/support/, or at 1-888-314-JTAC (within the United States) or 408-745-2121 (from outside the United States).

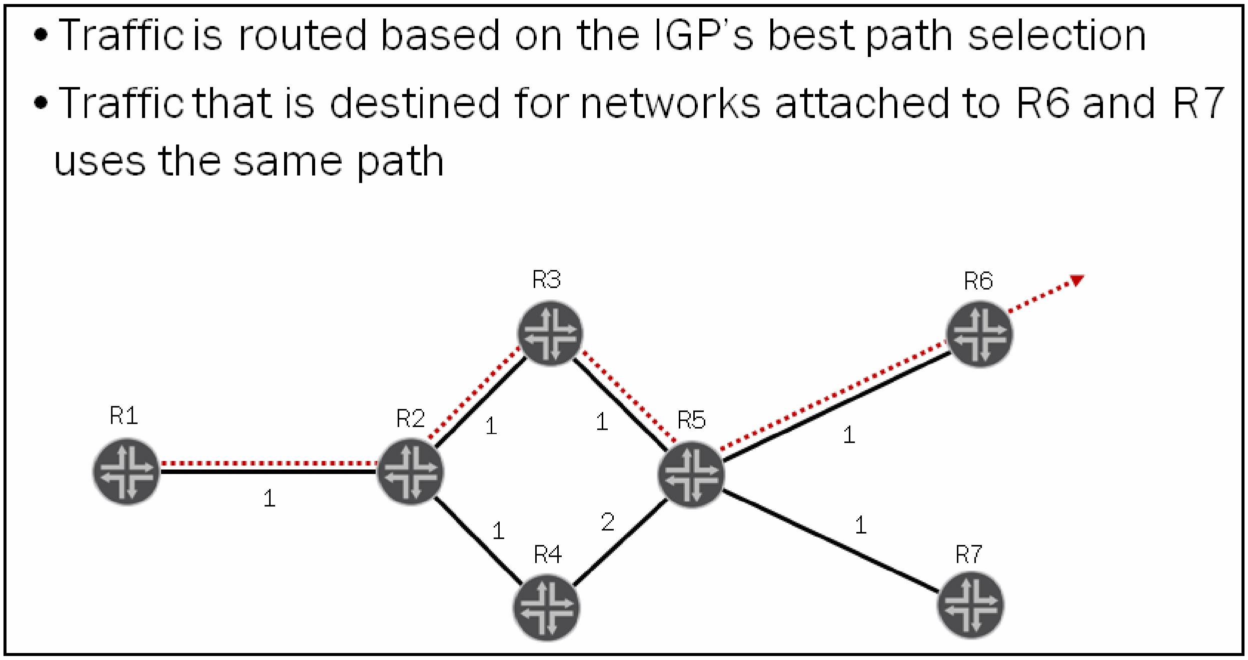
JNCIS-SP Study Guide—Part 3

Chapter 1: MPLS Fundamentals

This Chapter Discusses:

- Common terms relating to MPLS;
- Routers and the way they forward MPLS packets;
- Packet flow and handling through a label-switched path (LSP);
- Configuration and verification of MPLS forwarding; and
- Understanding the information in the Label Information Base (LIB).

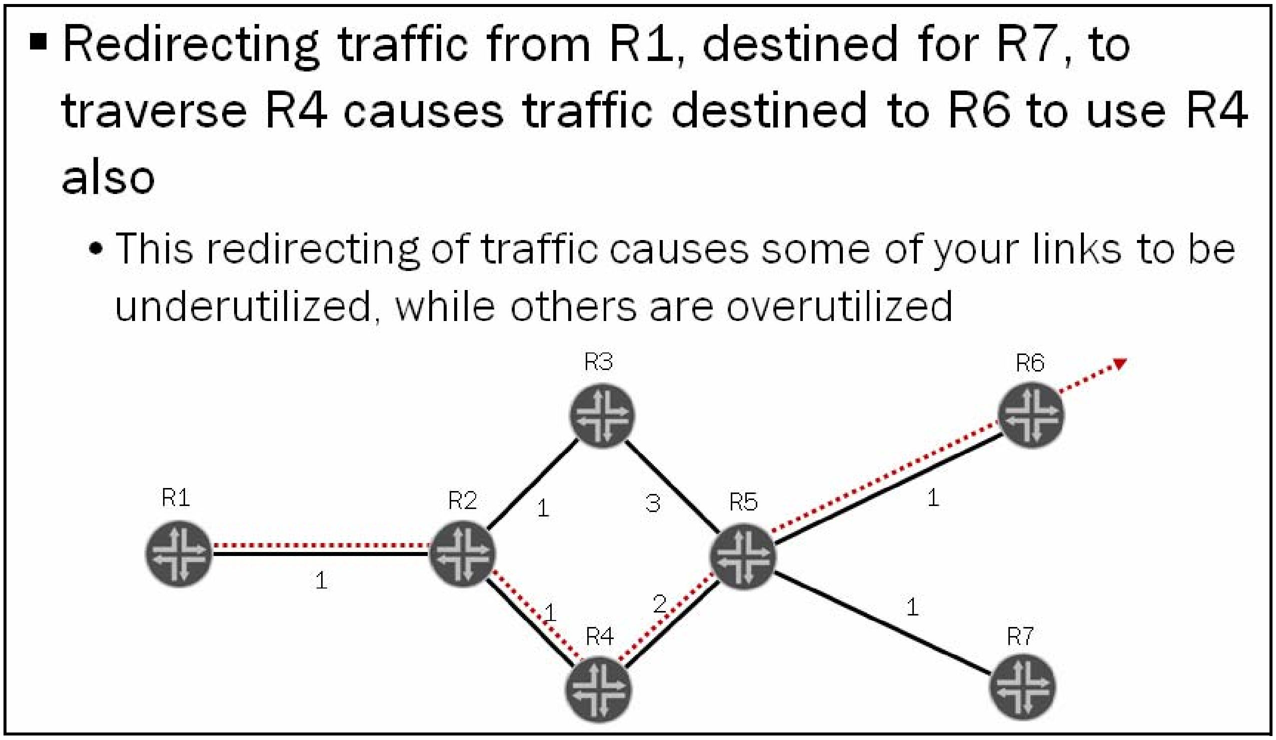
IGP Forwarding



- Traffic is routed based on the IGP's best path selection
- Traffic that is destined for networks attached to R6 and R7 uses the same path

The graphic shows metric-based traffic engineering in action. When sending traffic from a network connected to R1 to a network connected to R6, traffic is routed through R3 because it has a lower overall cost (3, as opposed to 4, through R4). Note that not only the traffic destined for networks connected to R6 follow the upper path, but also all traffic for networks connected to R7 and any routers downstream from these routers.

Redirecting Traffic



- Redirecting traffic from R1, destined for R7, to traverse R4 causes traffic destined to R6 to use R4 also
  - This redirecting of traffic causes some of your links to be underutilized, while others are overutilized

At some point, sending all of the traffic for R6 and R7 and points beyond through the R3 might not be the best idea. For example, a lot of local traffic to the R3 might exist, and this traffic might delay the traffic to R6 and R7 while the path through R4 is underutilized. Whatever the actual cause, you might want to route at least some of the traffic to some destinations over the lower links and through R4. Suppose traffic for R7 needs to be rerouted onto this lower path.

Rerouting traffic for R7 by raising metrics along the current path, as shown in the graphic, has the desired effect. Traffic to R7 now follows the path with cost 4 instead of cost 5. But forcing the traffic to use R4, by raising the metric on the upper path, has the unintended effect of causing traffic destined for R6 to do the same and flow through R4.

Because interior gateway protocol (IGP) route calculation is topology driven and based on a simple additive metric, such as the hop count or an administrative value, the traffic patterns on the network are not taken into account when the IGP calculates its forwarding table. As a result, traffic will not be evenly distributed across the network's links, causing inefficient use of expensive resources. Some links may become congested, while other links remain underutilized. This result might be satisfactory in a smaller network with less traffic, but in larger networks or networks with many connections, you must control the paths that traffic takes in order to balance the traffic load. In other words, you need more control for realistic traffic engineering than the usual IGP method of sending *all* traffic to a group of destinations over the same, single *best* path.

Possible Destabilization

- Adjusting the IGP metric might destabilize the network
  - Moves the problem to another section of the network
    - Some of the links will be underutilized
    - Some of the links will be congested and overutilized
  - Lacks control
    - All traffic flows over the IGP shortest path

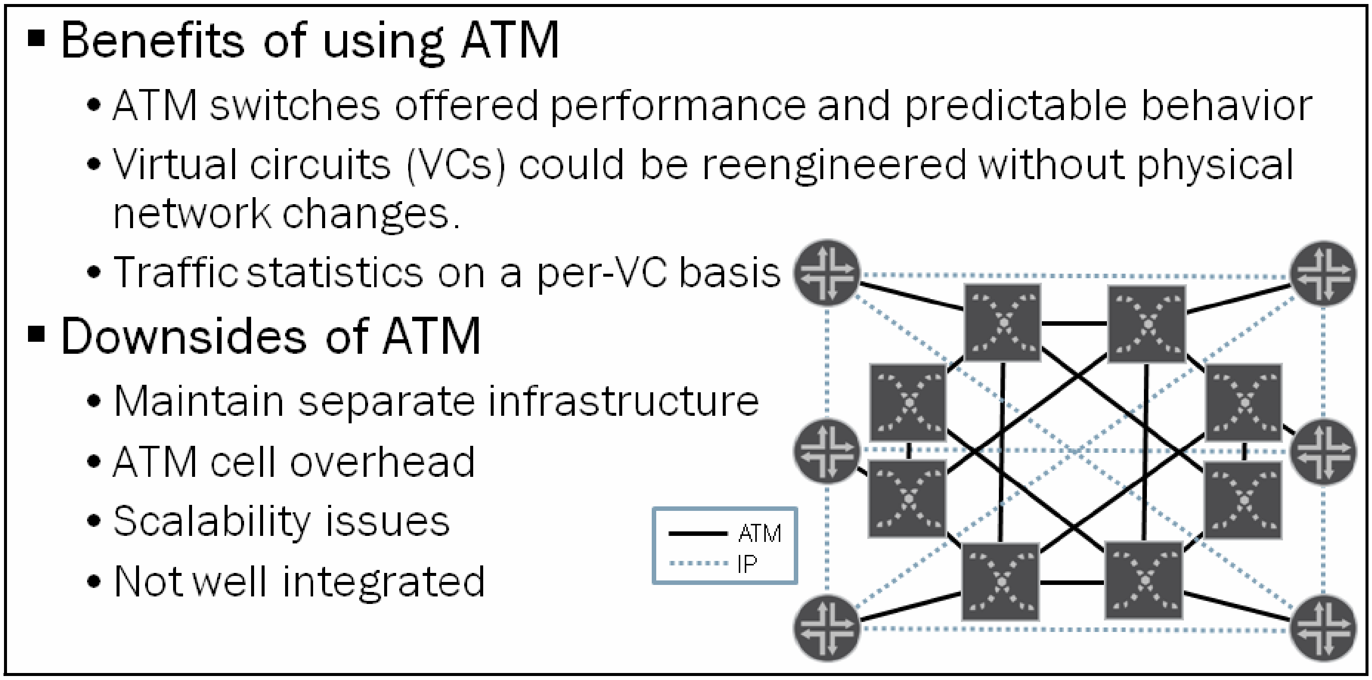
Changing of IGP metric to force traffic path movements has more drawbacks than just moving the traffic to downstream destinations along with the target to the new path. Adjusting metrics manually can have a severe destabilizing effect on a network, especially a large one. As Internet service provider (ISP) networks became more richly connected, it became more difficult to ensure that a metric adjustment in one part of the network did not cause problems in another part of the network. Adjusting metrics just tended to move problems around. The low-cost links and paths became saturated, while the higher-cost links and paths remained almost devoid of traffic.

There was little to no real control over the process. All traffic followed the path with the lowest IGP metric because no other standard mechanism to distribute traffic flow existed. There were no rules and few guidelines to follow about which metrics to adjust and by how much to adjust them. Traffic engineering based on metric manipulation offered a trial-and-error approach, rather than a scientific solution to an increasingly complex problem.

ATM Switched Networks

Despite the obvious drawbacks to manual traffic engineering through IGP metric adjustments, metric-based traffic controls continued to be an adequate traffic engineering solution until the mid-'90s. Most ISPs turned to Asynchronous Transfer Mode (ATM) as their core technology. ATM is also referred to as an *Overlay Network*, which indicates there are multiple networks working in parallel to forward traffic.

Benefits of ATM



- Benefits of using ATM
  - ATM switches offered performance and predictable behavior
  - Virtual circuits (VCs) could be reengineered without physical network changes.
  - Traffic statistics on a per-VC basis
- Downsides of ATM
  - Maintain separate infrastructure
  - ATM cell overhead
  - Scalability issues
  - Not well integrated

ATM switches use what are called virtual circuits (VCs) to logically connect the routers and forward traffic. From the perspective of the routers these VCs are viewed as point-to-point connections, but as you can see the physical topology can be much more complicated. If a section within the network is deemed to be overutilized then the VCs can be altered, moving traffic to a less utilized section, without changing the topology from the routers perspective. Another benefit for using an ATM network is the ability to gather statistics on a per-VC basis. With standard IGP routing there was no way to gather relevant statistics because all traffic either entering or leaving the router was counted. Being able to count the traffic entering or leaving a VC allowed the ISPs to evaluate the network load of each VC and engineer their network accordingly.

Downsides of ATM

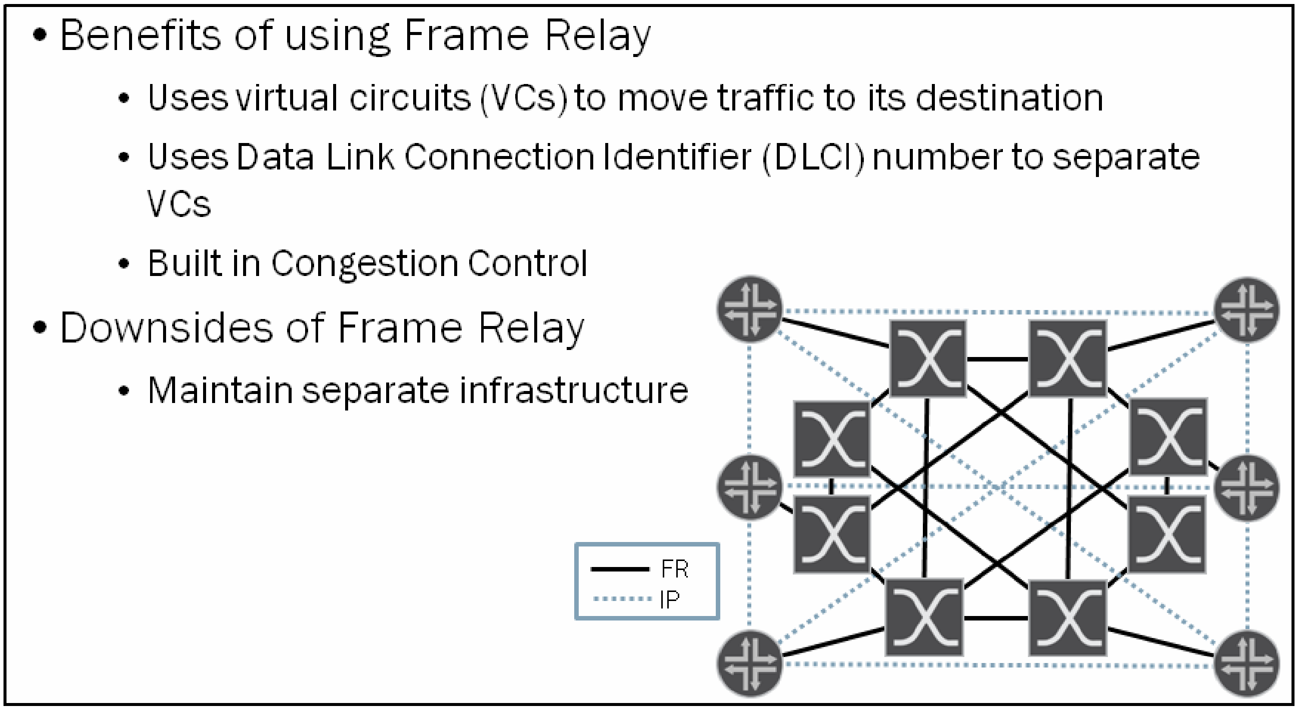
One of the downsides to running an ATM overlay network is that each of the different core technologies (ATM and IP) required separate expert engineers and support staff to address the problems in their platforms.

Another downside is that, ATM cell overhead (often called the ATM *cell tax*) is introduced when packet-oriented protocols, such as IP, are carried over an ATM infrastructure. ATM overhead is never less than about 10% and sometimes as high as 62% (a 40-byte TCP/IP acknowledgment packet requires 106 bytes of ATM on the wire when using AAL 5 and multi protocol encapsulation). Assuming 20% overhead for ATM running on a 2.488-Gbps OC-48 link, 1.99 Gbps is available for customer data, and 498 Mbps—almost a full OC-12—is required for the ATM overhead. On a 10-Gbps OC-192 interface, some 1.99 Gbps—almost a full OC-48 of the link's capacity—is consumed by ATM overhead!

A network that deploys a full mesh of ATM VCs exhibits the traditional n2 problem for the number of links to be maintained (*n* x (*n*-1))/2) where n is the number of routers. For relatively small or moderately sized networks, this problem is not a major issue. However, for core ISPs with hundreds of attached routers, the challenge is quite significant. For example, when expanding a network from five to six routers, an ISP must increase the number of VCs from 20 to 30. However, increasing the number of attached routers from 200 to 201 requires the addition of 400 new VCs—an increase from 39,800 to 40,200 VCs. These numbers do not include backup VCs or additional VCs for networks running multiple services that require more than one VC between any two routers.

ATM VCs are not integrated with the IGP either. Thus, deploying full-mesh of VCs also stresses the IGP. This stress results from the number of peer IGP relationships that must be maintained, the challenge of processing n^3 link-state updates in the event of a failure, and the complexity of performing the Dijkstra calculation over a topology containing a significant number of logical links. As an ATM core expands, the n^2 stress on the IGP compounds.

Frame Relay Networks

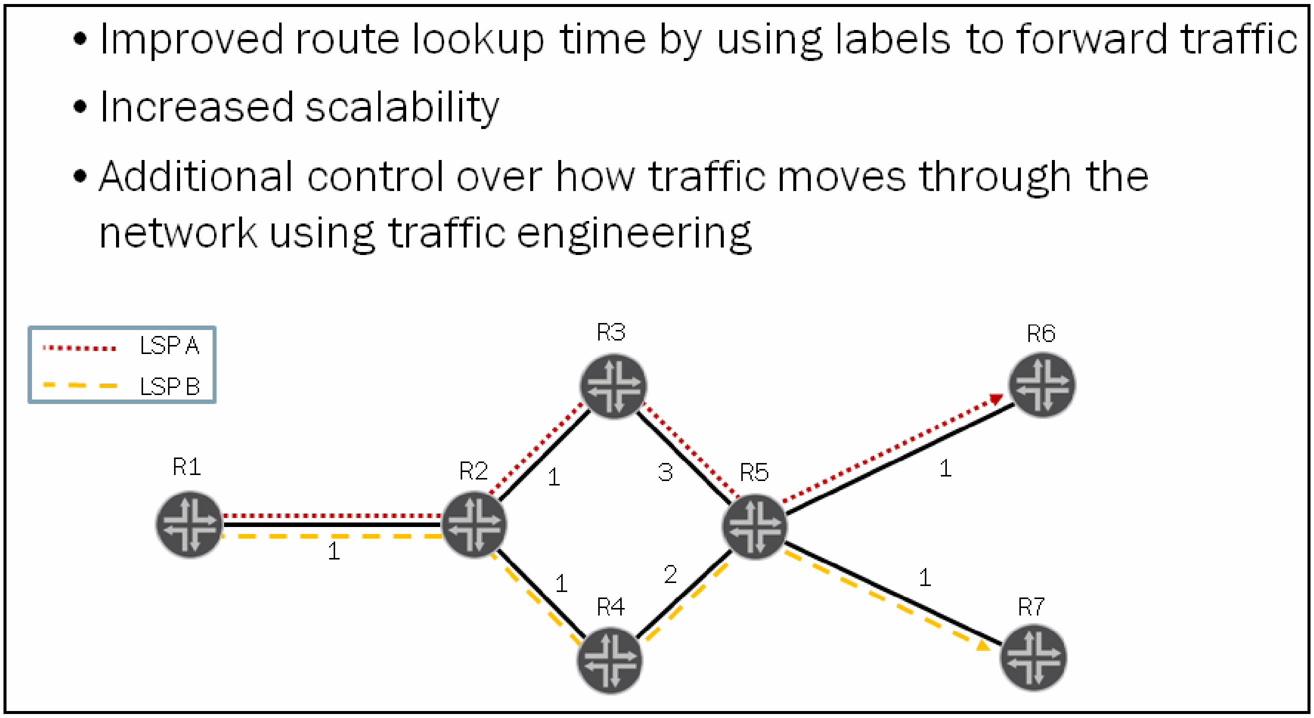


- Benefits of using Frame Relay
  - Uses virtual circuits (VCs) to move traffic to its destination
  - Uses Data Link Connection Identifier (DLCI) number to separate VCs
  - Built in Congestion Control
- Downsides of Frame Relay
  - Maintain separate infrastructure

Frame relay networks are also an overlay network. Frame Relay also use virtual circuits to create logical connections between routers. Frame Relay uses a unique data-link connection identifier (DLCI) number to separate one VC from another. Frame relay also has a built in congestion control mechanism.

Frame relay also has its downsides. Similar to ATM a Frame Relay switch network is running multiple core technologies (Frame Relay and IP) and each one required separate expert engineers and support staff to address the problems in their platforms.

Benefits of MPLS



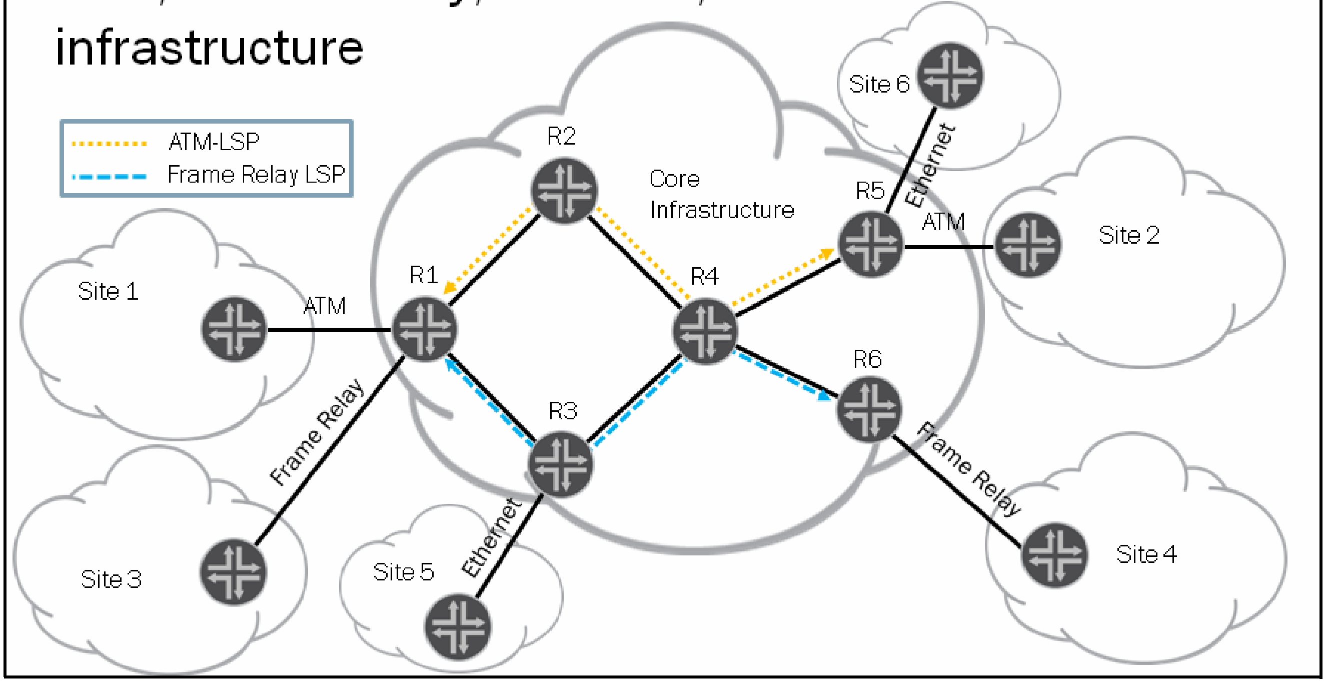
- Improved route lookup time by using labels to forward traffic
- Increased scalability
- Additional control over how traffic moves through the network using traffic engineering

Because core routing platforms and link speed increased so much within a few years the benefits of running a core of ATM switches was no longer being seen. Routers are as fast, if not faster, than the speediest ATM switch. High-speed interfaces, deterministic performance, and traffic engineering using VCs no longer distinguish ATM switches from Internet backbone routers. The deployment of a router-based core solves a number of inherent problems with the ATM model: the complexity and expense of coordinating two sets of equipment, the bandwidth limitations of ATM segmentation and reassembly (SAR) interfaces, the cell tax, the n^2 VC problem, the IGP stress, and the limitation of not being able to operate over a mixed-media infrastructure.

MPLS was originally designed to make IP routers as fast as ATM switches for handling traffic. MPLS uses label values to make its forwarding decisions as traffic traverses the network. It is still commonly believed that MPLS somehow significantly enhances the forwarding performance of label-switching routers. However, it is more accurate to say that exact-match lookups, such as those performed by MPLS and ATM switches, historically have been faster than the longest-match lookups performed by IP routers.

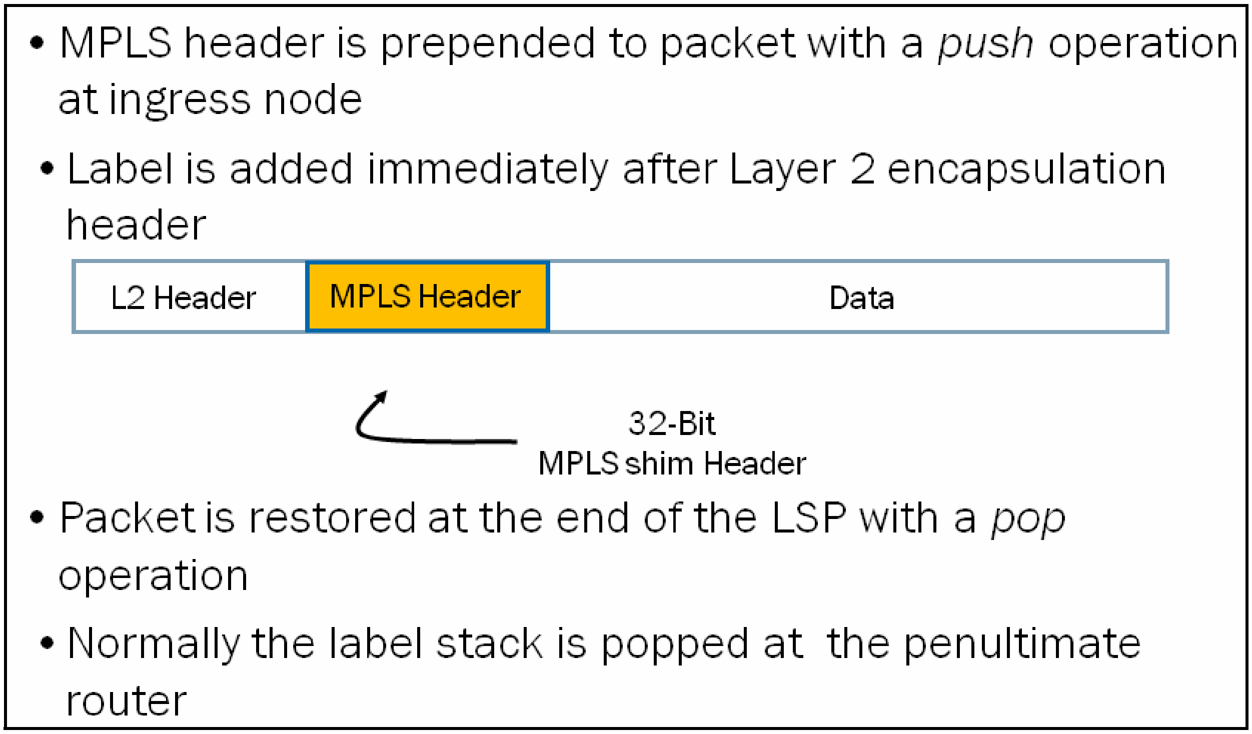
In any case, recent advances in silicon technology allow application-specific integrated circuit (ASIC)-based route-lookup engines to run just as fast as MPLS or ATM virtual path identifier (VPI)/virtual channel identifier (VCI) lookup engines, so MPLS is no longer seen as just a faster way of routing.

- Service Providers can offer different technologies like ATM, Frame Relay, Ethernet, and IPsec over the same infrastructure



The real benefit of MPLS is that it provides a clean separation between routing (that is, control) and forwarding (that is, moving data). This separation allows the deployment of a single forwarding algorithm—MPLS—that can be used for multiple services and traffic types. In the future, as ISPs must develop new revenue-generating services, the MPLS forwarding infrastructure can remain the same, while new services are built by simply changing the way packets are assigned to an LSP. For example, packets can be assigned to a label-switched path based on a combination of the destination subnetwork and application type, a combination of the source and destination subnetworks, a specific quality-of-service (QoS) requirement, an IP multicast group, or a virtual private network (VPN) identifier. In this manner, new services can be migrated easily to operate over the common MPLS forwarding infrastructure.

MPLS Packet Header



- MPLS header is prepended to packet with a *push* operation at ingress node
  - Label is added immediately after Layer 2 encapsulation header

L2 Header
MPLS Header
Data
32-Bit
MPLS shim Header

- Packet is restored at the end of the LSP with a *pop* operation
  - Normally the label stack is popped at the penultimate router

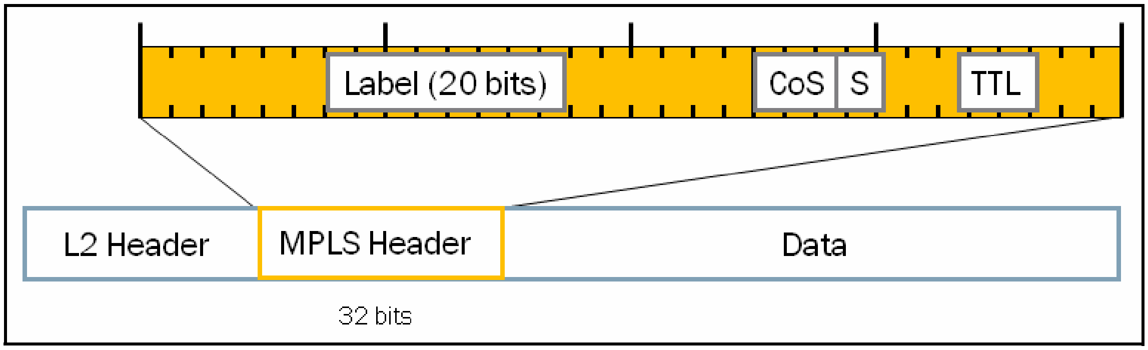
MPLS is responsible for directing a flow of IP packets along a predetermined path across a network. This path is the LSP, which is similar to an ATM VC in that it is unidirectional. That is, the traffic flows in one direction from the ingress router to an egress router. Duplex traffic requires two LSPs—that is, one path to carry traffic in each direction. An LSP is created by the concatenation of one or more label-switched hops that direct packets between LSRs to transit the MPLS domain.

When an IP packet enters a label-switched path, the ingress router examines the packet and assigns it a label based on its destination, placing a 32-bit (4-byte) label in front of the packet's header immediately after the Layer 2 encapsulation. The label transforms the packet from one that is forwarded based on IP addressing to one that is forwarded based on the fixed-length label. The graphic shows an example of a labeled IP packet. Note that MPLS can be used to label non-IP traffic, such as in the case of a Layer 2 VPN.

MPLS labels can be assigned per interface or per router. The Junos operating system currently assigns MPLS label values on a per-router basis. Thus, a label value of 10234 can only be assigned once by a given Juniper Networks router. Multicast and IPv6 labels are assigned independently of unicast packet labels. The Junos OS currently does not support labeled multicast or IPv6, except in the context of a Layer 2 or Layer 3 VPN.

At egress the IP packet is restored when the MPLS label is removed as part of a pop operation. The now unlabeled packet is routed based on a longest-match IP address lookup. In most cases, the penultimate (or second to last) router pops the label stack in penultimate hop popping. In some cases, a labeled packet is delivered to the ultimate router—the egress label-switching router (LSR)—when the stack is popped, and the packet is forwarded using conventional IP routing.

The MPLS Header (Label) Structure



The 32-bit MPLS header consists of the following four fields:

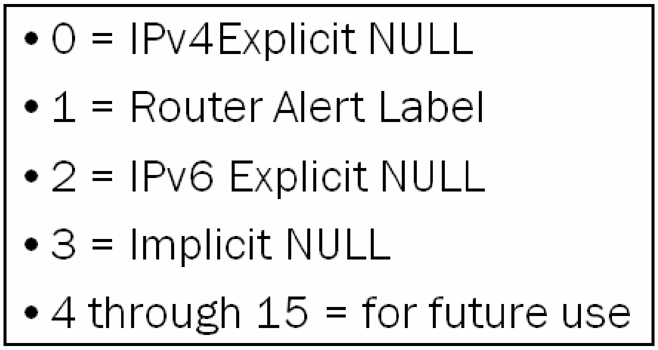
- *20-bit label*: Identifies the packet to a particular LSP. This value changes as the packet flows on the LSP from LSR to LSR.
- *Class of service (CoS) (experimental)*: Indicates queuing priority through the network. This field was initially just the CoS field, but lack of standard definitions and use led to the current designation of this field as experimental. In other words, this field was always intended for CoS, but which type of CoS is still experimental. At each hop along the way, the CoS value determines which packets receive preferential treatment within the tunnel.
- *Bottom of stack bit*: Indicates whether this MPLS packet has more than one label associated with it. The MPLS implementation in the Junos OS supports unlimited label stack depths for transit LSR operations. At ingress up to three labels can be pushed onto a packet. The *bottom* of the stack of MPLS labels is indicated by a 1 bit in this field; a setting of 1 tells the LSR that after popping the label stack an unlabeled packet will remain.
- *Time to live (TTL)*: Contains a limit on the number of router hops this MPLS packet can travel through the network. It is decremented at each hop, and if the TTL value drops below 1, the packet is discarded. The default behavior is to copy the value of the IP packet into this field at the ingress router.

Key Points to Remember about MPLS Labels

The following are some of the key points to remember about working with MPLS labels:

- MPLS labels can be either assigned manually or set up by a signaling protocol running in each LSR along the path of the LSP. Once the LSP is set up, the ingress router and all subsequent routers in the LSP do not examine the IP routing information in the labeled packet—they use the label to look up information in their label forwarding tables. Changing Labels by Segment
- Much as with ATM VCIs, MPLS label values change at each segment of the LSP. A single router can be part of multiple LSPs. It can be the ingress or egress router for one or more LSPs, and it also can be a transit router of one or more LSPs. The functions that each router supports depend on the network design.
- The LSRs replace the old label with a new label in a swap operation and then forward the packet to the next router in the path. When the packet reaches the LSP's egress point, it is forwarded again based on longest-match IP forwarding.
- There is nothing unique or special about most of the label values used in MPLS. We say that labels have *local significance*, meaning that a label value of 10254, for example, identifies one LSP on one router, and the same value can identify a different LSP on another router.

Reserved MPLS Label Values



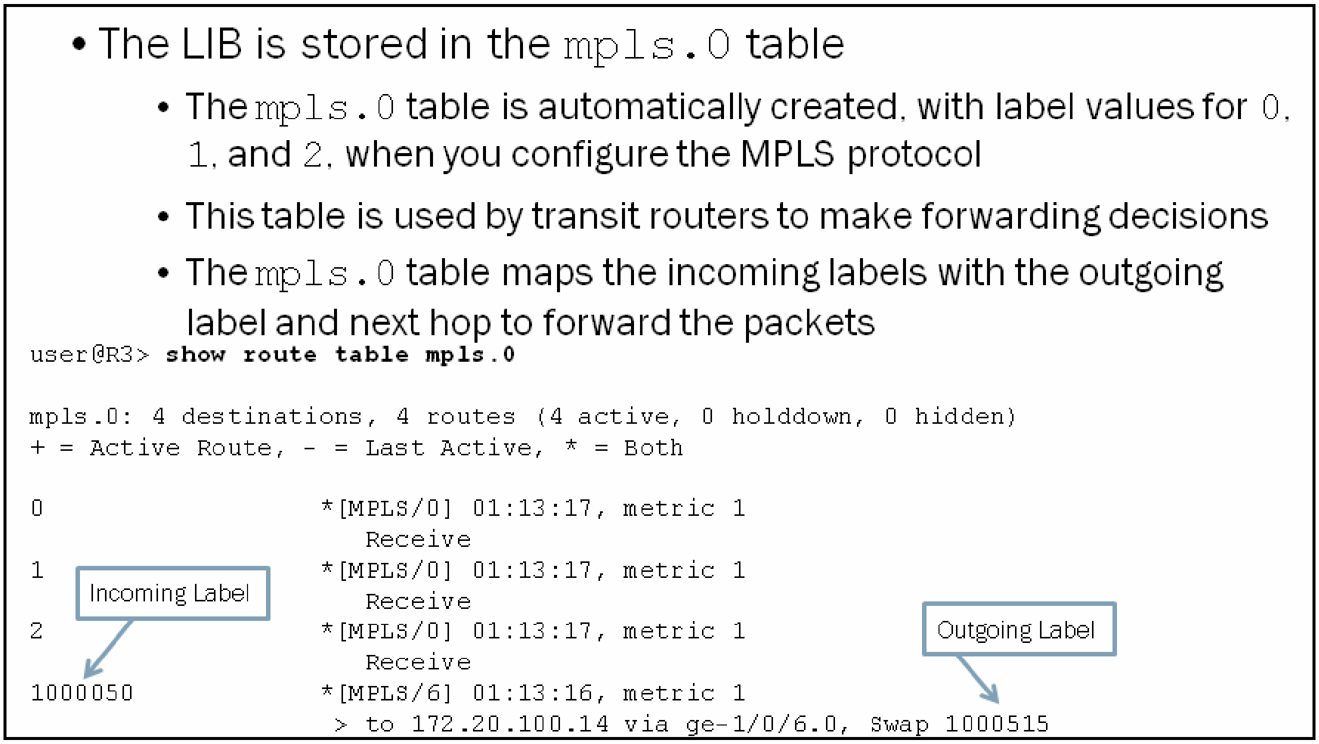
- 0 = IPv4Explicit NULL
- 1 = Router Alert Label
- 2 = IPv6 Explicit NULL
- 3 = Implicit NULL
- 4 through 15 = for future use

Labels 0 through 15 are reserved according to the procedures outlined in RFC 3032, *MPLS Label Stack Encoding*.

- A value of 0 represents the *IP version 4 (IPv4) explicit null label*. This label value is legal only when it is the sole label stack entry. It indicates that the label stack must be popped, and the forwarding of the packet must then be based on the IPv4 header.
- A value of 1 represents the *router alert label*. This label value is legal anywhere in the label stack except at the bottom. When a received packet contains this label value at the top of the label stack, it is delivered to a local software module for processing. The label beneath it in the stack determines the actual forwarding of the packet. However, if the packet is forwarded further, the router alert label should be pushed back onto the label stack before forwarding. The use of this label is analogous to the use of the *router alert option* in IP packets. Because this label cannot occur at the bottom of the stack, it is not associated with a particular network layer protocol. Essentially, label value 1 gives MPLS modules in different routers a way to communicate with each other.

- A value of 2 represents the *IP version 6 (IPv6) explicit null* label. This label value is legal only when it is the sole label stack entry. It indicates that the label stack must be popped, and the forwarding of the packet then must be based on the IPv6 header.
- A value of 3 represents the *implicit null label*. This is a label that an LSR can assign and distribute, but it never actually appears in the encapsulation. When an LSR would otherwise replace the label at the top of the stack with a new label, but the new label is implicit null, the LSR pops the stack instead of doing the replacement. Although this value might never appear in the encapsulation, it must be specified in the label signaling protocol, so a value is reserved.
- Values 4–15 are reserved for future use.

Label Information Base



- The LIB is stored in the `mpls.0` table
  - The `mpls.0` table is automatically created, with label values for `0`, `1`, and `2`, when you configure the MPLS protocol
  - This table is used by transit routers to make forwarding decisions
  - The `mpls.0` table maps the incoming labels with the outgoing label and next hop to forward the packets

```
user@R3> show route table mpls.0

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 01:13:17, metric 1
                      Receive
1                  *[MPLS/0] 01:13:17, metric 1
                      Receive
2                  *[MPLS/0] 01:13:17, metric 1
                      Receive
1000050            *[MPLS/6] 01:13:16, metric 1
                    > to 172.20.100.14 via ge-1/0/6.0, Swap 1000515
```

Incoming Label

Outgoing Label

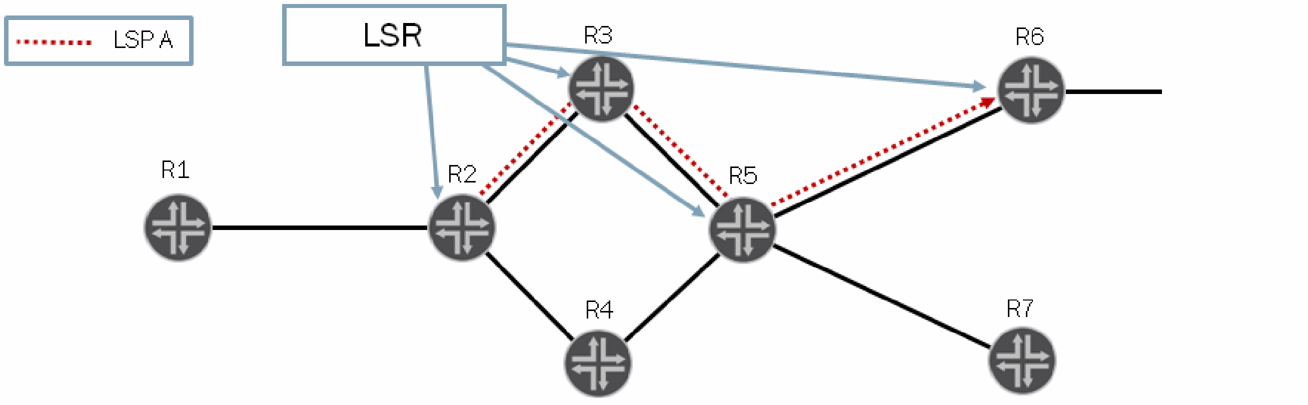
The LIB and mappings are stored in the `mpls.0` routing table. When you configure the MPLS protocol, the software automatically creates this table. When it creates this table it installs three default labels in this table. Packets received with these label values are sent to the Routing Engine for processing. As mentioned earlier, Label 0 is the IPv4 explicit null label, Label 1 is the MPLS equivalent of the IP Router Alert label and Label 2 is the IPv6 explicit null label.

The transit routers use this table to make forwarding decisions based on the incoming label. The router will consult this table and determine what the next-hop should be and what the outgoing label should be. This happens at each transit router to ensure the traffic is traversing the correct path through the network.

In the sample output you can see the three default labels are created with the action of `Receive`. You will also notice there is an incoming label value of 1000050, which indicates that the next-hop is 172.20.100.14 via interface ge-1/0/6. The output also indicates that this particular router will `swap` the label with 1000515 before sending the packet on to the next router in the path.

Label-Switching Routers

- All M Series Routers, T Series Routers, and MX Series Ethernet Services Routers support LSR capabilities
  - Simply called *routers* in this content

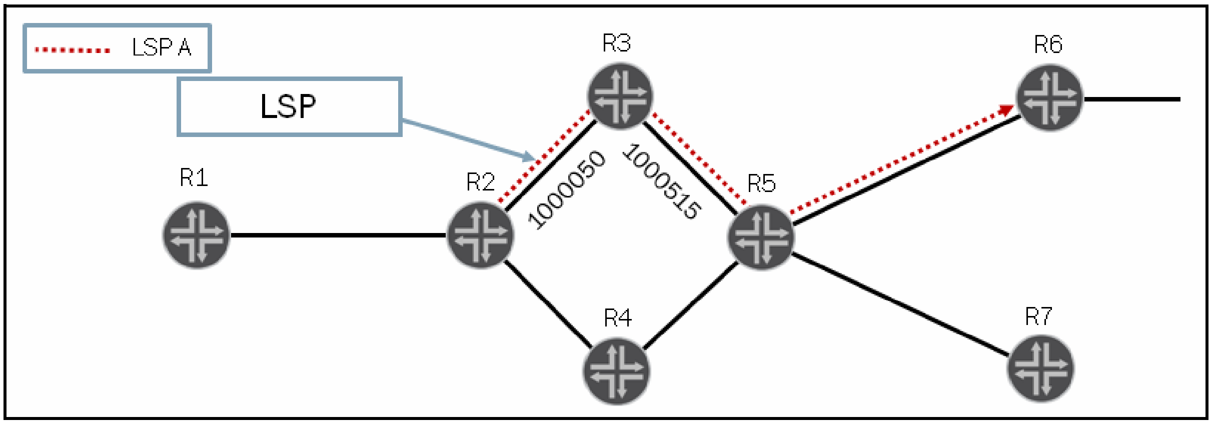


An LSR understands and forwards MPLS packets, which flow on, and are part of, an LSP. In addition, an LSR participates in constructing LSPs for the portion of each LSP entering and leaving the LSR. For a particular destination, an LSR can be at the start of an LSP, the end of an LSP, or in the middle of an LSP. An individual router can perform one, two, or all of these roles as required for various LSPs. However, a single router cannot be both entrance and exit points for any individual LSP.

Router = LSR

This study guide uses the terms *LSR* and *router* interchangeably because all Junos OS routers are capable of being an LSR.

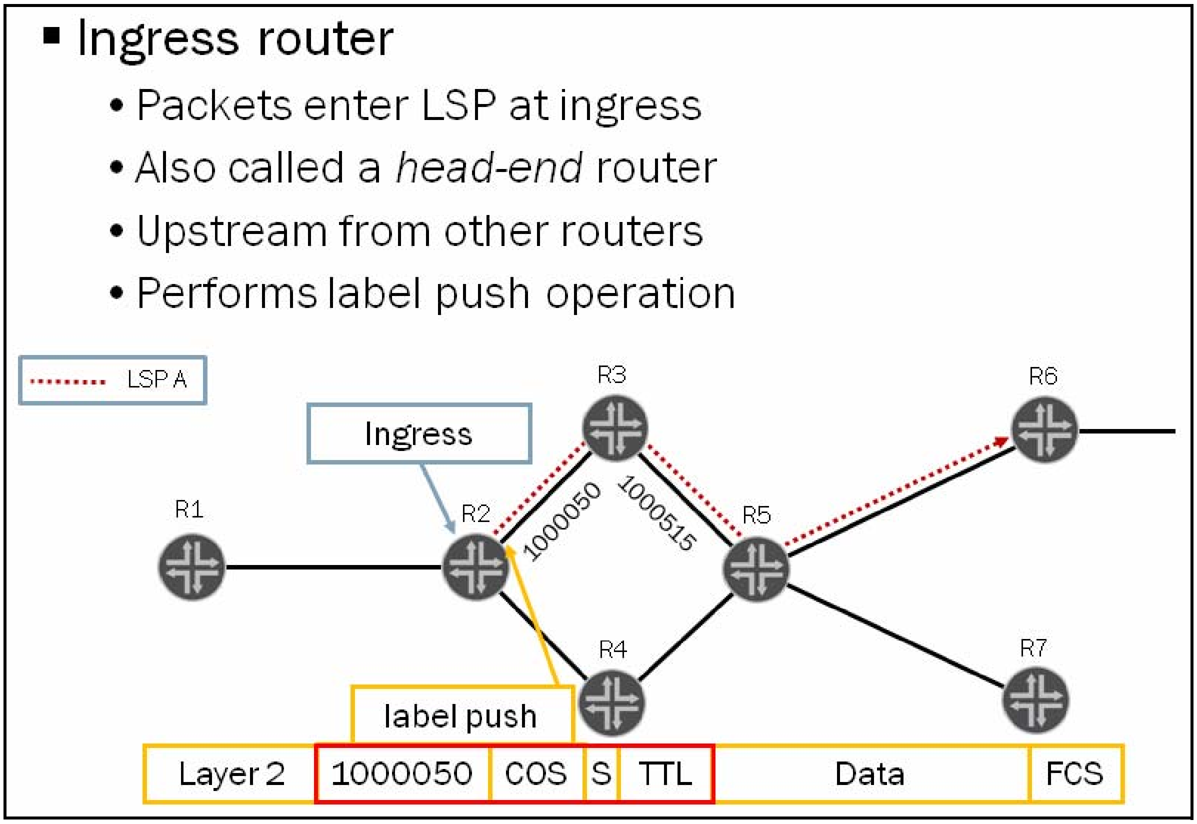
Label-Switched Path



An LSP is a one-way (unidirectional) flow of traffic, carrying packets from beginning to end. Packets must enter the LSP at the beginning (ingress) of the path, and can only exit the LSP at the end (egress). Packets cannot be injected into an LSP at an intermediate hop.

Generally, an LSP remains within a single MPLS domain. That is, the entrance and exit of the LSP, and all routers in between, are ultimately in control of the same administrative authority. This ensures that MPLS LSP traffic engineering is not done haphazardly or at cross purposes but is implemented in a coordinated fashion.

The Functions of the Ingress Router



- Ingress router
  - Packets enter LSP at ingress
  - Also called a *head-end* router
  - Upstream from other routers
  - Performs label push operation

Layer 2	1000050	COS	S	TTL	Data	FCS

Each router in an MPLS path performs a specific function and has a well-defined role based on whether the packet enters, transits, or leaves the router.

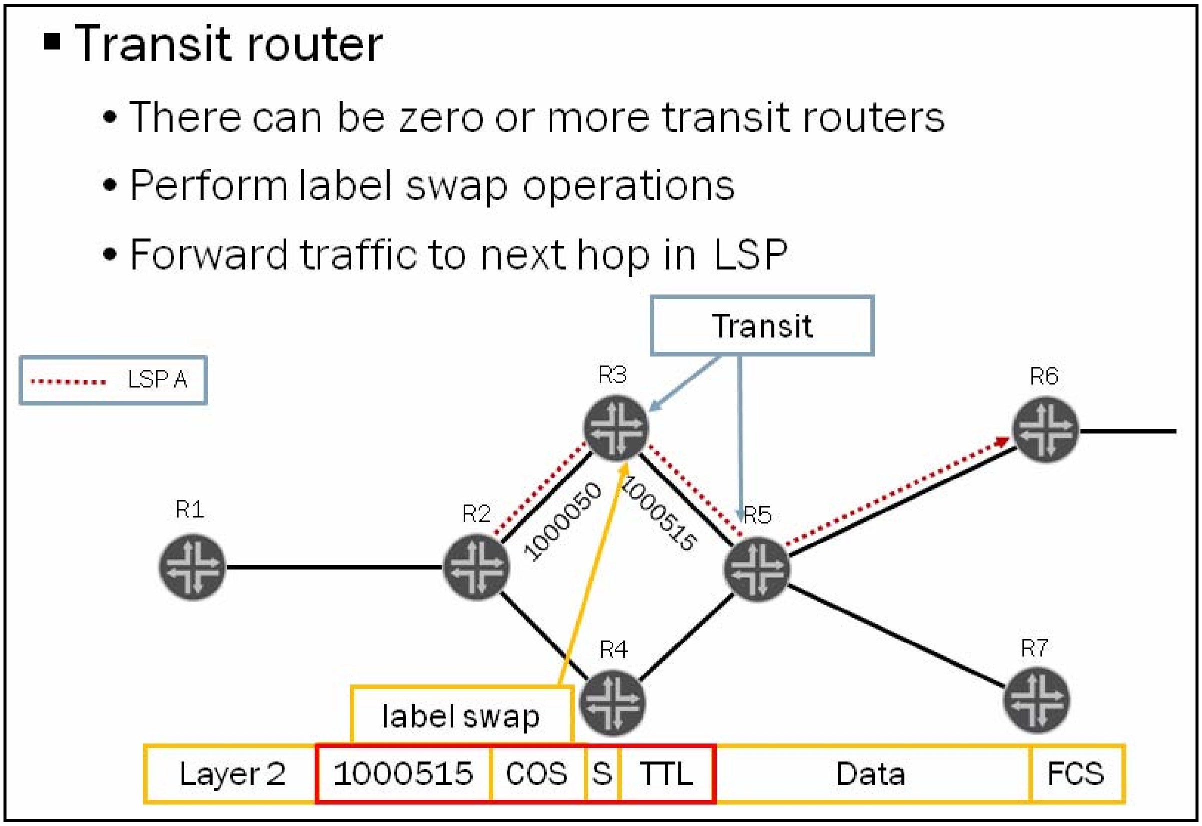
At the beginning of the tunnel, the ingress router encapsulates an IP packet that will use this LSP to R6 by adding the 32-bit MPLS shim header and the appropriate data link layer encapsulation before sending it to the first router in the path. Only one ingress router in a path can exist, and it is always at the beginning of the path. All packets using this LSP enter the LSP at the ingress router.

In some MPLS documents, this router is called the *head-end* router, or the label edge router (LER) for the LSP. In this study guide, we call it simply the ingress router for this LSP.

An ingress router always performs a push function, whereby an MPLS label is added to the label stack. By definition, the ingress router is upstream from all other routers on the LSP.

In our example we see the packet structure. We can identify that the label number is 1000050 and the ingress router action is to push this shim header in between the Layer 2 Frame and the IP header.

The Functions of the Transit Router



- Transit router
  - There can be zero or more transit routers
  - Perform label swap operations
  - Forward traffic to next hop in LSP

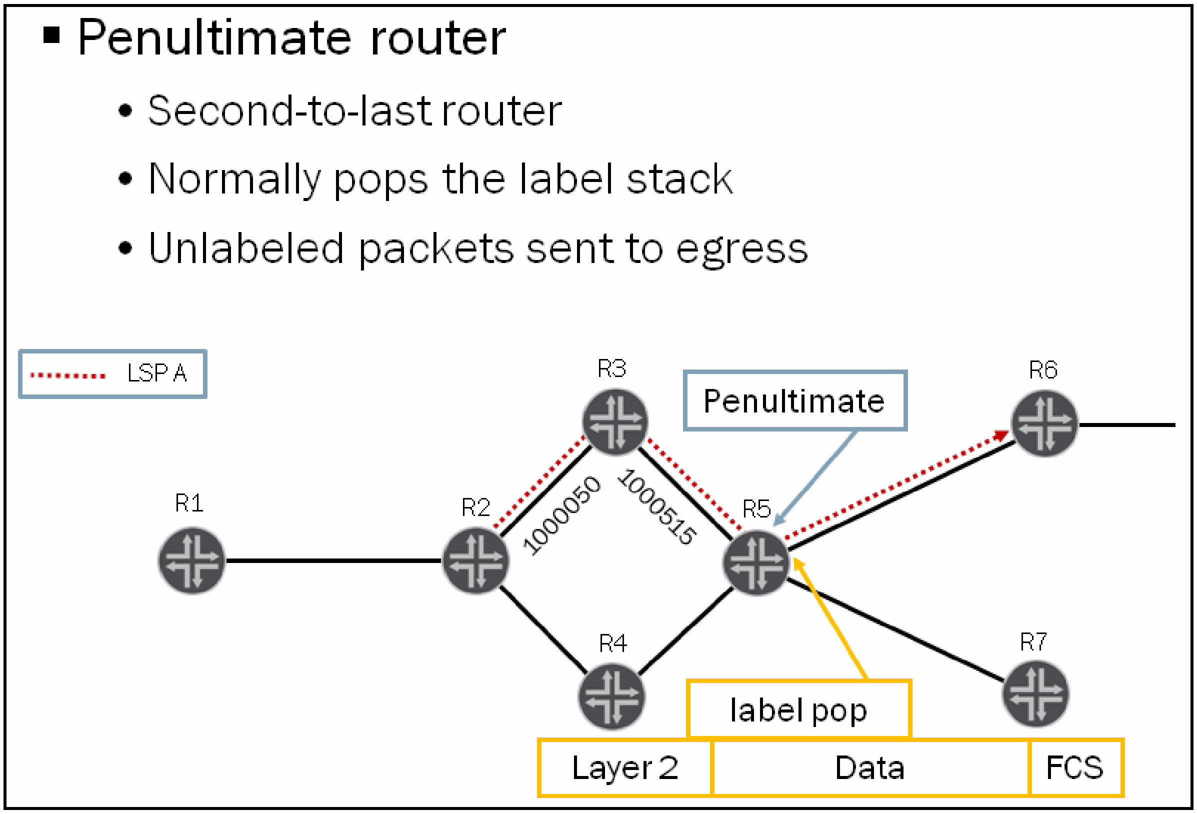
An LSP might have one or more transit routers along the path from ingress router to egress router. A transit router forwards a received MPLS packet to the next hop in the MPLS path. Zero or more transit routers in a path can exist. In a fully meshed collection of routers forming an MPLS domain, because each ingress router is connected directly to an exit point by definition, every LSP does not need a transit router to reach the exit point (although transit routers might still be configured, based on traffic engineering needs).

MPLS processing at each transit point is a simple swap of one MPLS label for another. In contrast to longest-match routing lookups, the incoming label value itself can be used as an index to a direct lookup table for MPLS forwarding, but this is strictly an MPLS protocol implementation decision.

The MPLS protocol enforces a maximum limit of 253 transit routers in a single path because of the 8 bit TTL field.

In our example we know that the packet was sent to us with the label value of 1000050 as the previous graphic indicated. Since this is a transit router we swap out the incoming label value with the outgoing label value for the next section of the LSP. We now see that the label has a value of 1000515.

The Function of the Penultimate Router



- Penultimate router
  - Second-to-last router
  - Normally pops the label stack
  - Unlabeled packets sent to egress

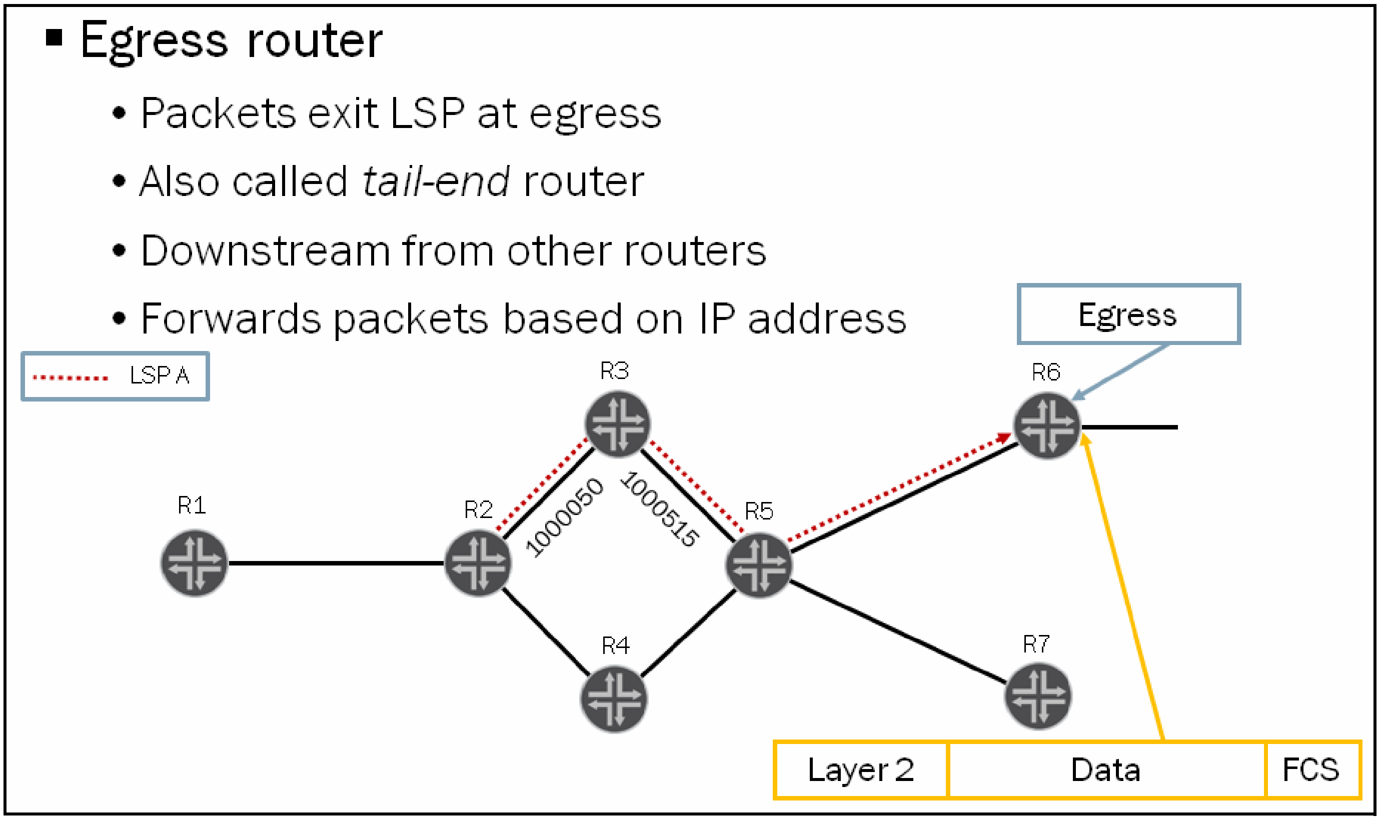
The second-to-last router in the LSP often is referred to as the penultimate hop—a term that simply means *second to the last*. In most cases the penultimate router performs a label pop instead of a label swap operation. This action results in the egress router receiving an unlabeled packet that then is subjected to a normal longest-match lookup.

Penultimate-hop popping (PHP) facilitates label stacking and *can* improve performance on some platforms because it eliminates the need for two lookup operations on the egress router. Juniper Networks routers perform equally well with, or without, PHP. Label stacking makes use of multiple MPLS labels to construct tunnels within tunnels. In these cases, having the penultimate node pop the label associated with the outer tunnel ensures that downstream nodes will be unaware of the outer tunnel's existence.

PHP behavior is controlled by the egress node by virtue of the label value that it assigned to the penultimate node during the establishment of the LSP.

In our example you can see that the MPLS header has been popped and the router is sending the packet on to the egress router without the MPLS information.

The Functions of the Egress Router



- Egress router
  - Packets exit LSP at egress
  - Also called *tail-end* router
  - Downstream from other routers
  - Forwards packets based on IP address

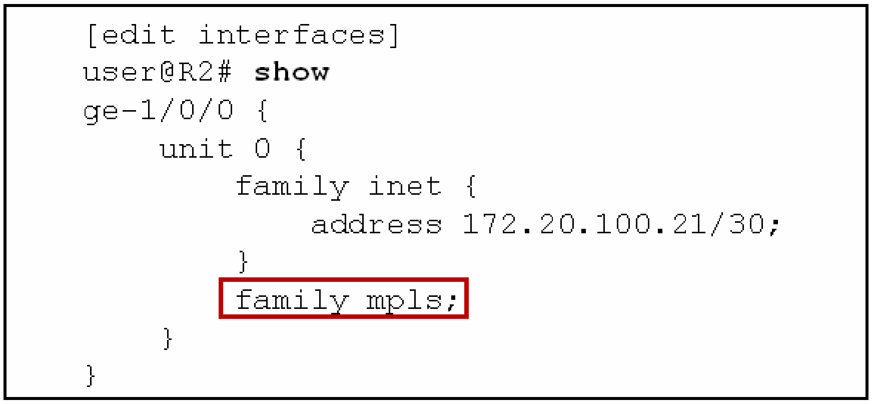
The final type of router defined in MPLS is the egress router. Packets exit the LSP at the egress router and revert to normal, IGP-based, next-hop routing outside the MPLS domain.

At the end of an LSP, the egress router routes the packet based on the native information and forwards the packet toward its final destination using the normal IP forwarding table. Only one egress router can exist in a path. In many cases, the use of PHP eliminates the need for MPLS processing at the egress node.

The egress router is sometimes called the *tail-end* router, or LER. We do not use these terms in this study guide. By definition, the egress router is located downstream from every other router on the LSP.

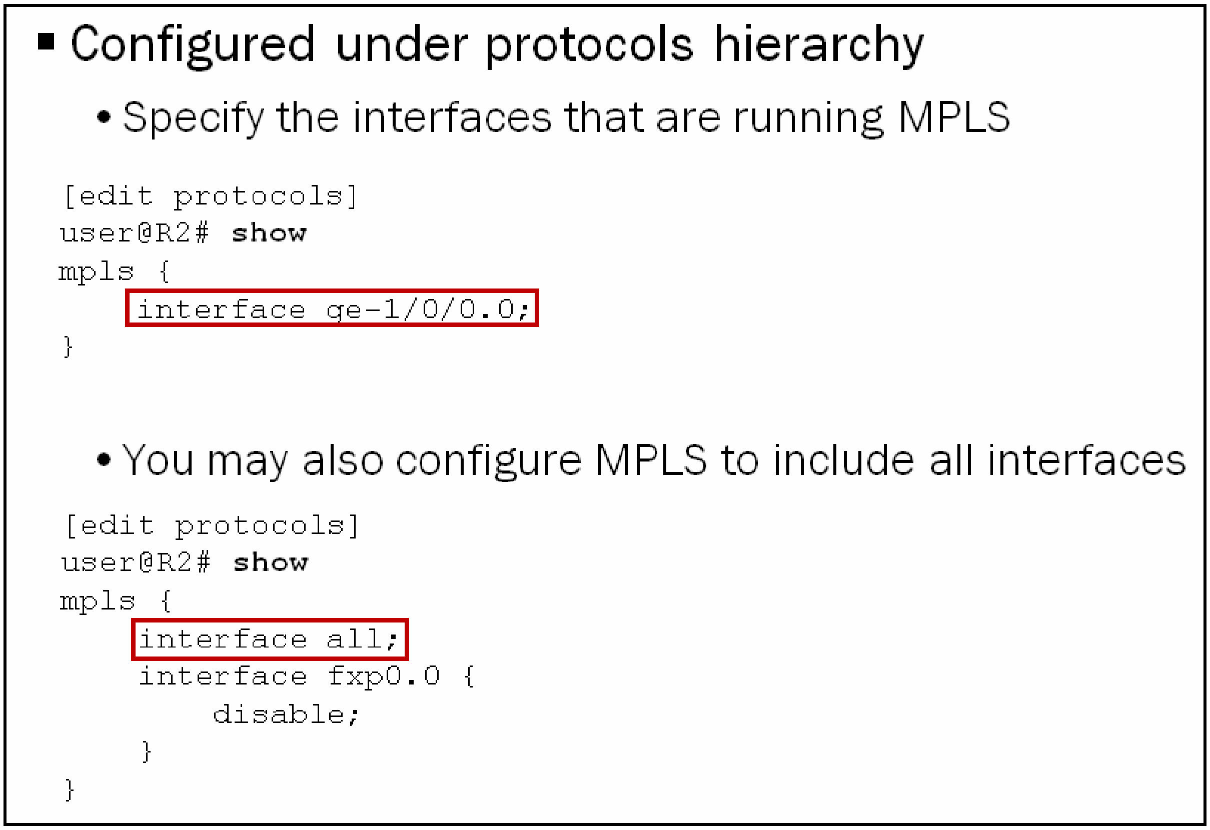
Interface Configuration

The default behavior of an interface is to accept IP packets. In the Junos OS, this is done by adding the protocol family inet with an IP address to the interface you are working with. In order for the interface to recognize and accept MPLS packets we have to also configure the MPLS protocol family under the interfaces that will be participating in your MPLS domain. Sample output demonstrates an interface configuration with both families applied.



```
[edit interfaces]
user@R2# show
ge-1/0/0 {
    unit 0 {
        family inet {
            address 172.20.100.21/30;
        }
        family mpls;
    }
}
```

MPLS is Configured Under Protocols Hierarchy



- Configured under protocols hierarchy
  - Specify the interfaces that are running MPLS

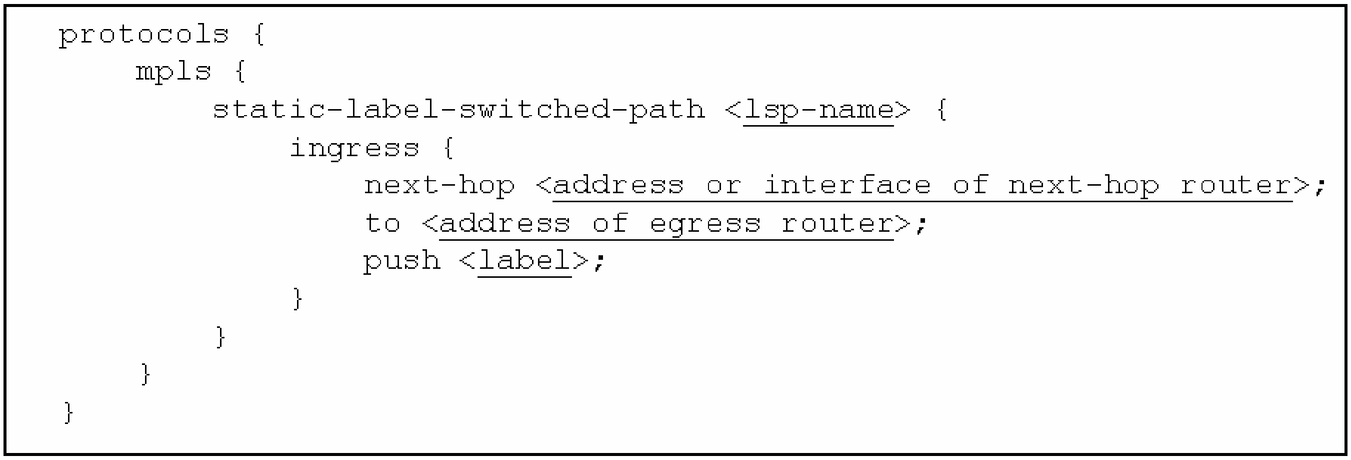
```
[edit protocols]
user@R2# show
mpls {
    interface ge-1/0/0.0;
}
```

  - You may also configure MPLS to include all interfaces

```
[edit protocols]
user@R2# show
mpls {
    interface all;
    interface fxp0.0 {
        disable;
    }
}
```

When configuring the router to support MPLS you must tell the protocol what interfaces it can use. In our example there is one interface on this router that will be participating in MPLS. In addition to specifying individual interfaces to participate in MPLS you can use the option to include all interfaces. Remember, that you have to enable the interface to recognize MPLS traffic. If the interface is not configured for protocol family MPLS, it will not send or receive MPLS packets. It is also good practice to disable the management interface (`FXP0`) from participating, since it is not a routable interface.

Configure a Static LSP on the Ingress Router



```
protocols {
    mpls {
        static-label-switched-path <lsp-name> {
            ingress {
                next-hop <address or interface of next-hop router>;
                to <address of egress router>;
                push <label>;
            }
        }
    }
}
```

The static LSP is configured under the protocols hierarchy.The first thing to configure is the LSP name. This allows you to configure multiple static LSPs between two specific routers. It is not necessary to configure unique names for static versus dynamic LSPs (a static LSP could have the same name as a dynamic LSP configured on the same router). Having named LSPs also allows you to configure a single-hop static LSP by specifying either an explicit null label or no label.

To configure a static LSP on an ingress router, include the `ingress` statement at the `[edit protocols mpls static-label-switched-path` *`lsp-name`*`]` hierarchy level.You must also configure the `to` (address of egress router) and `next-hop` (address or interface name of next-hop to reach next router) statements under the ingress statement. You can optionally configure the push statement. If you configure the push statement, you must specify a non-reserved label in the range of 0 through 1,048,575. You can also apply preference, CoS values, node protection, and link protection to the packets under the ingress configuration.

Configure a Static LSP on the Transit Router

```
protocols {
    mpls {
        static-label-switched-path <lsp-name> {
            transit <incoming-label> {
                next-hop <address or interface of next-hop router>;
                swap <outgoing label>;
            }
        }
    }
}
```

To configure a static LSP on a transit router, include the `transit` statement at the `[edit protocols mpls static-label-switched-path` ***<static-lsp-name>***`]` hierarchy level. You must include the expected incoming label directly after the transmit statement.

Under the transit hierarchy you must include the next-hop statement and either the swap or pop action. If you configure the swap statement, you must specify a non-reserved label in the range of 0 through 1,048,575.

The transit static LSP is added to the mpls.0 routing table. You should configure each static LSP using a unique name and at least one unique incoming label on the router. Each transit static LSP can have one or more incoming labels configured. If a transit LSP has more than one incoming label, each would effectively operate as an independent LSP, meaning you could configure all of the related LSP attributes for each incoming label. The range of incoming labels available is limited to the standard static LSP range of labels (1,000,000 through 1,048,575). To verify that a static LSP has been added to the routing table, issue the show route table mpls.0 command.

Because you must configure the pop action at the penultimate router, you do not need to configure the static LSP on the egress router. The packet coming into the egress router will be routed based on its Layer 3 information.

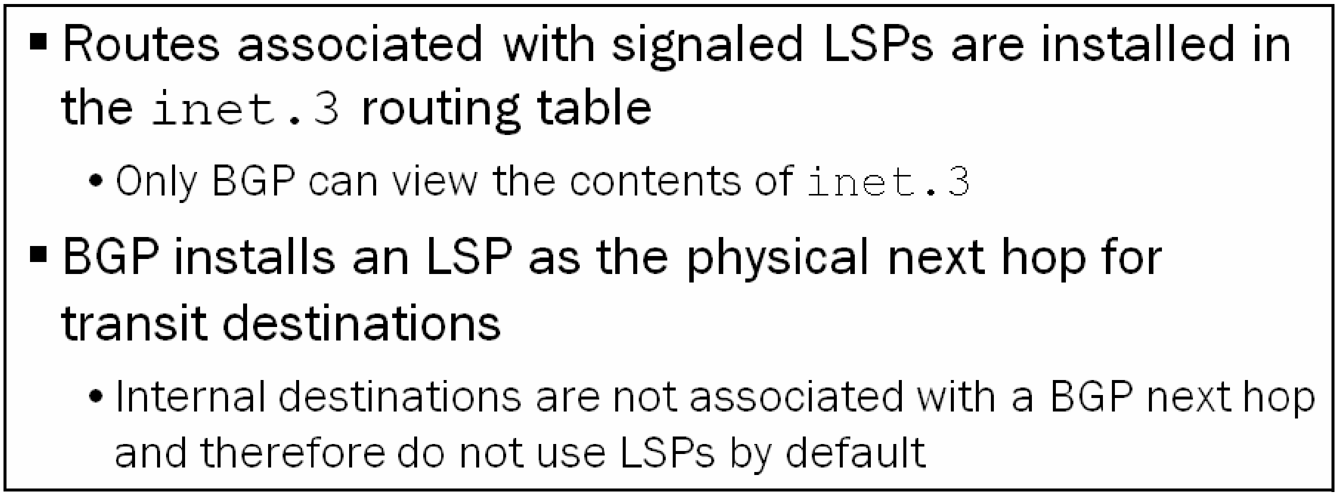
Additional Information on Static LSPs

It is best practice to make your LSP names unique to the path. This allows you to quickly identify the path you are looking for when troubleshooting or making alterations to the configuration.

In the Junos OS you can configure your outgoing label with values from 0 to 1,048,575 but will only accept a incoming label between 1,000,000 and 1,048,575 on the transit router. The Junos OS allows the outgoing label to be configured this way to allow interoperability with other vendor equipment that might not have the same static label restriction on the transit routers.

The Junos OS will also allow the static label to be swapped and sent with a label value of 0 from the penultimate router. This will allow the egress router to honor the EXP bits when queuing traffic through the static LSP.

The Use of the `inet.3` Routing Table



- Routes associated with signaled LSPs are installed in the `inet.3` routing table
  - Only BGP can view the contents of `inet.3`
- BGP installs an LSP as the physical next hop for transit destinations
  - Internal destinations are not associated with a BGP next hop and therefore do not use LSPs by default

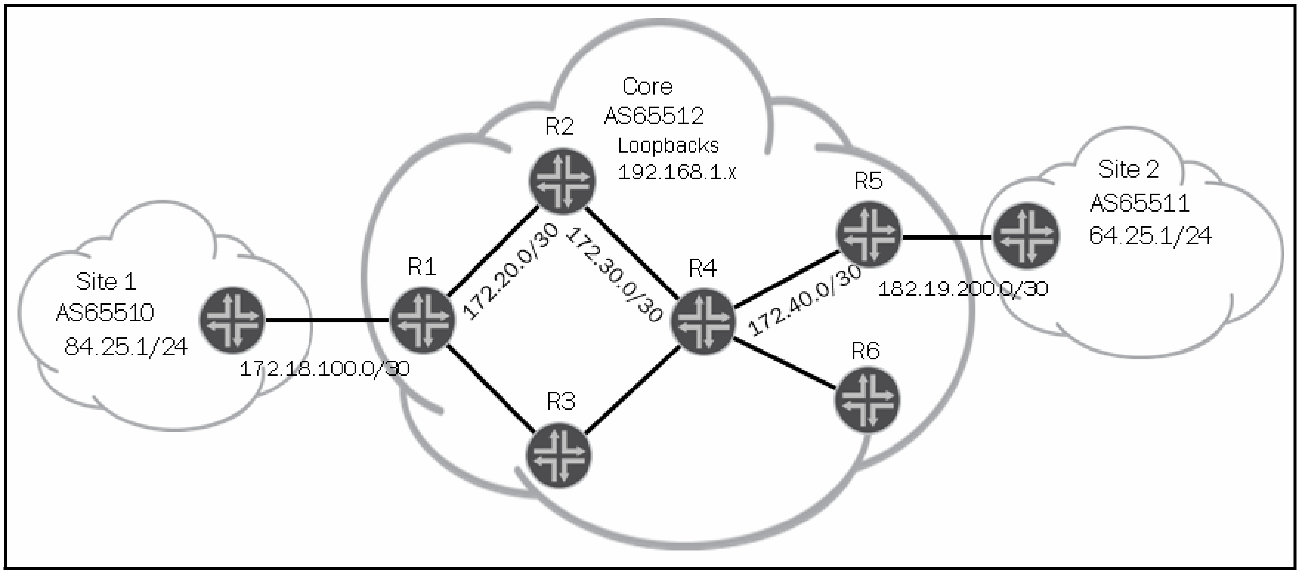
In the Junos OS implementation of MPLS, the default behavior makes BGP the only protocol that is aware of the presence of LSPs, and only then when BGP attempts to resolve the next hops associated with advertised prefixes.

Because MPLS LSPs are often used to engineer and direct transit traffic across an ISP's backbone, the default behavior results in internal traffic, which is not associated with a BGP next hop, continuing to use IGP forwarding. The result is that transit traffic associated with a BGP next hop that resolves through the `inet.3` table is subjected to LSP forwarding while all other traffic remains unaware of the LSP's presence. To maintain this separation from the normal IGP routing table, LSPs are normally installed in the `inet.3` table only.

BGP Installs LSP as Next Hop

When attempting to resolve the BGP next hop associated with a given prefix, BGP first looks in the `inet.3` table. If the next hop can be resolved in the `inet.3` table, the resulting LSP is installed into the forwarding table as the next hop for that BGP prefix. If the next hop cannot be resolved in `inet.3`, BGP next attempts to resolve the next hop through the main `inet.0` table.

Route Resolution



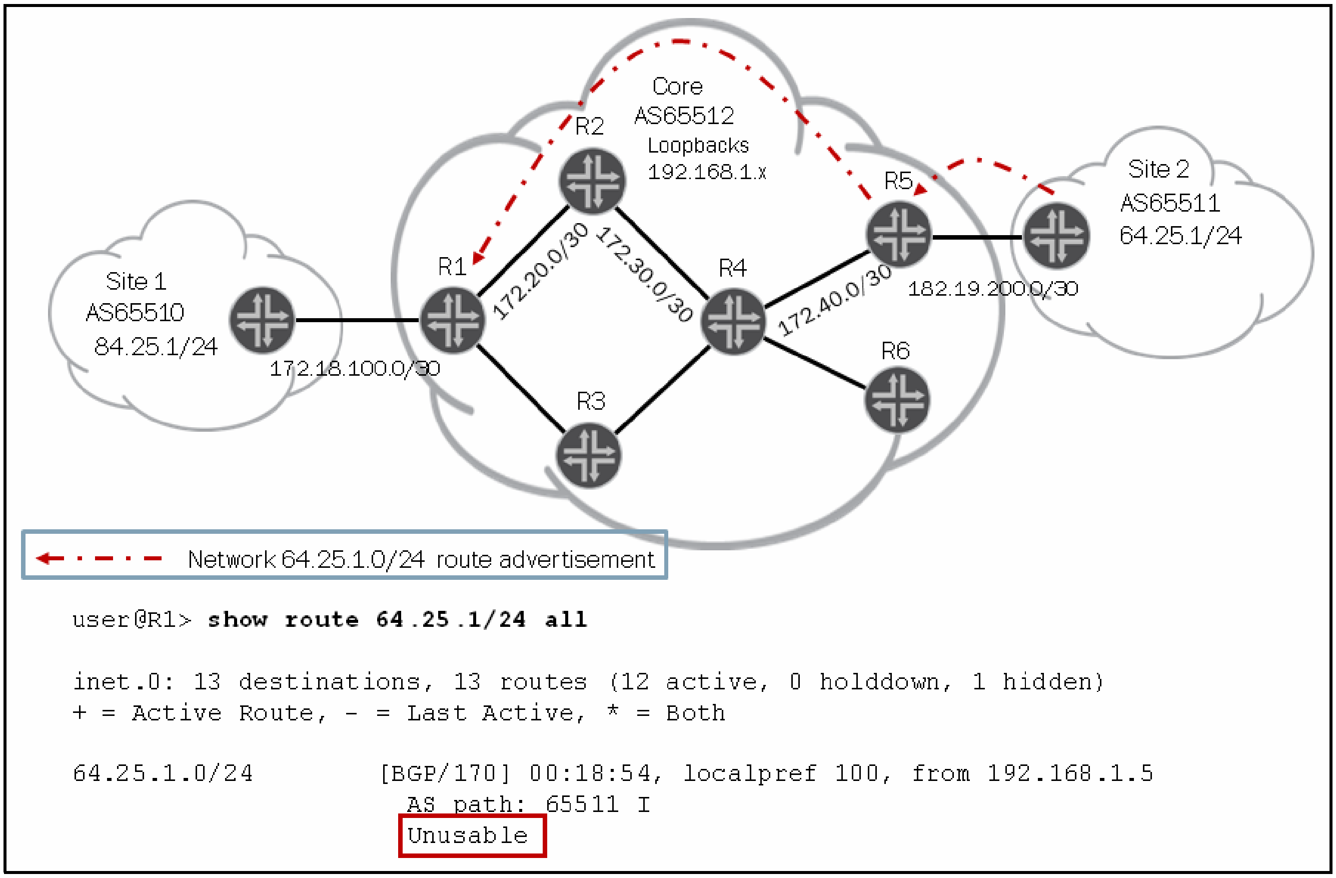
The example in the next series of graphics shows how a router uses the information learned by BGP to forward transit traffic into a LSP. We begin by examining how traffic is forwarded to the 64.25.1/24 network from the perspective of the Core.

Things start with the 64.25.1/24 prefix being learned by the R5 router through its EBGP session to Site2. The R1 router then learns about 64.25.1/24 through its internal BGP (IBGP) session to R5. R1 installs the prefix as active and readvertises the prefix to the Site1 router, again using EBGP.

In this example, routers in Site1 begin sending traffic to 64.25.1/24 prefixes through R1. When this transit traffic arrives at the R1 router, it must decide how to forward this transit traffic to 64.25.1/24.

So far, nothing in this example has anything to do with MPLS or traffic engineering. This has simply been a recap of conventional BGP operation.

Unusable BGP Next Hop



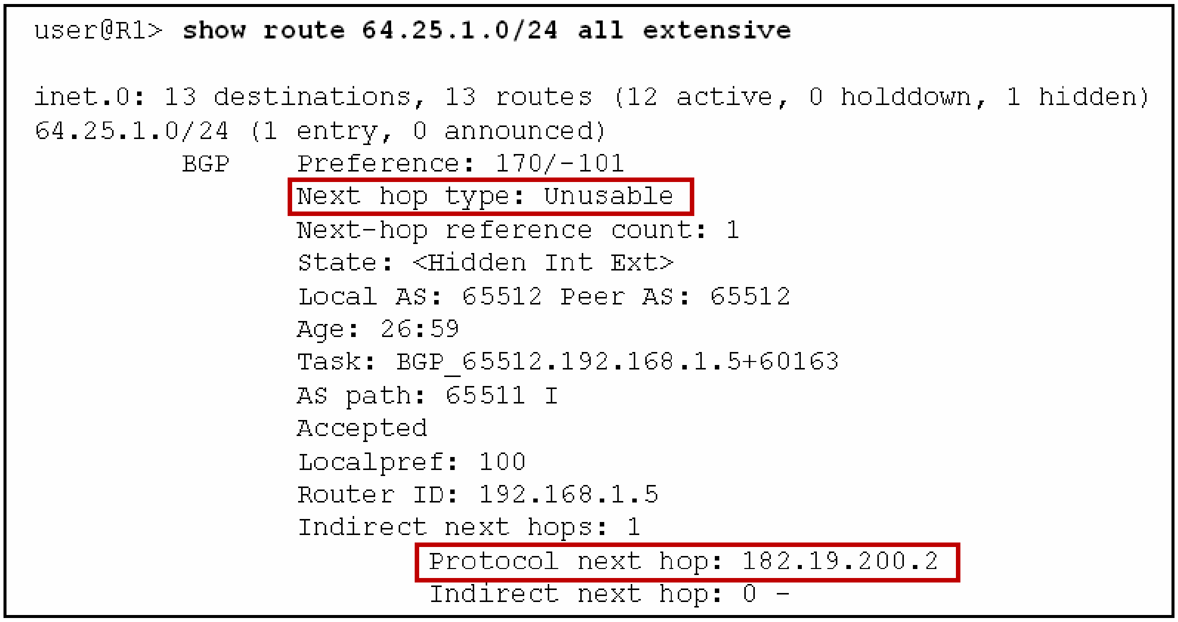
```
user@R1> show route 64.25.1/24 all

inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1 hidden)
+ = Active Route, - = Last Active, * = Both

64.25.1.0/24        [BGP/170] 00:18:54, localpref 100, from 192.168.1.5
                      AS path: 65511 I
                      Unusable
```

This example backs up the process a bit and looks at a common problem with BGP routes: unusable next hops. This discussion helps reinforce the interaction of BGP and LSP routing table integration. Note that the previous graphic shows the R1 router advertising the 64.25.1/24 prefix to a router in Site1. A route with an unusable next hop cannot be active and, therefore, cannot be exported from the routing table.

A look at the routing table on the R1 router reveals that the router has learned the 64.25.1/24 through BGP, but it also shows that the route cannot be used. The route learned from the R5 router to
64.25.1/24 is hidden, which is why the **`all`** switch was added to the **`show route`** command in the example. More investigation is necessary to determine why.

Again, there is no MPLS traffic engineering in this example.

Why the Route Is Hidden

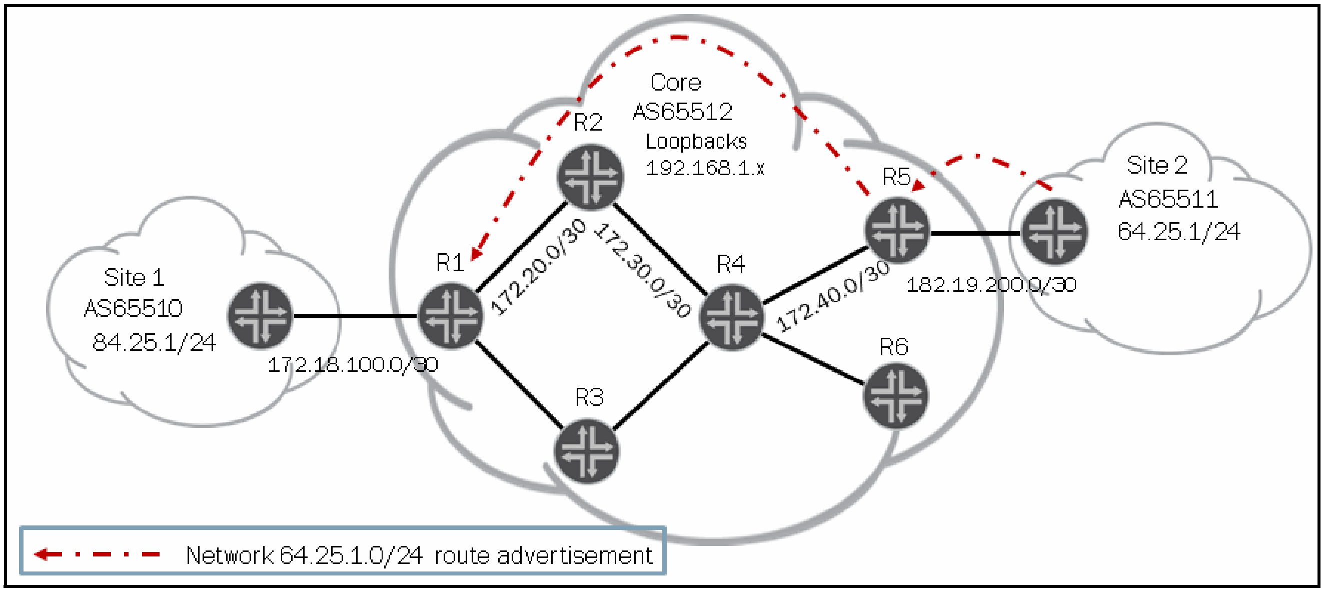


```
user@R1> show route 64.25.1.0/24 all extensive

inet.0: 13 destinations, 13 routes (12 active, 0 holddown, 1 hidden)
64.25.1.0/24 (1 entry, 0 announced)
         BGP    Preference: 170/-101
                Next hop type: Unusable
                Next-hop reference count: 1
                State: <Hidden Int Ext>
                Local AS: 65512 Peer AS: 65512
                Age: 26:59
                Task: BGP_65512.192.168.1.5+60163
                AS path: 65511 I
                Accepted
                Localpref: 100
                Router ID: 192.168.1.5
                Indirect next hops: 1
                        Protocol next hop: 182.19.200.2
                        Indirect next hop: 0 -
```

An `extensive` view of the 64.25.1/24 prefix at the R1 router shows that the next hop for this route is unusable, even though the correct next hop for the route is listed. At this point, it appears that the R1 router does not know how to get to the next hop 182.19.200.2 connected to R5.

The problem here is that the 182.19.200/30 prefix used to support the EBGP peering session between R5 and Site2 is not advertised by the core IGP. Put simply, the problem is that R1 does not have a route to 182.19.200/30.

Suggested Resolution

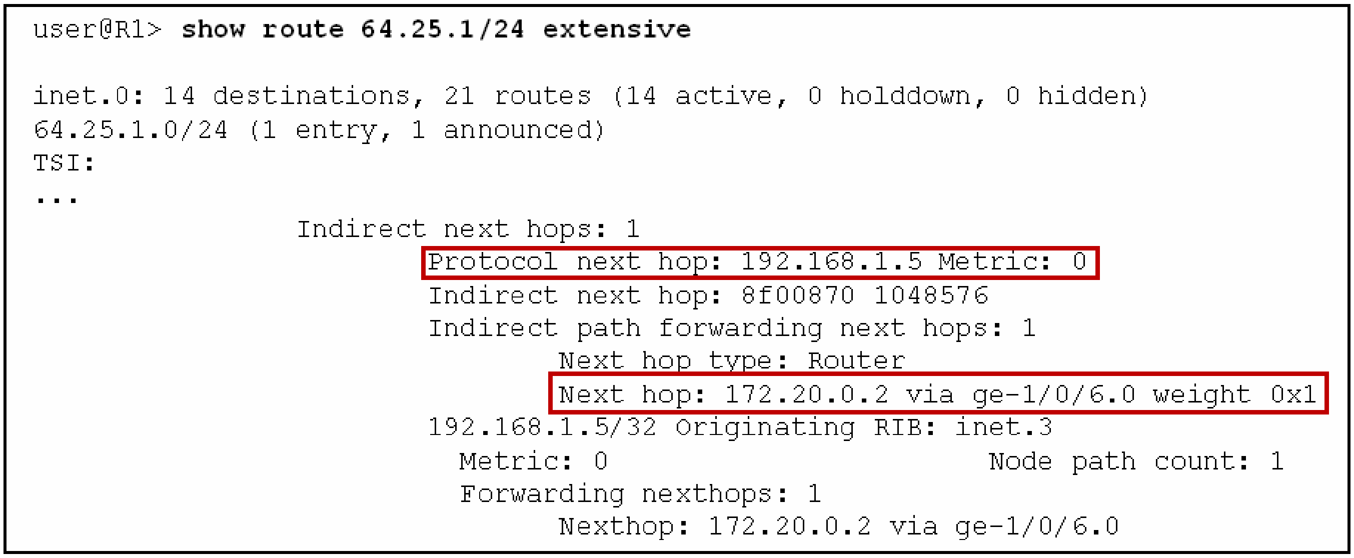


This graphic solves the 64.25.1/24 hidden route problem at R1 through a next-hop self policy applied at the R5 router. A next-hop self solution is one of several viable ways to correct unreachable BGP next hops.

Setting next-hop self on the R5 router results in the route advertisement for 64.25.1/24 arriving at the R1 router with a BGP next hop that represents the R5 router's loopback address. The R1 router can resolve the R5 router's loopback address because the IGP running throughout the autonomous system (AS) advertises that information.

At this stage, transit connectivity to 64.25.1/24 destinations is now provided by the core. This connectivity is based on IGP forwarding.

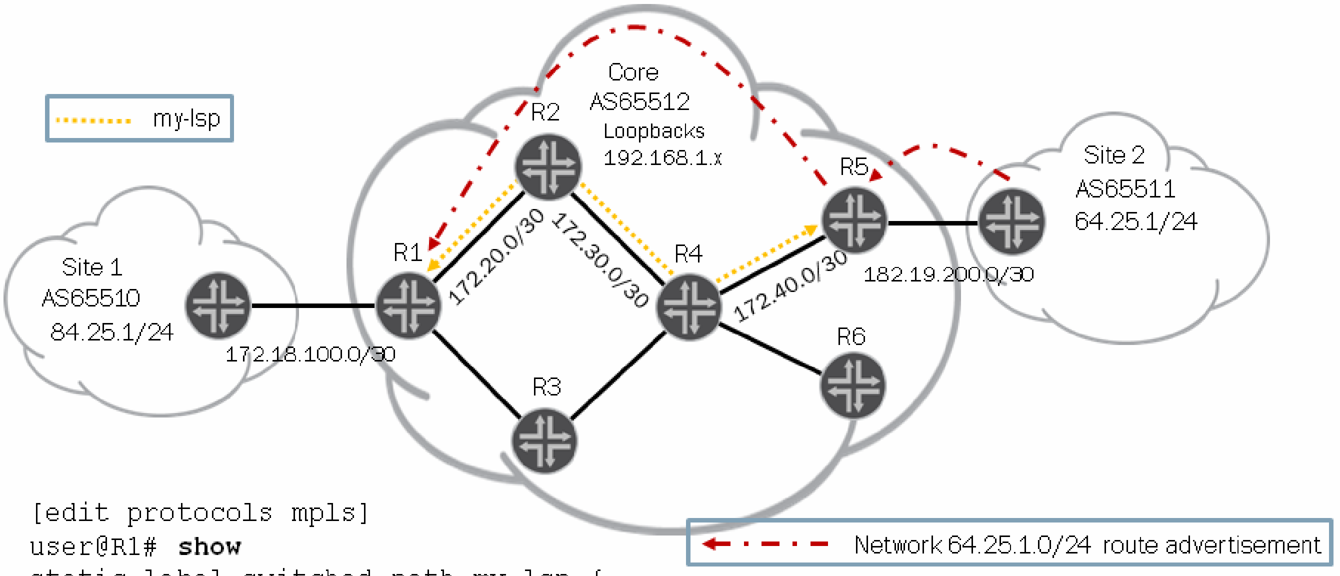
Verifying the Route Is Usable



```
user@R1> show route 64.25.1/24 extensive

inet.0: 14 destinations, 21 routes (14 active, 0 holddown, 0 hidden)
64.25.1.0/24 (1 entry, 1 announced)
TSI:
...
                Indirect next hops: 1
                        Protocol next hop: 192.168.1.5 Metric: 0
                        Indirect next hop: 8f00870 1048576
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 172.20.0.2 via ge-1/0/6.0 weight 0x1
                        192.168.1.5/32 Originating RIB: inet.3
                          Metric: 0                       Node path count: 1
                          Forwarding nexthops: 1
                                Nexthop: 172.20.0.2 via ge-1/0/6.0
```

Quick review of the extensive information for the 64.25.1/24 route reveals that the route now has a usable protocol next hop to R5s loopback interface and we have the physical next hop to R2.

Static LSP From R1 to R5 Is Configured



```
[edit protocols mpls]
user@R1# show
static-label-switched-path my-lsp {
    ingress {
        next-hop 172.20.0.2;
        to 192.168.1.5;
        push 1000050;
    }
}
…
```

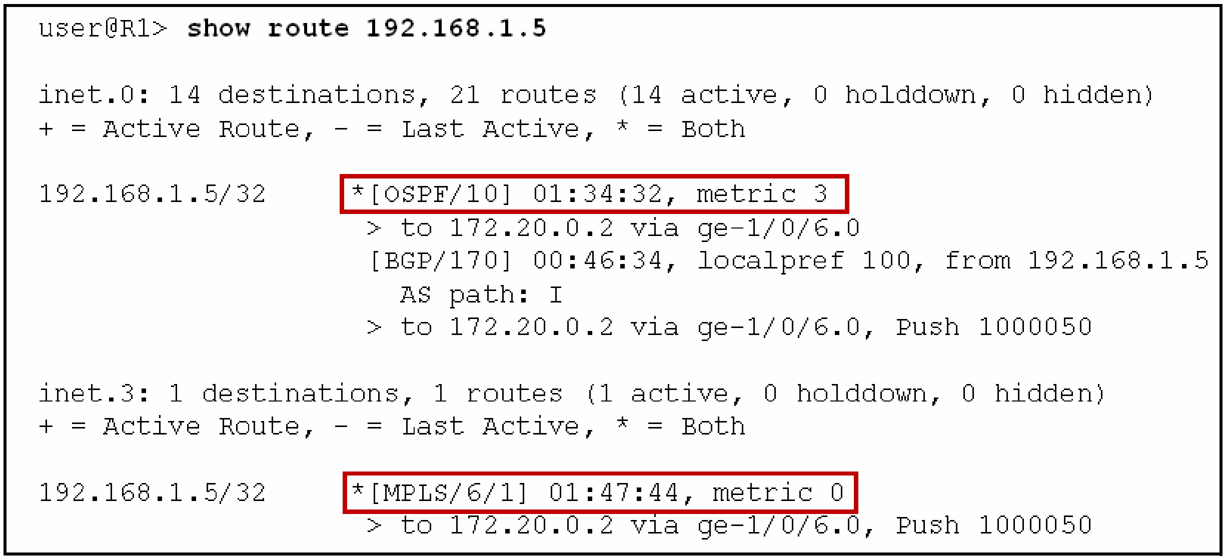
Now that we applied next-hop self to the BGP route making it usable. The sample network is ready for us to configure a Static LSP.

The key aspects of the configuration at R1 that define a static LSP to R5 are shown on the graphic. The LSP is the preferred route to 64.25.1/24 because BGP looks up a prefix in the `inet.3` table (for LSPs) before looking in the `inet.0` table (used by IGPs) and because LSPs are preferred over IGP routes due to route preference.

Now that we have configured the ingress router to forward the traffic into the LSP, we need to configure the transit LSRs to swap the labels through the rest of the path. The minimum configuration is displayed above. Notice that we do not define the egress router. As discussed previously we do not need to configure the static LSP on the egress router because we are popping the MPLS header at the penultimate router and sending the packet to the egress router in its native form to be routed based on the Layer 3 header.

```
[edit protocols mpls]
user@R2# show
static-label-switched-path my-lsp {
    transit 1000050 {
        next-hop 172.30.0.2;
        swap 1000515;
    }
}
…

[edit protocols mpls]
user@R4# show
static-label-switched-path my-lsp {
    transit 1000515 {
        next-hop 172.40.0.2;
        pop;
    }
}
…
```

Comparing Route Preferences



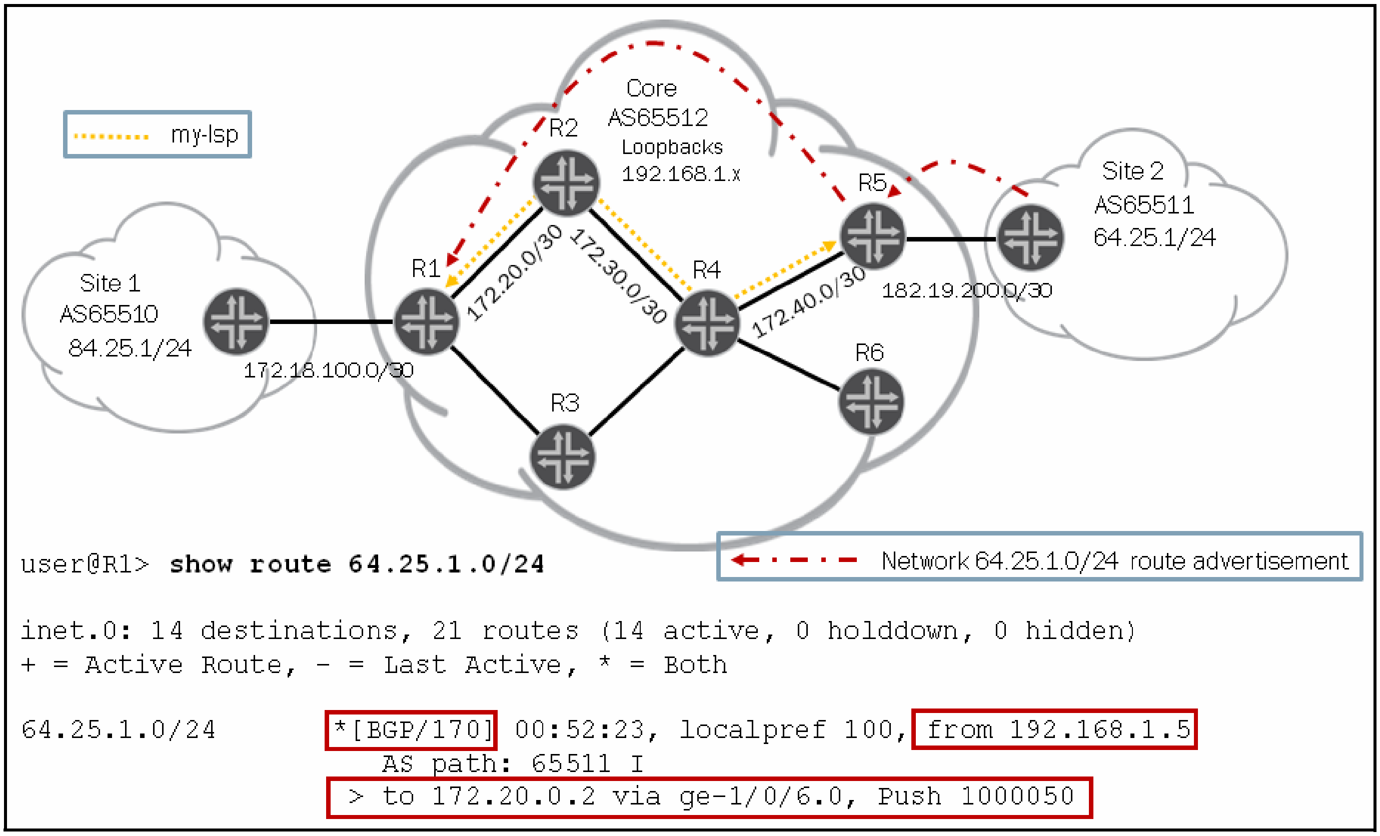
```
user@R1> show route 192.168.1.5

inet.0: 14 destinations, 21 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.5/32     *[OSPF/10] 01:34:32, metric 3
                    > to 172.20.0.2 via ge-1/0/6.0
                    [BGP/170] 00:46:34, localpref 100, from 192.168.1.5
                      AS path: I
                    > to 172.20.0.2 via ge-1/0/6.0, Push 1000050

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.5/32     *[MPLS/6/1] 01:47:44, metric 0
                    > to 172.20.0.2 via ge-1/0/6.0, Push 1000050
```

The graphic shows that once the LSP to R5 is established, information about the route to the R5 router's loopback address (the next hop for the 64.25.1/24 prefix) is present in not one but *two* routing tables. These are `inet.0`, used by all routing protocols, and `inet.3`, which is used by BGP.

But what ensures that BGP resolves its next hop through the LSP to R5 instead of the IGP route? The answer is simple: the preference assigned to LSPs is lower than the preference assigned to IGP routes. This causes the router to prefer LSPs over IGP routes. Should preferences be set the same, entries in the `inet.3` table are preferred over entries in the `inet.0` table.

A key point on the graphic is that both the IGP and MPLS routes are active, albeit in separate tables. The result is that traffic addressed to 192.168.1.5 uses the IGP route, while traffic associated with a BGP next hop of 192.168.1.5 uses the LSP.

BGP Installs LSP as Forwarding Next Hop for 64.25.1/24



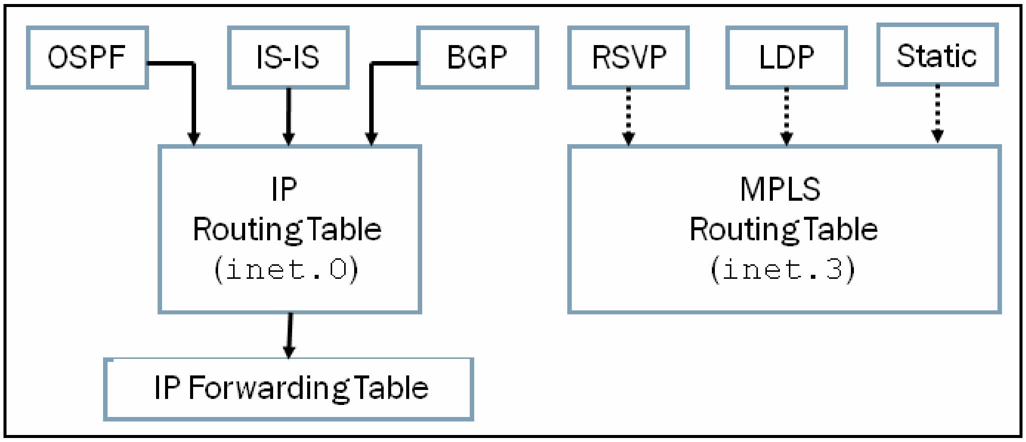
```
user@R1> show route 64.25.1.0/24

inet.0: 14 destinations, 21 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

64.25.1.0/24       *[BGP/170] 00:52:23, localpref 100, from 192.168.1.5
                      AS path: 65511 I
                    > to 172.20.0.2 via ge-1/0/6.0, Push 1000050
```

It helps to keep in mind that the goal of this whole exercise is not to use the LSP to reach the R5 router's loopback address; the goal is to direct transit traffic associated with the 64.25.1/24 prefix through a LSP for transport across the Core AS.

The graphic shows that the BGP route to 64.25.1/24 is present in the `inet.0` routing table as a BGP route. However, the results of BGP next-hop resolution through the `inet.3` table results in the static LSP being installed as the forwarding next hop for traffic associated with the 64.25.1/24 prefix.

Packets destined to 64.25.1/24 that arrive at the R1 router are forwarded over the LSP. No other traffic will use this LSP with the current configuration. Note also that packets can only enter an LSP at the ingress node.

Ingress Router



The previous example demonstrated how signaled LSPs are installed in the `inet.3` routing table. RSVP, LDP, and Static LSPs install the IP prefix to the egress router into the `inet.3` table on the ingress router. We will discuss RSVP and LDP in later chapters.

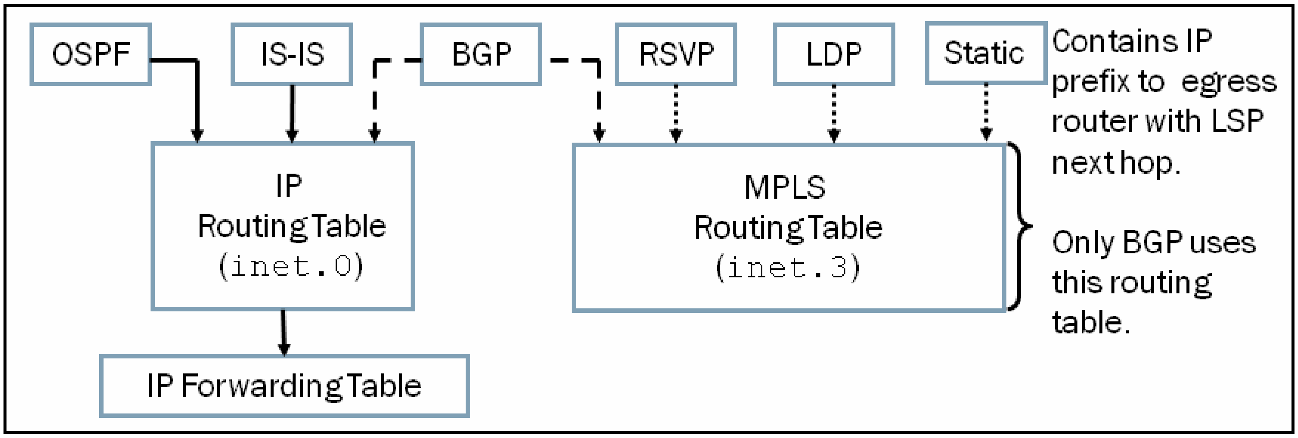
The next-hop data for entries in the `inet.3` table consist of the LSP's egress interface and the label value assigned by the LSP's first downstream router.

Note that the various routing protocols continue to use the inet.0 IP routing table to determine the current active route to IP destinations.

A sample inet.3 entry that shows the next-hop forwarding information for an LSP is shown here. Note the presence of a label push operation and egress interface:

```
user@R1> show route table inet.3 detail

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
192.168.1.5/32 (1 entry, 0 announced)
        *MPLS   Preference: 6/1
                Next hop type: Router
                Next-hop reference count: 2
                Next hop: 172.20.0.2 via ge-1/0/6.0 weight 0x1, selected
                Label operation: Push 1000050
                State: <Active Int>
                Local AS: 65512
                Age: 39         Metric: 0
                Task: MPLS
                AS path: I
```

BGP Resolves Its Next Hop Using Both Tables


OSPF
IS-IS
BGP
RSVP
LDP
Static
IP Routing Table (inet.0)
MPLS Routing Table (inet.3)
IP Forwarding Table
Contains IP prefix to egress router with LSP next hop.
Only BGP uses this routing table.

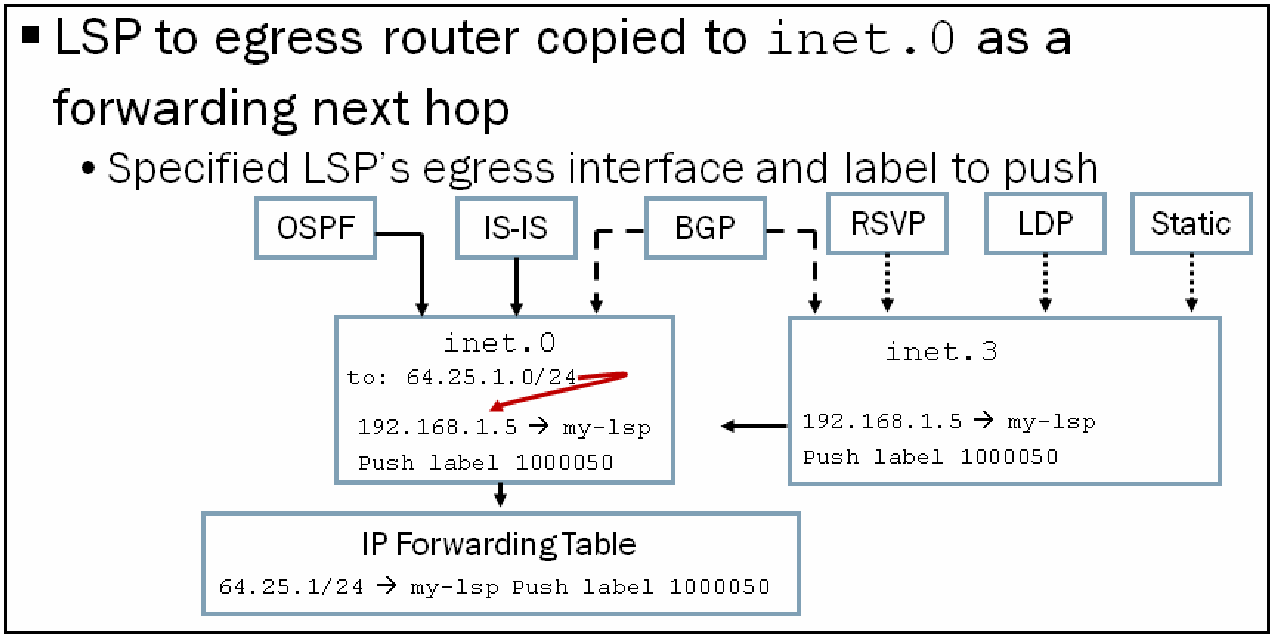
BGP must resolve a protocol next hop to a forwarding next hop in a process known as a *recursive route lookup*. The goal of this process is to resolve the advertised BGP next hop to a directly connected forwarding next hop.

BGP uses both the inet.0 and inet.3 tables when attempting to resolve a next-hop address. When the same prefix appears in both inet.0 and inet.3 tables, as is the case of this example, BGP chooses the route with the lowest preference. This results in the selection of the LSP when default preference values are at play. In the event of a preference tie, entries in inet.3 are preferred over inet.0 entries.

BGP Selects inet.3 over inet.0

By default, BGP prefers LSP-based resolution of a BGP prefix's next hop. If there is a preference tie between inet.3 and inet.0, BGP selects the inet.3 route over the inet.0 route.

LSP Installed as Forwarding Next Hop in inet.0



- LSP to egress router copied to `inet.0` as a forwarding next hop
  - Specified LSP's egress interface and label to push

OSPF

IS-IS

BGP

RSVP

LDP

Static

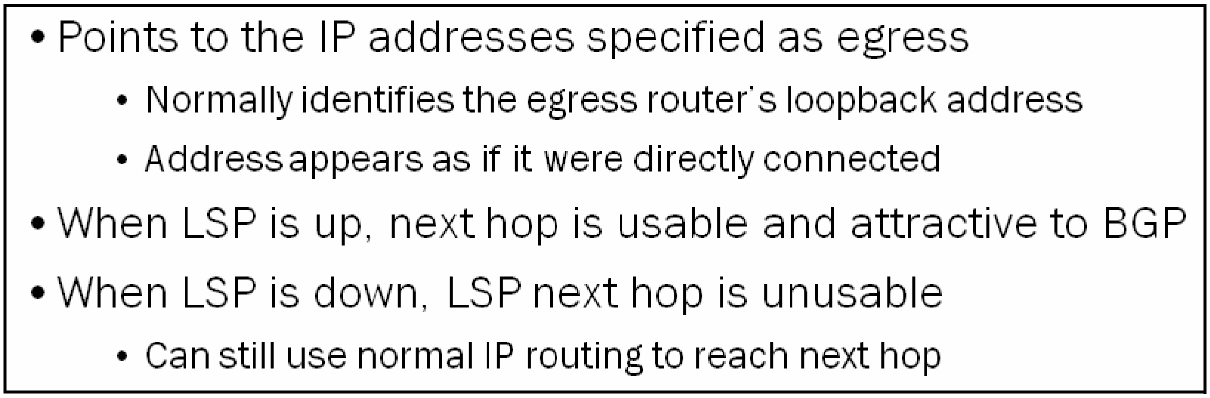
inet.0
to: 64.25.1.0/24
192.168.1.5 → my-lsp
Push label 1000050

inet.3
192.168.1.5 → my-lsp
Push label 1000050

IP Forwarding Table
64.25.1/24 → my-lsp Push label 1000050

The LSP's forwarding information is copied into `inet.0` when BGP can resolve its next hop though the LSP. A key point is that the LSP itself is not installed into `inet.0`. Rather, the BGP prefix is installed into `inet.0` with a forwarding next hop that points to the LSP. Thus, traffic not associated with the BGP next hop in question continues to be unaware of the LSP's presence. The graphic shows how the forwarding table is ultimately configured to forward over the LSP for traffic associated with the BGP prefix 64.25.1/24.

LSPs Are Installed in Ingress Router's `inet.3` Table



- Points to the IP addresses specified as egress
  - Normally identifies the egress router's loopback address
  - Address appears as if it were directly connected
- When LSP is up, next hop is usable and attractive to BGP
- When LSP is down, LSP next hop is unusable
  - Can still use normal IP routing to reach next hop

In summary, LSPs appear in the ingress router's `inet.3` table. The next-hop address points to the egress router as if it were directly connected and not at the end of an LSP passing through many transit points. Normally the LSP's egress address is specified as the egress router's loopback address, which also serves as the router ID.

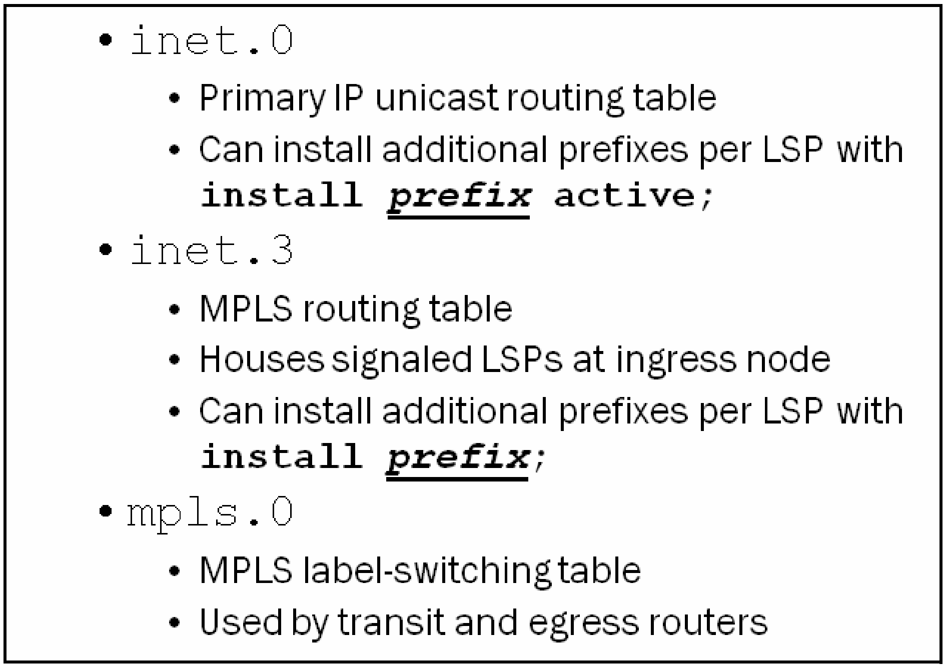
When the LSP is up, this next hop is usable by BGP for next hop route resolution. When the LSP is down, the next hop referenced by the LSP is unusable. Packets still can use normal IGP routing information to resolve the next hop, however.

Only BGP Is Aware of inet.3

Only BGP pays attention to the entries housed in `inet.3` and only then when it is resolving a BGP next hop. LSPs are hidden from the main IP routing table, which allows non-BGP traffic to continue to use the IGP forwarding path.

Note that the examples we discuss in this section are based on the default LSP routing table integration behavior. With additional configuration, the presence of LSPs can be made known to non-BGP traffic. The decision to traffic engineer internal traffic requires careful consideration because MPLS, and even basic router troubleshooting, might be complicated when pings and traceroutes begin using LSPs.

Routing Tables Used in MPLS

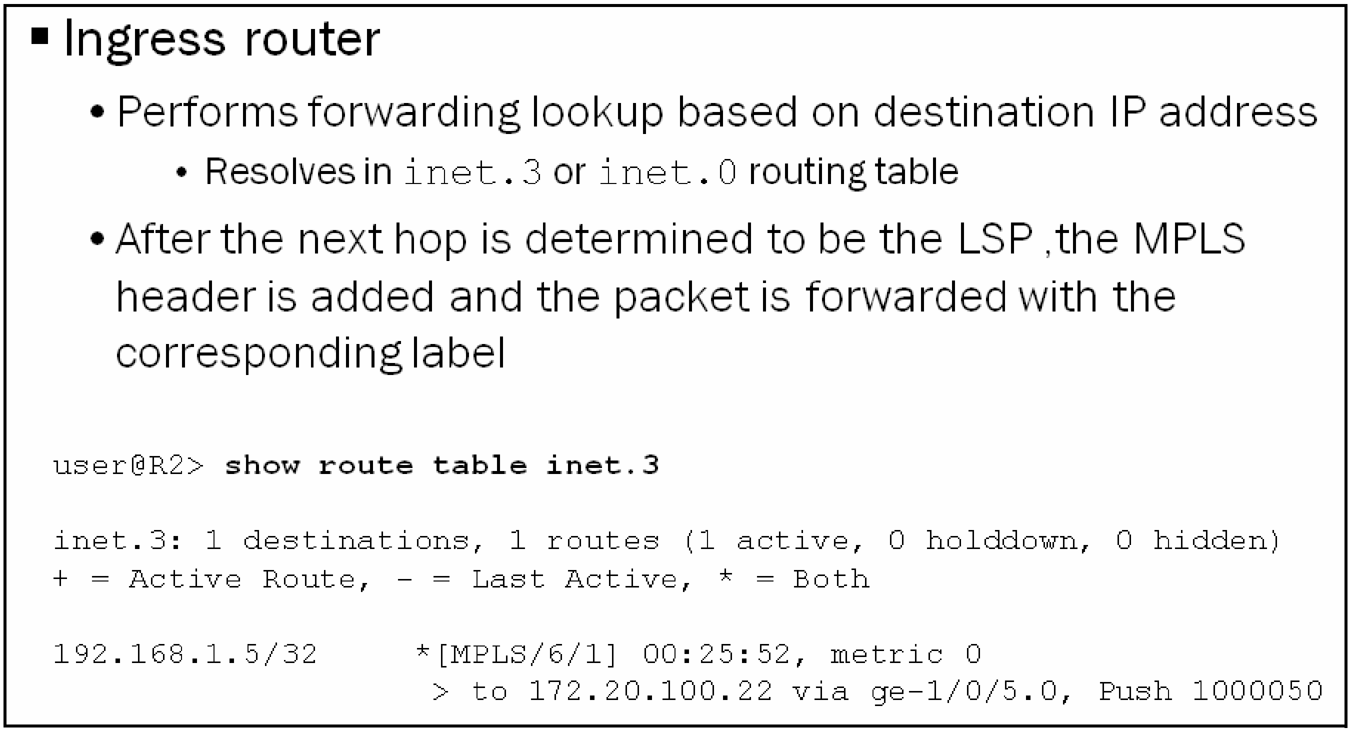


- inet.0
  - Primary IP unicast routing table
  - Can install additional prefixes per LSP with **install *prefix* active;**
- inet.3
  - MPLS routing table
  - Houses signaled LSPs at ingress node
  - Can install additional prefixes per LSP with **install *prefix*;**
- mpls.0
  - MPLS label-switching table
  - Used by transit and egress routers

In summary, three tables are important when you configure MPLS along with normal routing protocols. These are inet.0, inet.3, and mpls.0:

- inet.0: The primary IP unicast routing table. Normally, IGPs only look in this table to resolve next hops. You can allow IGPs to access LSP information available in inet.3 on an LSP-by-LSP basis by installing LSP prefixes into inet.0 using the **install *prefix* active** CLI configuration command in the configuration for that LSP.
- inet.3: The MPLS routing table. All signaled LSPs are installed in this table where only BGP can find them. You can add prefixes associated with the LSP to the inet.3 table by using the **install *prefix*** CLI configuration command in the configuration for that LSP. This knob is useful when you want BGP to use the LSP and next-hop self is not in effect at the egress node.
- mpls.0: The MPLS label switching table. Transit and egress routers use the contents of this table to swap and pop labels as needed when handling the LSP.

MPLS Forwarding—Ingress Router

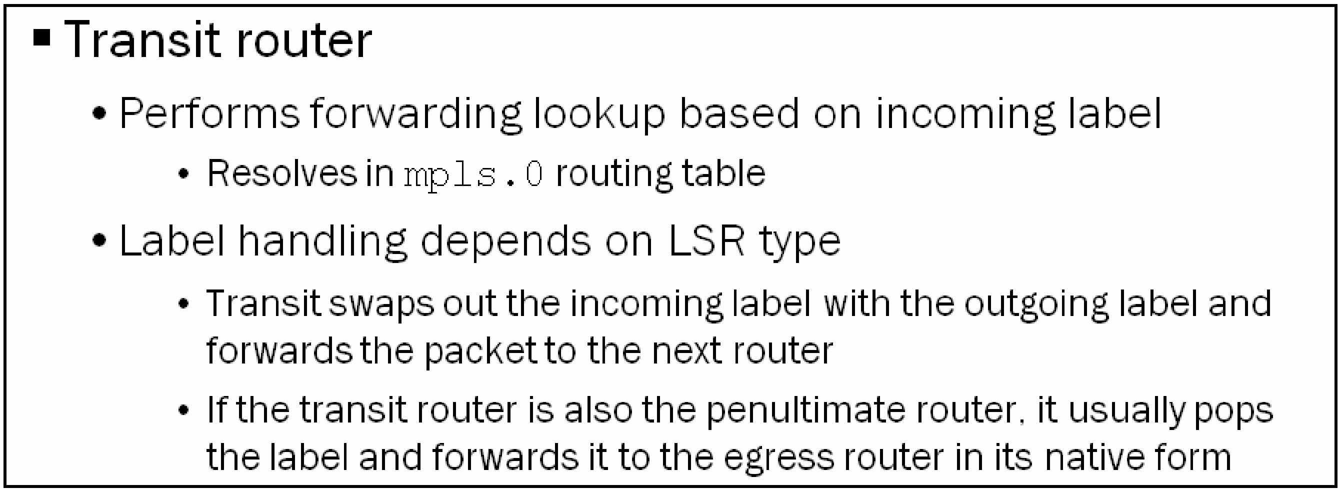


- Ingress router
  - Performs forwarding lookup based on destination IP address
    - Resolves in `inet.3` or `inet.0` routing table
  - After the next hop is determined to be the LSP ,the MPLS header is added and the packet is forwarded with the corresponding label

```
user@R2> show route table inet.3

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.5/32     *[MPLS/6/1] 00:25:52, metric 0
                    > to 172.20.100.22 via ge-1/0/5.0, Push 1000050
```

The ingress router makes its forwarding decisions based on the destination address. It will resolve the next hop by inspecting the `inet.3` and the `inet.0` routing tables. After the next hop is determined to be the LSP the MPLS shim header is added and the packet is forwarded on to the appropriate next hop router.

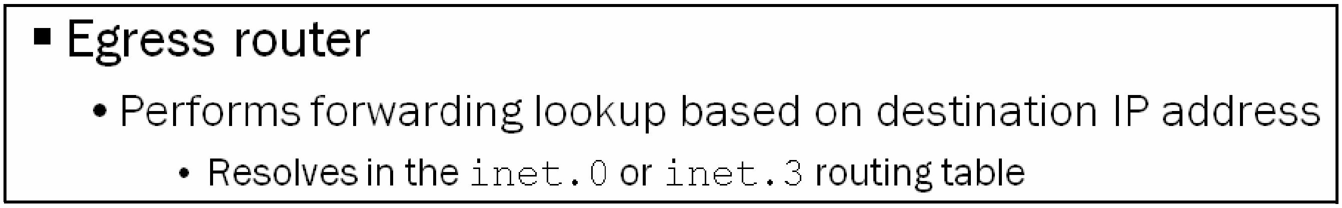
MPLS Forwarding—Transit Router



- Transit router
  - Performs forwarding lookup based on incoming label
    - Resolves in `mpls.0` routing table
  - Label handling depends on LSR type
    - Transit swaps out the incoming label with the outgoing label and forwards the packet to the next router
    - If the transit router is also the penultimate router, it usually pops the label and forwards it to the egress router in its native form

The transit router makes its forwarding decision based on the incoming label and refers to the information stored in the `mpls.0` routing table. If the transit router is more than two hops away from the egress router it will perform a label swap operation in the MPLS header. The router will then forward the packet on to next hop router with the new label to continue through the LSP. If the transit router is also the penultimate router it will pop the MPLS header and forward the packet on to the egress router in its native form.

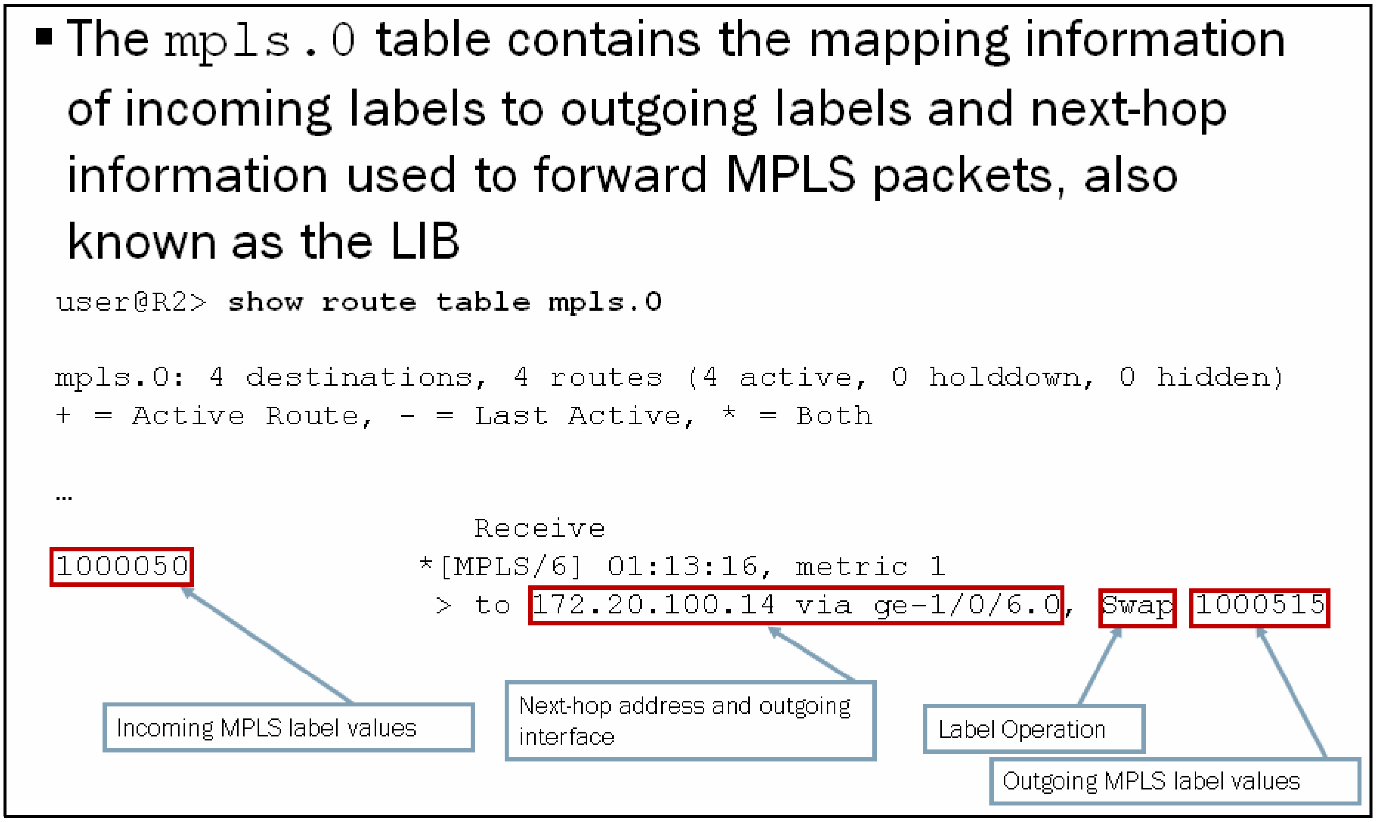
MPLS Forwarding—Egress Router



- Egress router
  - Performs forwarding lookup based on destination IP address
    - Resolves in the `inet.0` or `inet.3` routing table

The egress router forwards traffic based on the destination IP address and consults both the `inet.0` and `inet.3` routing tables to accomplish this.

MPLS Label Mapping



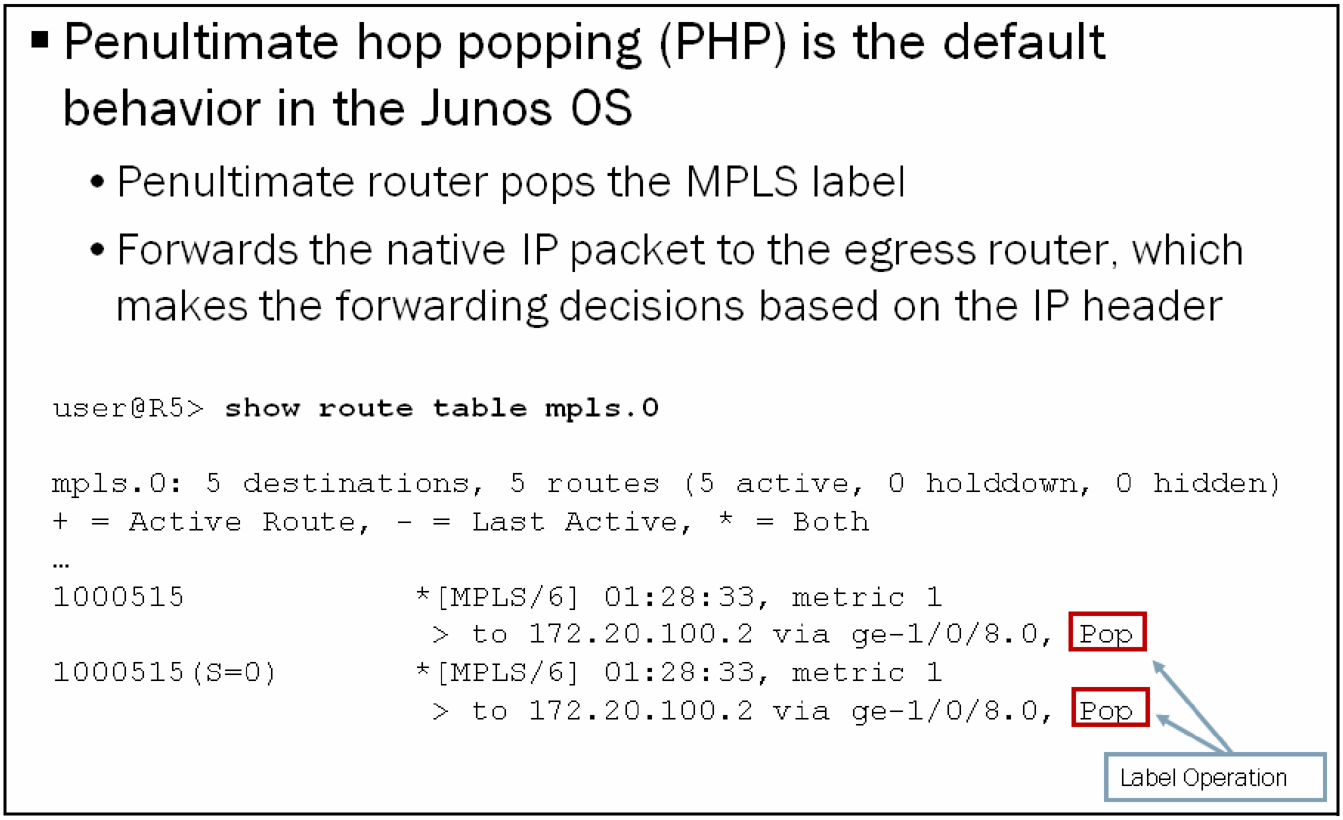
- The `mpls.0` table contains the mapping information of incoming labels to outgoing labels and next-hop information used to forward MPLS packets, also known as the LIB

```
user@R2> show route table mpls.0

mpls.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

...
                         Receive
1000050                 *[MPLS/6] 01:13:16, metric 1
                         > to 172.20.100.14 via ge-1/0/6.0, Swap 1000515
```

Incoming MPLS label values
Next-hop address and outgoing interface
Label Operation
Outgoing MPLS label values

The mpls.0 routing table contains the mapping information used to forward traffic by the transit routers. A quick review of this table will show you the possible incoming label values and indicate what protocol they were learned by. As you can see in the example you can see local next hop information. The output also shows us what the label action is and what the next label value will be when the packet is forwarded on to the next router downstream. This table is sometimes referred to as the LIB.

Penultimate Hop Popping



- Penultimate hop popping (PHP) is the default behavior in the Junos OS
  - Penultimate router pops the MPLS label
  - Forwards the native IP packet to the egress router, which makes the forwarding decisions based on the IP header

```
user@R5> show route table mpls.0

mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
...
1000515            *[MPLS/6] 01:28:33, metric 1
                    > to 172.20.100.2 via ge-1/0/8.0, Pop
1000515(S=0)       *[MPLS/6] 01:28:33, metric 1
                    > to 172.20.100.2 via ge-1/0/8.0, Pop
```

PHP is the default behavior in the Junos OS. As discussed earlier the penultimate router is the router directly upstream from the egress router. It will pop the label and forward the packet on to the egress router without the MPLS header.

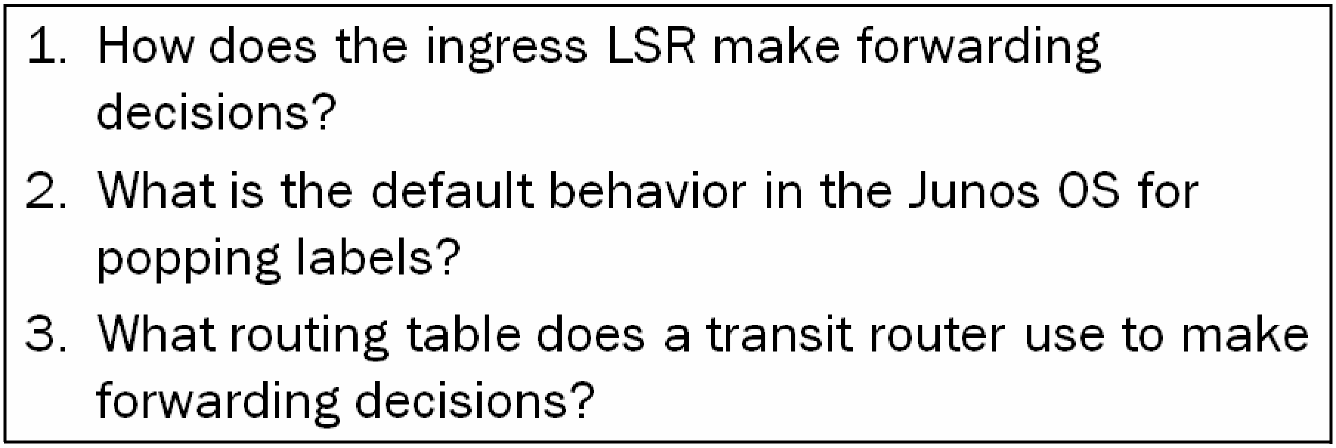
Implicit NULL

Implicit Null is the default behavior in the Junos OS from the perspective of the egress router. This operation tells the upstream router (penultimate) that they should pop the label and forward the packet without a MPLS header.

Explicit NULL

Explicit Null can be configured on the egress router under the `[edit protocols mpls]` hierarchy. This operation tells the upstream router (penultimate) that it must forward the packets on to the egress router with a MPLS header and the egress router will handle the pop action.

Review Questions



1. How does the ingress LSR make forwarding decisions?
2. What is the default behavior in the Junos OS for popping labels?
3. What routing table does a transit router use to make forwarding decisions?

Answers to Review Questions

1.

The ingress router uses the destination IP address of the packet to make its forwarding decision. It will consult the inet.0 and inet.3 routing table to resolve the next-hop.

2.

The default behavior in the Junos OS is for the egress router to signal the penultimate hop router to pop the mpls header and send the packet downstream without a mpls header. This is known as penultimate hop popping.

3.

The transit router will use the mpls.0 routing table to make its forwarding decisions. These decisions are made based on the incoming label value.

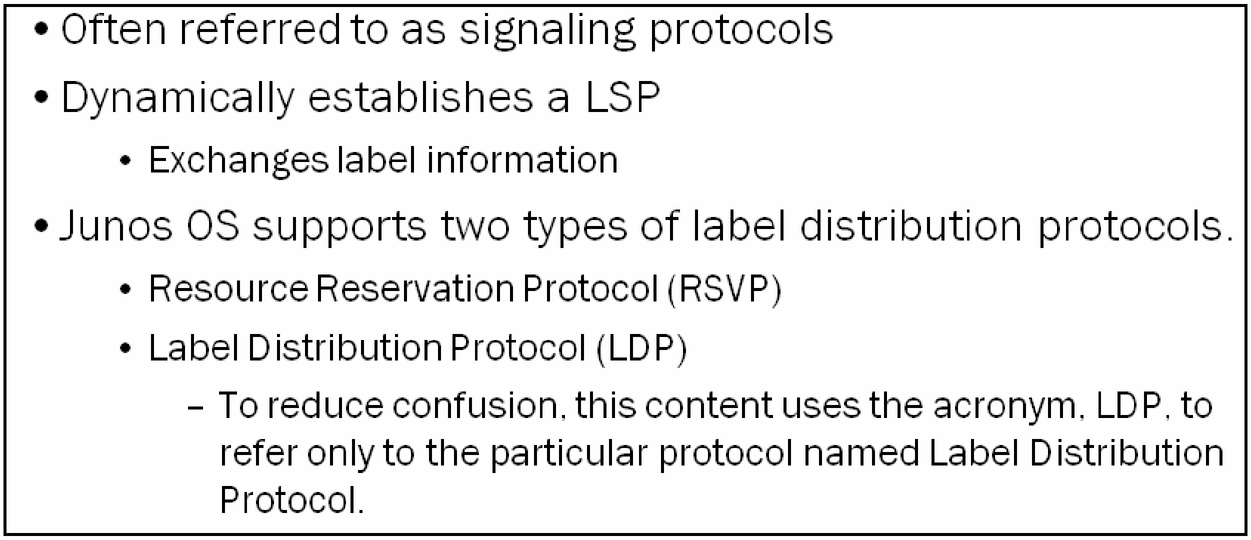
JNCIS-SP Study Guide—Part 3

Chapter 2: Label Distribution Protocols

This Chapter Discusses:

- Two label distribution protocols used by the Junos operating system;
- Configuration and verification of RSVP-signaled and LDP-signaled label-switched paths (LSPs); and
- Understanding the constraints of both RSVP and LDP.

Label Distribution Protocols



- Often referred to as signaling protocols
- Dynamically establishes a LSP
  - Exchanges label information
- Junos OS supports two types of label distribution protocols.
  - Resource Reservation Protocol (RSVP)
  - Label Distribution Protocol (LDP)
    - To reduce confusion, this content uses the acronym, LDP, to refer only to the particular protocol named Label Distribution Protocol.

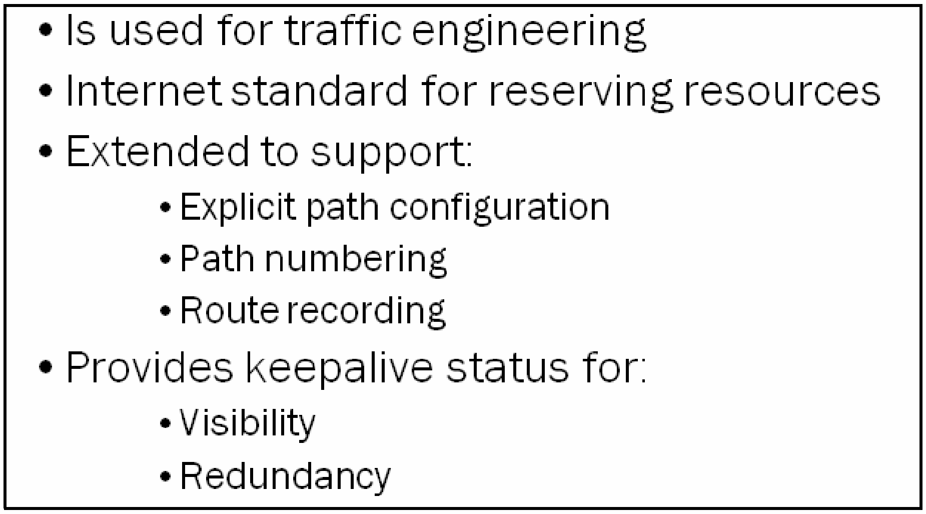
Label distribution protocols create and maintain the label-to-forwarding equivalence class (FEC) bindings along an LSP from the MPLS ingress label-switching router (LSR) to the MPLS egress LSP. A label distribution protocol is a set of procedures by which one LSR informs a peer LSR of the meaning of the labels used to forward traffic between them. MPLS uses this information to create the forwarding tables in each LSR.

Label distribution protocols are often referred to as signaling protocols. However, label distribution is a more accurate description of their function and is preferred in this study guide.

Unlike the static LSP we discussed in the last chapter, the label distribution protocols create and maintain an LSP dynamically with little or no user intervention. Once the label distribution protocols are configured for the signaling of an LSP, the egress router of an LSP will send label (and other) information in the upstream direction towards the ingress router based on the configured options.

The Junos OS supports two different label distribution protocols: RSVP and LDP. To reduce the chance of confusion this study guide will use the acronym LDP when referring to the particular protocol. RSVP is a generic label distribution protocol that was adapted for use in MPLS. LDP on the other hand was developed specifically to be used with MPLS.

RSVP



- Is used for traffic engineering
- Internet standard for reserving resources
- Extended to support:
  - Explicit path configuration
  - Path numbering
  - Route recording
- Provides keepalive status for:
  - Visibility
  - Redundancy

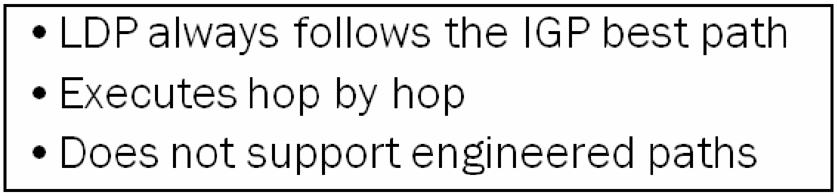
The Junos OS uses RSVP as the label distribution protocol for traffic engineered LSPs.

- RSVP was designed to be the resource reservation protocol of the Internet and “provide a general facility for creating and maintaining distributed reservation state across a set of multicast or unicast delivery paths” (RFC 2205). Reservations are an important part of traffic engineering, so it made sense to continue to use RSVP for this purpose rather than *reinventing the wheel*.
- RSVP was explicitly designed to support extensibility mechanisms by allowing it to carry what are called *opaque objects*. Opaque objects make no real sense to RSVP itself but are carried with the understanding that some adjunct protocol (such as MPLS) might find the information in these objects useful. This encourages RSVP extensions that create and maintain distributed state for information other than pure resource reservation. The designers believed that extensions could be developed easily to add support for explicit routes and label distribution.
- Extensions do not make the enhanced version of RSVP incompatible with existing RSVP implementations. An RSVP implementation can differentiate between LSP signaling and standard RSVP reservations by examining the contents of each message.
- With the proper extensions, RSVP provides a tool that consolidates the procedures for a number of critical signaling tasks into a single message exchange:
  - Extended RSVP can establish an LSP along an explicit path that would not have been chosen by the interior gateway protocol (IGP);
  - Extended RSVP can distribute label-binding information to LSRs in the LSP;
  - Extended RSVP can reserve network resources in routers comprising the LSP (the traditional role of RSVP); and
  - Extended RSVP permits an LSP to be established to carry best-effort traffic without making a specific resource reservation.

Thus, RSVP provides MPLS-signaled LSPs with a method of support for explicit routes (“go here, then here, finally here...”), path numbering through label assignment, and route recording (where the LSP actually goes from ingress to egress, which is very handy information to have).

RSVP also gives MPLS LSPs a keepalive mechanism to use for visibility (“this LSP is still here and available”) and redundancy (“this LSP appears dead... is there a secondary path configured?”).

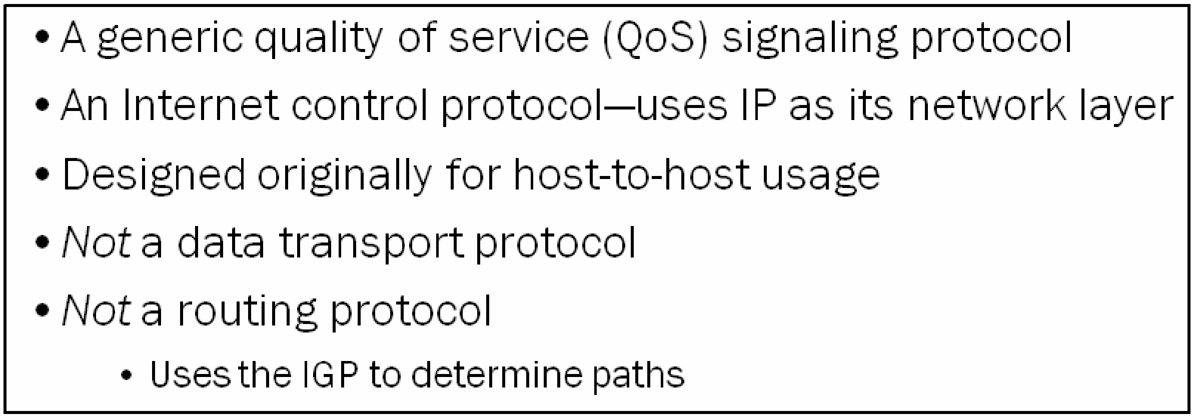
LDP



- LDP always follows the IGP best path
- Executes hop by hop
- Does not support engineered paths

LDP associates a set of destinations (prefixes) with each data link layer LSP. This set of destinations is called the FEC. These destinations all share a common data LSP path egress and a common unicast routing path. LDP supports topology-driven MPLS networks in best-effort, hop-by-hop implementations. The LDP signaling protocol always establishes LSPs that follow the contours of the IGP's shortest path. Traffic engineering is not possible with LDP.

RSVP

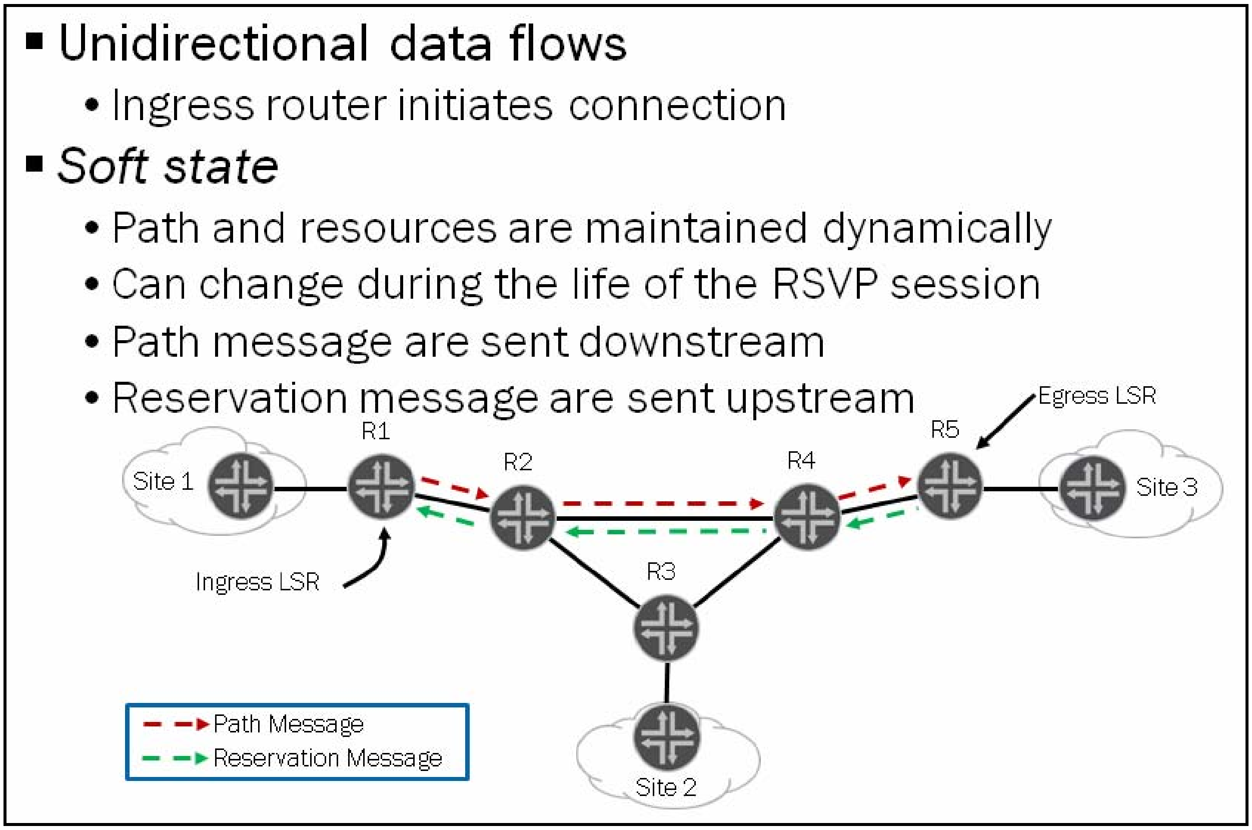


- A generic quality of service (QoS) signaling protocol
- An Internet control protocol—uses IP as its network layer
- Designed originally for host-to-host usage
- *Not* a data transport protocol
- *Not* a routing protocol
  - Uses the IGP to determine paths

RSVP is a generic signaling protocol designed originally to be used by applications to request and reserve specific quality-of-service (QoS) requirements across an internetwork. Resources are reserved hop by hop across the internetwork; each router receives the resource reservation request, establishes and maintains the necessary state for the data flow (if the requested resources are available), and forwards the resource reservation request to the next router along the path. As this behavior implies, RSVP is an internetwork control protocol, similar to Internet Control Message Protocol (ICMP), Internet Group Management Protocol (IGMP), and routing protocols.

RSVP does not transport application data, nor is it a routing protocol. It is simply a label distribution protocol. RSVP uses unicast and multicast IGP routing protocols to discover paths through the internetwork by consulting existing routing tables.

RSVP Data Flows



- Unidirectional data flows
  - Ingress router initiates connection
- *Soft state*
  - Path and resources are maintained dynamically
  - Can change during the life of the RSVP session
  - Path message are sent downstream
  - Reservation message are sent upstream

RSVP requests resources for unidirectional data flows. Each reservation is made for a data flow from a specific sender to a specific receiver. While RSVP messages are exchanged between the sender and receiver, the resulting path itself is unidirectional.

Although the application data flow is from the sender to the receiver, the reservation itself is initiated by the receiver. The sender notifies the receiver of a pending flow and characterizes the flow, and the receiver is responsible for requesting the resources. This design choice was made to accommodate heterogeneous receiver requirements and for multicast flows in which multiple receivers join and leave a multicast group.

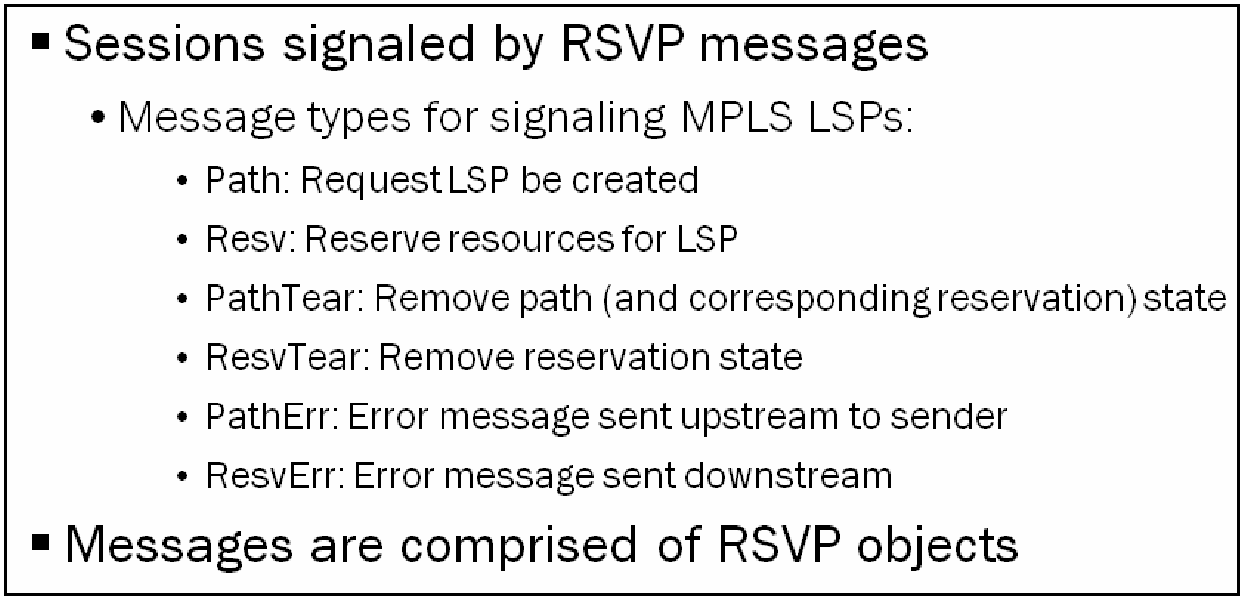
RSVP Is a Soft State Protocol

RSVP requests made to routers along the transit path cause each router either to reject the request for lack of resources or establish a *soft state*. This is in contrast to a *hard state*, which is associated with virtual connections that remain established for the duration of the data transfer. Soft state means that the logical path set up by RSVP is not associated necessarily with a physical path through the internetwork. The logical path might change during its lifetime as the result of the sender's changing the characterization of the traffic, causing the receiver to modify its reservation request, or causing the failure of a transit router.

Refreshing the soft state periodically maintains it. In standard RSVP implementations, sending path (downstream) and reservation (upstream) messages along the path accomplishes this refreshing. The Junos OS also uses the hello extension to maintain state and detect router failures; by default, hello messages are sent every nine seconds. The hello protocol can detect state changes within seconds, instead of several minutes as with standard RSVP implementations. The hello protocol is fully backward-compatible with older RSVP implementations. If a neighboring router does not support hello messages, The Junos OS RSVP uses standard soft-state procedures.

Note that path messages are sent *all the way* to the egress before they trigger the generation of a reservation message in the upstream direction. Messages are not paired in time on the link.

RSVP Messages Signal Sessions



- Sessions signaled by RSVP messages
  - Message types for signaling MPLS LSPs:
    - Path: Request LSP be created
    - Resv: Reserve resources for LSP
    - PathTear: Remove path (and corresponding reservation) state
    - ResvTear: Remove reservation state
    - PathErr: Error message sent upstream to sender
    - ResvErr: Error message sent downstream
- Messages are comprised of RSVP objects

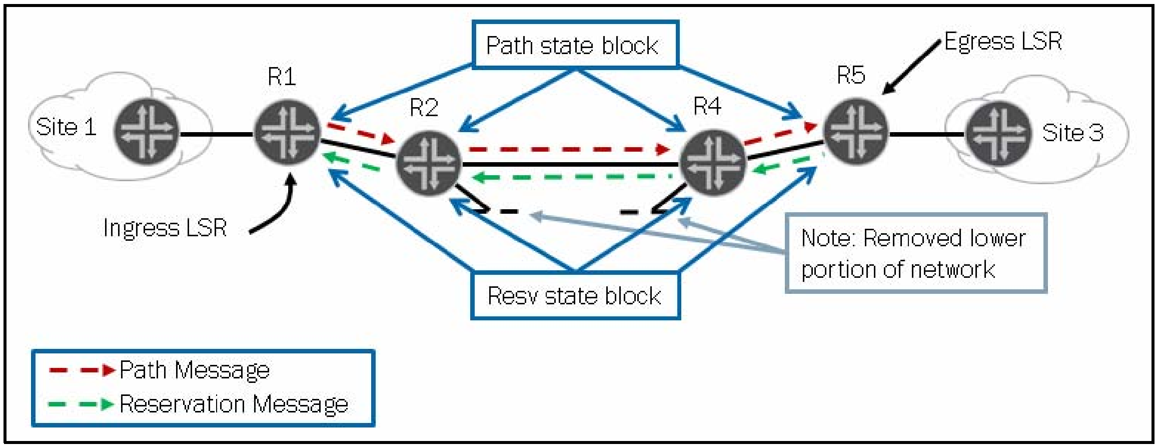
Different RSVP message types are used to establish and remove the RSVP state necessary for signaling MPLS LSPs. This graphic describes some of these message types:

- *Path* messages are sent by the ingress router and request that a path (LSP) be created. Path messages contain a destination address of the egress router but are processed by all RSVP routers along the requested path.
- *Resv* messages reserve resources—including label assignments—for the LSP. These messages are sent from the egress router and are forwarded hop by hop along the reverse path to the ingress router.
- *PathTear* messages delete the path and all dependant reservation state from routers that receive them. The PathTear message is originated by the sender, or by a router whose path state has timed out, and it always travels downstream toward the receiver.
- *PathErr* messages report errors in processing path messages, and travel upstream to the sender along the reverse route of path messages.
- *Resv Err* messages report errors in the processing of Resv messages, and travel hop by hop downstream to the receiver.

Messages Comprised of RSVP Objects

All RSVP messages share a common RSVP header that includes a field for identifying the message type. The messages is comprised of this common header followed by one or more RSVP objects. Later sections provide details on the RSVP objects used in the signaling of MPLS LSPs.

Soft State Information



To maintain a reservation state, RSVP tracks soft state in each router. The RSVP soft state is created and periodically refreshed by path and reservation-request messages. The state information is *soft* because it is deleted if no matching refresh messages arrive before the expiration of a cleanup timeout interval. This state can also be deleted as the result of an explicit teardown message. RSVP periodically scans the soft state to build and forward path and reservation-request refresh messages to succeeding hops.

Path State Block

Each path state block contains information about a particular session or LSP. This state is derived from information received in path messages and is stored in each router along the path.

Reservation State Block

Each reservation state block holds a reservation request that arrived in a particular Resv message, corresponding to the following three objects: (session, next hop, Filter_spec_list).

Traffic Engineering Extensions to RSVP

RSVP has been extended in several ways to support the establishment and maintenance of MPLS LSPs. For example, a *hello* mechanism has been added to RSVP to speed up router-to-router failure detection, and a label object has been added so that RSVP can signal label bindings.

Now Positioned as a Router Signaling Protocol

Extended RSVP is well suited as a router-to-router signaling protocol. Recall that in its original form RSVP was intended as a host-to-host signaling protocol. This change in signaling role makes ongoing sense as RSVP becomes the signaling protocol of choice for MPLS traffic engineering.

We discuss RSVP extensions in support of traffic engineering in subsequent pages.

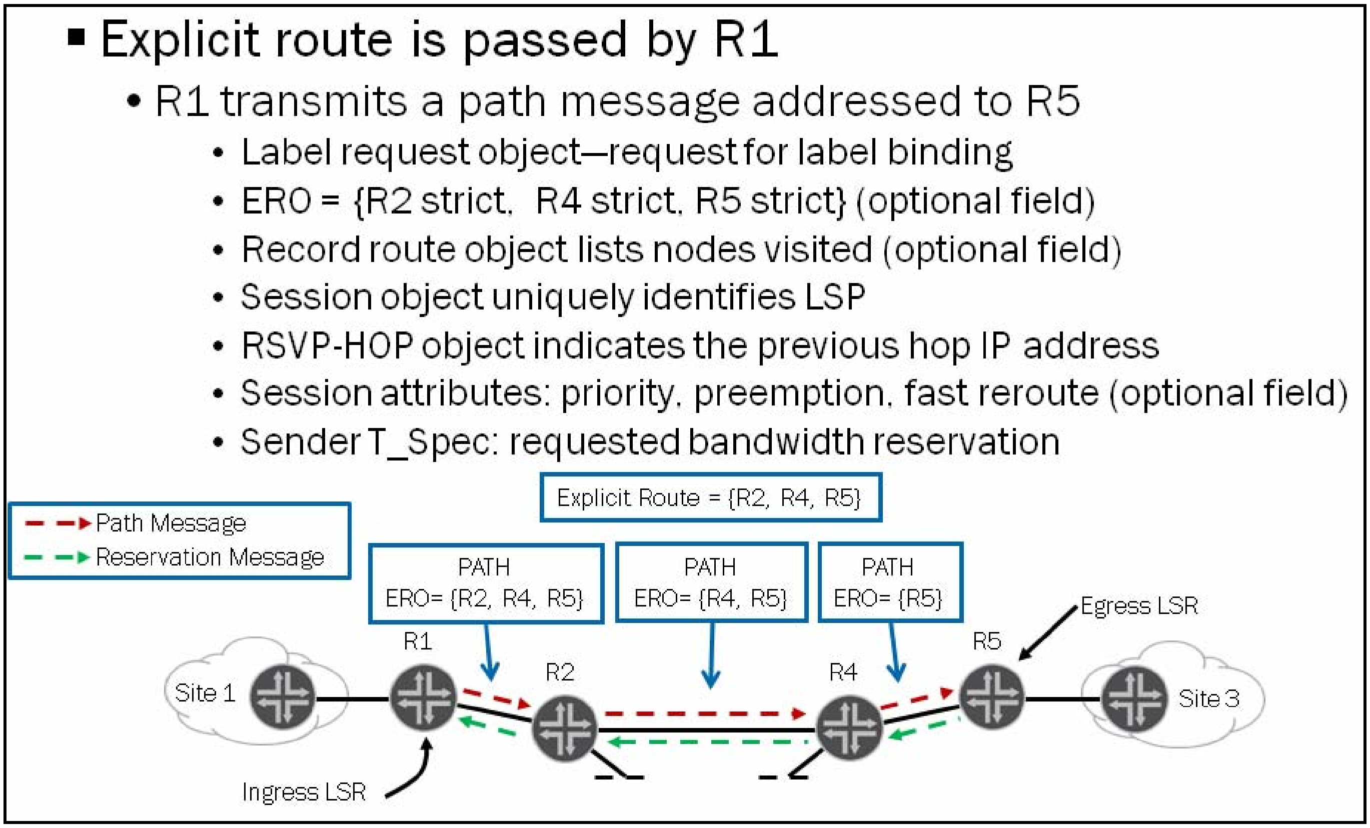
Path Message Extensions

A path message is transmitted by the ingress LSR toward the egress LSR when it wants to establish an LSP tunnel. The path message is addressed to the egress LSR, but it contains the router alert IP option (RFC 2113) in its IP header to indicate that the datagram requires special processing by intermediate routers. The path message can include a number of RSVP objects that provide TE-related signaling capabilities:

- LABEL_REQUEST object: Request for label mapping from downstream router.
- EXPLICIT_ROUTE object (ERO): Strict or loose list of routers that RSVP path messages must visit. The object can contain strict hops which indicate the exact path to use. In addition to strict hops the ERO can also contain loose hops which indicate the LSP must traverse this hop before reaching the egress. You can also use both strict and loose hops in the same ERO.
- RECORD_ROUTE object: List of addresses of all routers visited by the path message.

- SESSION_ATTRIBUTE object: Characteristics of the session.
- CoS FLOWSPEC object: Identify the resources that will be allocated.
- RSVP-HOP object: Indicates the IP address of the interface on the router sending the Path message. At the next router, the Hop object contains the previous hop IP address.

RSVP Path Message



- Explicit route is passed by R1
  - R1 transmits a path message addressed to R5
    - Label request object—request for label binding
    - ERO = {R2 strict, R4 strict, R5 strict} (optional field)
    - Record route object lists nodes visited (optional field)
    - Session object uniquely identifies LSP
    - RSVP-HOP object indicates the previous hop IP address
    - Session attributes: priority, preemption, fast reroute (optional field)
    - Sender T_Spec: requested bandwidth reservation

The ingress LSR generates an RSVP path message with a SESSION type of LSP_TUNNEL_IPv4. The path message contains a LABEL_REQUEST object that asks intermediate LSRs and the egress LSR to provide a label binding for this path. If the LABEL_REQUEST object is not supported by each LSR along the path, the ingress LSR is notified by the first LSR on the path that does not support the LABEL_REQUEST object. In addition to the LABEL_REQUEST object, an RSVP path message can also contain a number of optional objects:

- EXPLICIT_ROUTE object (ERO): Can be added to specify a predetermined path for the LSP across the service provider's network. When the ERO is present, the RSVP path message is forwarded towards the egress LSR along the path specified by the ERO, independent of the IGP’s shortest path.
- RECORD_ROUTE object (RRO): Allows the ingress LSR to receive a listing of the LSRs that the LSP tunnel traverses across the service provider's network.
- SESSION_ATTRIBUTE object: Can be included in the RSVP path message to aid in session identification and diagnosis. The SESSION_ATTRIBUTE object also controls the path setup priority, holding priority, and local-rerouting features.

The path message also contains the following standard RSVP objects (as opposed to extended objects):

- RSVP-HOP: Identifies the IP address of the neighboring RSVP router. It allows each device in the network to store information in both the path and Resv state blocks. Assists in properly routing RSVP messages.
- SENDER_TEMPLATE: Contains the sender's IP address and perhaps some additional information to identify the sender of the path message.
- SENDER_TSPEC: Describes the traffic characteristics of the flow that will be sent along the LSP. R5 uses this information to construct an appropriate RECEIVER_TSPEC (describing the traffic flow) and RSPEC (defining the desired QoS). The format and content of the TSPEC and RSPEC are opaque to RSVP.

RSVP Processing at Each Router

Assume that MPLS and RSVP are configured and enabled on R1, R2, R4, and R5.

R1 knows that the LSP should follow the explicit route (R1 to R2 to R4 to R5). Each router in the ERO has the L-bit cleared (making them strict hops) and is identified by a 32-bit IP version 4 (IPv4) prefix.

Required behavior: We want to establish an LSP to carry transit traffic that enters the service provider's network at R1 and exits at R5. Transit traffic should follow the physical path of the LSP rather than the route calculated by the IGP through the network. The result is that all transit traffic entering the network at R1 (with R5 as its BGP next hop) is forwarded along the LSP.

The physical path for the LSP has been specifically selected by another process to: 1) Reduce the amount of traffic flowing along the IGP route, 2) Optimize the overall utilization of network resources, 3) Enhance the traffic-oriented performance characteristics for the traffic flow, and 4) Enhance the traffic-oriented performance characteristics for the entire network.

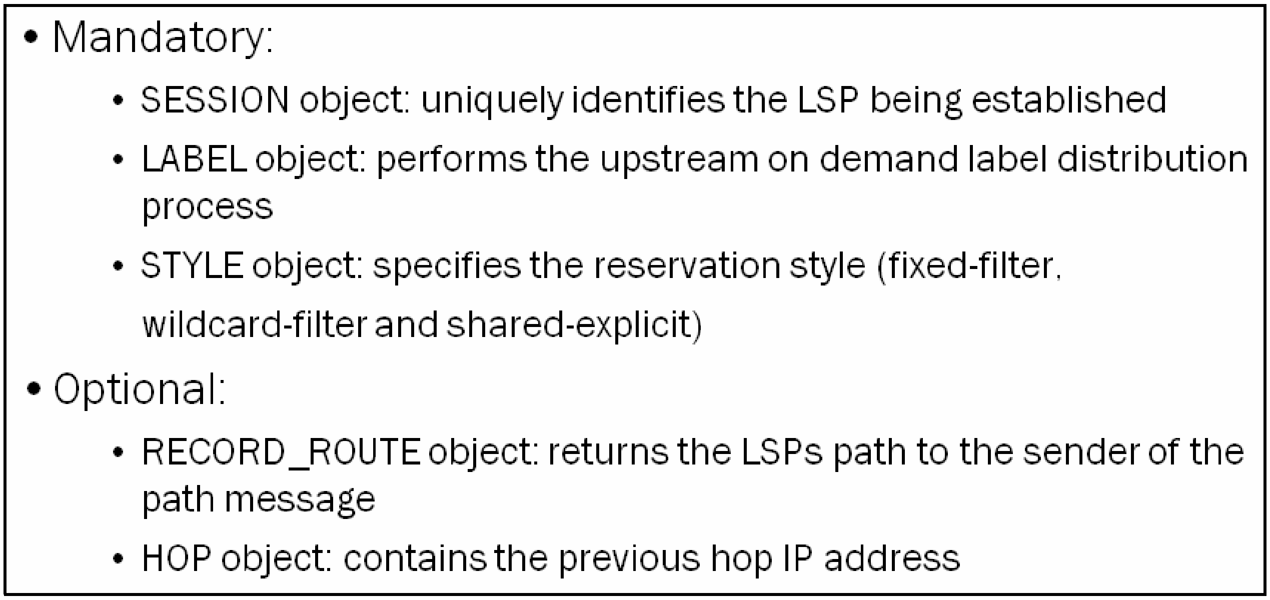
Processing at R1: The path message is transmitted toward R5 along the path specified by the ERO. Recall that the path message is addressed to the egress LSR but contains the router alert IP option to indicate that the datagram requires special processing by intermediate routers.

Processing at R2: When the path message arrives at R2, it records the LABEL_REQUEST object and the ERO in its path state block. The path state block also contains the IP address of the previous hop, the session, the sender, and the TSPEC. This information is used to route the corresponding Resv message back to R1. R2 forwards the path message toward R5 along the path specified in the ERO. If R2 cannot allocate a label for the LSP, it responds by sending a PathErr message with an unknown object class error to R1.

Processing at R4: When the path message arrives at R4, it records the LABEL_REQUEST object and ERO in its path state block. The path state block also contains the previous hop, session, sender, and TSPEC. The path message is forwarded toward R5 along the path specified in the ERO.

Processing at R5 (Egress LSR): When the path message arrives at R5, it notices from the LABEL_REQUEST object that it is the egress LSR for the LSP.

Resv Message Extensions



- Mandatory:
  - SESSION object: uniquely identifies the LSP being established
  - LABEL object: performs the upstream on demand label distribution process
  - STYLE object: specifies the reservation style (fixed-filter, wildcard-filter and shared-explicit)
- Optional:
  - RECORD_ROUTE object: returns the LSPs path to the sender of the path message
  - HOP object: contains the previous hop IP address

A Resv message is transmitted from the egress LSR toward the ingress in response to the receipt of a path message. The Resv message establishes path state in each LSR by distributing label bindings, requesting resource reservations along the path, and specifying the reservation style. The reservation style object can be a fixed-filter (FF), wildcard-filter (WF), or a shared-explicit (SE) value.

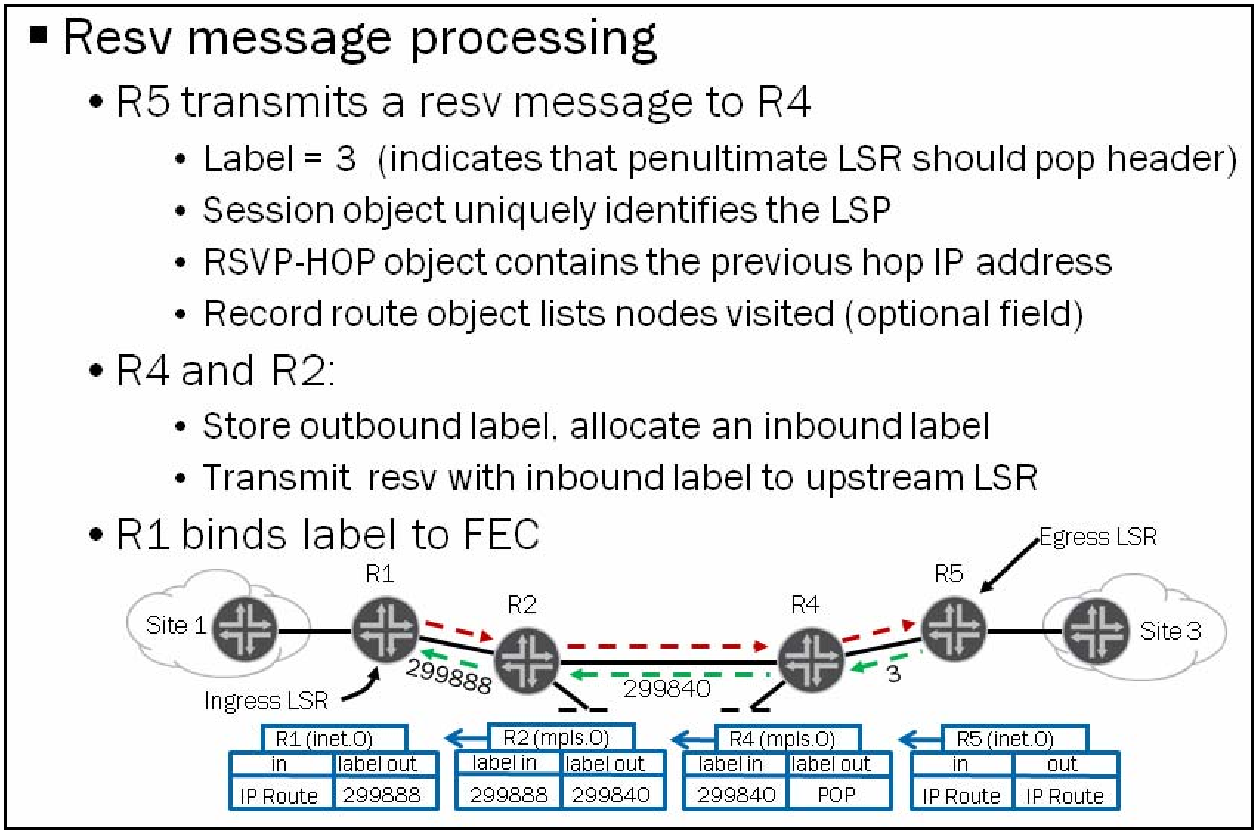
The fixed-filter reservation style consists of distinct reservations among explicit senders. Examples of applications that use fixed-filter-style reservations are video applications and unicast applications, which both require flows that have a separate reservation for each sender. The fixed filter reservation is the default style in RSVP LSPs.

The wildcard-filter reservation style consists of shared reservations among wildcard senders. This type of reservation reserves bandwidth for any and all senders, and propagates upstream toward all senders. A sample application for wildcard filter reservations is an audio application in which each sender transmits a distinct data stream. Typically, only a few senders are transmitting at any one time and does not require a separate reservation for each sender.

The shared-explicit reservation style consists of shared reservations among explicit senders. This type of reservation reserves bandwidth for a limited group of senders. A sample application is an audio application similar to wildcard filter reservations.

Like the path message, the Resv message can contain the RSVP-Hop object to identify the previous hops IP address and a record route object that lists all routers visited by the Resv message.

RSVP Processing at Each Router



- Resv message processing
  - R5 transmits a resv message to R4
    - Label = 3 (indicates that penultimate LSR should pop header)
    - Session object uniquely identifies the LSP
    - RSVP-HOP object contains the previous hop IP address
    - Record route object lists nodes visited (optional field)
  - R4 and R2:
    - Store outbound label, allocate an inbound label
    - Transmit resv with inbound label to upstream LSR
  - R1 binds label to FEC

R1 (inet.0)

in	label out
IP Route	299888

R2 (mpls.0)

label in	label out
299888	299840

R4 (mpls.0)

label in	label out
299840	POP

R5 (inet.0)

in	out
IP Route	IP Route

Following standard RSVP procedures, R5 generates a Resv message for the session to distribute labels and establish forwarding state for the LSP tunnel. The IP destination address of the Resv message is the unicast address of the previous-hop router, as obtained from the LSR's local path state block.

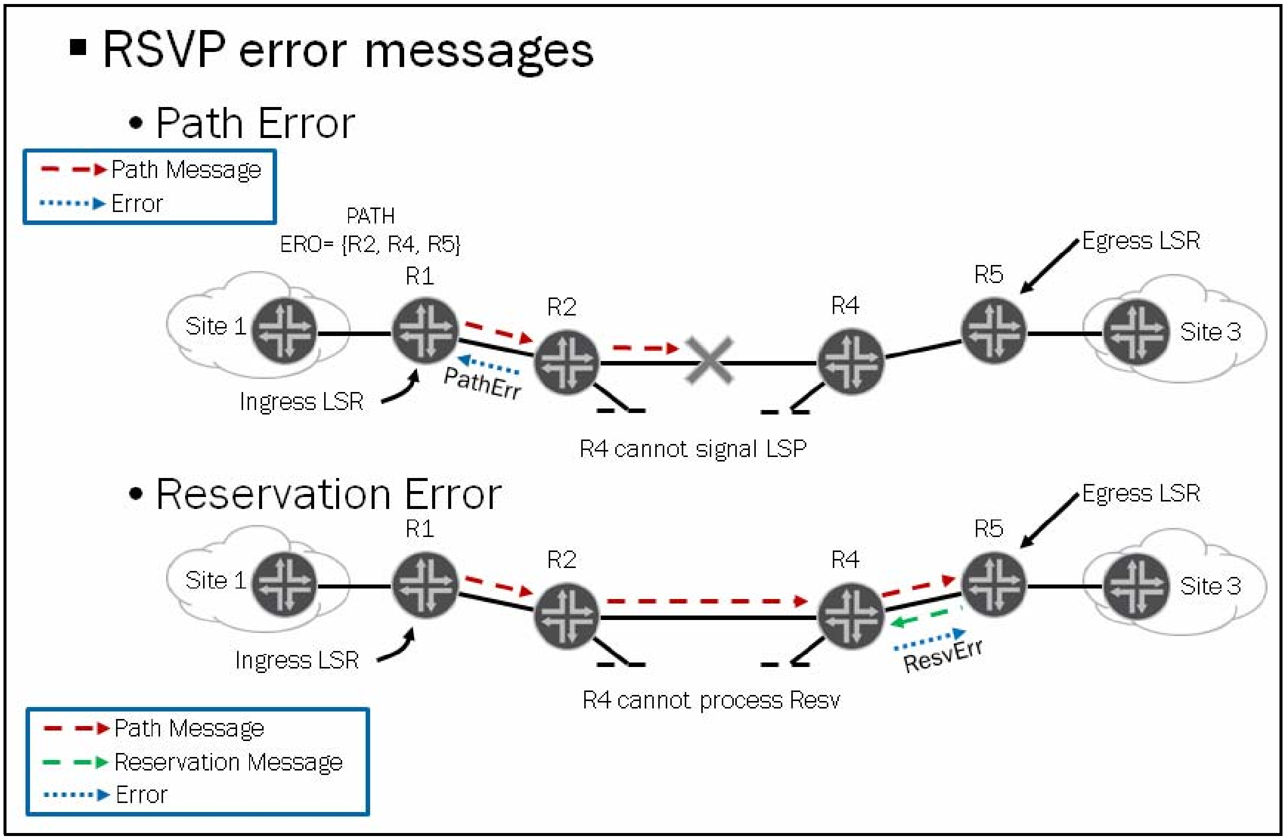
Processing at R5: R5 allocates a label with a value of 3 and places it in the LABEL object of the Resv message. The value of 3 has a special meaning to R4; when R4 receives an indication that is should assign label value 3, it knows that it is the penultimate LSR for the LSP. In this case, R4 pops the top label off the label stack and forwards the packet to R5, the egress router. If R1 inserted a TSPEC in the path message, R5 uses this information to construct an appropriate receiver TSPEC and RSPEC. The Resv message is transmitted back toward R1 through R4. The Resv message does not carry a reverse ERO to find its way back along the path to R1. Instead, the Resv message follows the reverse path using the RSVP-HOP object, which is set up in the path state block.

Processing at R4: R4 receives the Resv message containing the label assigned by R5. R4 stores the label (3) as part of the reservation state for the LSP. R4 uses this label when forwarding outgoing traffic along the LSP to R5. R4 allocates a new label (299840) and places it in the LABEL object (replacing the received label) of the Resv message that it sends upstream to R2. This is the label that R4 uses to identify incoming traffic on the LSP from R2.

Processing at R2: R2 receives the Resv message containing the label assigned by R4. R2 stores the label (299840) as part of the reservation state for the LSP. R2 uses this label when forwarding outgoing traffic along the LSP to R4. R2 allocates a new label (299888) and places it in the LABEL object (replacing the label received from R4) of the Resv message that it sends upstream to R1. R2 uses this label to identify incoming traffic on the LSP from R1.

Processing at R1: R1 receives the Resv message that contains the label assigned by LSR. It uses this label for all outgoing traffic that it maps to the LSP. Because of these operations, the LSP is established from R1 to R5 following the explicitly routed path specified in the ERO. R1 forwards traffic for prefix X through R2 by pushing the label signaled by R2 (299888) before forwarding the packet to R2.

Path and Resv Error Messages

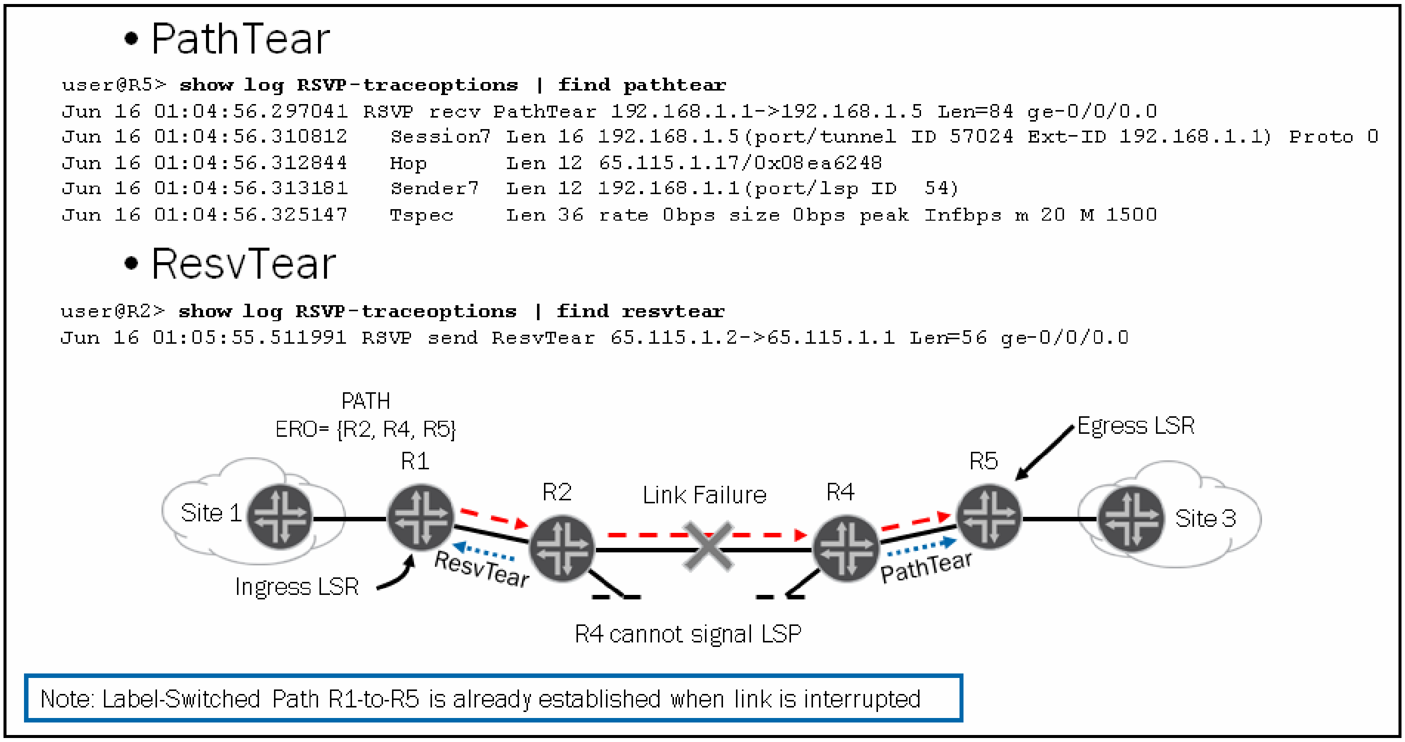


- RSVP error messages
  - Path Error
  - Reservation Error

There are two RSVP error messages, ResvErr and PathErr. PathErr messages are very simple; they are sent upstream to the sender that created the error, and they do not change path state in the routers though which they pass. There are only a few possible causes of path errors. However, a number of ways exist for a valid reservation request to fail at some router along the path. A router might also decide to preempt an established reservation. Because a reservation request that fails might be the result of merging a number of requests, a reservation error must be reported to all of the responsible receivers. The handling of ResvErr messages is somewhat complex as a result.

The upper portion of the graphic shows how an error in handing a path message results in the generation of a PathErr message in the upstream direction. The bottom portion of the graphic demonstrates typical Resv message error handling. In this example the egress router (R5) responds to the receipt of a path message with a Resv message that is sent to R4. R4 generates a ResvErr message back towards the source of the reservation request (in the down stream direction) because it is unable to accommodate the reservation request from some reason. Note that a path or reservation error message is *not* sent upstream in this case because R4 has not signaled any reservation state for the associated path message in this example. Therefore, there is no need to generate a ResvTear message. If R4 had previously signaled a reservation to its upstream neighbor (R2), the error that causes R4 to send a ResvErr back to R5 also results in the generation of a ResvTear message in the upstream direction to remove the existing reservation state.

RSVP Teardown Messages



- PathTear

```
user@R5> show log RSVP-traceoptions | find pathtear
Jun 16 01:04:56.297041 RSVP recv PathTear 192.168.1.1->192.168.1.5 Len=84 ge-0/0/0.0
Jun 16 01:04:56.310812   Session7 Len 16 192.168.1.5(port/tunnel ID 57024 Ext-ID 192.168.1.1) Proto 0
Jun 16 01:04:56.312844   Hop      Len 12 65.115.1.17/0x08ea6248
Jun 16 01:04:56.313181   Sender7  Len 12 192.168.1.1(port/lsp ID  54)
Jun 16 01:04:56.325147   Tspec    Len 36 rate 0bps size 0bps peak Infbps m 20 M 1500
```

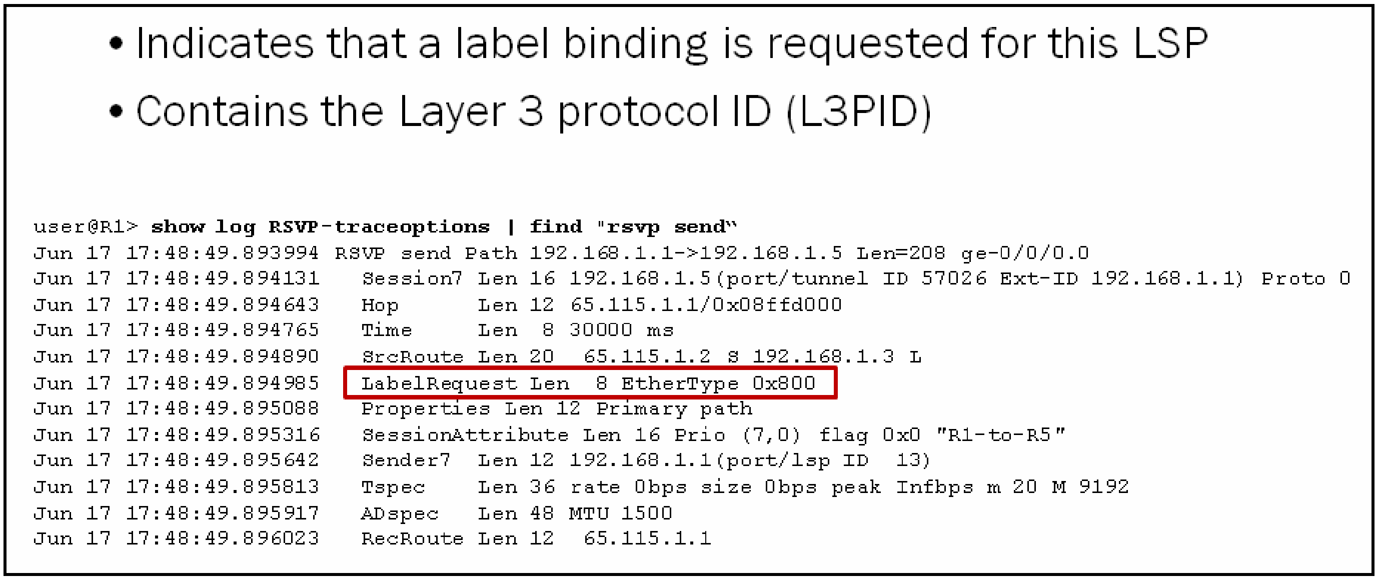
- ResvTear

```
user@R2> show log RSVP-traceoptions | find resvtear
Jun 16 01:05:55.511991 RSVP send ResvTear 65.115.1.2->65.115.1.1 Len=56 ge-0/0/0.0
```

Note: Label-Switched Path R1-to-R5 is already established when link is interrupted

RSVP *teardown* messages remove path or reservation state immediately. The two types of RSVP teardown messages are PathTear and ResvTear. A PathTear message travels towards all receivers downstream from its point of initiation and deletes path state, as well as all dependent reservation state, along the way. A ResvTear message deletes reservation state and travels towards all senders upstream from its point of initiation. You can conceptualize a PathTear (ResvTear) message as a reversed-sense path message (Resv message, respectively). A teardown request can be initiated either by an application in an end system (sender or receiver), or by a router as the result of state timeout or service preemption. Once initiated, a teardown request must be forwarded hop by hop without delay.

The Label Request Object



- Indicates that a label binding is requested for this LSP
- Contains the Layer 3 protocol ID (L3PID)

```
user@R1> show log RSVP-traceoptions | find "rsvp send"
Jun 17 17:48:49.893994 RSVP send Path 192.168.1.1->192.168.1.5 Len=208 ge-0/0/0.0
Jun 17 17:48:49.894131   Session7 Len 16 192.168.1.5(port/tunnel ID 57026 Ext-ID 192.168.1.1) Proto 0
Jun 17 17:48:49.894643   Hop      Len 12 65.115.1.1/0x08ffd000
Jun 17 17:48:49.894765   Time     Len  8 30000 ms
Jun 17 17:48:49.894890   SrcRoute Len 20  65.115.1.2 S 192.168.1.3 L
Jun 17 17:48:49.894985   LabelRequest Len  8 EtherType 0x800
Jun 17 17:48:49.895088   Properties Len 12 Primary path
Jun 17 17:48:49.895316   SessionAttribute Len 16 Prio (7,0) flag 0x0 "R1-to-R5"
Jun 17 17:48:49.895642   Sender7  Len 12 192.168.1.1(port/lsp ID  13)
Jun 17 17:48:49.895813   Tspec    Len 36 rate 0bps size 0bps peak Infbps m 20 M 9192
Jun 17 17:48:49.895917   ADspec   Len 48 MTU 1500
Jun 17 17:48:49.896023   RecRoute Len 12  65.115.1.1
```

To establish an LSP tunnel, the ingress LSR generates a path message that contains a LABEL_REQUEST object. The presence of a LABEL_REQUEST object indicates that a label binding is requested for this LSP. The LABEL_REQUEST object also contains the Layer 3 protocol ID (L3PID) that identifies the Layer 3 protocol that will traverse the LSP tunnel. The L3PID is required because it is not possible to assume that an LSP tunnel transports IPv4 traffic—the Layer 3 protocol cannot be inferred from the Layer 2 header, which simply identifies the higher layer protocol as MPLS. In this example, the EtherType is set to 0x0800, which indicates that IPv4 will be transported over the LSP.

The three possible LABEL_REQUEST types are the following:

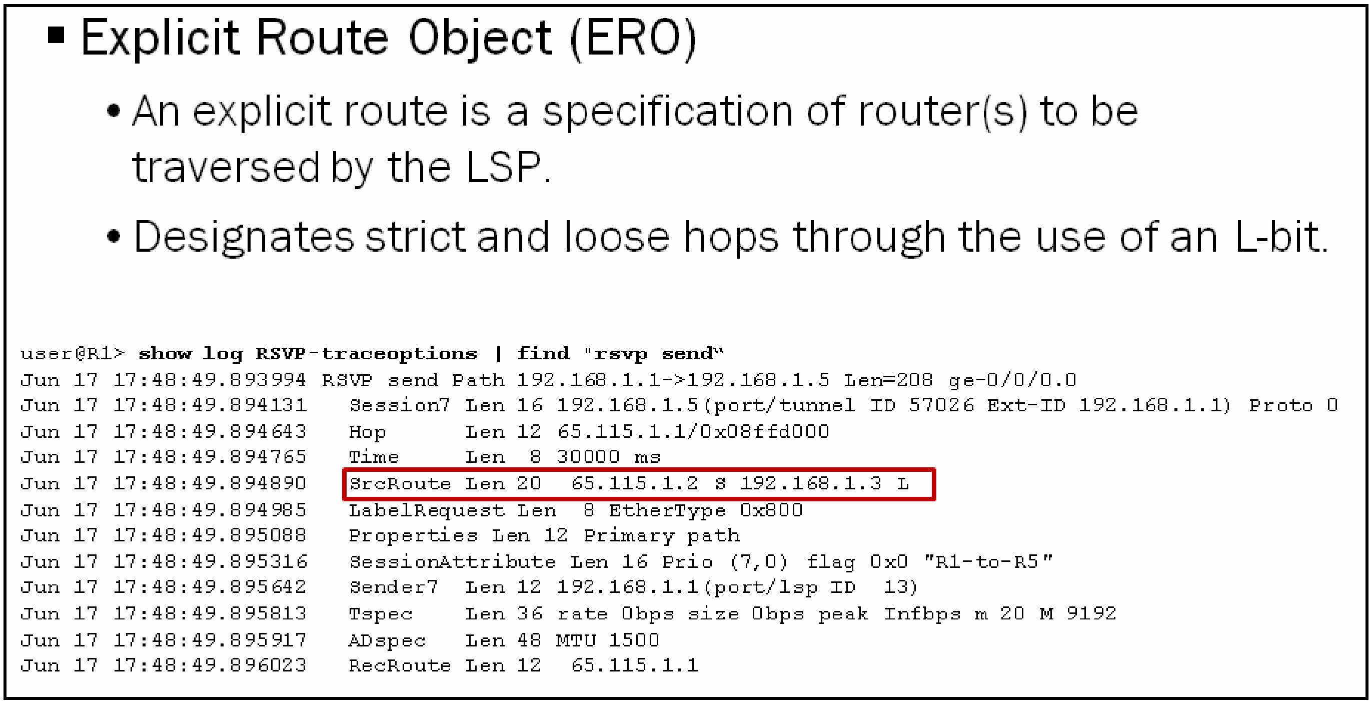
- *Request for a label that does not specify a specific label range*: This is the common case in which the MPLS label is carried in a standard MPLS shim header that sits between the data link and network layer headers. The Junos OS always uses this type of label request, as shown in the trace output on the graphic.
- *Request for a label with an ATM label range that specifies both the minimum and maximum virtual path identifier (VPI) and virtual channel identifier (VCI) values*: This type of request is useful when the MPLS label is carried in a Layer 2 Asynchronous Transfer Mode (ATM) header.
- *Request for a label with a Frame Relay label range that specifies the minimum and maximum data-link connection identifier (DLCI) values*: This type of request is useful when the MPLS label is carried in a Layer 2 Frame Relay header.

When a path message arrives at an LSR, the LSR stores the LABEL_REQUEST object in the local path state block for the LSP. If a label range is specified, the label allocation process must assign a label from that range.

Potential error conditions include the following:

- If the LSR receiving the path message recognizes the LABEL_REQUEST object but is unable to assign a label, it sends a PathErr message (indicating a routing problem or MPLS label allocation failure) toward the ingress LSR.
- If the receiver cannot support the L3PID, it sends a PathErr (routing problem/unsupported L3PID) toward the ingress LSR. This error causes the LSP setup request to fail.
- If the LSR receiving the message does not recognize the LABEL_REQUEST object, it sends a PathErr (unknown object class) toward the ingress LSR. This error also causes LSP setup to fail.

The Explicit Route Object



- Explicit Route Object (ERO)
  - An explicit route is a specification of router(s) to be traversed by the LSP.
  - Designates strict and loose hops through the use of an L-bit.

```
user@R1> show log RSVP-traceoptions | find "rsvp send"
Jun 17 17:48:49.893994 RSVP send Path 192.168.1.1->192.168.1.5 Len=208 ge-0/0/0.0
Jun 17 17:48:49.894131   Session7 Len 16 192.168.1.5(port/tunnel ID 57026 Ext-ID 192.168.1.1) Proto 0
Jun 17 17:48:49.894643   Hop      Len 12 65.115.1.1/0x08ffd000
Jun 17 17:48:49.894765   Time     Len  8 30000 ms
Jun 17 17:48:49.894890   SrcRoute Len 20  65.115.1.2 S 192.168.1.3 L
Jun 17 17:48:49.894985   LabelRequest Len  8 EtherType 0x800
Jun 17 17:48:49.895088   Properties Len 12 Primary path
Jun 17 17:48:49.895316   SessionAttribute Len 16 Prio (7,0) flag 0x0 "R1-to-R5"
Jun 17 17:48:49.895642   Sender7  Len 12 192.168.1.1(port/lsp ID  13)
Jun 17 17:48:49.895813   Tspec    Len 36 rate 0bps size 0bps peak Infbps m 20 M 9192
Jun 17 17:48:49.895917   ADspec   Len 48 MTU 1500
Jun 17 17:48:49.896023   RecRoute Len 12  65.115.1.1
```

By adding the EXPLICIT_ROUTE object to the path message, the ingress LSR can specify a predetermined explicit route for the LSP that is independent of the IGP's shortest path view. The ERO is intended to be used only for unicast applications and only when all routers along the explicit route support RSVP and the ERO.

An explicit route is encoded as a series of subobjects contained in the ERO. Each subobject can identify a group of routers in the explicit route or can specify an operation to be performed along the path. Hence, an explicit route is a specification of groups of routers to be traversed and a set of operations to be performed along the path. The ERO portion of the path message is highlighted in trace output shown on the graphic.

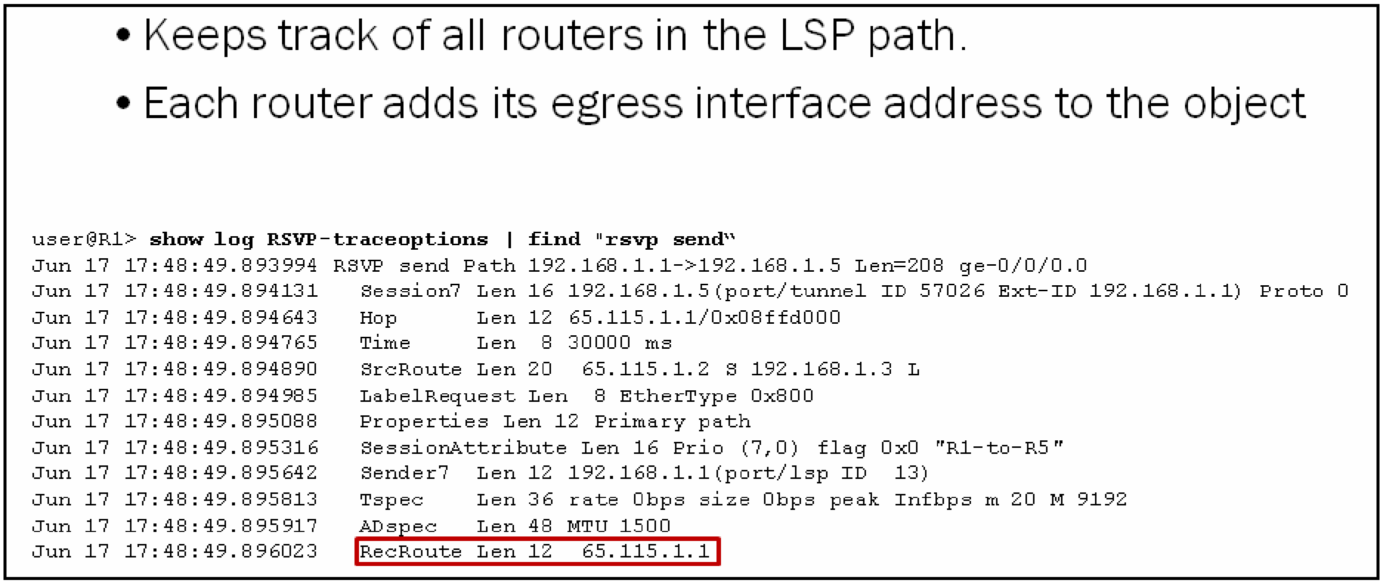
ERO coding designates strict and loose hops through the use of an L bit. If the L-bit is set, the subobject represents a loose hop, if set to a 0, the corresponding hop is considered to be strict.

Currently, four types of EXPLICIT_ROUTE subobjects are defined:

- *IPv4 Prefix*: Identifies an abstract router consisting of the set of routers that have an IP prefix that lies within this IPv4 prefix. A prefix with a length of 32 bits indicates a single IPv4 router.
- *IP version 6 (IPv6) Prefix*: Identifies an abstract router consisting of the set of routers that have an IP prefix that lies within this IPv6 prefix. A prefix with a length of 128 bits indicates a single IPv6 router.
- *Autonomous System Number*: Identifies an abstract router consisting of the set of routers belonging to the autonomous system.
- *MPLS LSP Termination*: Indicates that the prior abstract router should remove one level of the label stack from all packets following this LSP tunnel.

The use of EROs can lead to loops in the forwarding of the RSVP path message. Such a loop will cause LSP setup to fail. Loops are detected with the RRO, as described on the next page.

The Record Route Object



- Keeps track of all routers in the LSP path.
- Each router adds its egress interface address to the object

```
user@R1> show log RSVP-traceoptions | find "rsvp send"
Jun 17 17:48:49.893994 RSVP send Path 192.168.1.1->192.168.1.5 Len=208 ge-0/0/0.0
Jun 17 17:48:49.894131   Session7 Len 16 192.168.1.5(port/tunnel ID 57026 Ext-ID 192.168.1.1) Proto 0
Jun 17 17:48:49.894643   Hop      Len 12 65.115.1.1/0x08ffd000
Jun 17 17:48:49.894765   Time     Len  8 30000 ms
Jun 17 17:48:49.894890   SrcRoute Len 20  65.115.1.2 S 192.168.1.3 L
Jun 17 17:48:49.894985   LabelRequest Len  8 EtherType 0x800
Jun 17 17:48:49.895088   Properties Len 12 Primary path
Jun 17 17:48:49.895316   SessionAttribute Len 16 Prio (7,0) flag 0x0 "R1-to-R5"
Jun 17 17:48:49.895642   Sender7  Len 12 192.168.1.1(port/lsp ID  13)
Jun 17 17:48:49.895813   Tspec    Len 36 rate 0bps size 0bps peak Infbps m 20 M 9192
Jun 17 17:48:49.895917   ADspec   Len 48 MTU 1500
Jun 17 17:48:49.896023   RecRoute Len 12  65.115.1.1
```

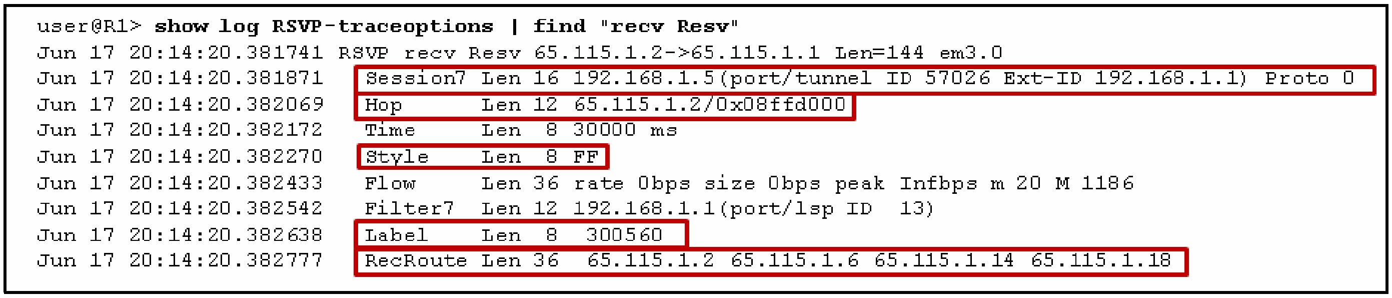
By adding the record route object (RRO) to the path message, the ingress LSR can receive information about the actual route that the LSP tunnel traverses. The contents of a RRO is a series of data items called subobjects. Two types of subobjects are currently defined: IPv4 addresses and IPv6 addresses.

Three possible applications for the RRO in RSVP signaling include the following:

- Discover Layer 3 routing loops or loops inherent in the explicit route because the RRO is analogous to a path vector.
- Collect up-to-date detailed path information about the LSP setup session.
- Define the contents of an EXPLICIT_ROUTE object to be used in the next path message. Using an EXPLICIT_ROUTE object derived from the previous RRO allows the session path to be pinned down. When pinned down, the path will not change even if a better path becomes available.

When an ingress LSR attempts to establish an LSP tunnel, it creates a path message that contains a LABEL_REQUEST object. The path message can also contain an RRO object. The initial RRO contains the ingress LSR's IP address, as shown on the graphic. When a path message containing an RRO is received by an intermediate router, the router stores a copy of the RRO in its path state block and adds its own IP address to the RRO. When the egress LSR receives a path message with an RRO, it adds the received RRO to its subsequent Resv message. After the exchange of path and Resv messages, each router along the path will have the complete route of the LSP from ingress to egress, which is extremely useful for network management purposes.

Resv Message Objects

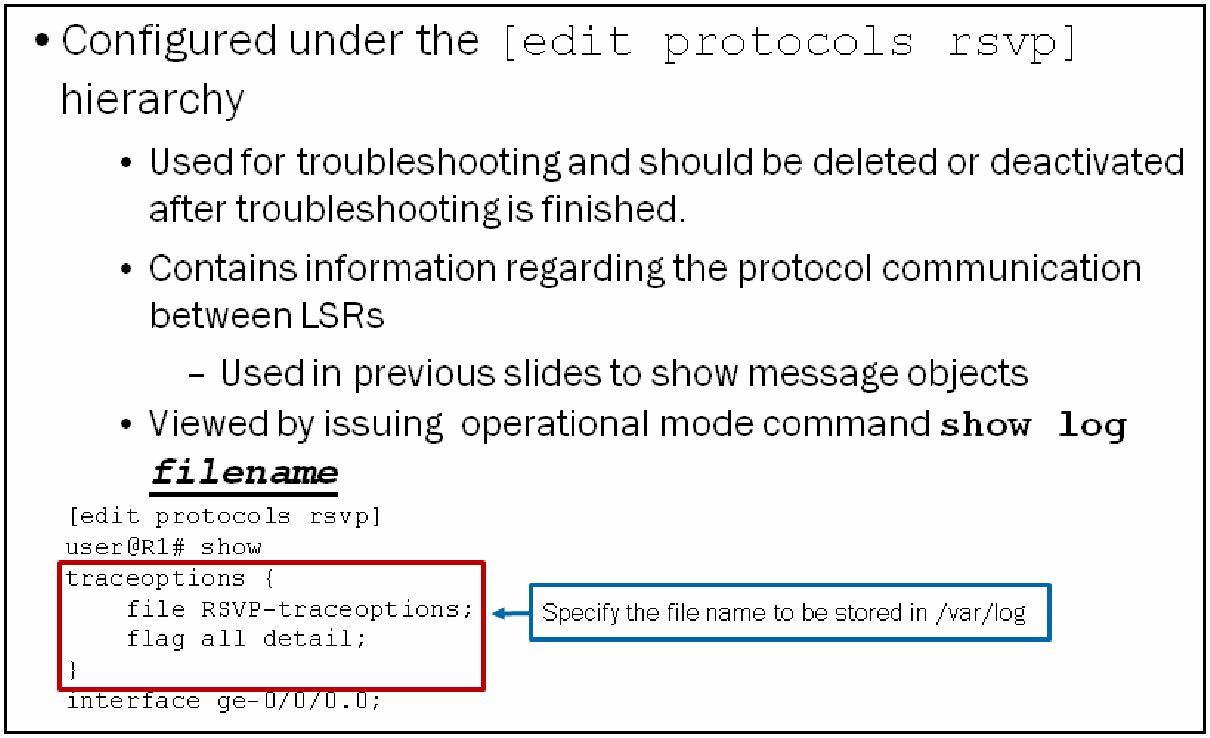


```
user@R1> show log RSVP-traceoptions | find "recv Resv"
Jun 17 20:14:20.381741 RSVP recv Resv 65.115.1.2->65.115.1.1 Len=144 em3.0
Jun 17 20:14:20.381871  Session7 Len 16 192.168.1.5(port/tunnel ID 57026 Ext-ID 192.168.1.1) Proto 0
Jun 17 20:14:20.382069  Hop      Len 12 65.115.1.2/0x08ffd000
Jun 17 20:14:20.382172  Time     Len  8 30000 ms
Jun 17 20:14:20.382270  Style    Len  8 FF
Jun 17 20:14:20.382433  Flow     Len 36 rate 0bps size 0bps peak Infbps m 20 M 1186
Jun 17 20:14:20.382542  Filter7  Len 12 192.168.1.1(port/lsp ID  13)
Jun 17 20:14:20.382638  Label    Len  8  300560
Jun 17 20:14:20.382777  RecRoute Len 36  65.115.1.2 65.115.1.6 65.115.1.14 65.115.1.18
```

The RSVP Resv message can contain a number of different RSVP objects:

- *LABEL object*: Performs the upstream on-demand label distribution process. This object can contain either a single MPLS label or a stack of labels. If an MPLS implementation does not support a label stack, only the top label is examined. When the LSR receives a Resv message corresponding to a previous path message, it confirms that the Resv message was transmitted by the next hop in the LSP. The LSR then binds a locally allocated label (300560 in this example) to the LSP's incoming interface. The incoming interface is the one over which the LSR received the LSP's corresponding path message.
- *RECORD_ROUTE object*: Returns the LSPs path to the sender of the original path message.
- *STYLE object*: Carries the value for shared explicit, wildcard or fixed filter reservations.
- *HOP object:* Indicates the previous hops IP address.
- *SESSION object*: Carries the parameters that uniquely identify the session (port, protocol, and destination address).

Traceoptions



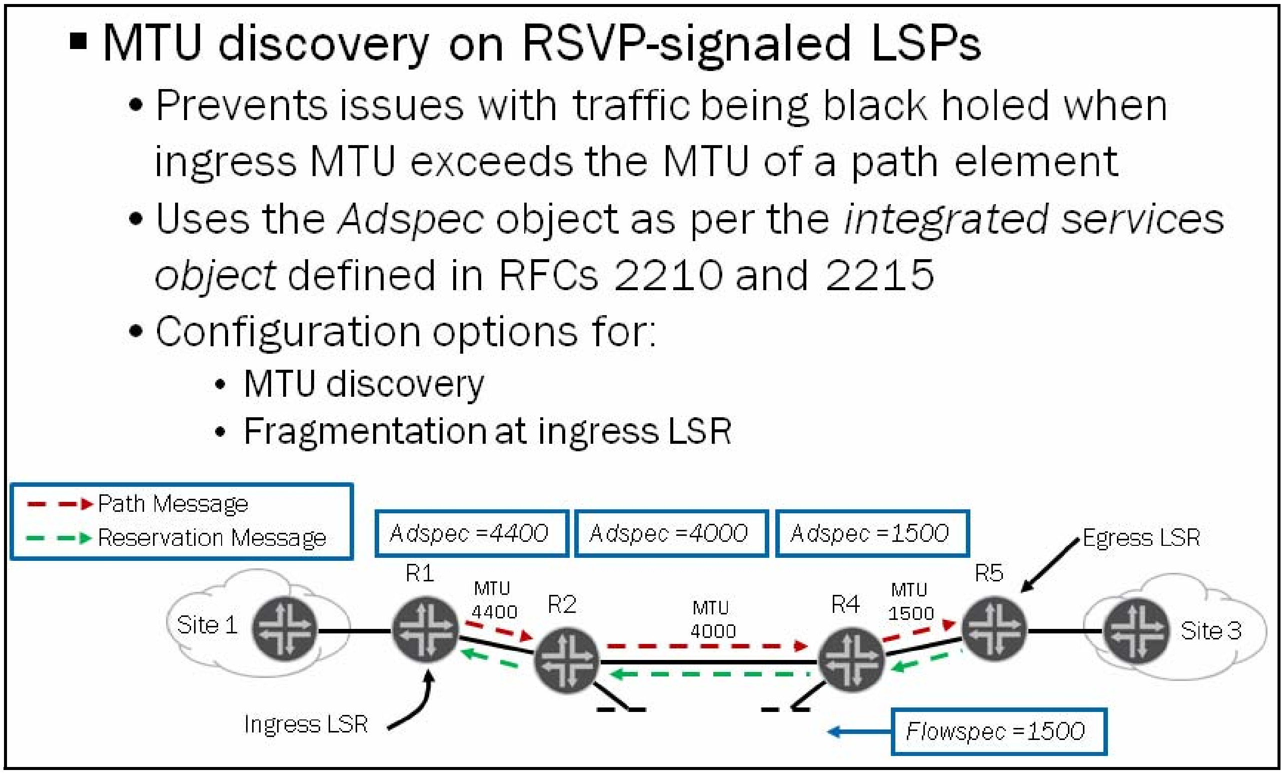
- Configured under the `[edit protocols rsvp]` hierarchy
  - Used for troubleshooting and should be deleted or deactivated after troubleshooting is finished.
  - Contains information regarding the protocol communication between LSRs
    - Used in previous slides to show message objects
  - Viewed by issuing operational mode command **`show log`** ***`filename`***

```
[edit protocols rsvp]
user@R1# show
traceoptions {
    file RSVP-traceoptions;
    flag all detail;
}
interface ge-0/0/0.0;
```

Specify the file name to be stored in /var/log

Traceoptions are configured under the protocol that you want to troubleshoot. Because the information gathered can be very extensive it is recommended that you only turn on traceoptions when troubleshooting a specific issue. The more specific you can make the flag options, the easier it is to locate the information you need to review. As displayed in earlier graphics you can also use match and find conditions to narrow down the information displayed when looking at the file. In our example we capture all available information for RSVP communication. We also included the detailed tag to increase the detail of information captured in the RSVP-traceoptions file.

MTU Discovery for RSVP-Signaled LSPs



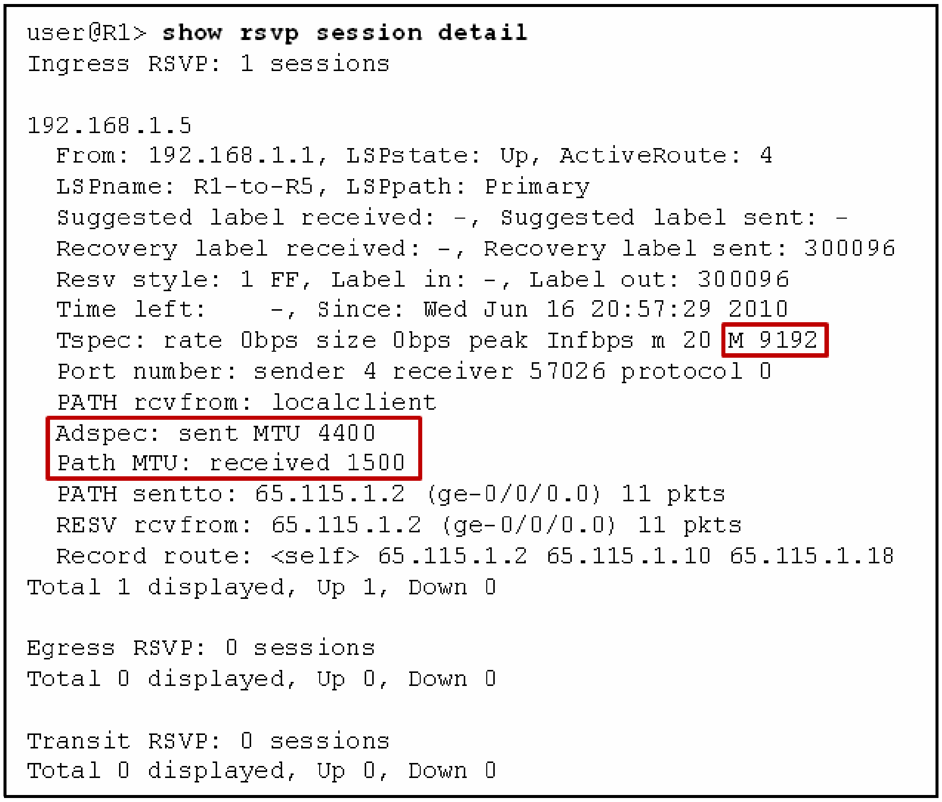
- MTU discovery on RSVP-signaled LSPs
  - Prevents issues with traffic being black holed when ingress MTU exceeds the MTU of a path element
  - Uses the *Adspec* object as per the *integrated services object* defined in RFCs 2210 and 2215
  - Configuration options for:
    - MTU discovery
    - Fragmentation at ingress LSR

The Junos OS supports maximum transmission unit (MTU) discovery when using RSVP signaling. The discovery mechanism is performed according to the integrated services object as defined in RFCs 2210 and 2215. This feature helps to prevent the black hole condition that is normally associated with mismatched MTUs along an the elements that make up an LSP.

MTU discovery signaling can be configured independently of ingress LSR fragmentation, but you must have mtu-signaling configured if you are configuring the allow fragmentation option. Both options are configured at the `[edit protocols mpls path-mtu]` hierarchy.

In operation, the ingress LSR sets the M value in the TSPEC to 9192 and codes the egress interface's IP MTU in the ADSPEC object in the path message. At each hop transit LSRs update the MTU value in the ADSPEC object with the minimum of the incoming value and egress interface MTU. When the path message is received by the egress LSR the smaller of the two values coded in the TSPEC and ADSPEC objects is signaled back to the ingress router using the Flowspec object in the Resv message. This behavior is shown on the graphic where the 1500-byte MTU is correctly reported to the egress router in the ADSPEC object.

MTU Discovery and Fragmentation



```
user@R1> show rsvp session detail
Ingress RSVP: 1 sessions

192.168.1.5
  From: 192.168.1.1, LSPstate: Up, ActiveRoute: 4
  LSPname: R1-to-R5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 300096
  Resv style: 1 FF, Label in: -, Label out: 300096
  Time left:    -, Since: Wed Jun 16 20:57:29 2010
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 9192
  Port number: sender 4 receiver 57026 protocol 0
  PATH rcvfrom: localclient
  Adspec: sent MTU 4400
  Path MTU: received 1500
  PATH sentto: 65.115.1.2 (ge-0/0/0.0) 11 pkts
  RESV rcvfrom: 65.115.1.2 (ge-0/0/0.0) 11 pkts
  Record route: <self> 65.115.1.2 65.115.1.10 65.115.1.18
Total 1 displayed, Up 1, Down 0

Egress RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0

Transit RSVP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

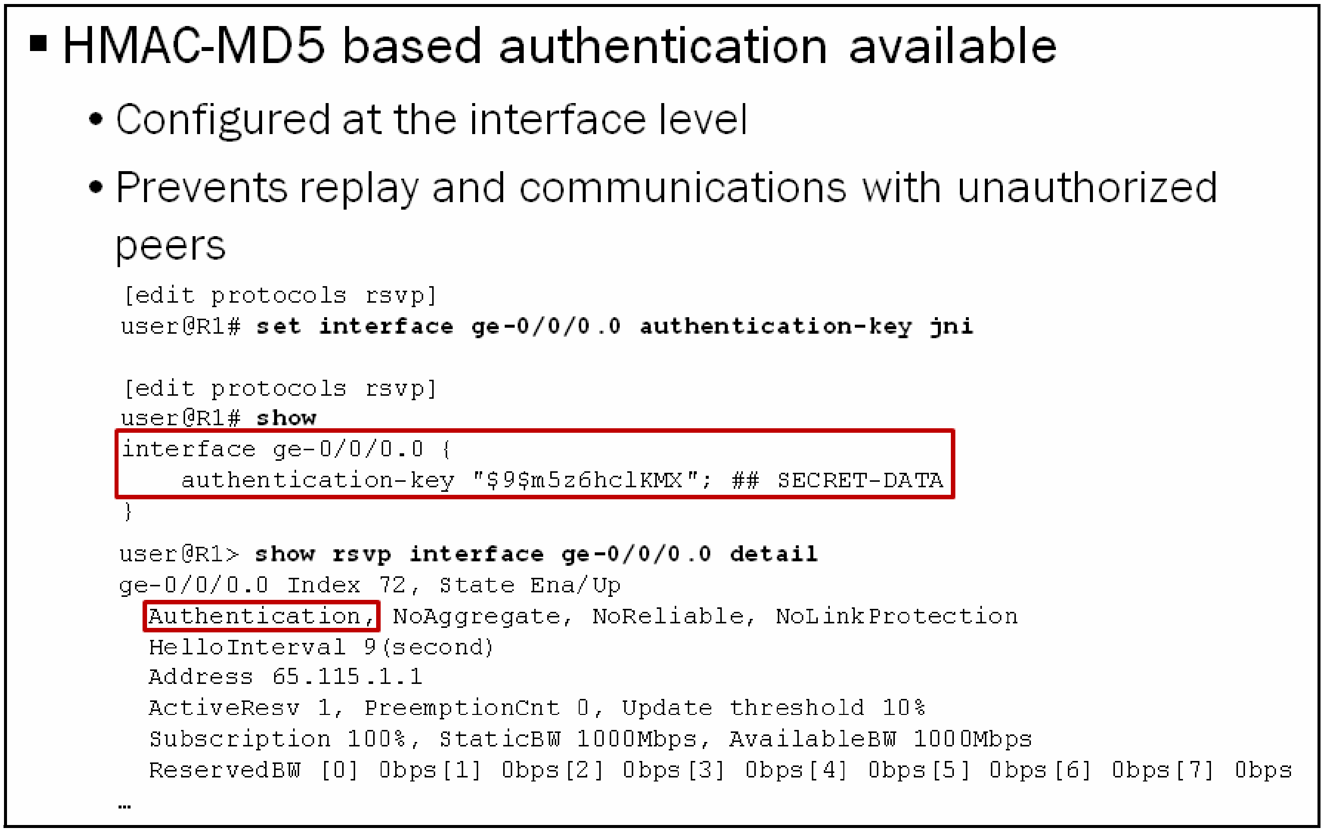
You can confirm proper operation with the output of a **show rsvp session** command when the **detail** switch is added as reflected in the graphic.

For proper operation all routers along the LSP path must support MTU signaling. In a network where there are devices that do not support MTU signaling in RSVP, you might have the following behaviors:

- If the egress router does not support MTU signaling in RSVP, the MTU is set to 1,500 bytes by default.
- A Juniper Networks transit router that does not support MTU signaling in RSVP sets an MTU value of 1,500 bytes in the ADSPEC object by default.

Note that for link/node protection, the MTU of the bypass is only signaled at the time the bypass becomes active. Thus, during the time it takes for the new path MTU to be propagated, there might be packet loss due to MTU mismatch. Similarly for fast-reroute, the MTU of the path will be updated only after the detour becomes active; thus, there will be a delay in the update on the head end.

Authenticate RSVP Exchanges



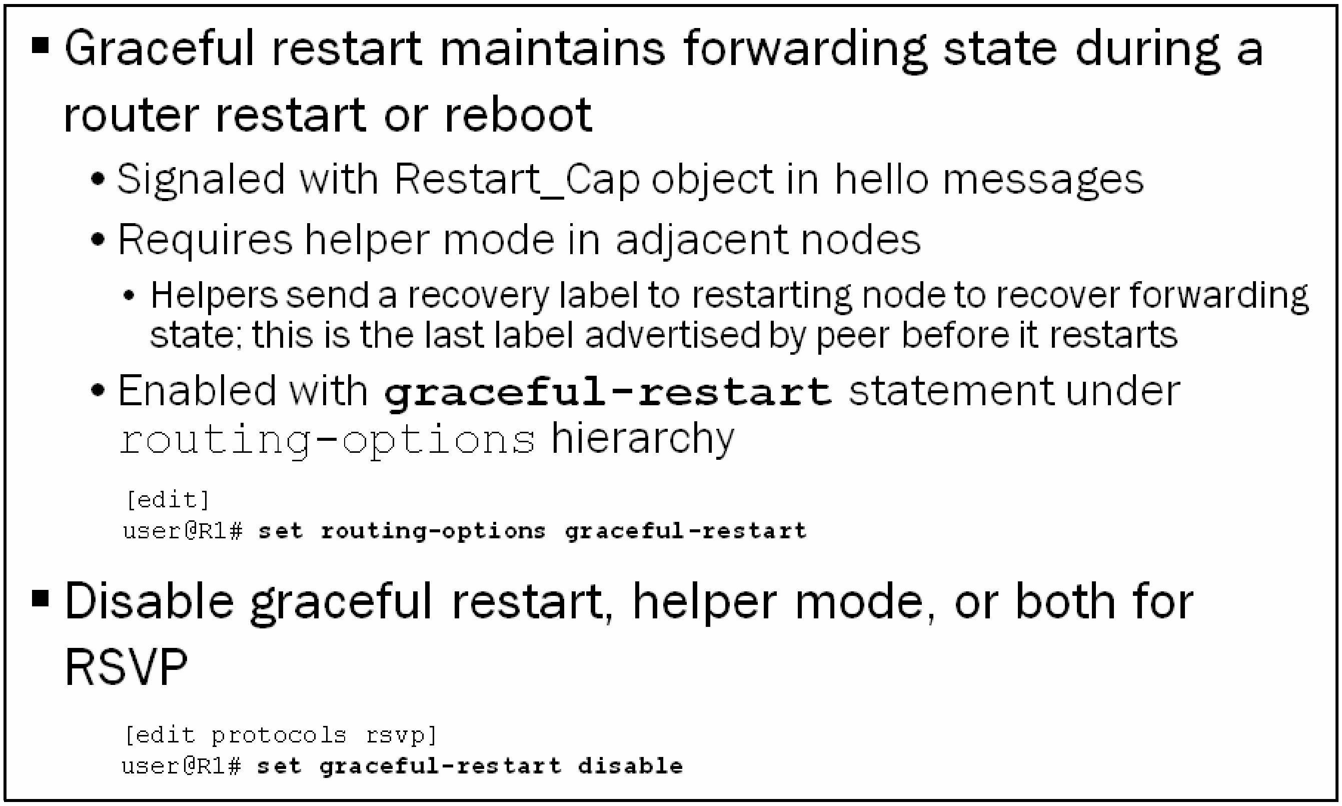
- HMAC-MD5 based authentication available
  - Configured at the interface level
  - Prevents replay and communications with unauthorized peers

```
[edit protocols rsvp]
user@R1# set interface ge-0/0/0.0 authentication-key jni

[edit protocols rsvp]
user@R1# show
interface ge-0/0/0.0 {
    authentication-key "$9$m5z6hclKMX"; ## SECRET-DATA
}

user@R1> show rsvp interface ge-0/0/0.0 detail
ge-0/0/0.0 Index 72, State Ena/Up
  Authentication, NoAggregate, NoReliable, NoLinkProtection
  HelloInterval 9(second)
  Address 65.115.1.1
  ActiveResv 1, PreemptionCnt 0, Update threshold 10%
  Subscription 100%, StaticBW 1000Mbps, AvailableBW 1000Mbps
  ReservedBW [0] 0bps[1] 0bps[2] 0bps[3] 0bps[4] 0bps[5] 0bps[6] 0bps[7] 0bps
…
```

When desired, you can configure Hashed Message Authentication Code (HMAC)-Message Digest 5 (MD5) authentication for RSVP exchanges based on the procedures defined in RFC 2747. RSVP authentication is configured on a per-interface basis, as shown in the graphic for the router's `ge-0/0/0` interface. Once configured, all RSVP messages are authenticated using a message digest based on a shared secrete key. Sequence numbers are added to all messages to prevent replay attacks.

The graphic shows the command used to configure RSVP authentication and the resulting RSVP configuration stanza. As the graphic also shows, you confirm that authentication is in effect using the **`show rsvp interface`** ***`interface-name`*** **`detail`** command.

RSVP Graceful Restart



- Graceful restart maintains forwarding state during a router restart or reboot
  - Signaled with Restart_Cap object in hello messages
  - Requires helper mode in adjacent nodes
    - Helpers send a recovery label to restarting node to recover forwarding state; this is the last label advertised by peer before it restarts
  - Enabled with **`graceful-restart`** statement under `routing-options` hierarchy

```
[edit]
user@R1# set routing-options graceful-restart
```

- Disable graceful restart, helper mode, or both for RSVP

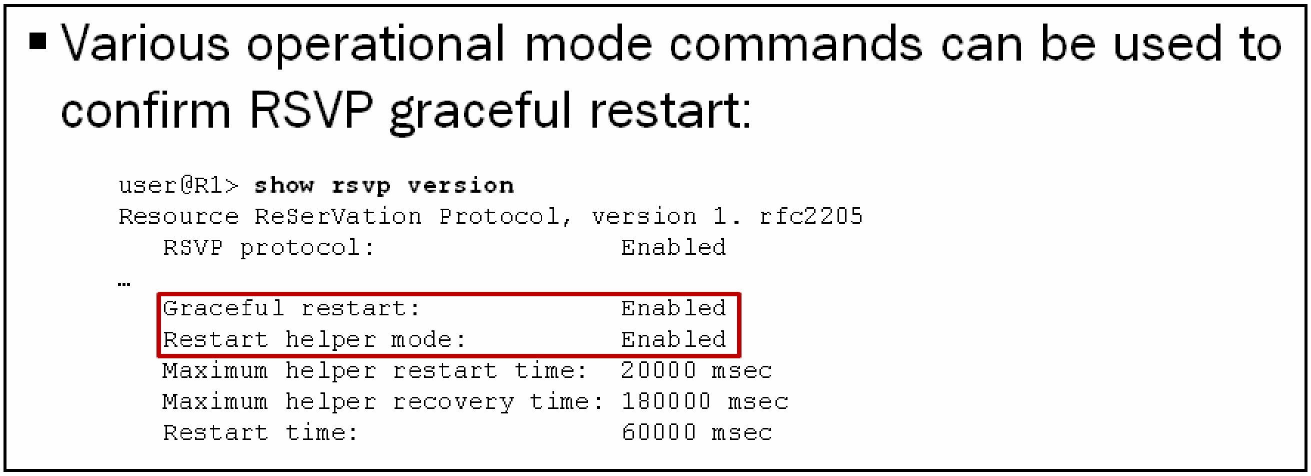
```
[edit protocols rsvp]
user@R1# set graceful-restart disable
```

The Junos OS supports RSVP graceful restart procedures as defined in RFC 3473. To enable graceful restart, add the **`graceful-restart`** statement to the main routing instance at the `[edit routing-options]` hierarchy. This global configuration statement enables graceful restart on all protocols that support the capability, for example LDP and OSPF. To specifically disable RSVP graceful restart, helper mode, or both, add explicit configuration to the RSVP stanza as shown on the graphic.

In operation, a router makes its RSVP graceful restart capabilities known through the inclusion of a Restart_Cap object in its hello messages. By default, both graceful restart and helper mode are enabled for RSVP when the **`graceful-restart`** statement is added to the main routing instance. After a restart, the local router signals that it was able to preserve its forwarding state by sending a Restart_Cap object with a `recovery-time` value that is not zero. Neighbors with helper mode enabled respond to this message by sending back the labels that were last advertised by the restarting router. The result is that the restarting router's signaling plane can be *bootstrapped* back into its pre-restart state, while forwarding continues unabated.

Note that you cannot modify the timers associated with RSVP graceful restart at this time. Also note that RSVP helper mode is enabled by default, even when the **`graceful-restart`** option is not specified in the main routing instance. Therefore, a Junos OS RSVP implementation will always try to help another RSVP peer restart, unless you explicitly disable helper mode.

Verify RSVP Graceful Restart

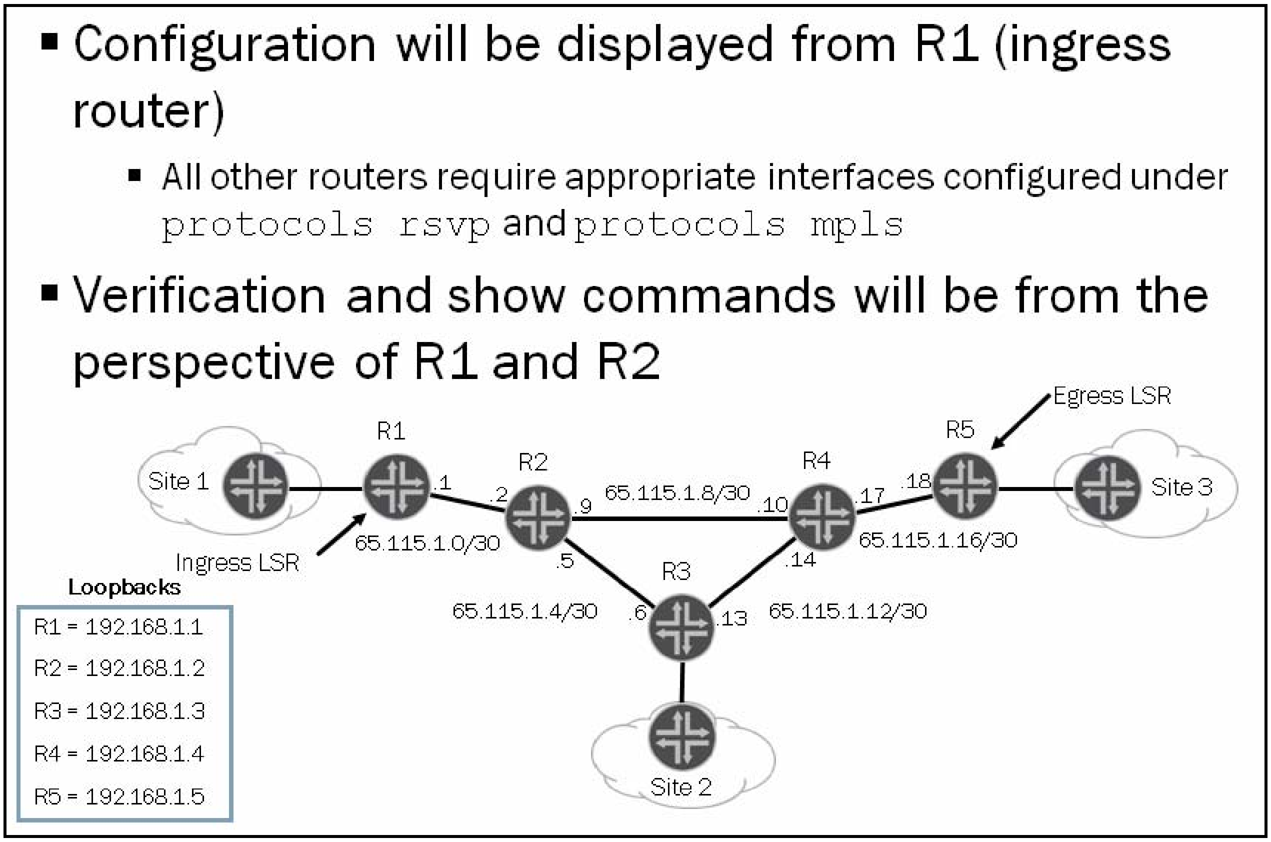


- Various operational mode commands can be used to confirm RSVP graceful restart:

```
user@R1> show rsvp version
Resource ReSerVation Protocol, version 1. rfc2205
   RSVP protocol:                Enabled
…
   Graceful restart:             Enabled
   Restart helper mode:          Enabled
   Maximum helper restart time:  20000 msec
   Maximum helper recovery time: 180000 msec
   Restart time:                 60000 msec
```

Use the **`show rsvp version`** command to determine the global RSVP settings for graceful restart and helper mode. Remember that helper mode is always enabled unless it is explicitly disabled.

Configuration Example

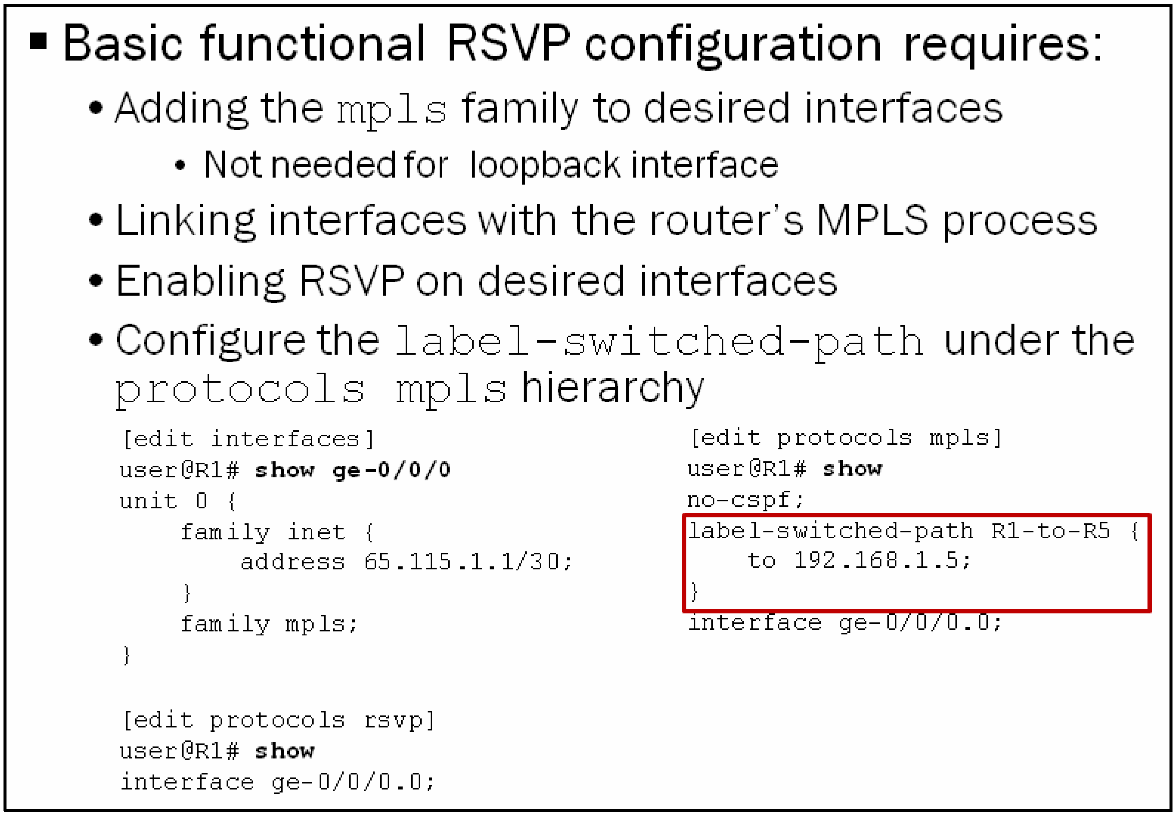


- Configuration will be displayed from R1 (ingress router)
  - All other routers require appropriate interfaces configured under `protocols rsvp` and `protocols mpls`
- Verification and show commands will be from the perspective of R1 and R2

As noted in the graphic, all configuration examples will be from the perspective of the ingress router (R1). We will use the topology and IP addressing illustrated on this graphic to demonstrate the configuration required to create an LSP that egresses at R5.

All verification and show commands will be displayed from the ingress router (R1) or one of the transit routers (R2).

Basic RSVP Configuration



- Basic functional RSVP configuration requires:
  - Adding the `mpls` family to desired interfaces
    - Not needed for loopback interface
  - Linking interfaces with the router's MPLS process
  - Enabling RSVP on desired interfaces
  - Configure the `label-switched-path` under the `protocols mpls` hierarchy

```
[edit interfaces]
user@R1# show ge-0/0/0
unit 0 {
    family inet {
        address 65.115.1.1/30;
    }
    family mpls;
}

[edit protocols rsvp]
user@R1# show
interface ge-0/0/0.0;
```

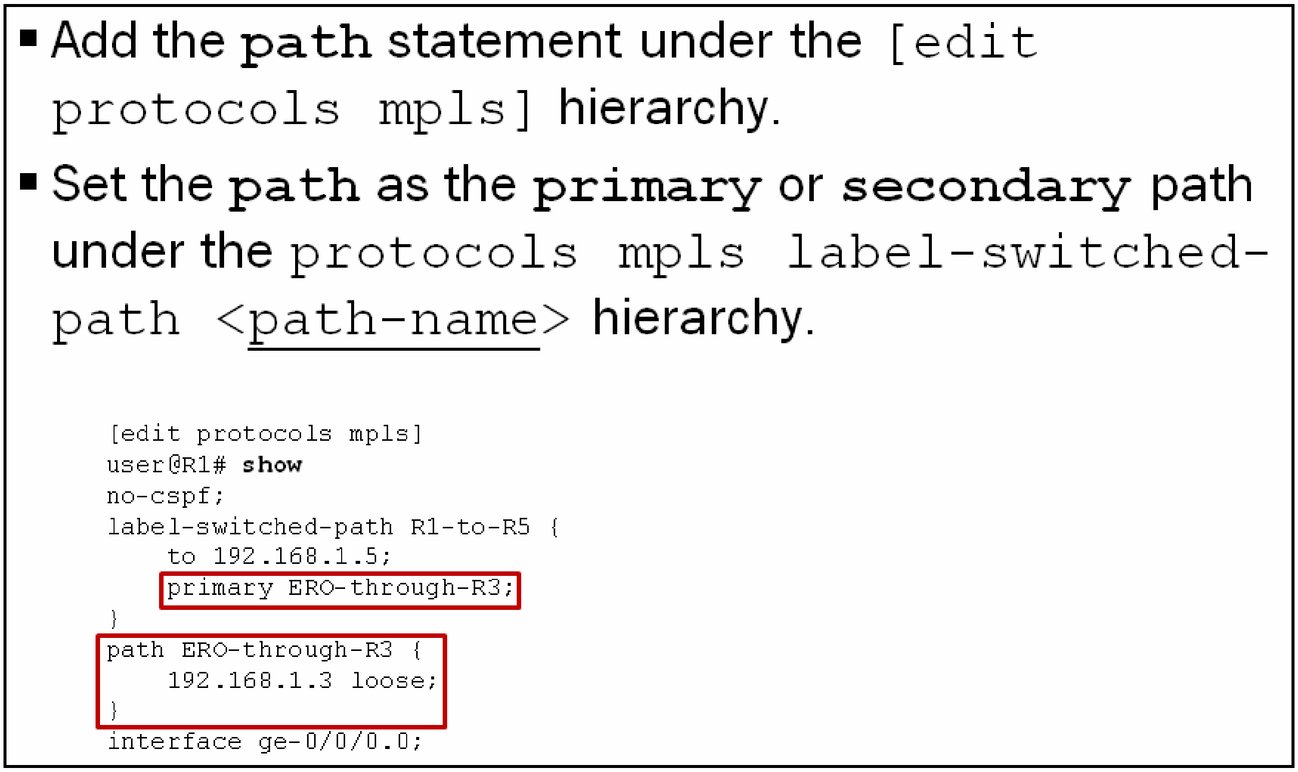
```
[edit protocols mpls]
user@R1# show
no-cspf;
label-switched-path R1-to-R5 {
    to 192.168.1.5;
}
interface ge-0/0/0.0;
```

You must add `family mpls` to all appropriate interfaces on each router throughout the network. You must also add these interfaces to both protocols MPLS and RSVP for the LSP to establish correctly.

The label-switched-path is configured under the MPLS protocol hierarchy. For a basic RSVP LSP to signal through the network you must define the egress address under the `label-switched-path` ***path-name*** hierarchy by adding the **`to`** ***ip-address*** statement. In the example on the graphic, the LSP is named `R1-to-R5` and the egress address of the LSP is the loopback address on the egress router. Any traffic from R1 with a BGP protocol next-hop of this loopback address will traverse the network through this LSP.

As you might have noticed in the example configuration, the statement **`no-cspf`** was also used. CSPF has been turned off because we have not discussed this topic and is not required to signal the LSP. CSPF will be covered in detail in the next chapter.

Configuring an Explicit Route Object



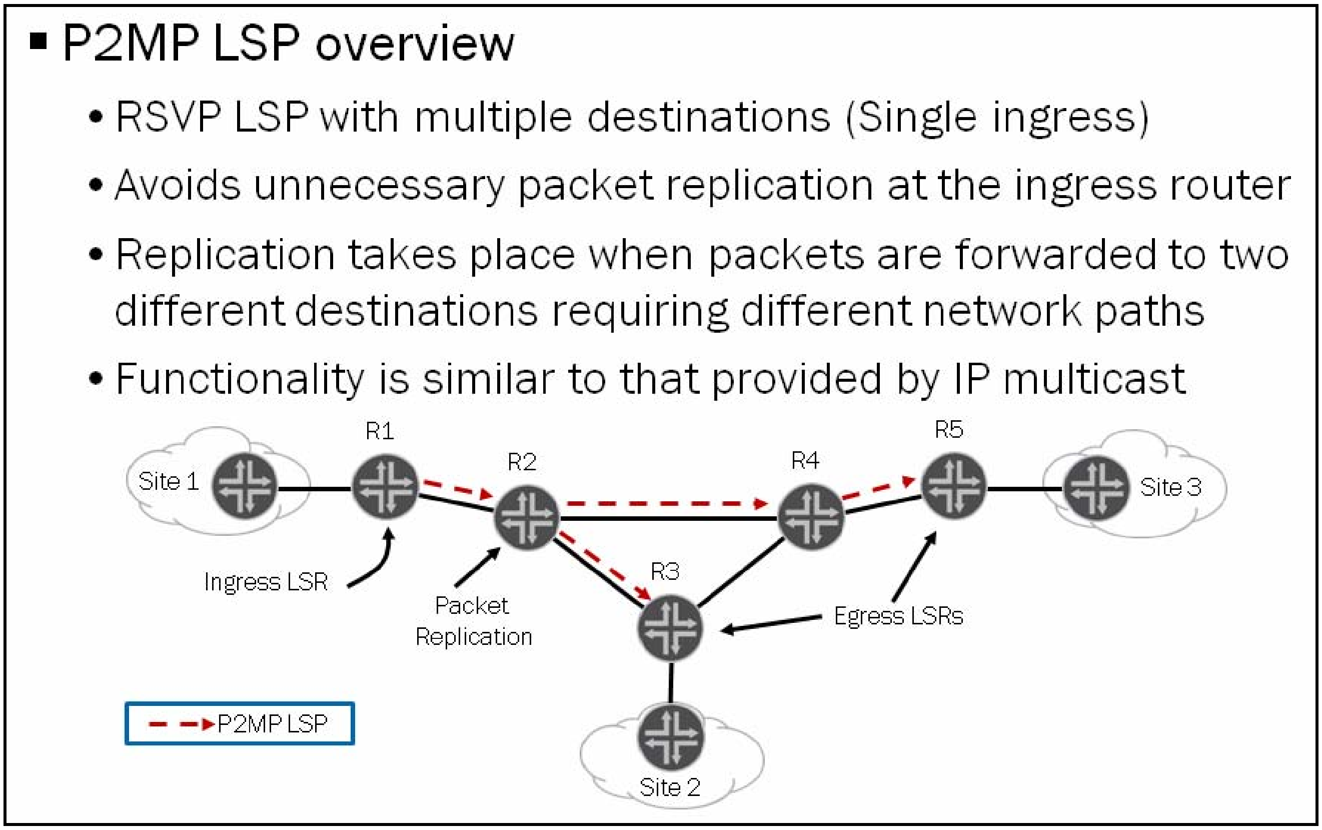
- Add the **`path`** statement under the `[edit protocols mpls]` hierarchy.
- Set the **`path`** as the **`primary`** or **`secondary`** path under the `protocols mpls label-switched-path <path-name>` hierarchy.

```
[edit protocols mpls]
user@R1# show
no-cspf;
label-switched-path R1-to-R5 {
    to 192.168.1.5;
    primary ERO-through-R3;
}
path ERO-through-R3 {
    192.168.1.3 loose;
}
interface ge-0/0/0.0;
```

To configure an explicit path trough the network you begin by configuring the **`path`** ***`path-name`*** statement under `protocols mpls`. The path statement is where you indicate what routers you want the LSP to traverse. You can either specify **`strict`** or **`loose`** hops. After defining the path to be used you must add this path to the `label-switched-path` as either a **`primary`** or **`secondary`** path.

In the example illustrated in the graphic, the `path` is named `ERO-through-R3` and has defined that the LSP must be signaled through `192.168.1.3`, which is the loopback address of R3. The path is also applied as the `primary` path under the `R1-to-R5` LSP.

Point-to-Multipoint LSPs

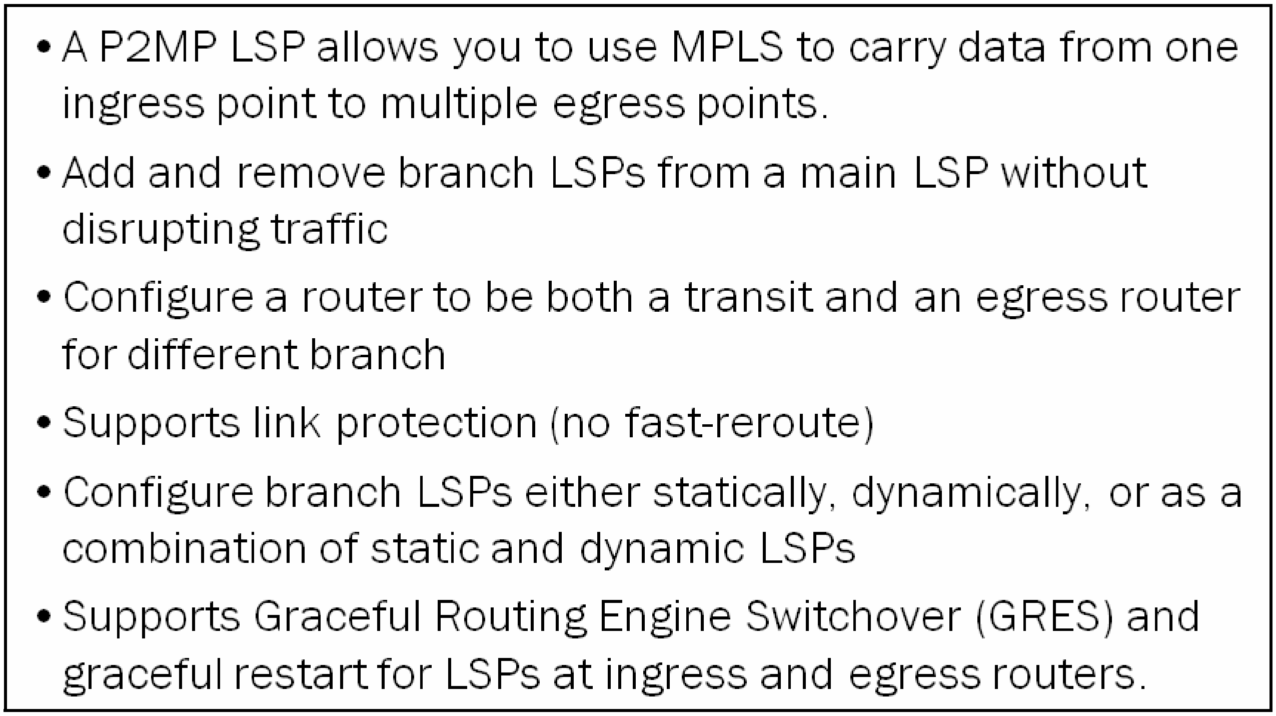


- P2MP LSP overview
  - RSVP LSP with multiple destinations (Single ingress)
  - Avoids unnecessary packet replication at the ingress router
  - Replication takes place when packets are forwarded to two different destinations requiring different network paths
  - Functionality is similar to that provided by IP multicast

A point-to-multipoint MPLS LSP is an RSVP LSP with multiple destinations. By taking advantage of the MPLS packet replication capability of the network, point-to-multipoint LSPs avoid unnecessary packet replication at the ingress router.

Let’s walk through the packet processing detailed in the graphic. Router R1 is configured with a point-to-multipoint LSP to routers R3 and R5. When router R1 sends a packet through the point-to-multipoint LSP, router R2 replicates the packet and forwards it on to routers R3 and R5.

Point-to-Multipoint Details

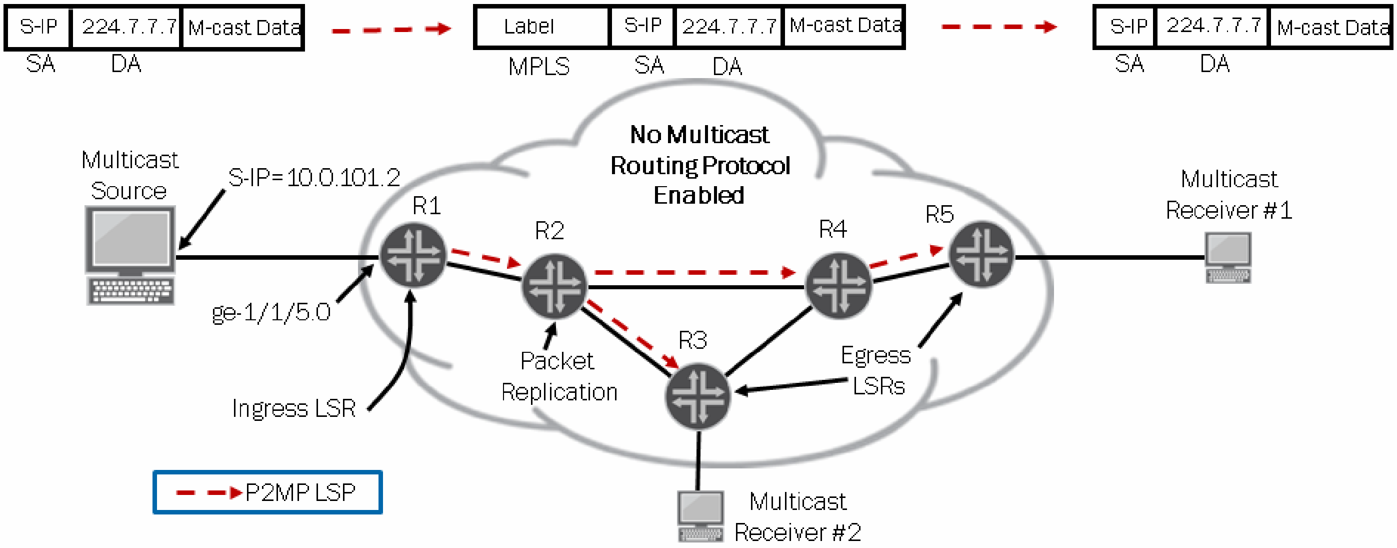


- A P2MP LSP allows you to use MPLS to carry data from one ingress point to multiple egress points.
- Add and remove branch LSPs from a main LSP without disrupting traffic
- Configure a router to be both a transit and an egress router for different branch
- Supports link protection (no fast-reroute)
- Configure branch LSPs either statically, dynamically, or as a combination of static and dynamic LSPs
- Supports Graceful Routing Engine Switchover (GRES) and graceful restart for LSPs at ingress and egress routers.

A point-to-multipoint LSP allows you to use MPLS for point-to-multipoint data distribution. This functionality is similar to that provided by IP multicast. A branch can be added or removed from the main point-to-multipoint LSP without disrupting traffic. The unaffected parts of the LSP continue to function normally. A router can be configured as both a transit and an egress router for different branch LSPs of the same point-to-multipoint LSP. Link protection can also be used with point-to-multipoint LSPs. Link protection can provide a bypass LSP for each of the branch LSPs that make up the LSP. If any of the primary paths fail, traffic can be quickly switched to the bypass. Point-to-multipoint LSPs can be configured either statically, dynamically, or as a combination of static and dynamic LSPs. You can enable graceful Routing Engine switchover (GRES) and graceful restart for point-to-multipoint LSPs at ingress and egress routers.

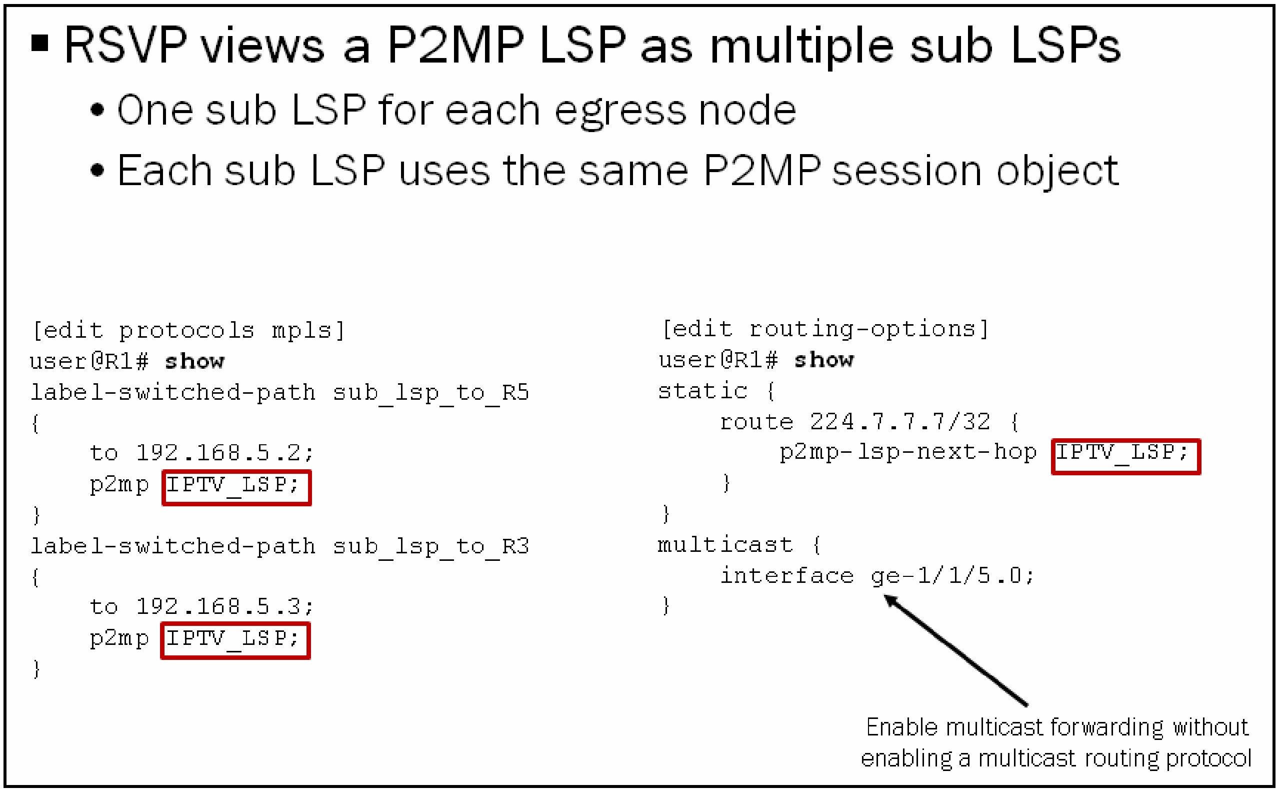
Multicast Example

- Replace PIM in provider core with P2MP LSP
  - Core routers can forward multicast data (encapsulated in MPLS) without the need for a multicast routing protocol



One of the obvious benefits of using point-to-multipoint LSPs is that its forwarding properties are very multicast-like. The Junos OS allows for the configuration of point-to-multipoint LSPs as a replacement for multicast routing protocols in the core of a network. In the example on this graphic and the next, P1 is configured for a point-to-multipoint that terminates on both R3 and R5. A multicast source is attached to R1 and a receiver is attached to both R3 and R5. As multicast data arrives from the source to R1, R1 encasulates the multicast traffic into an MPLS header and sends the MPLS packet into the core. R2 will then receive that traffic, replicate the traffic, swap the labels, and send the traffic out of its two outgoing interfaces. R3 and R5 will eventually receive the multicast traffic even without a multicast routing protocol running in the core.

Point-to-Multipoint LSP Configuration



- RSVP views a P2MP LSP as multiple sub LSPs
  - One sub LSP for each egress node
  - Each sub LSP uses the same P2MP session object

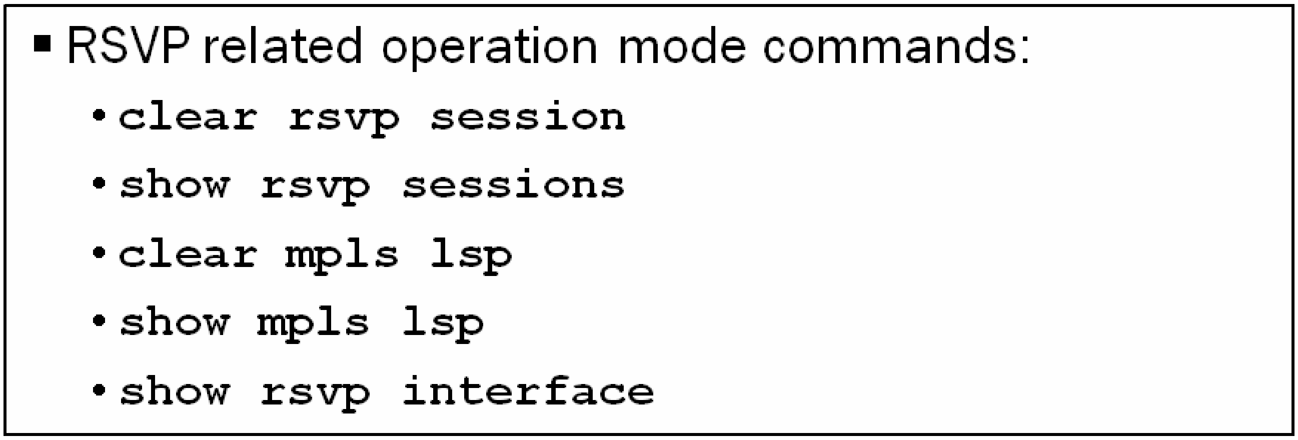
```
[edit protocols mpls]
user@R1# show
label-switched-path sub_lsp_to_R5
{
    to 192.168.5.2;
    p2mp IPTV_LSP;
}
label-switched-path sub_lsp_to_R3
{
    to 192.168.5.3;
    p2mp IPTV_LSP;
}
```

```
[edit routing-options]
user@R1# show
static {
    route 224.7.7.7/32 {
        p2mp-lsp-next-hop IPTV_LSP;
    }
}
multicast {
    interface ge-1/1/5.0;
}
```

Enable multicast forwarding without enabling a multicast routing protocol

The graphic shows the steps needed to build a point-to-multipoint LSP that ingresses at R1 and terminates at R3 and R5. To enable the forwarding of multicast traffic on R1 without enabling a multicast routing protocol, the source facing interface must be configured for multicast as shown in the graphic. Also, a multicast static route must be configured with the point-to-multipoint LSP listed as the next hop. This will allow R1 to know where to send the multicast traffic.

RSVP Operational Mode Commands



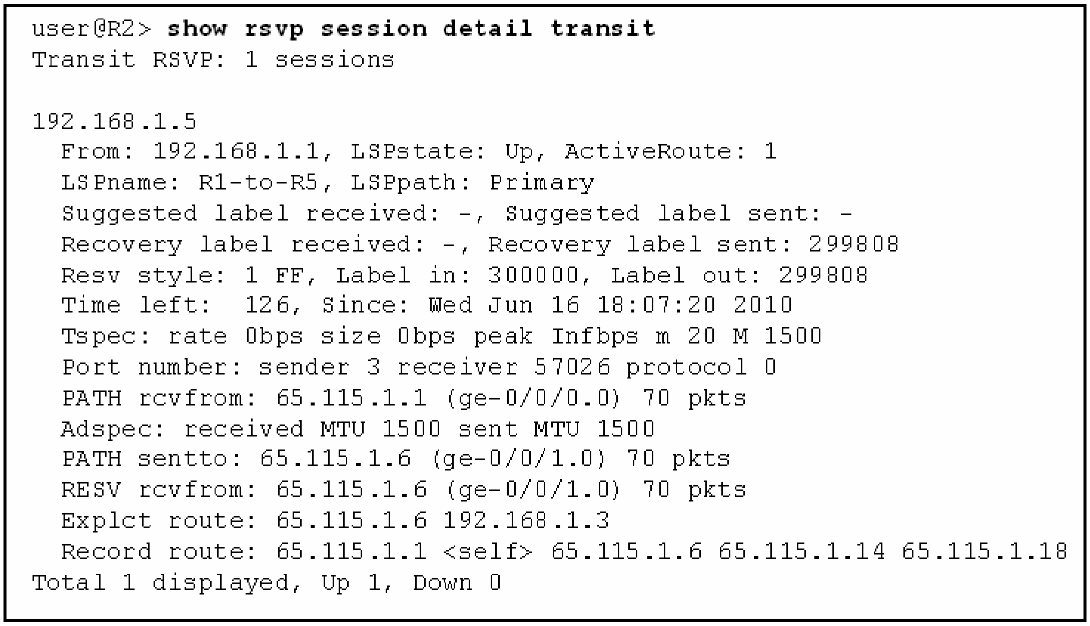
- RSVP related operation mode commands:
  - **clear rsvp session**
  - **show rsvp sessions**
  - **clear mpls lsp**
  - **show mpls lsp**
  - **show rsvp interface**

This graphic reviews some of the more common operational mode commands used to monitor the status and operation of the RSVP protocol and RSVP-signaled LSPs. Each command is briefly described here:

- **clear rsvp session**: This command is used to clear the named RSVP session, or all RSVP sessions (ingress, transit, and egress) when no session name is specified.
- **show rsvp session**: Displays current RSVP session status. Use the ingress, egress, or transit switch to limit command output to the type of session that is of interest.
- **clear mpls lsp**: Used to clear the named LSP session. You can only clear ingress LSPs with this command.

- **show mpls lsp**: Many operators prefer the **show mpls lsp extensive** command when troubleshooting LSP establishment problems because the command's output provides additional error information when compared to the output of the **show rsvp session detail** command.
- **show rsvp interface**: Use this command to display RSVP-enabled interfaces, along with their operational status and reservation state. Any link coloring (administrative groups) associated with RSVP interfaces is also displayed.

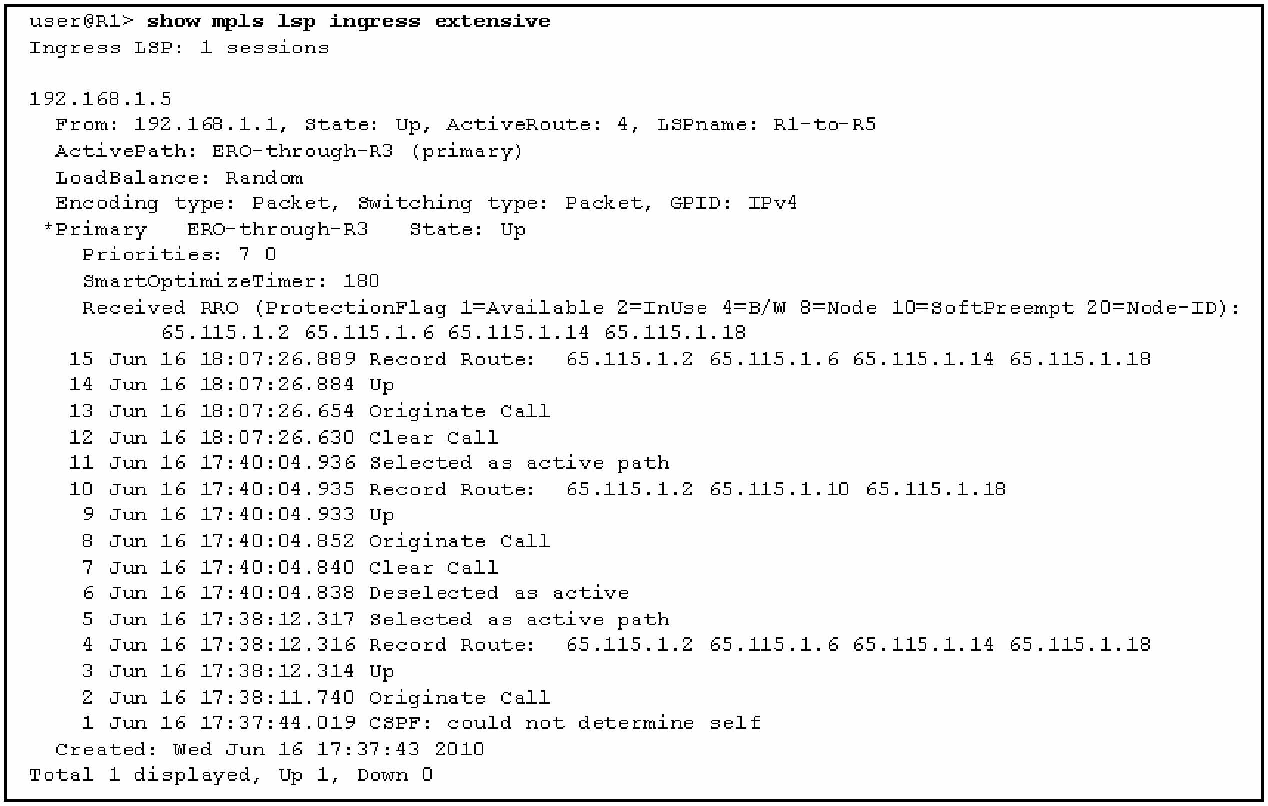
RSVP Session Status Example



```
user@R2> show rsvp session detail transit
Transit RSVP: 1 sessions

192.168.1.5
  From: 192.168.1.1, LSPstate: Up, ActiveRoute: 1
  LSPname: R1-to-R5, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  Recovery label received: -, Recovery label sent: 299808
  Resv style: 1 FF, Label in: 300000, Label out: 299808
  Time left:  126, Since: Wed Jun 16 18:07:20 2010
  Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
  Port number: sender 3 receiver 57026 protocol 0
  PATH rcvfrom: 65.115.1.1 (ge-0/0/0.0) 70 pkts
  Adspec: received MTU 1500 sent MTU 1500
  PATH sentto: 65.115.1.6 (ge-0/0/1.0) 70 pkts
  RESV rcvfrom: 65.115.1.6 (ge-0/0/1.0) 70 pkts
  Explct route: 65.115.1.6 192.168.1.3
  Record route: 65.115.1.1 <self> 65.115.1.6 65.115.1.14 65.115.1.18
Total 1 displayed, Up 1, Down 0
```

This graphic provides an example of the output associated with a **show rsvp session** command using the **detail** and **transit** switches.

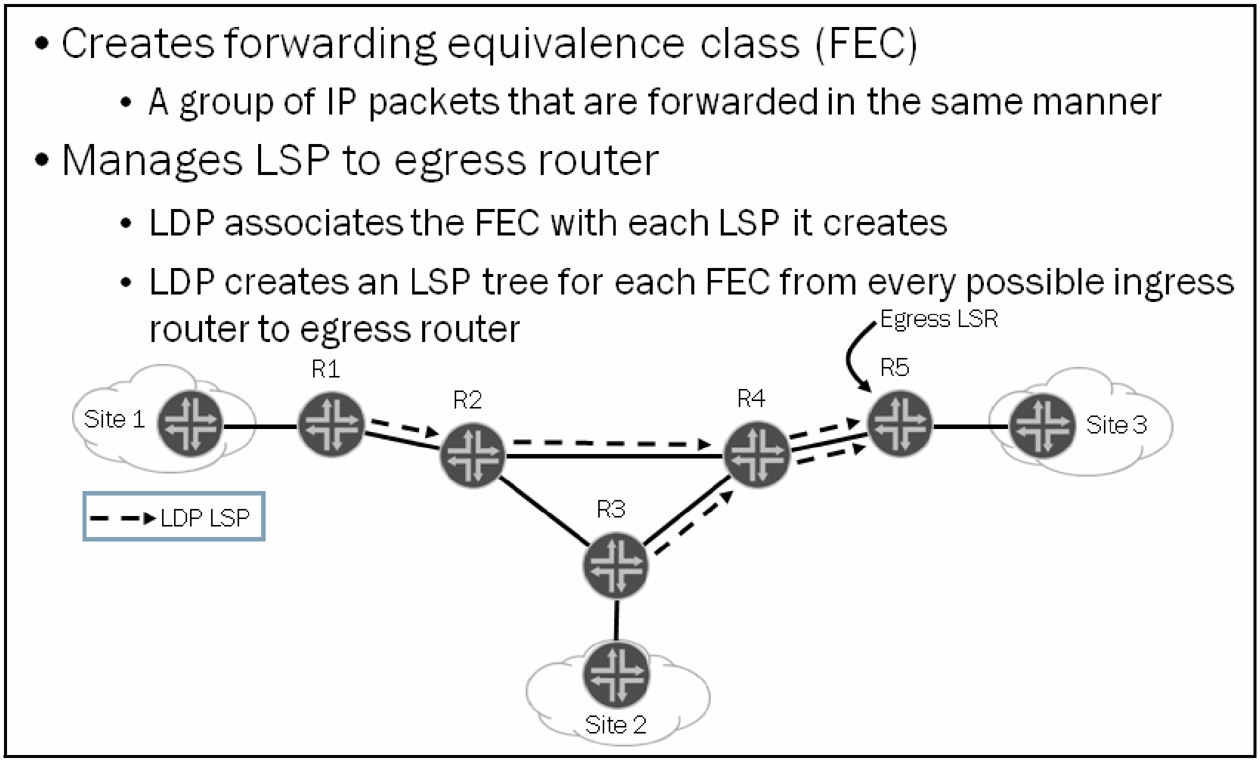
MPLS LSP Status Example



```
user@R1> show mpls lsp ingress extensive
Ingress LSP: 1 sessions

192.168.1.5
  From: 192.168.1.1, State: Up, ActiveRoute: 4, LSPname: R1-to-R5
  ActivePath: ERO-through-R3 (primary)
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary   ERO-through-R3   State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt 20=Node-ID):
          65.115.1.2 65.115.1.6 65.115.1.14 65.115.1.18
   15 Jun 16 18:07:26.889 Record Route:  65.115.1.2 65.115.1.6 65.115.1.14 65.115.1.18
   14 Jun 16 18:07:26.884 Up
   13 Jun 16 18:07:26.654 Originate Call
   12 Jun 16 18:07:26.630 Clear Call
   11 Jun 16 17:40:04.936 Selected as active path
   10 Jun 16 17:40:04.935 Record Route:  65.115.1.2 65.115.1.10 65.115.1.18
    9 Jun 16 17:40:04.933 Up
    8 Jun 16 17:40:04.852 Originate Call
    7 Jun 16 17:40:04.840 Clear Call
    6 Jun 16 17:40:04.838 Deselected as active
    5 Jun 16 17:38:12.317 Selected as active path
    4 Jun 16 17:38:12.316 Record Route:  65.115.1.2 65.115.1.6 65.115.1.14 65.115.1.18
    3 Jun 16 17:38:12.314 Up
    2 Jun 16 17:38:11.740 Originate Call
    1 Jun 16 17:37:44.019 CSPF: could not determine self
  Created: Wed Jun 16 17:37:43 2010
Total 1 displayed, Up 1, Down 0
```

This graphic provides an example of the output associated with a **`show mpls lsp`** command using the **`extensive`** and **`ingress`** switches. Note that the display contains a time-stamped log of significant events in the life of the LSP. This information often proves invaluable when troubleshooting RSVP control plane problems.

Purpose of LDP



- Creates forwarding equivalence class (FEC)
  - A group of IP packets that are forwarded in the same manner
- Manages LSP to egress router
  - LDP associates the FEC with each LSP it creates
  - LDP creates an LSP tree for each FEC from every possible ingress router to egress router

LDP associates a set of destinations (prefixes) with each LSP. This set of destinations is called the FEC. These destinations share a common LSP path and egress router, as well as a common unicast routing path.

LDP maps groups of prefixes to an egress router at the end of an LSP. LDP manages the LSP to the egress router for each FEC. LDP is not related to RSVP or traffic engineering concepts from previous lectures.

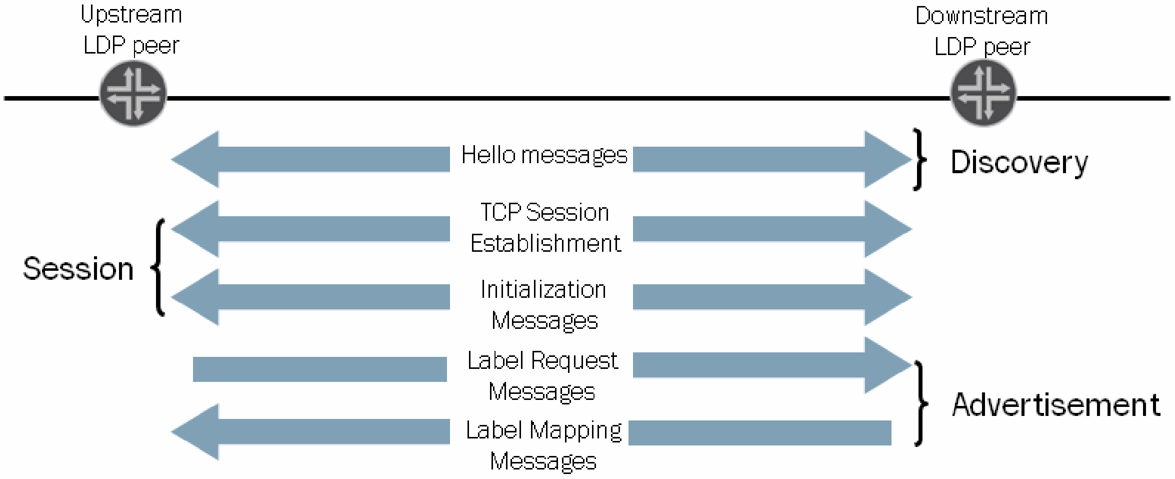
LDP maps the FECs (prefixes) to label values. The LSP forwarding paths look like a unicast forwarding path, in that MPLS traffic for the ultimate destination is forwarded along the unicast forwarding tree.

LDP allows multiple prefixes to share the same label mapping. No constraints are allowed when signaling the LSPs. The LSPs must follow the IGP best path. LDP merges together traffic from different tunnels, which results in fewer total tunnels than would be required with RSVP.

LDP will create a LSP tree for each FEC from every possible ingress point in the LDP network to the egress point. Each LDP speaking router will advertise the addresses reachable via a MPLS label into the LDP domain. The label information is exchanged in a hop by hop fashion so every LSR in the domain will become an ingress router to all other routers in the network. This process creates a full mesh LDP environment. The graphic displays what LSPs will be generated for the FEC egressing at R5.

LDP Message Types

- Discovery: Locate potential LDP peers
- Session: Manage peer-to-peer TCP sessions
- Advertisement: Create, change, or delete label mappings
- Notification: Provide advisory information

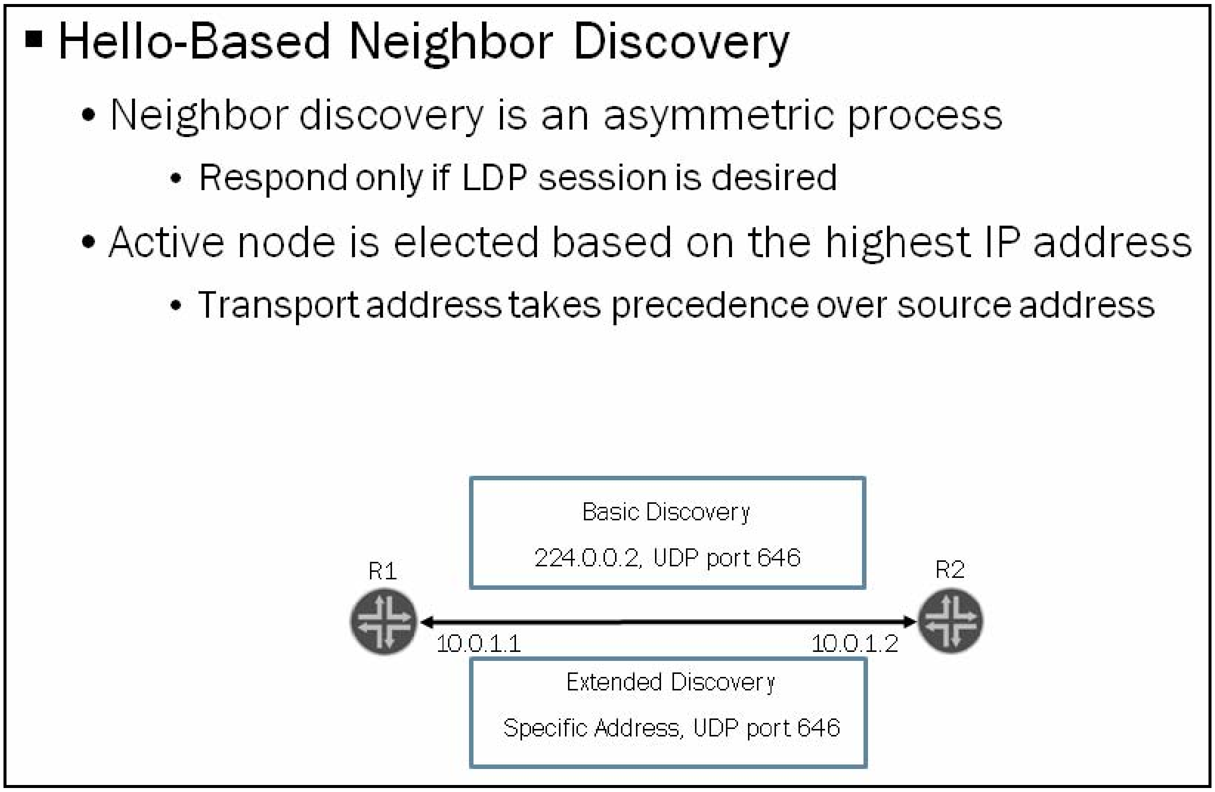


LDP uses several types of messages to establish, remove mappings and to report errors. All LDP messages have a common structure that incorporate a type/length/value (TLV) encoding scheme.

- *Discovery Messages*: Discovery messages announce and maintain the presence of a router in a network. Routers indicate their presence in a network by periodically sending hello messages. This hello message is encapsulated within a User Datagram Protocol (UDP) packet that is sent to the LDP port (port 646) using the multicast all routers group address. The use of the 224.0.0.2 multicast address limits neighbor discovery to directly connected peers by default. Extended LDP neighbor discovery is discussed on a subsequent page.
- *Session Messages*: Session messages establish, maintain, and terminate sessions between LDP peers. When a router establishes a session with another router learned through hello exchanges, it uses the LDP initialization procedure over a Transmission Control Protocol (TCP) transport. Note that the higher IP address is responsible for establishing the TCP session. When the initialization procedure completes successfully the two routers are LDP peers and they can begin the exchange of advertisement messages.
- *Advertisement Messages*: Advertisement messages create, change, and delete label mappings for FECs. Requesting a label or advertising a label mapping to a peer is a decision made by the local router. In general, the router requests a label mapping from a neighboring router when it needs one and it advertises a label mapping to a neighboring router when it wants the neighbor to use a label.

- *Notification Messages*: Notification messages convey advisory and error related information. LDP sends notification messages to report errors and other events of interest. The two kinds of LDP notification messages are the following:
  - *Error notifications* signal fatal errors. If a router receives an error notification from a peer it terminates the LDP session by closing the TCP transport connection and discarding all label mappings learned that were learned through that session.
  - *Advisory notifications* pass information to the router about the LDP session.

Neighbor Discovery



- Hello-Based Neighbor Discovery
  - Neighbor discovery is an asymmetric process
    - Respond only if LDP session is desired
  - Active node is elected based on the highest IP address
    - Transport address takes precedence over source address

Basic Discovery

224.0.0.2, UDP port 646

R1

R2

10.0.1.1

10.0.1.2

Extended Discovery

Specific Address, UDP port 646

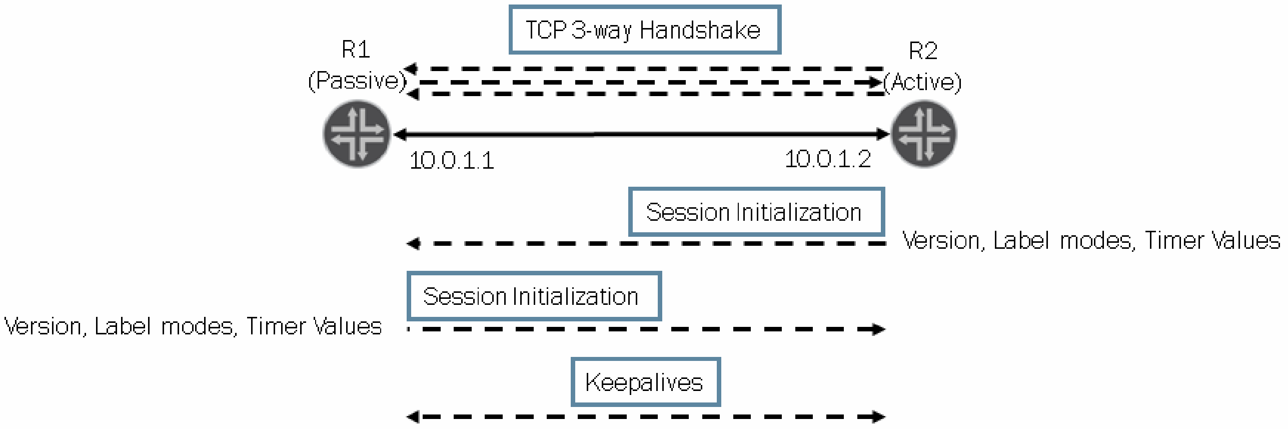
The discovery process can either send a hello message to 224.0.0.2 (*basic discovery*) or to a specific address (*extended discovery*), in both cases using UDP encapsulation and port 646. 224.0.0.2 is the *all routers on this subnet* multicast address. Note that a station's response to a hello message indicates its desire to establish an LDP session with the neighboring router.

Transport Address

The transport address is used to determine which side is *active*. The transport address is placed into the Hello message as a transport address object. If the transport address object is not specified, the source address of the hello packet is used to determine the active router.

Transport Connection Establishment

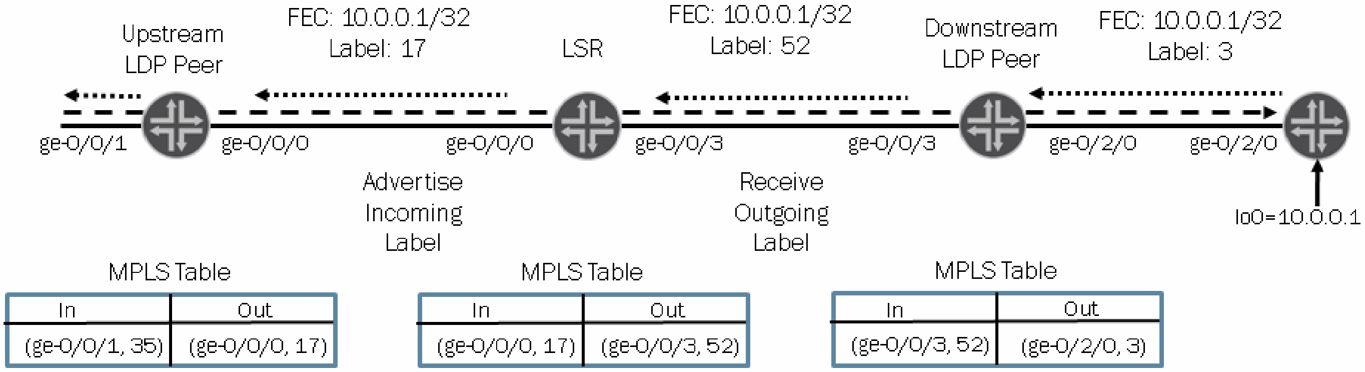
- Active Node initiates TCP session
  - LDP Session initiated after TCP session established



The router with the numerically highest IP address is responsible for initiating the TCP session. After successful TCP connection establishment, the LDP session can be established.

LDP Label Mapping

- Downstream peer assigns labels
- Benefits:
  - Traffic engineering information is not piggybacked on routing protocols
- Limitations:
  - LSPs follow the conventional IGP path
  - Does not support explicit routing



In	Out
(ge-0/0/1, 35)	(ge-0/0/0, 17)

In	Out
(ge-0/0/0, 17)	(ge-0/0/3, 52)

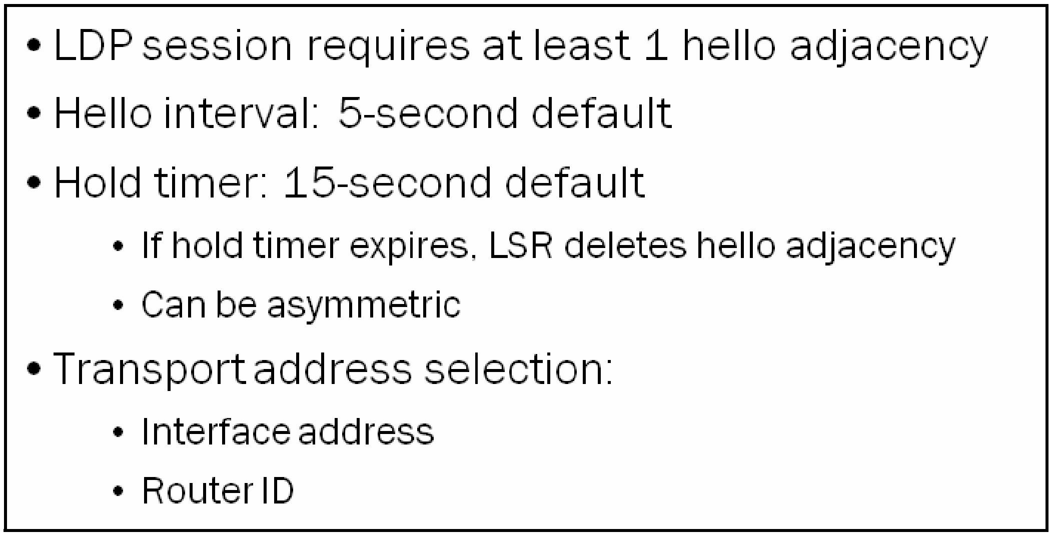
In	Out
(ge-0/0/3, 52)	(ge-0/2/0, 3)

Label Request and Label Map messages are used to associate FECs with labels. In this example on the graphic, the router on the right has knowledge of network 10.0.0.1/32, and it is running LDP with its upstream neighbors. The router in the middle receives a FEC mapping of 10.0.0.1/32 to Label 52 over its ge-0/0/3 interface.

The middle router now advertises the FEC for 10.0.0.1/32 *upstream*, which is to the router on the left in this case, with a label mapping of 17. The process continues until there are no more LDP adjacencies to which the FEC can be advertised.

The graphic focuses on the 10.0.0.1/32 FEC. There may be additional FECs within this same network.

Session Maintenance



- LDP session requires at least 1 hello adjacency
- Hello interval: 5-second default
- Hold timer: 15-second default
  - If hold timer expires, LSR deletes hello adjacency
  - Can be asymmetric
- Transport address selection:
  - Interface address
  - Router ID

An LDP peer must receive an LDP packet every keepalive period to prevent the tear down of neighbor state. Any LDP protocol message is acceptable for keepalive purposes, so keepalive messages are sent only in the absence of other LDP traffic. Either end can shut down the session by issuing a *shutdown* message. If a router has multiple links to an LDP peer then hellos are sent across all of the links. As long as one of the links can continue to exchange hellos, the LDP session remains active. See the last section on the next page for more detail on choosing an LDP transport address.

The LDP hello messages enable LDP routers to discover one another and to detect the failure of a neighbor, or the link, to that neighbor. Hello messages are sent periodically on all interfaces on which LDP is enabled. By default, LDP sends hello messages every 5 seconds. This value can be configured depending on the network requirements.

The hold time determines how long a router can wait for a hello message before declaring the neighbor lost. The configured value is sent inside of hello messages to inform the receiving router how often it should expect to receive a hello; this mechanism means that hello intervals do not be the same between neighbors. The default hold time is 15 seconds; this value represents the recommended setting of three times the hello interval.

You can control the transport address used by LDP. The transport address is the IP address used to support the TCP session. You can configure the transport address globally for all LDP sessions or for each interface independently. If you select interface, the interface address is used as the transport address for any LDP sessions to neighbors that are reachable over that interface.

You cannot specify interface when there are multiple parallel links to the same LDP neighbor because the LDP specification requires that the same transport address be advertised on all interfaces to the same neighbor. If LDP detects multiple parallel links to the same neighbor, it disables interfaces to that neighbor one by one until the condition is cleared.

Junos OS LDP Implementation

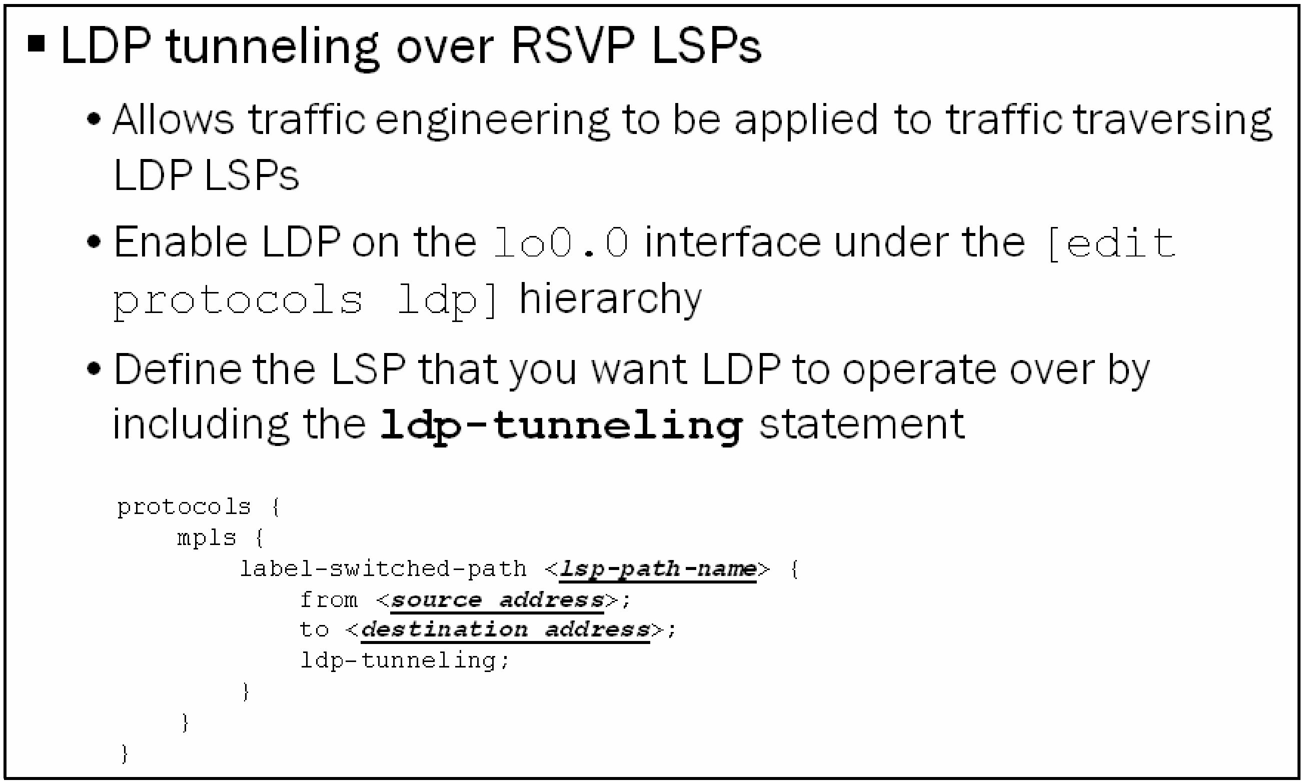
The Junos OS implementation of LDP supports LDP Version 1. Constraint-Based Routed Label Distribution Protocol (CR-LDP) is not supported. The Junos OS implementation of LDP supports the "ordered downstream unsolicited with liberal label retention" mode defined in RFC 3036. This means that each LDP peer will store all label bindings received (liberal retention), that each *downstream* peer will advertise all FECs for which it is prepared to receive labeled traffic (downstream unsolicited), and that FECs are only advertised when the router is the traffic's egress point, or it has received a label mapping for the traffic's next hop (ordered).

With the Junos OS using the minimum LDP configuration, LSRs will form LSPs to the /32 router ID of all LDP capable routers that are reachable.

Basic neighbor discovery forms an LDP session with a directly connected neighbor because the hello messages have a destination address of 224.0.0.2; messages sent to these addresses are not routed.

Extended discovery allows peers to establish LDP sessions through an RSVP-signaled LSP, thus allowing some level of traffic engineering for LDP traffic. You explicitly configure the destination address of the hello messages when using extended discovery; because a routable IP address is specified, the LDP peer can be reached via IP routing and no longer needs to be directly connected. Extended discovery enables LDP to be tunneled over RSVP. This is explained in more detail in the following sections.

LDP Tunneling

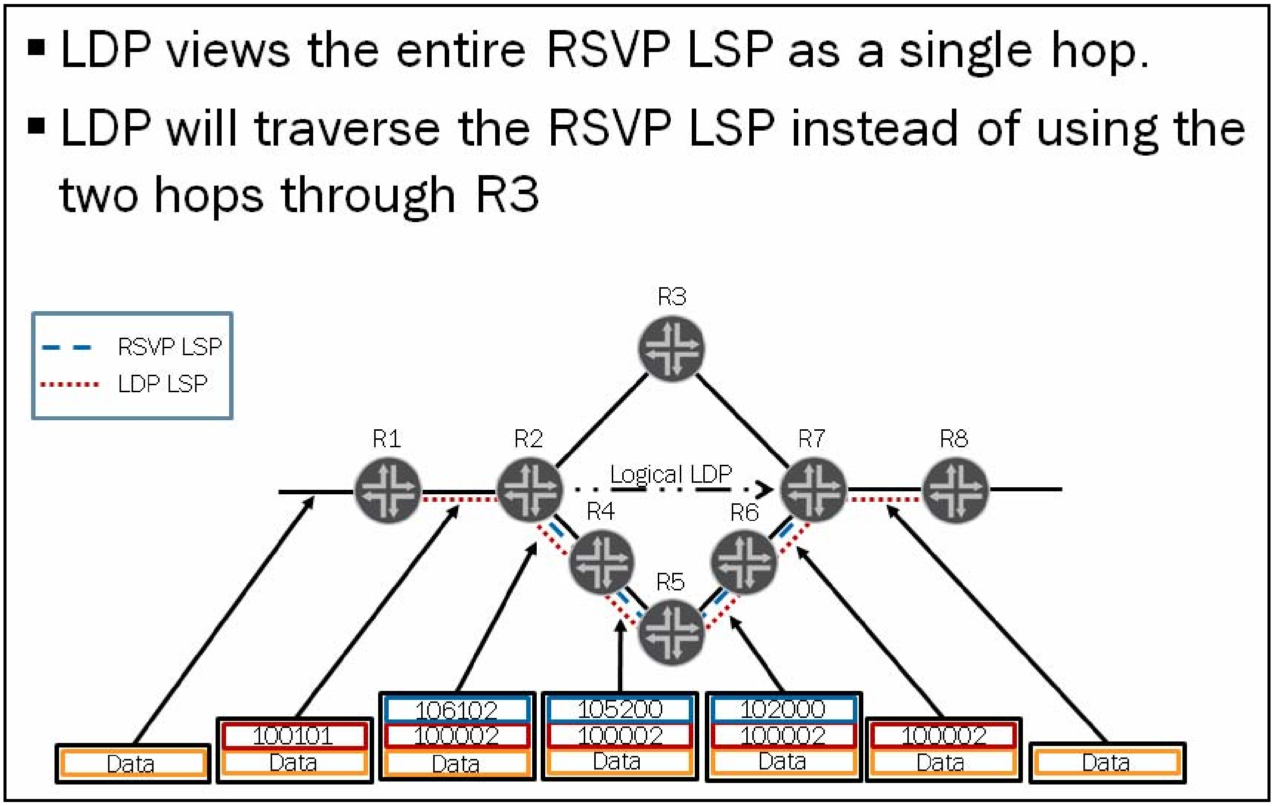


- LDP tunneling over RSVP LSPs
  - Allows traffic engineering to be applied to traffic traversing LDP LSPs
  - Enable LDP on the `lo0.0` interface under the `[edit protocols ldp]` hierarchy
  - Define the LSP that you want LDP to operate over by including the **`ldp-tunneling`** statement

```
protocols {
    mpls {
        label-switched-path <lsp-path-name> {
            from <source address>;
            to <destination address>;
            ldp-tunneling;
        }
    }
}
```

You can tunnel LDP LSPs over RSVP-signaled LSPs using label stacking. Note that you *must* enable LDP on the `lo0.0` interface to support extended neighbor discovery needed for this application. Additionally, you must configure the LSPs over which you want LDP to operate by including the **`ldp-tunneling`** statement as shown.

LDP over RSVP

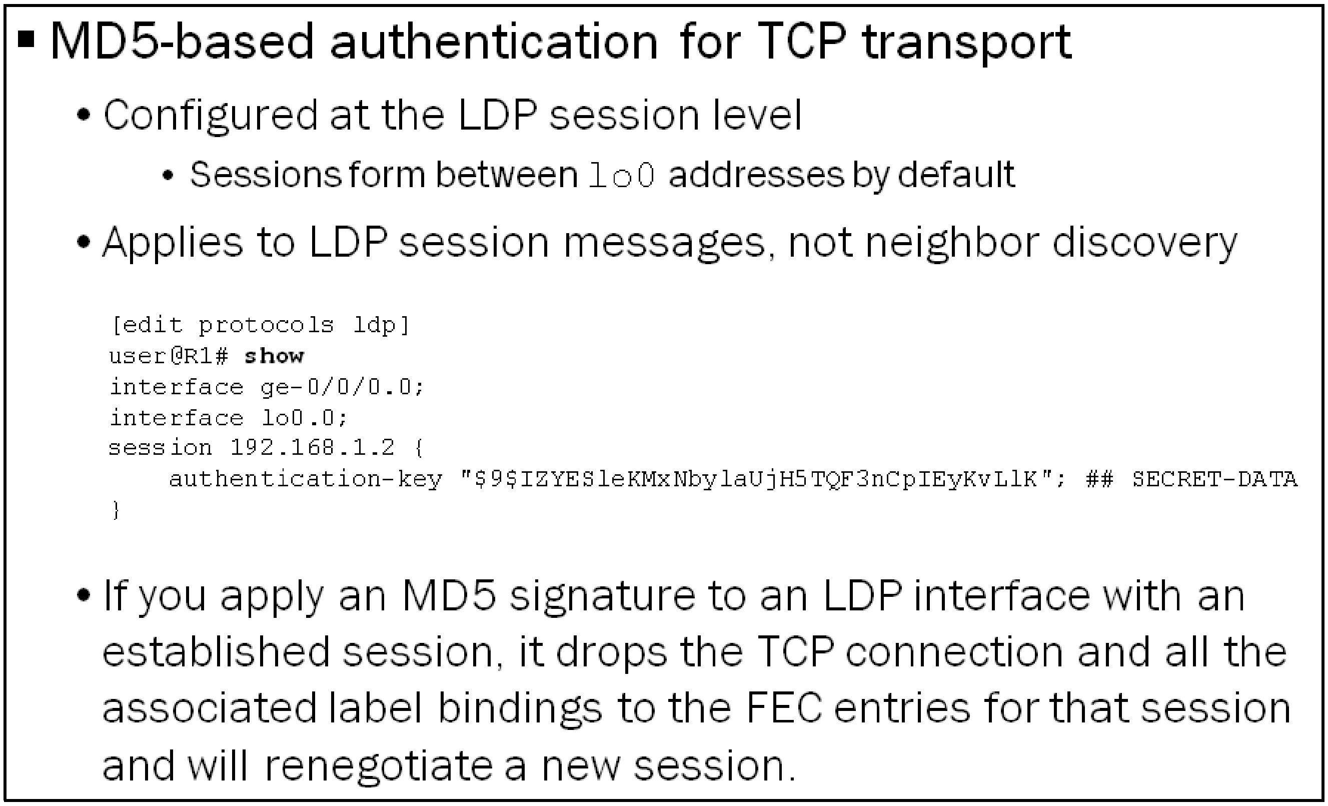


- LDP views the entire RSVP LSP as a single hop.
- LDP will traverse the RSVP LSP instead of using the two hops through R3

This graphic shows that LDP-over-RSVP tunneling results in LDP traffic being forwarded through the RSVP tunnel, which itself takes a traffic engineered path. By default, LDP always follows the IGP's shortest path, which in this case, would be the 3-hop path at the top of the graphic. LDP views the RSVP LSP as a single hop, therefore the RSVP path becomes the more preferred path even though the LSP actually traverses 5 hops.

The label assignment is also shown on this graphic. When the traffic enters into the LDP LSP it pushes a label value of 100101. When received on R2 it accepts the packet based on the assigned incoming label. R2 will lookup the route and identify that the route will be sent over the RSVP LSP. R2 pushes on the LDP label value of 100002 and then stacks an outer label value of 106102 for the RSVP label. When the packet is received at R4 it accepts the packet based on the RSVP label and swaps the outer label with the label assigned to the outgoing interface (105200) and forwards the traffic to the next LSR. R5 received the packet based on the incoming RSVP label. R5 swaps the RSVP label value with the next label (102000) and forwards it on to R6. R6 will also process the packet based on the RSVP label. Since R6 is the penultimate LSP for the RSVP LSP it will pop the RSVP label and forward to the next LSR with the LDP label and R7 will accept and forward the pact based on the LDP label.

Authenticate LDP Sessions



- MD5-based authentication for TCP transport
  - Configured at the LDP session level
    - Sessions form between `lo0` addresses by default
  - Applies to LDP session messages, not neighbor discovery

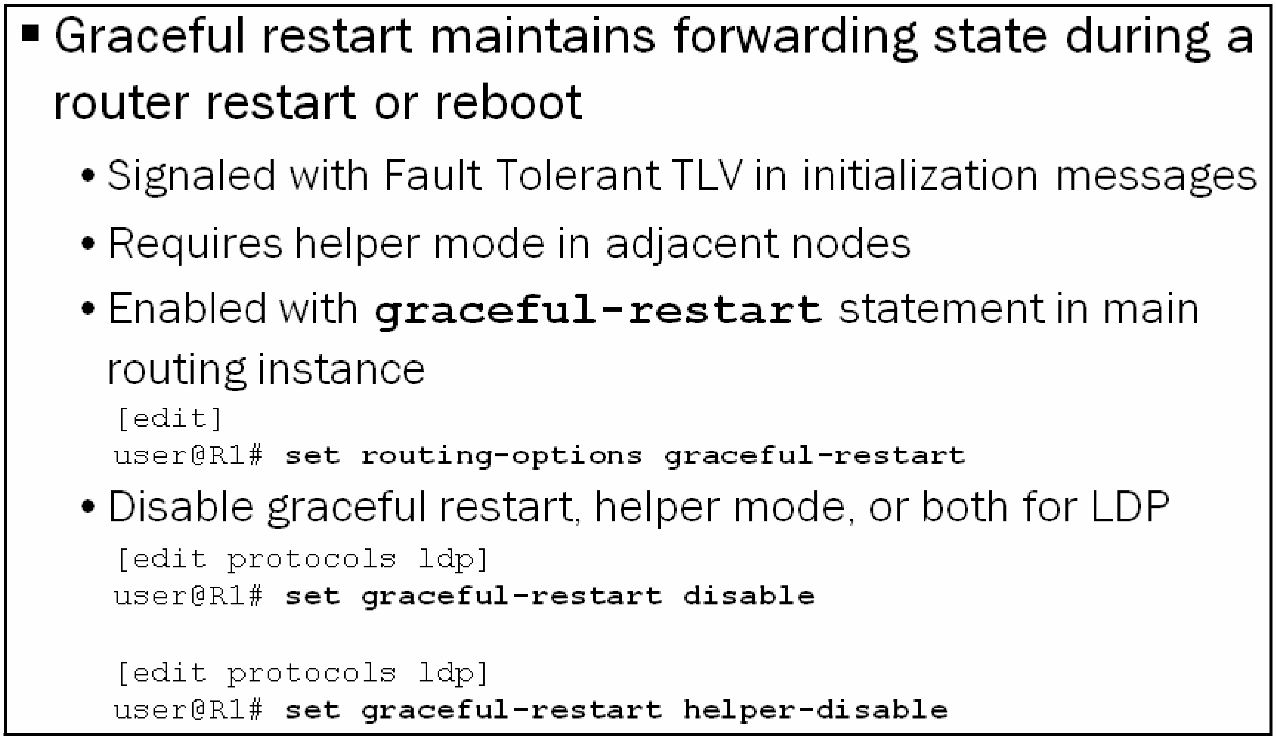
```
[edit protocols ldp]
user@R1# show
interface ge-0/0/0.0;
interface lo0.0;
session 192.168.1.2 {
    authentication-key "$9$IZYESleKMxNbylaUjH5TQF3nCpIEyKvLlK"; ## SECRET-DATA
}
```

  - If you apply an MD5 signature to an LDP interface with an established session, it drops the TCP connection and all the associated label bindings to the FEC entries for that session and will renegotiate a new session.

When you want, you can configure MD5-based authentication for the TCP transport protocol that supports LDP sessions. LDP session authentication is configured on a per-session basis, as shown on the graphic for the R1's LDP session to R2 (192.168.1.2). Note that LDP session authentication does not apply to the UDP-based neighbor discovery mechanism. Thus, mismatched LDP authentication settings permit LDP neighbor discovery and adjacency formation, but the LDP session will not establish without compatible authentication values. Note that specifying interface addresses under the session stanza requires the use of transport-address interface for authentication to be in effect because LDP sessions form between `lo0` addresses by default.

The graphic shows the command used to configure LDP authentication and the resulting LDP configuration stanza.

LDP Graceful Restart



- Graceful restart maintains forwarding state during a router restart or reboot
  - Signaled with Fault Tolerant TLV in initialization messages
  - Requires helper mode in adjacent nodes
  - Enabled with **`graceful-restart`** statement in main routing instance

```
[edit]
user@R1# set routing-options graceful-restart
```

  - Disable graceful restart, helper mode, or both for LDP

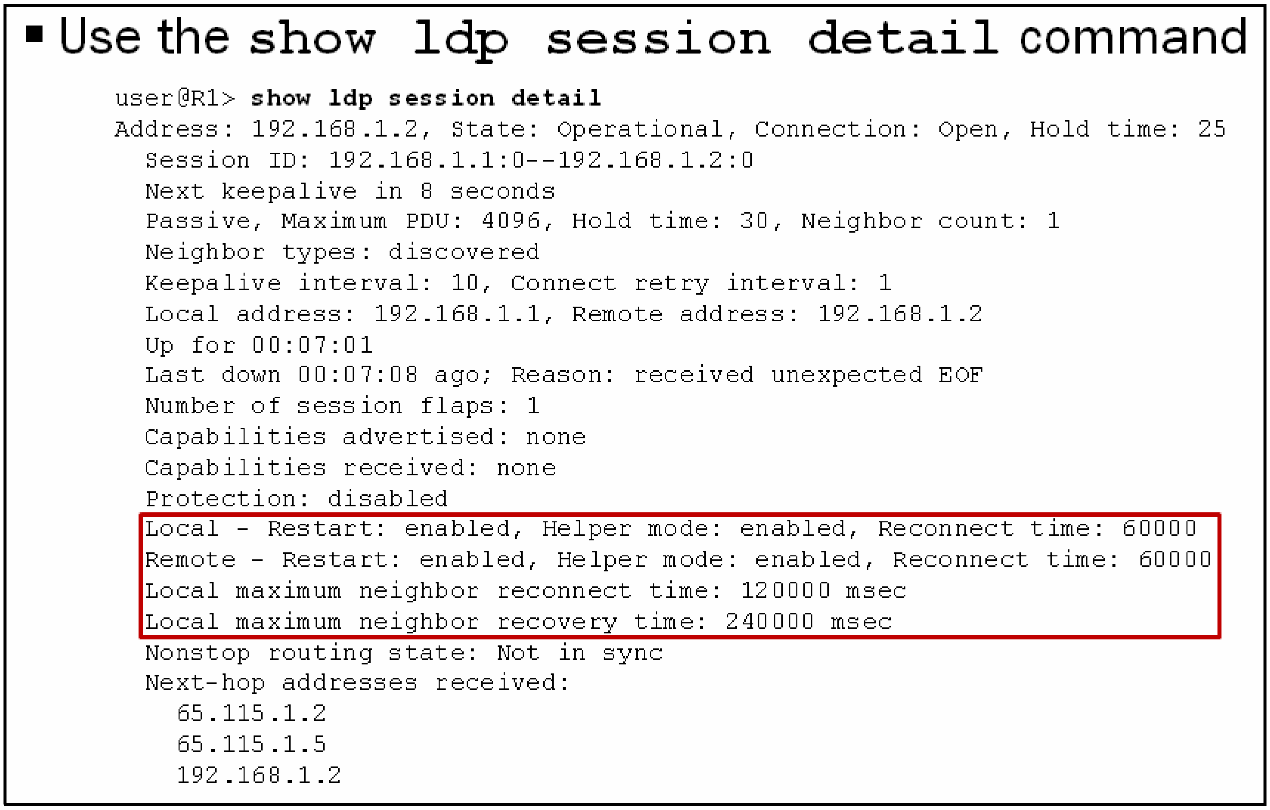
```
[edit protocols ldp]
user@R1# set graceful-restart disable

[edit protocols ldp]
user@R1# set graceful-restart helper-disable
```

The Junos OS supports LDP graceful restart procedures as defined in RFC 3478: *Graceful Restart Mechanism for Label Distribution Protocol*. LDP graceful restart is enabled when you add the **`graceful-restart`** statement to the main routing instance at the `[edit routing-options]` hierarchy. This global configuration statement enables graceful restart on all protocols that support the capability. In operation, a router makes its LDP graceful restart capabilities known through the inclusion of the Fault Tolerant TLV in its session initialization messages. By default, both graceful restart and helper mode are enabled for LDP when the **`graceful-restart`** statement is added to the main routing instance. After a restart, the local router signals that it was able to preserve its forwarding state by sending a nonzero `recovery-time` TLV in session messages to all neighbors. Neighbors with LDP helper mode enabled maintain the label mappings they last advertised to the restarting peer. When the LDP session is reestablished, the retained labels are readvertised (using mapping messages), which allows the restarting router to refresh all label bindings that are still valid (nonrefreshed labels are marked as stale and flushed).

The result is that the restarting router's signaling plane can be *bootstrapped* back into its pre-restart state, while forwarding continues unabated.

As mentioned earlier, LDP graceful restart and helper modes are enabled by default when graceful restart is configured. You can explicitly disable of LDP graceful restart and recovery, as well as prevent the router from performing helper mode function to a restarting router.

View LDP Graceful Restart

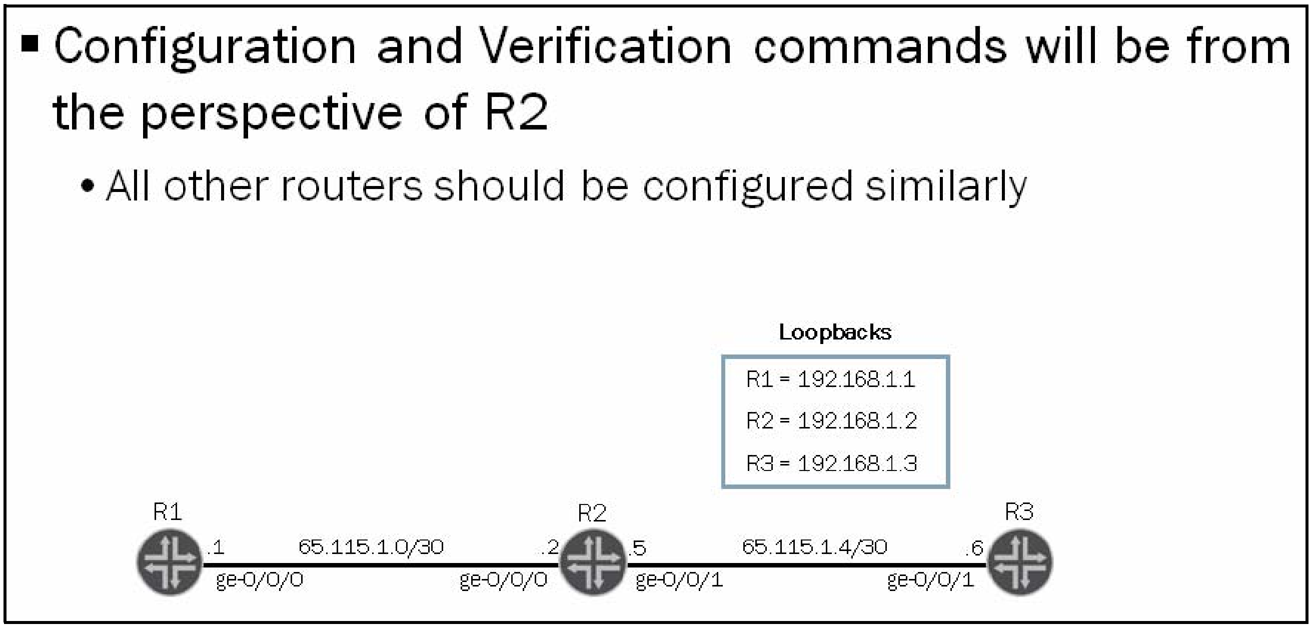


- Use the `show ldp session detail` command

```
user@R1> show ldp session detail
Address: 192.168.1.2, State: Operational, Connection: Open, Hold time: 25
  Session ID: 192.168.1.1:0--192.168.1.2:0
  Next keepalive in 8 seconds
  Passive, Maximum PDU: 4096, Hold time: 30, Neighbor count: 1
  Neighbor types: discovered
  Keepalive interval: 10, Connect retry interval: 1
  Local address: 192.168.1.1, Remote address: 192.168.1.2
  Up for 00:07:01
  Last down 00:07:08 ago; Reason: received unexpected EOF
  Number of session flaps: 1
  Capabilities advertised: none
  Capabilities received: none
  Protection: disabled
  Local - Restart: enabled, Helper mode: enabled, Reconnect time: 60000
  Remote - Restart: enabled, Helper mode: enabled, Reconnect time: 60000
  Local maximum neighbor reconnect time: 120000 msec
  Local maximum neighbor recovery time: 240000 msec
  Nonstop routing state: Not in sync
  Next-hop addresses received:
    65.115.1.2
    65.115.1.5
    192.168.1.2
```

Use the **`show ldp session`** command with the **`detail`** switch to confirm LDP graceful restart settings.

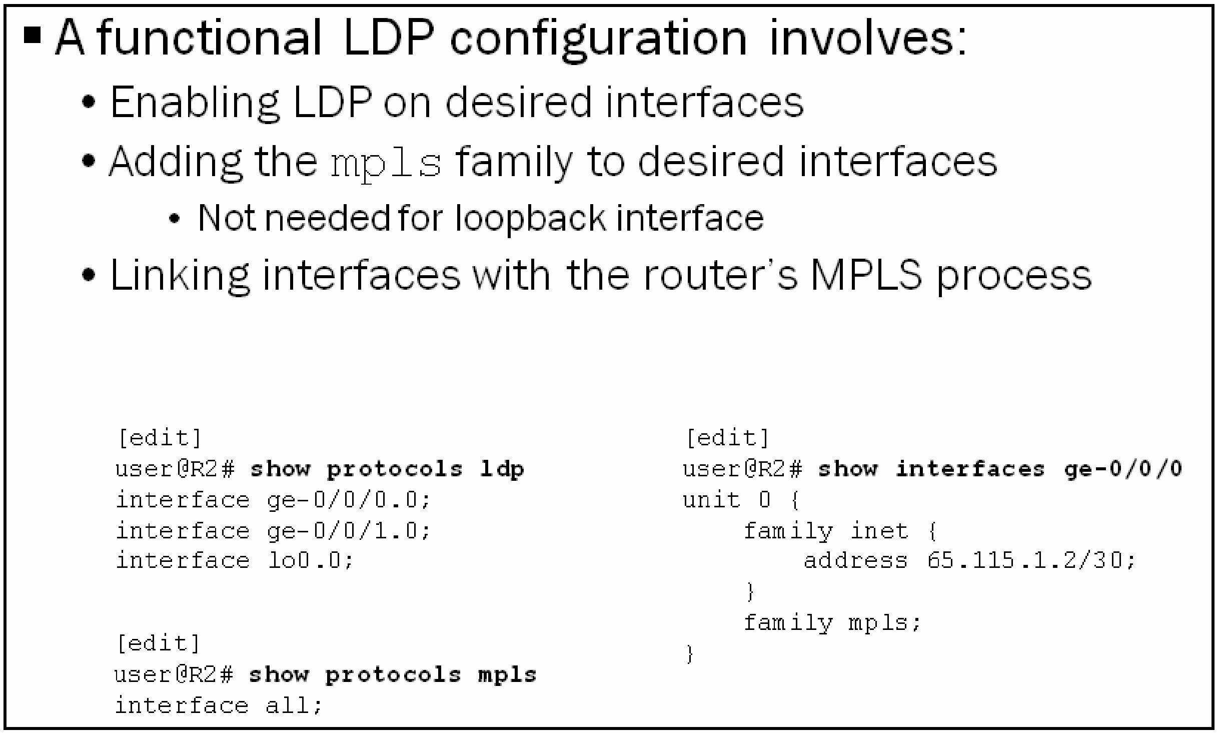
Sample LDP Topology



- Configuration and Verification commands will be from the perspective of R2
  - All other routers should be configured similarly

This graphic serves to establish a simple LDP topology that we use to drive the configuration examples and screen captures shown on subsequent pages.

Minimum LDP Configuration



- A functional LDP configuration involves:
  - Enabling LDP on desired interfaces
  - Adding the `mpls` family to desired interfaces
    - Not needed for loopback interface
  - Linking interfaces with the router’s MPLS process

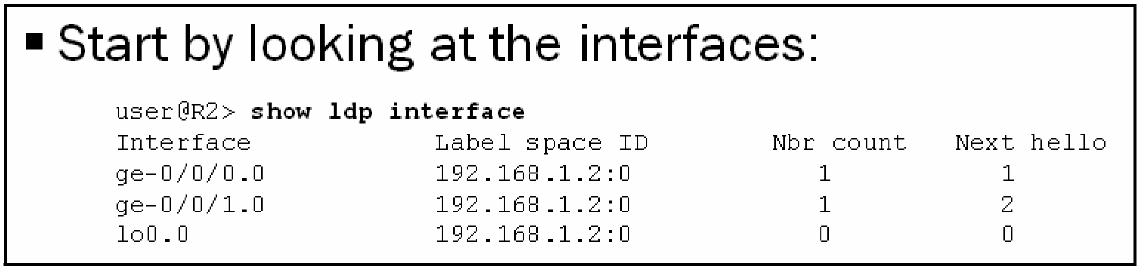
```
[edit]
user@R2# show protocols ldp
interface ge-0/0/0.0;
interface ge-0/0/1.0;
interface lo0.0;


[edit]
user@R2# show protocols mpls
interface all;
```

```
[edit]
user@R2# show interfaces ge-0/0/0
unit 0 {
    family inet {
        address 65.115.1.2/30;
    }
    family mpls;
}
```

You must configure LDP for each interface on which you want LDP to run. Further, you must also add **`family mpls`** to all interfaces that are expected to handle labeled packets, and you must associate these interfaces with the router’s MPLS process under the `[edit protocols mpls]` hierarchy. A minimum LDP configuration that is functional at both the signaling and data planes is shown on the graphic. Note that you should specifically disable LDP on the router’s out-of-band (OoB) interface when using the **`interface all`** option instead of manually specifying the interfaces.

By default, the Junos OS implementation of LDP only advertises the /32 router ID (RID) address, which is normally obtained from the `lo0` interface. Note that you must enable LDP on the `lo0` interface to achieve this behavior.

Confirm LDP Interfaces

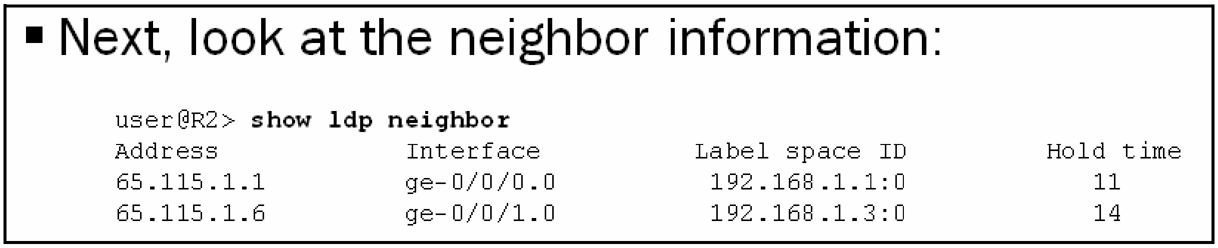


- Start by looking at the interfaces:

```
user@R2> show ldp interface
Interface             Label space ID         Nbr count   Next hello
ge-0/0/0.0            192.168.1.2:0             1           1
ge-0/0/1.0            192.168.1.2:0             1           2
lo0.0                 192.168.1.2:0             0           0
```

The **`show ldp interface`** command is an excellent place to begin LDP verification. In this example, all expected interfaces are listed, including the routers’ `lo0` interface. Note that the display also shows a count of neighbors detected on that interface.

Confirm LDP Neighbors

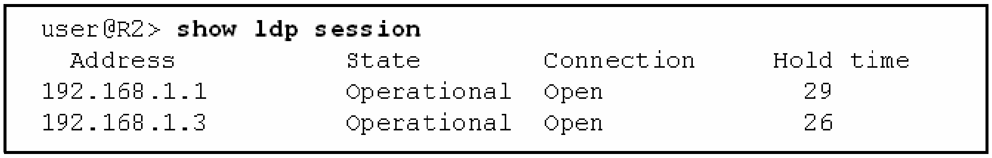


- Next, look at the neighbor information:

```
user@R2> show ldp neighbor
Address            Interface            Label space ID          Hold time
65.115.1.1         ge-0/0/0.0            192.168.1.1:0              11
65.115.1.6         ge-0/0/1.0            192.168.1.3:0              14
```

With LDP interfaces confirmed, you move on to verify that neighbor discovery is operational with the **`show ldp neighbors`** command. Note that each neighbor's RID is used to uniquely identify the label space for that neighbor.

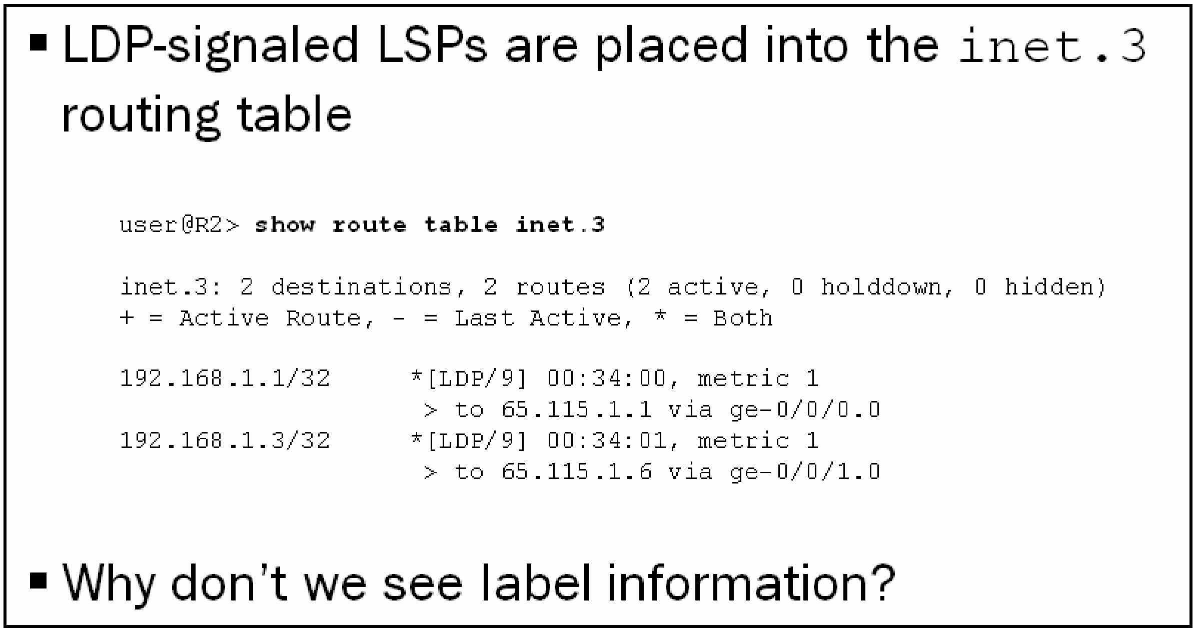
Verify Session State



```
user@R2> show ldp session
  Address           State        Connection     Hold time
192.168.1.1         Operational  Open             29
192.168.1.3         Operational  Open             26
```

With LDP interfaces and neighbors confirmed, you move on to verify that the sessions have been established using the **`show ldp sessions`** command.

Confirm the LDP Control Plane



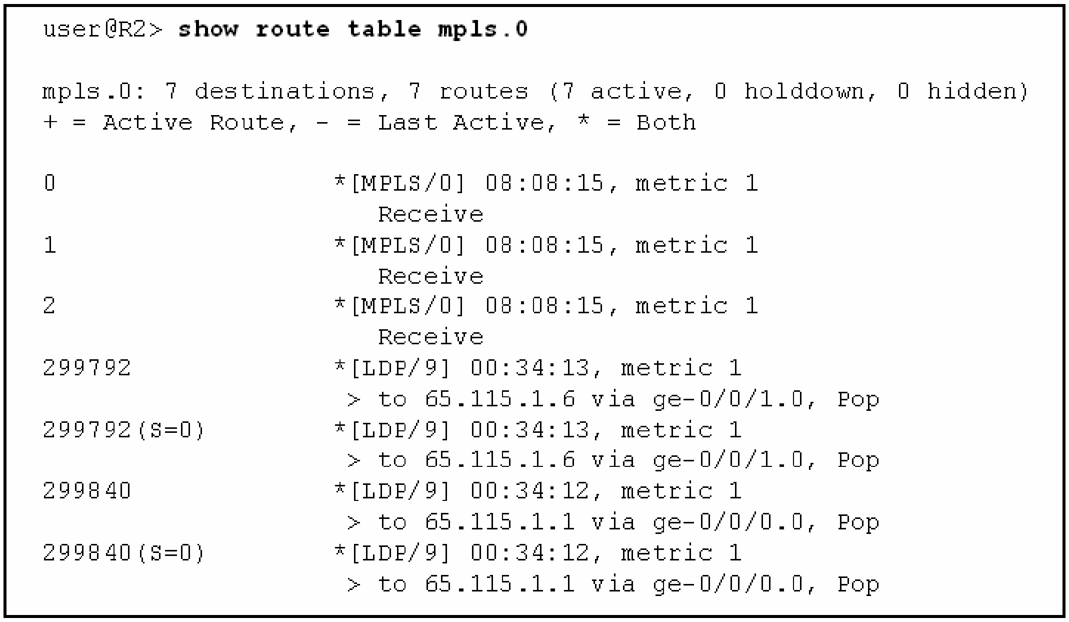
- LDP-signaled LSPs are placed into the `inet.3` routing table

```
user@R2> show route table inet.3

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.1/32      *[LDP/9] 00:34:00, metric 1
                     > to 65.115.1.1 via ge-0/0/0.0
192.168.1.3/32      *[LDP/9] 00:34:01, metric 1
                     > to 65.115.1.6 via ge-0/0/1.0
```

- Why don't we see label information?

A quick way to verify that LDP signaling is operational is to look for established LSPs in the routers `inet.3` routing table. Because the default behavior of LDP in the Junos OS is to establish tunnels to the /32 RID of all reachable routers, there should be at least one LSP if LDP is working at all. Note that the LSP to R3 is not associated with a label operation due to penultimate-hop popping (PHP) behavior.

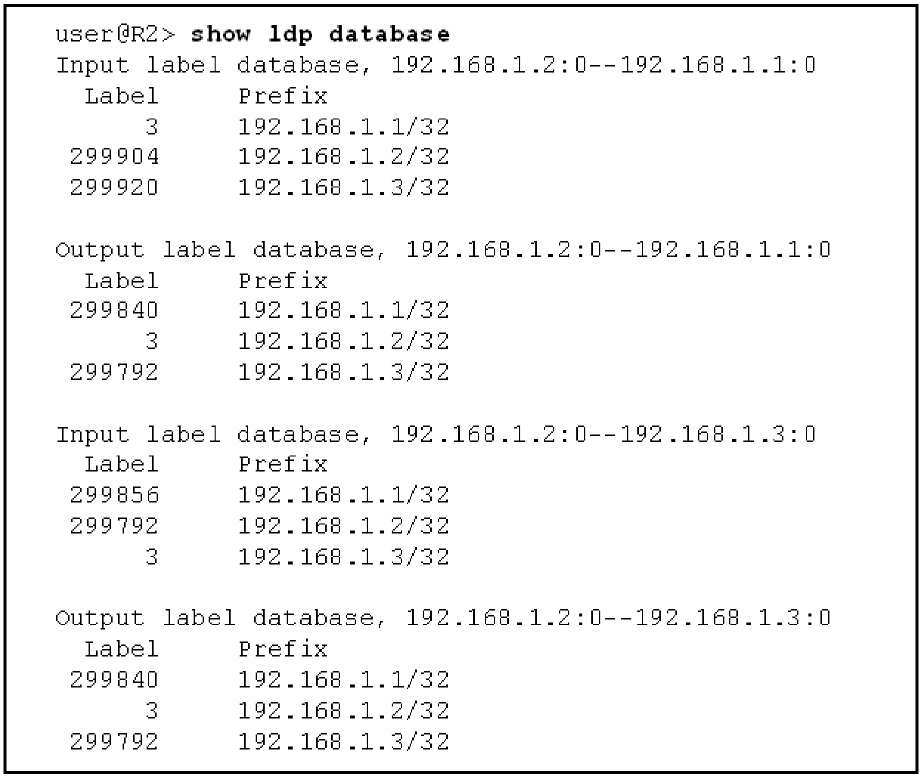
Verify Label Information Base



```
user@R2> show route table mpls.0

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 08:08:15, metric 1
                      Receive
1                  *[MPLS/0] 08:08:15, metric 1
                      Receive
2                  *[MPLS/0] 08:08:15, metric 1
                      Receive
299792             *[LDP/9] 00:34:13, metric 1
                    > to 65.115.1.6 via ge-0/0/1.0, Pop
299792(S=0)        *[LDP/9] 00:34:13, metric 1
                    > to 65.115.1.6 via ge-0/0/1.0, Pop
299840             *[LDP/9] 00:34:12, metric 1
                    > to 65.115.1.1 via ge-0/0/0.0, Pop
299840(S=0)        *[LDP/9] 00:34:12, metric 1
                    > to 65.115.1.1 via ge-0/0/0.0, Pop
```

Verify the expected labels are correctly mapped in the Label Information Base (LIB) by reviewing the contents of the `mpls.0` route table. Take note of the incoming label values on the left and ensure the router's egress interface is correct with the correct label operation.

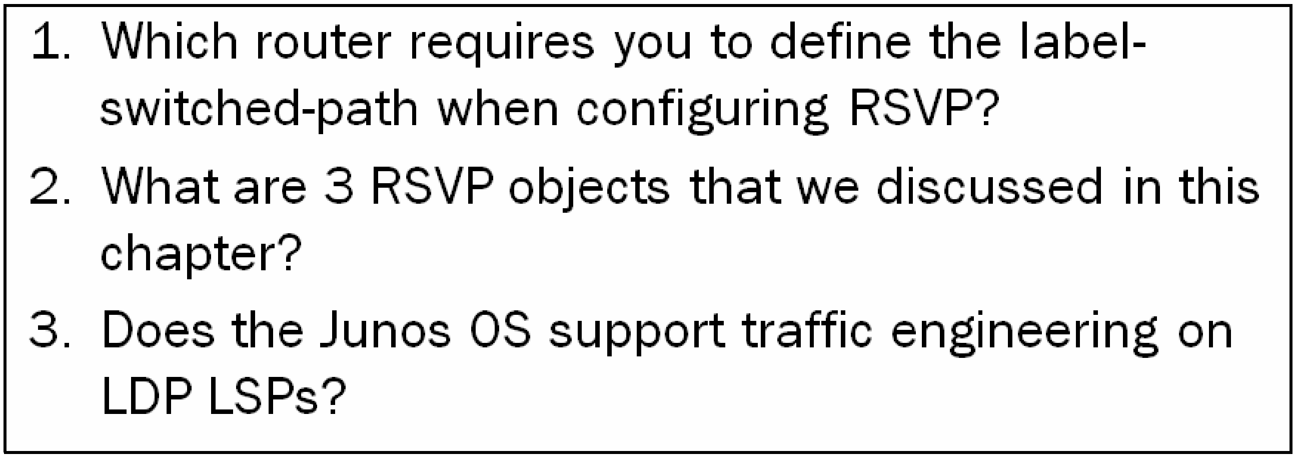
LDP Label Database



```
user@R2> show ldp database
Input label database, 192.168.1.2:0--192.168.1.1:0
  Label     Prefix
      3     192.168.1.1/32
 299904     192.168.1.2/32
 299920     192.168.1.3/32

Output label database, 192.168.1.2:0--192.168.1.1:0
  Label     Prefix
 299840     192.168.1.1/32
      3     192.168.1.2/32
 299792     192.168.1.3/32

Input label database, 192.168.1.2:0--192.168.1.3:0
  Label     Prefix
 299856     192.168.1.1/32
 299792     192.168.1.2/32
      3     192.168.1.3/32

Output label database, 192.168.1.2:0--192.168.1.3:0
  Label     Prefix
 299840     192.168.1.1/32
      3     192.168.1.2/32
 299792     192.168.1.3/32
```

You can view the label databases of LDP neighbors with the **`show ldp database`** command, as demonstrated on the graphic. Note that there will be a separate entry for each LDP peer. The LDP session ID delineates the entries learned over each session, with all the labels for either the input or output direction displayed. For example, LDP session `192.168.1.2:0`—`192.168.1.1:0` has three labels in the input label database and three labels in the output label database.

Review Questions



1. Which router requires you to define the label-switched-path when configuring RSVP?
2. What are 3 RSVP objects that we discussed in this chapter?
3. Does the Junos OS support traffic engineering on LDP LSPs?

Answers to Review Questions

1.

The only router that requires configuration is the ingress router. The other routers need to have protocols MPLS and RSVP configured but do not need information about the label-switched-path.

2.

There are many RSVP Objects. We discussed the SESSION, LABEL_REQUEST, EXPLICIT_ROUTE (ERO), RECORD_ROUTE (RRO), SESSION_ATTRIBUTE, RSVP-HOP, LABEL, and the STYLE objects within this chapter.

3.

No. The Junos OS does not support traffic engineering for LDP signaled LSPs. You can however, us LDP tunneling over RSVP traffic engineered LSPs to apply traffic engineering to the LDP traffic.

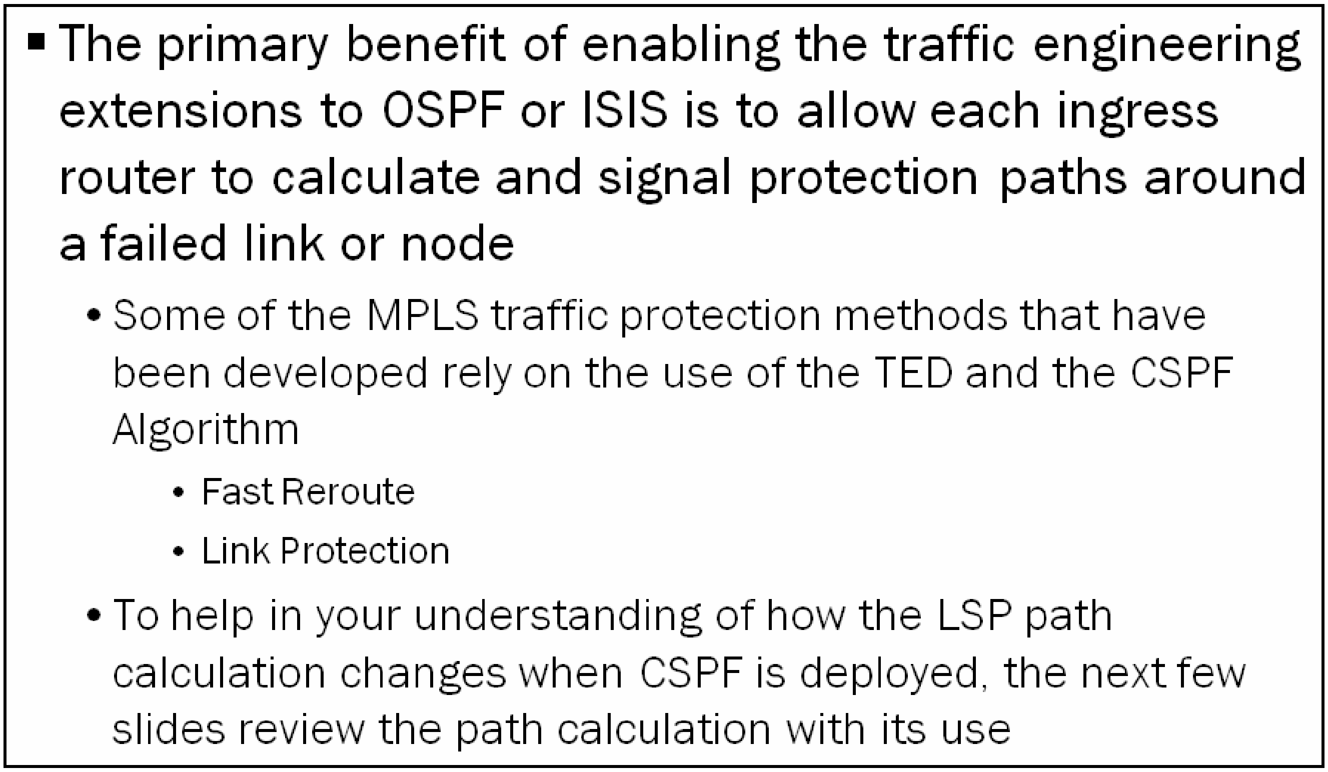
JNCIS-SP Study Guide—Part 3

Chapter 3: Constrained Shortest Path First

This Chapter Discusses:

- The path selection process of RSVP without the use of the Constrained Shortest Path First (CSPF) algorithm;
- The interior gateway protocol (IGP) extensions used to build the traffic engineering database (TED);
- The CSPF algorithm and its path selection process; and
- Administrative groups and how they can be used to influence path selection.

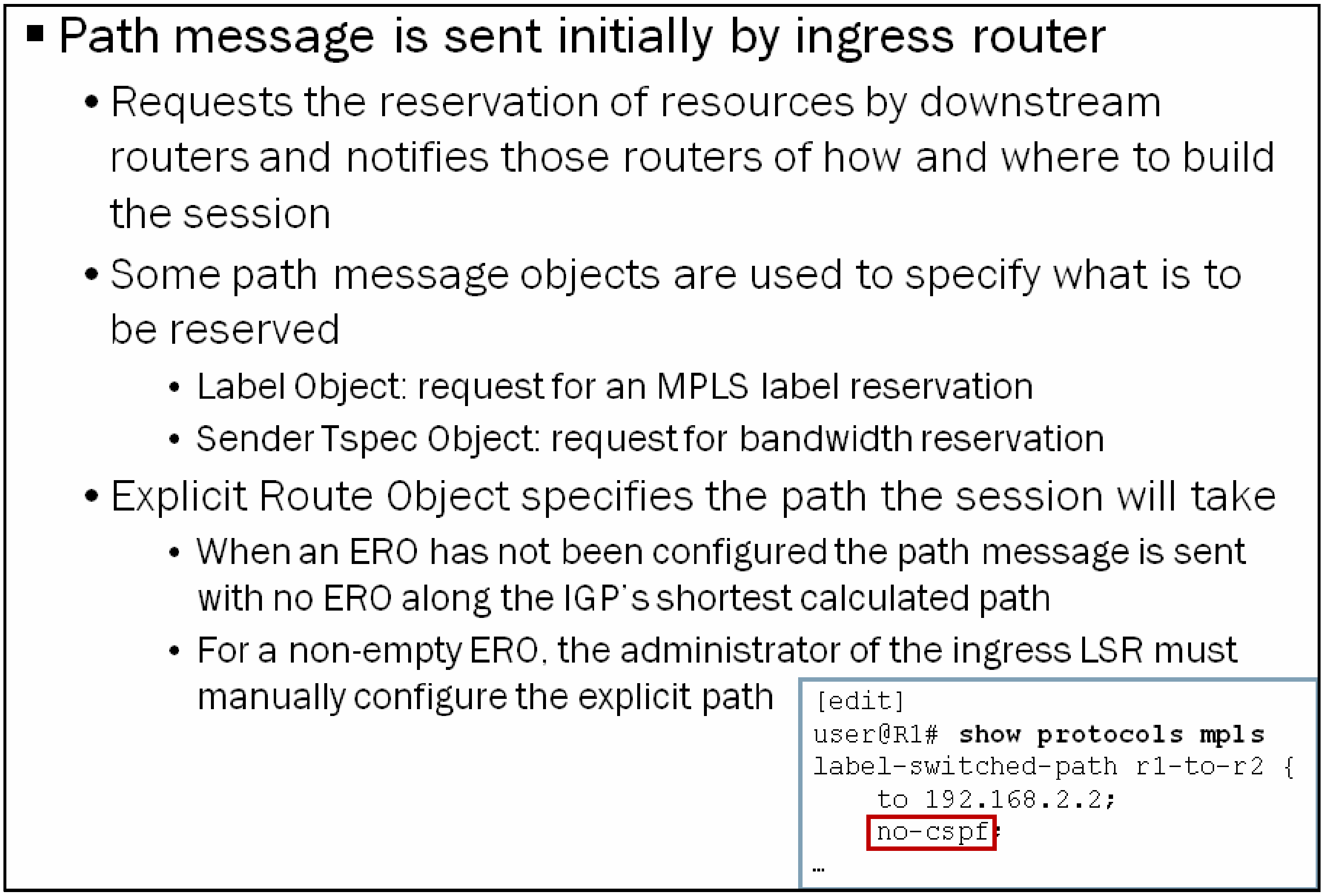
Protecting the MPLS Network



- The primary benefit of enabling the traffic engineering extensions to OSPF or ISIS is to allow each ingress router to calculate and signal protection paths around a failed link or node
  - Some of the MPLS traffic protection methods that have been developed rely on the use of the TED and the CSPF Algorithm
    - Fast Reroute
    - Link Protection
  - To help in your understanding of how the LSP path calculation changes when CSPF is deployed, the next few slides review the path calculation with its use

This chapter discusses the details of how an ingress router for an RSVP-signaled label-switched path (LSP) can use the CSPF algorithm and the TED to calculate its path through the network. As you read, keep in mind that it is this functionality that can provide protection against packet loss in an MPLS network. Some of the widely used protection methods for RSVP-signaled LSPs, like fast reroute and link protection (described in the next chapter), use the CSPF algorithm. The next few sections will prepare you for the CSPF topic by reviewing the path selection behavior of RSVP when CSPF has not been deployed.

Initial Path Message

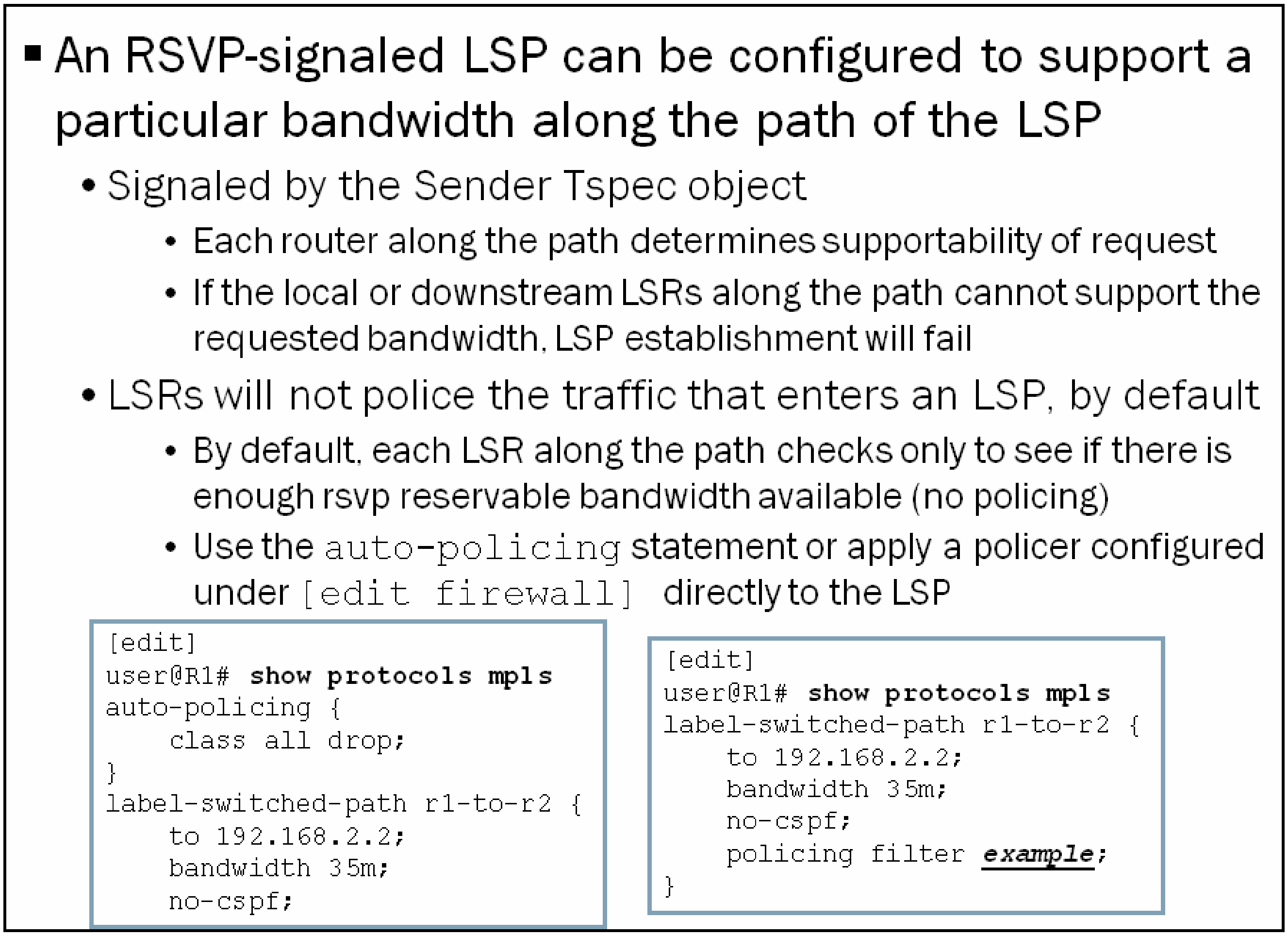


- Path message is sent initially by ingress router
  - Requests the reservation of resources by downstream routers and notifies those routers of how and where to build the session
  - Some path message objects are used to specify what is to be reserved
    - Label Object: request for an MPLS label reservation
    - Sender Tspec Object: request for bandwidth reservation
  - Explicit Route Object specifies the path the session will take
    - When an ERO has not been configured the path message is sent with no ERO along the IGP's shortest calculated path
    - For a non-empty ERO, the administrator of the ingress LSR must manually configure the explicit path

```
[edit]
user@R1# show protocols mpls
label-switched-path r1-to-r2 {
    to 192.168.2.2;
    no-cspf;
…
```

As it was described in the previous chapter, the initial path message related to the setup of an LSP is sent by the ingress router. The path message will contain objects. The objects within the path message are used to either request the reservation of resources by downstream routers (MPLS label, bandwidth, and so forth) or inform those same downstream routers of the direction in which they should be building the RSVP session (MPLS LSP). The path the session will take depends on both the IGP shortest path calculation along with what is contained in the Explicit Route Object (ERO) of the path message. When CSPF is not in use (**`no-cspf`** option) and no ERO has been manually configured by the administrator of the ingress router, the initial path message will be sent with no ERO. For a path message to have a non-empty ERO, it must be manually configured by the administrator of the ingress router.

Bandwidth Reservation



- An RSVP-signaled LSP can be configured to support a particular bandwidth along the path of the LSP
  - Signaled by the Sender Tspec object
    - Each router along the path determines supportability of request
    - If the local or downstream LSRs along the path cannot support the requested bandwidth, LSP establishment will fail
  - LSRs will not police the traffic that enters an LSP, by default
    - By default, each LSR along the path checks only to see if there is enough rsvp reservable bandwidth available (no policing)
    - Use the `auto-policing` statement or apply a policer configured under `[edit firewall]` directly to the LSP

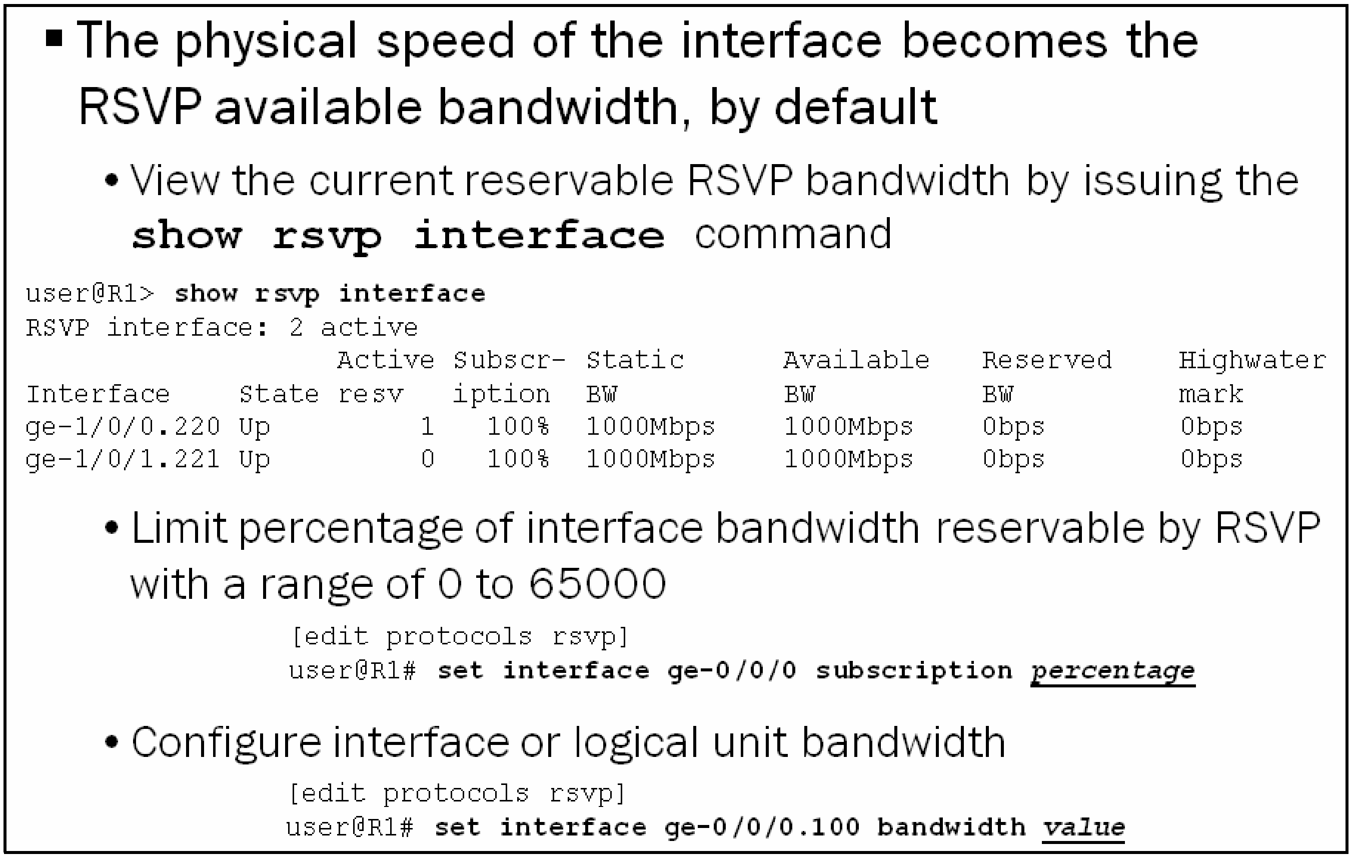
```
[edit]
user@R1# show protocols mpls
auto-policing {
    class all drop;
}
label-switched-path r1-to-r2 {
    to 192.168.2.2;
    bandwidth 35m;
    no-cspf;
```

```
[edit]
user@R1# show protocols mpls
label-switched-path r1-to-r2 {
    to 192.168.2.2;
    bandwidth 35m;
    no-cspf;
    policing filter example;
}
```

It is possible for an LSP to be configured to reserve bandwidth along the path of the LSP. During the setup process for an LSP configured for bandwidth (as shown in graphic), each downstream router will receive a request to reserve bandwidth for the LSP in the form of the traffic specification (TSpec) object. Each router along the path will make its own individual decision as whether it has enough available bandwidth on its egress interface for the LSP. To determine whether or not there is enough available bandwidth, a router will sum the bandwidth of all LSPs traversing the egress interface and subtract it from the total bandwidth for the interface. If there is not enough available bandwidth, the LSP will fail to be instantiated and the upstream routers will be informed with a PathErr message.

By default, the bandwidths described on the graphic are only logical and used for LSP setup. The amount of traffic that actually traverses an LSP is not enforced. It is possible, however, to override the default behavior and have the ingress router police the traffic that enters an LSP however. This can be done by configuring **auto-policing** or configuring a firewall filter an applying directly to the specific LSP.

RSVP Bandwidth



- The physical speed of the interface becomes the RSVP available bandwidth, by default
  - View the current reservable RSVP bandwidth by issuing the **show rsvp interface** command

```
user@R1> show rsvp interface
RSVP interface: 2 active
                   Active Subscr- Static       Available    Reserved     Highwater
Interface     State resv   iption  BW           BW           BW           mark
ge-1/0/0.220 Up         1   100%  1000Mbps     1000Mbps     0bps         0bps
ge-1/0/1.221 Up         0   100%  1000Mbps     1000Mbps     0bps         0bps
```

  - Limit percentage of interface bandwidth reservable by RSVP with a range of 0 to 65000

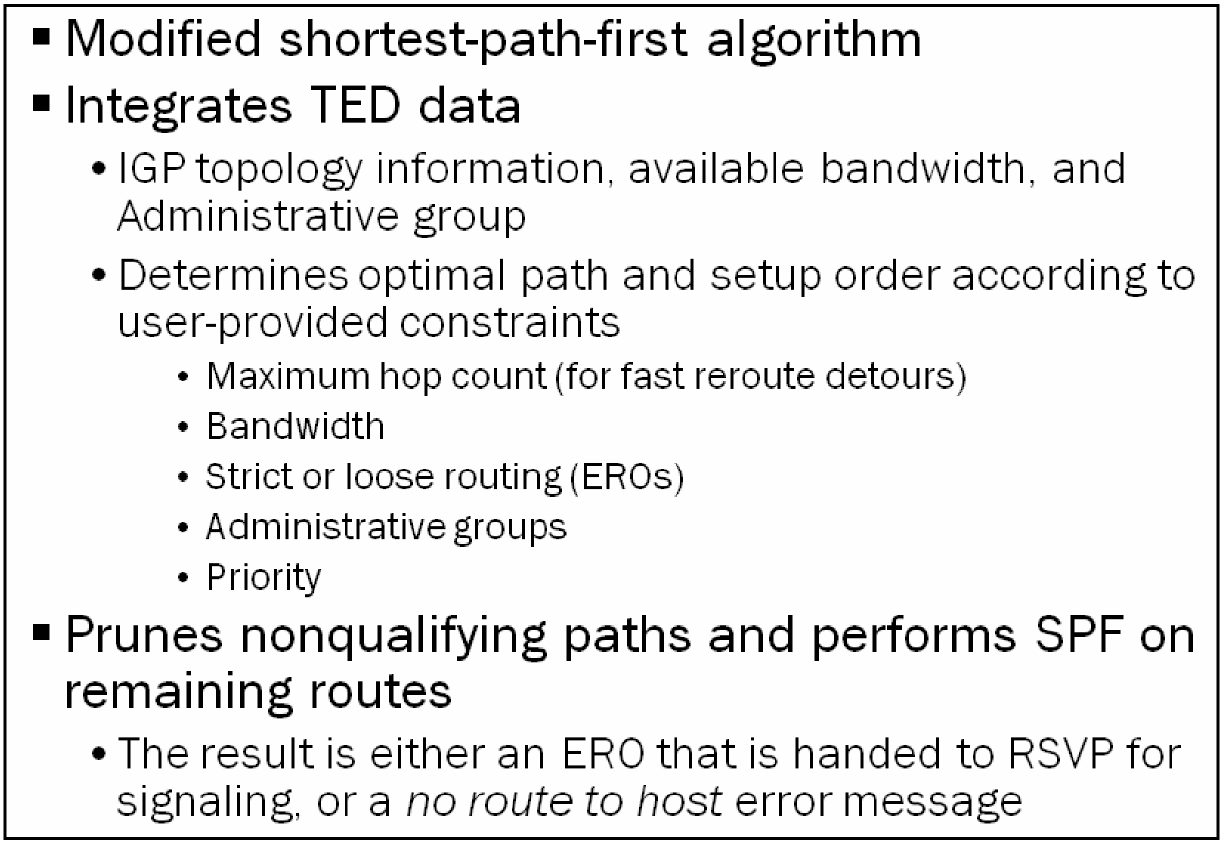
```
[edit protocols rsvp]
user@R1# set interface ge-0/0/0 subscription percentage
```

  - Configure interface or logical unit bandwidth

```
[edit protocols rsvp]
user@R1# set interface ge-0/0/0.100 bandwidth value
```

Whether the CSPF algorithm is being used, when RSVP is being used for LSP signaling in a a network, every interface on every router will have an associated available bandwidth associated with it. By default, the available bandwidth for LSP reservation is equivalent to the physical speed of the interface, which can be overridden by one of the methods shown on the graphic.

A Modified Shortest-Path-First Algorithm



- Modified shortest-path-first algorithm
- Integrates TED data
  - IGP topology information, available bandwidth, and Administrative group
  - Determines optimal path and setup order according to user-provided constraints
    - Maximum hop count (for fast reroute detours)
    - Bandwidth
    - Strict or loose routing (EROs)
    - Administrative groups
    - Priority
- Prunes nonqualifying paths and performs SPF on remaining routes
  - The result is either an ERO that is handed to RSVP for signaling, or a *no route to host* error message

The ingress router determines the physical path for each LSP by applying a CSPF algorithm to the information in the TED. CSPF is a shortest-path-first (SPF) algorithm that has been modified to take into account specific restrictions when calculating the shortest path across the network. Links that do not comply with the restrictions are removed from the tree and cannot be factored into the resulting SPF calculations.

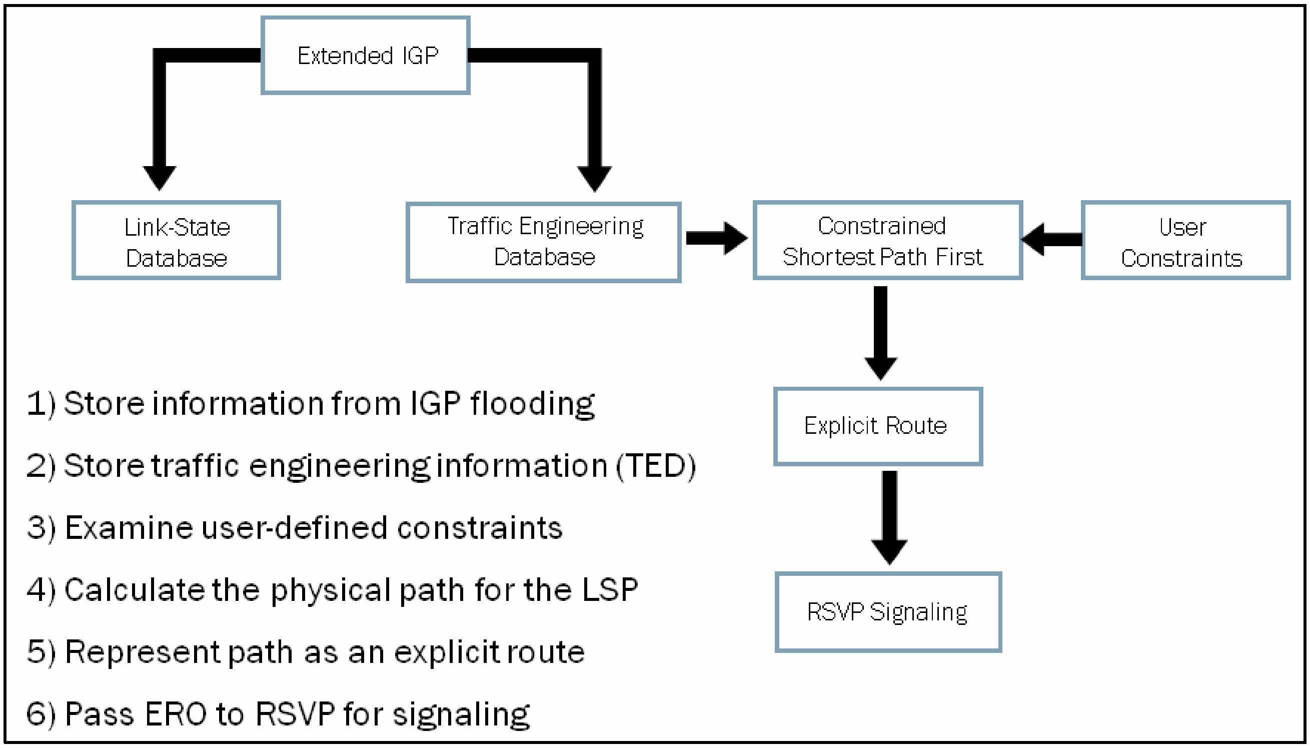
TED and User Constraint Integration

CSPF integrates topology link-state information learned from IGP traffic engineering extensions and is maintained in the TED. The information stored in the TED includes attributes associated with the state of network resources (such as total link bandwidth, reserved link bandwidth, available link bandwidth, and link color). When calculating a path, the CSPF algorithm factors in user-provided constraints such as bandwidth requirements, maximum allowed hop count, and administrative groups, all of which are obtained from user configuration.

Prune Nonqualifying Links

As CSPF considers each candidate node and link for a new LSP, it either accepts or rejects a specific path component based on resource availability and whether selecting the component violates a user provided constraint. The output of a successful CSPF calculation is an explicit route consisting of a sequence of router addresses that provides the shortest path through the network that meets all provided constraints. This explicit route is then passed to the RSVP-signaling component, which establishes forwarding state in the routers along the LSP. When no compliant route can be found, the output of the CSPF algorithm is a rather generic *no route* to host error message.

Ingress LSR Operations



1) Store information from IGP flooding

2) Store traffic engineering information (TED)

3) Examine user-defined constraints

4) Calculate the physical path for the LSP

5) Represent path as an explicit route

6) Pass ERO to RSVP for signaling

The graphic lists the six primary aspects of the CSPF process from the perspective of the ingress label-switching router (LSR). We describe each CSPF component in the following list:

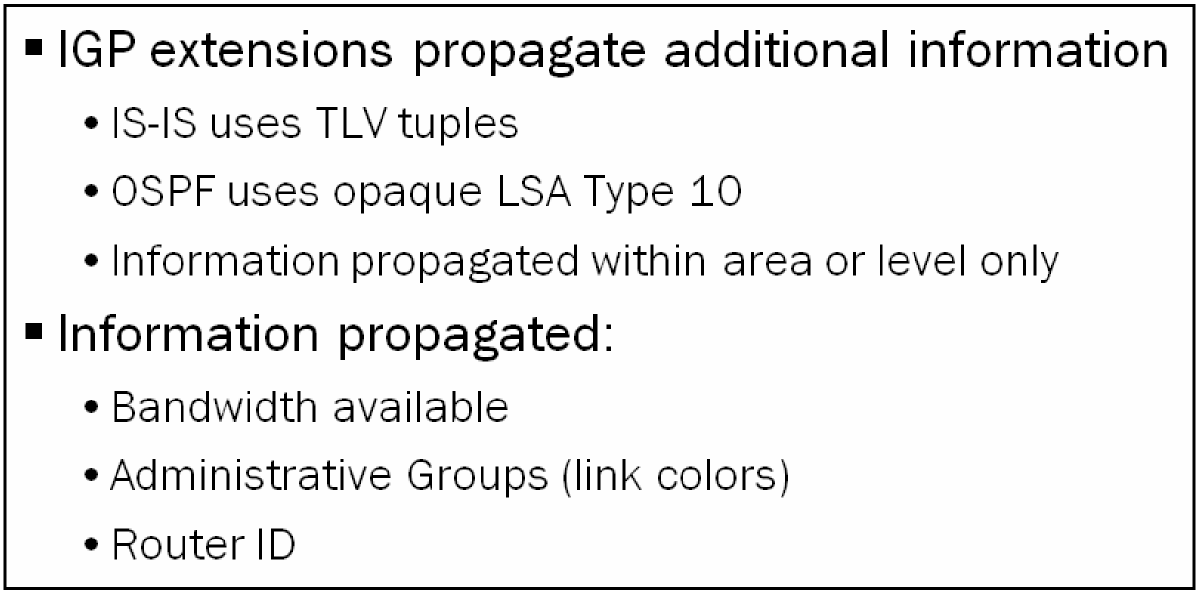
1. *Information Propagation*: Traffic engineering extensions to either Intermediate System-to-Intermediate System (IS-IS) or Open Shortest Path First (OSPF) carry traffic engineering topology information.
2. *Information Storage*: The router stores traffic engineering link-state information in the TED.
3. *User Constraints*: The user specifies constraints for a specific LSP through configuration settings.
4. *Physical Path Calculation*: The CSPF algorithm finds the shortest path based on links that comply with user-provided constraints.
5. *Explicit Route Generation*: The router forms a complete list of EROs that describes the sequence of nodes and links representing the shortest compliant path between ingress and egress LSRs.
6. *RSVP-signaling*: The router passes the computed ERO list to RSVP for LSP signaling. Note that because the TED contains a relatively up-to-date view of the entire network's current state, a high probability exists that the RSVP-signaled LSP will succeed. Put another way, if no path in the network meets a provided constraint, CSPF does not compute an ERO list, and RSVP does not even attempt to signal an LSP that would be doomed to failure anyway. Last minute changes in the state of the network might result in the TED being slightly out of date, and this can lead to RSVP path signaling failures until the TED is again synchronized with the true state of the network.

IGP Extensions

- IS-IS uses TLV tuples
- OSPF uses opaque LSA Type 10
- Information propagated within area or level only

Both OSPF and IS-IS can propagate additional information through some form of extension. IS-IS carries different parameters in type/length/value (TLV) tuples, which are propagated within a level; these TLVs do not propagate between levels. OSPF, on the other hand, uses Type 10 opaque LSAs to carry traffic engineering extensions. Type 10 LSAs have an area flooding scope, meaning that the information is propagated within a given area only; OSPF traffic engineering extensions do not cross area border routers (ABRs). The MPLS Traffic Engineering Information carried by these IGP extensions is defined in RFC 3630 and RFC 4203 for OSPF, and RFC 3784 and RFC 4205 for IS-IS.

Information Propagated



- IGP extensions propagate additional information
  - IS-IS uses TLV tuples
  - OSPF uses opaque LSA Type 10
  - Information propagated within area or level only
- Information propagated:
  - Bandwidth available
  - Administrative Groups (link colors)
  - Router ID

The TLVs listed here are based on IS-IS traffic engineering extensions. OSPF supports more or less the same parameters; the primary difference is how the extended information is propagated (TLV versus opaque LSA).

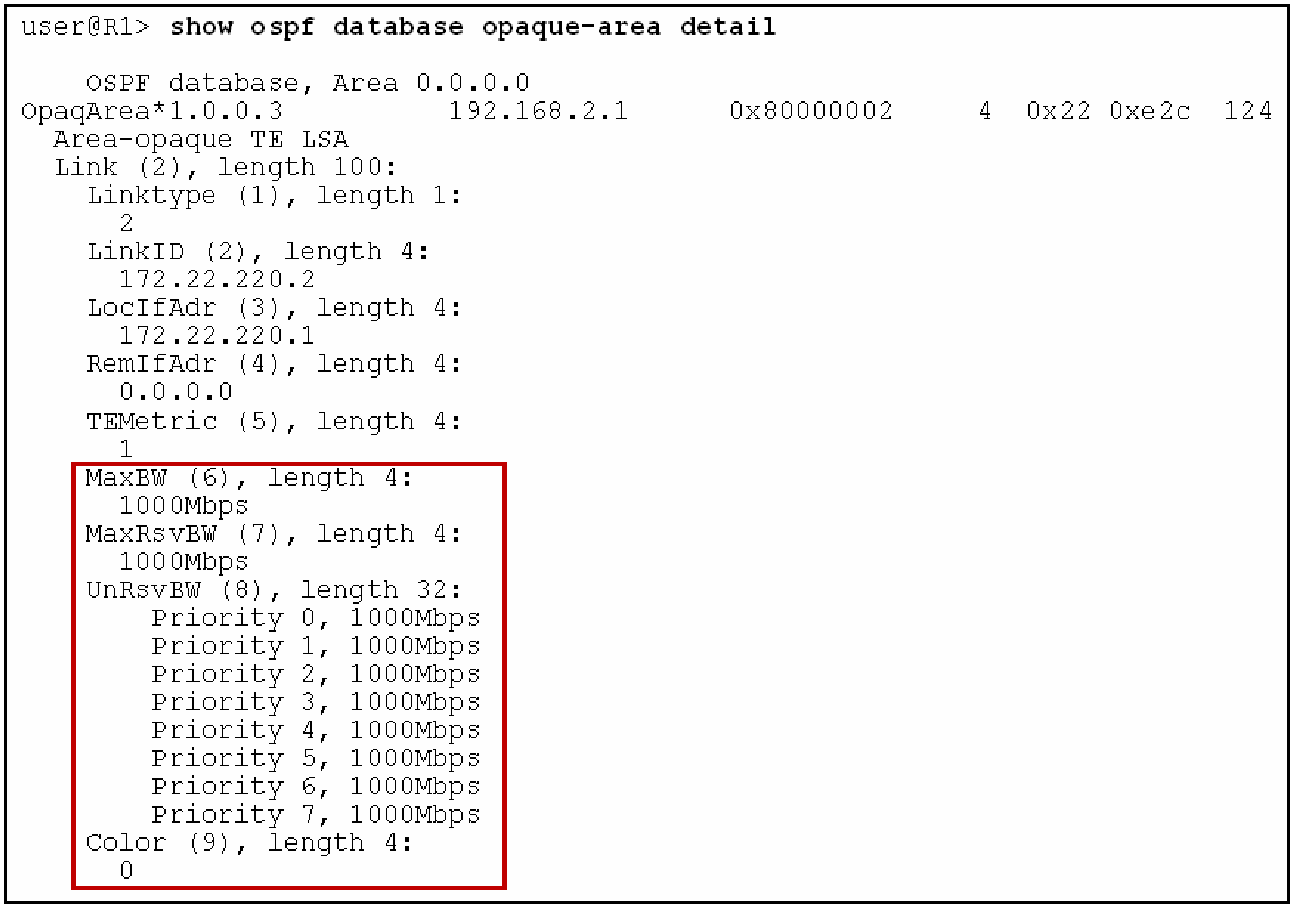
- Router ID (TLV 134): Single stable address, regardless of node's interface state. The /32 prefix for router ID should not be installed into the forwarding table or it can lead to forwarding loops for systems that do not support this TLV.
- Extended IP Reachability (TLV 135): One bit used for route leaking (up/down bit); extends metrics from 6 bits to 32 bits.
- Extended IS Reachability (TLV 22): Contains information about a series of neighbors. Consists of the following sub-TLVs:
  - IPv4 Neighbor Address (Sub-TLV 8).
  - Maximum Link Bandwidth (Sub-TLV 9): A 32-bit field, Institute of Electrical and Electronics Engineers (IEEE) floating point format. Units are bytes per second and unidirectional.
  - Maximum Reservable Bandwidth (Sub-TLV 10): A 32-bit field, IEEE floating point format. Units are bytes per second and unidirectional. Supports over subscription (can be greater than link bandwidth).
  - Unreserved Bandwidth (Sub-TLV 11): A 32-bit field, IEEE floating point format. Units are bytes per second. A value is specified for each priority level 0 through 7.
  - Traffic Engineering Default Metric (Sub-TLV 18): A 24-bit unsigned integer.
  - Resource Class/Color (Sub-TLV 3): Specifies administrative group membership (also known as affinity class). Up to 32 different groups. Each group is represented by a different bit.

OSPF traffic engineering extensions include the following TLVs. Note that these extensions are silently discarded by non-traffic-engineering-aware routers in accordance with opaque LSA processing rules.

- Router TLV: Stable IP address of advertising router.
- Link TLV: Composed of the following sub-TLVs:
  - Link Type: PP or multi-access.
  - Link ID: Identifies other end of link. A designated router is identified if the link is used for multi-access.
  - Local Interface Address: IP address of the link; advertising router address if unnumbered link.

- Remote Interface IP Address: Neighbors' IP address; first two octets 0 if unnumbered, remaining octets are local interface index assignment. This sub-TLV and local address used to discern multiple parallel links between systems.
- Traffic Engineering Metric: Link metric for traffic engineering. Might be different than the OSPF link metric.
- Maximum Bandwidth (Unidirectional): A 32-bit IEEE floating point format. Bytes per second.
- Maximum Reservable Bandwidth: A 32-bit IEEE floating point format. Over subscription supported. Bytes per second.
- Unreserved Bandwidth: Unreserved bandwidth for each of the eight priority levels. Bytes per second. 32-bit IEEE floating point format. Each value less than or equal to maximum reservable bandwidth.
- Resource Class/Color: Specifies administrative group membership (also known as affinity class). Up to 32 different groups. Each group is represented by a different bit.

IGP Extensions: OSPF

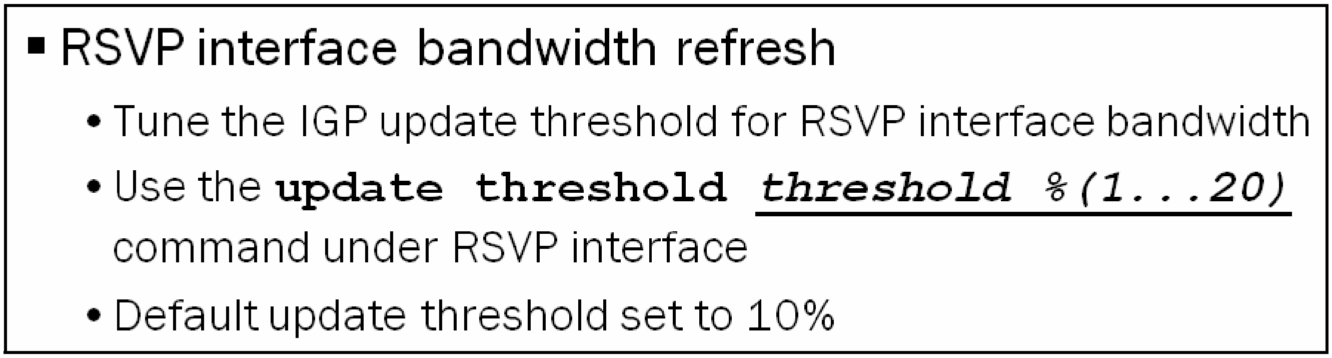


```
user@R1> show ospf database opaque-area detail

    OSPF database, Area 0.0.0.0
OpaqArea*1.0.0.3          192.168.2.1      0x80000002     4  0x22 0xe2c  124
  Area-opaque TE LSA
  Link (2), length 100:
    Linktype (1), length 1:
      2
    LinkID (2), length 4:
      172.22.220.2
    LocIfAdr (3), length 4:
      172.22.220.1
    RemIfAdr (4), length 4:
      0.0.0.0
    TEMetric (5), length 4:
      1
    MaxBW (6), length 4:
      1000Mbps
    MaxRsvBW (7), length 4:
      1000Mbps
    UnRsvBW (8), length 32:
        Priority 0, 1000Mbps
        Priority 1, 1000Mbps
        Priority 2, 1000Mbps
        Priority 3, 1000Mbps
        Priority 4, 1000Mbps
        Priority 5, 1000Mbps
        Priority 6, 1000Mbps
        Priority 7, 1000Mbps
    Color (9), length 4:
      0
```

The capture shown on the graphic provides an example of an IGP update that also carries traffic engineering extensions for distribution to the TED. By default, the IGP only sends an update message if the available link bandwidth changes by greater than 10%. The **update threshold** command (covered later in this chapter) allows you to alter this default behavior. The highlighted portion of the graphic calls out the following attributes:

- Unreserved bandwidth indicates the reservable bandwidth by priority level for a given link;
- Maximum reservable bandwidth indicates the total reservable bandwidth for a given link;
- Maximum bandwidth communicates the total bandwidth for the link; and
- Color identifies the hexadecimal bit mask used to associate affinity classes (administrative groups) with this link.

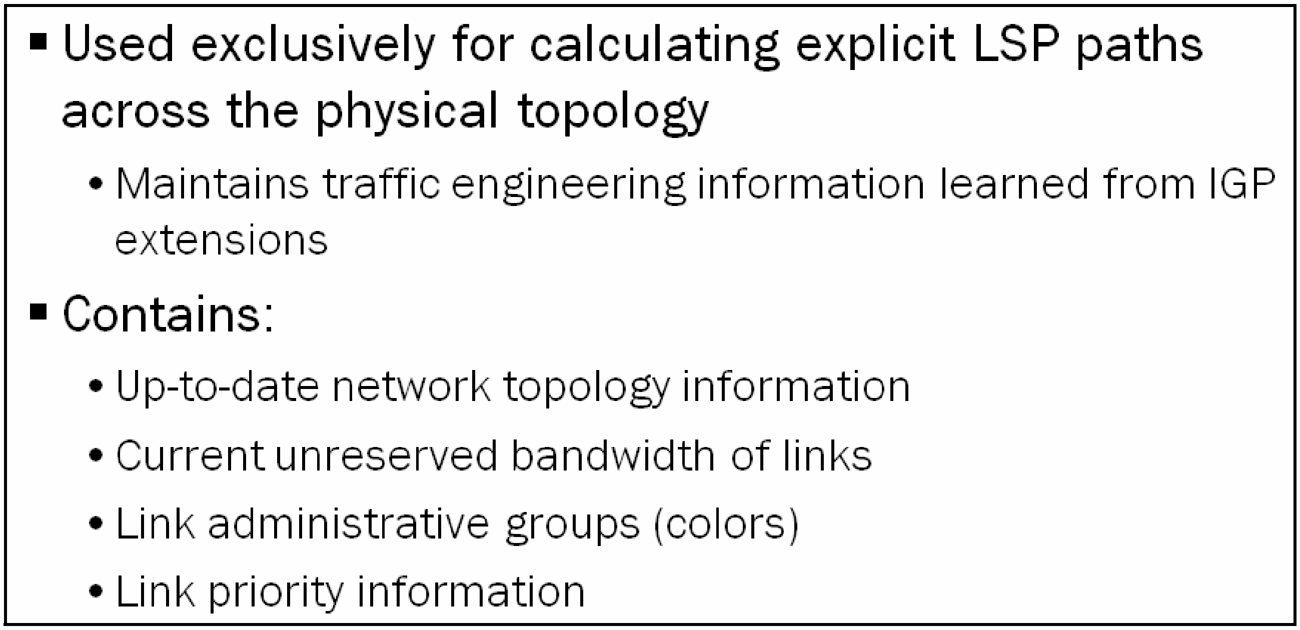
Interface Bandwidth Refresh



- RSVP interface bandwidth refresh
  - Tune the IGP update threshold for RSVP interface bandwidth
  - Use the **update threshold** ***threshold %(1...20)*** command under RSVP interface
  - Default update threshold set to 10%

The Junos operating system propagates changes in bandwidth according to a configured threshold percentage. By default, updates are sent only if the bandwidth changes by 10%. However, you can configure the update threshold to be a percentage from 1 to 20.

Used Exclusively for LSP Path Computation



- Used exclusively for calculating explicit LSP paths across the physical topology
  - Maintains traffic engineering information learned from IGP extensions
- Contains:
  - Up-to-date network topology information
  - Current unreserved bandwidth of links
  - Link administrative groups (colors)
  - Link priority information

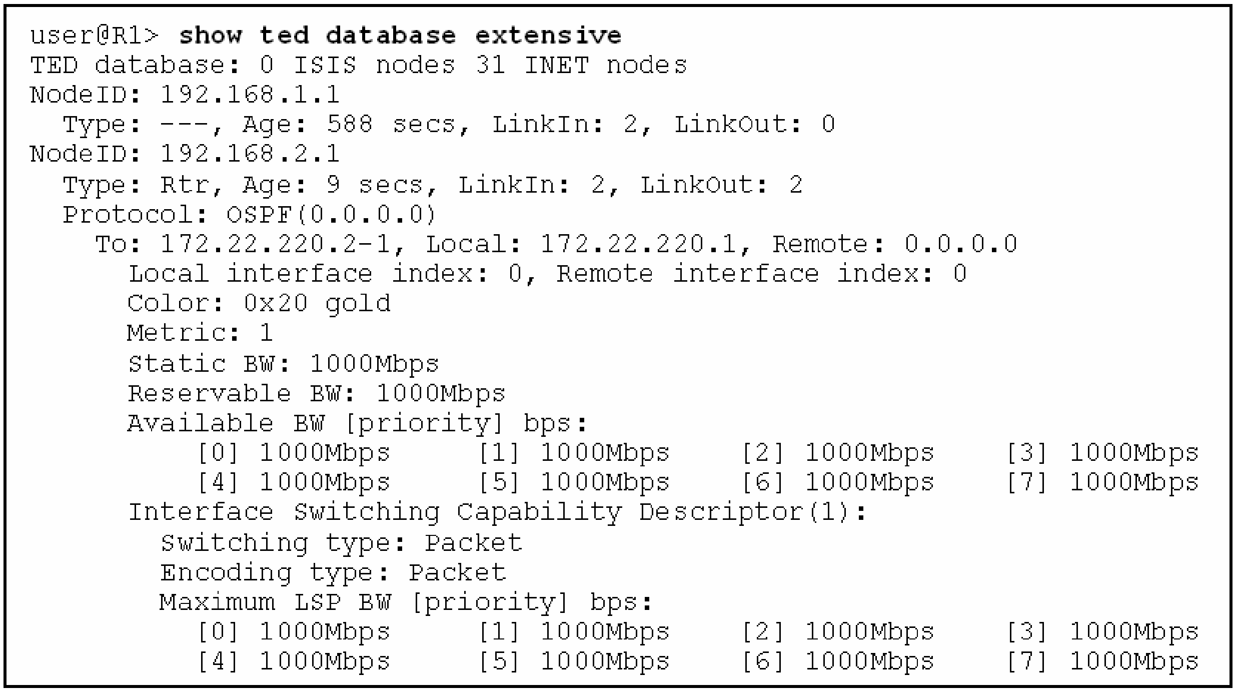
Each router maintains network link attributes and topology information in its TED. The TED is used exclusively for calculating explicit paths for the placement of LSPs across the physical topology. Because the TED does not know about existing LSPs, the TED does not allow a CSPF LSP to form over an LSP (because a non-CSPF LSP consults the routing table on a hop-by-hop basis to forward the RSVP messages, a non-CSPF LSP might try to form over an existing LSP) if features like forwarding adjacencies or traffic engineering shortcuts are enabled.

TED Contents

CSPF uses the TED to calculate explicit paths across the physical topology. It is similar to IGP link-state database (LSDB) and relies on extensions to the IGP, but it is stored independently of the IGP database.

Traffic engineering requires detailed knowledge about the network topology as well as dynamic information about network loading. The information distribution component is implemented by defining relatively simple extensions to the IGPs so that link attributes are included as part of each router's link-state advertisement (LSA). The standard flooding algorithm used by the link-state IGPs ensures that link attributes are distributed to all routers in the routing domain. Some of the traffic engineering extensions to be added to the IGP link-state advertisement include maximum link bandwidth, maximum reserved link bandwidth, current bandwidth reservation levels, and link coloring.

Analyzing the TED



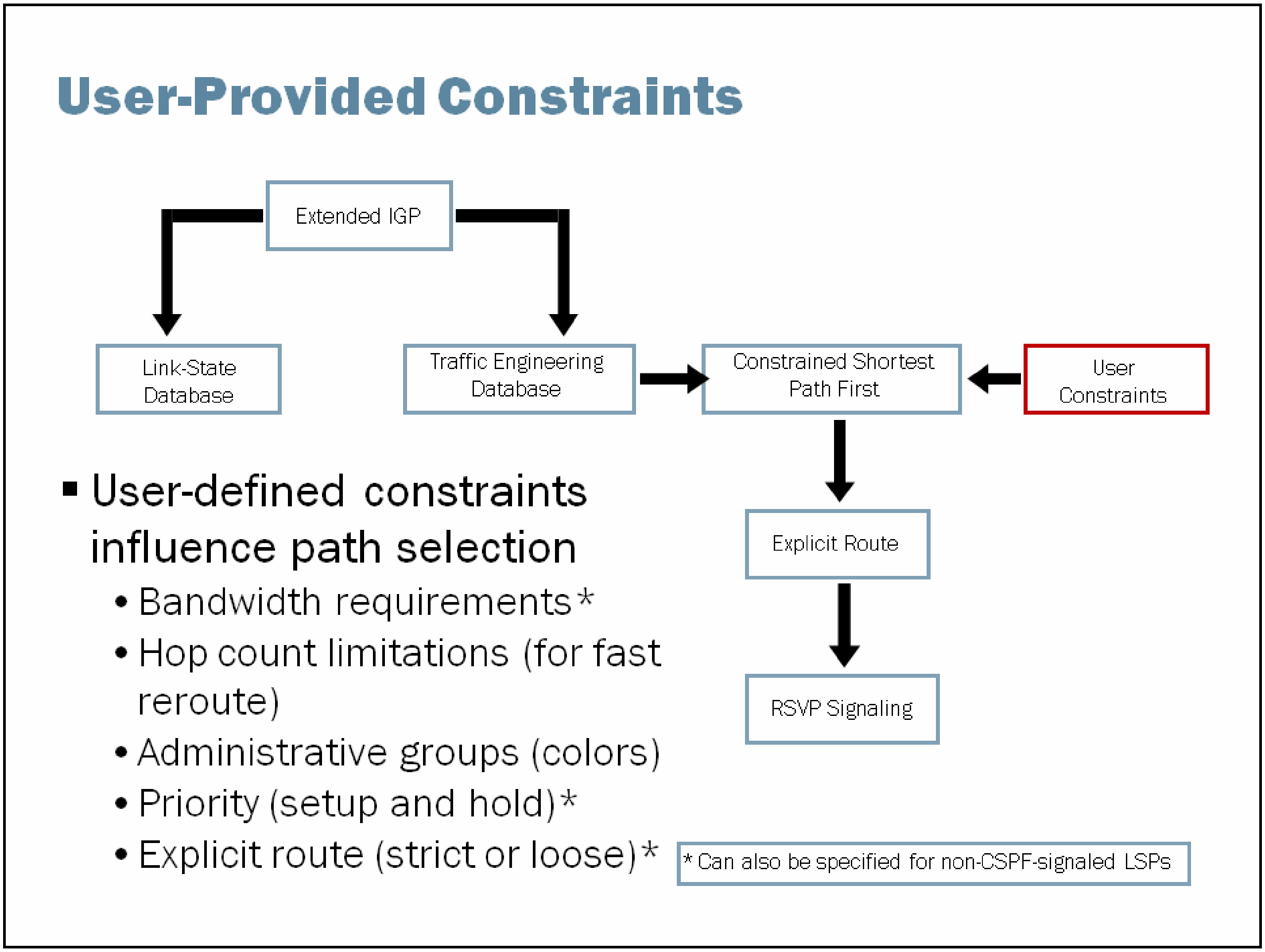
```
user@R1> show ted database extensive
TED database: 0 ISIS nodes 31 INET nodes
NodeID: 192.168.1.1
  Type: ---, Age: 588 secs, LinkIn: 2, LinkOut: 0
NodeID: 192.168.2.1
  Type: Rtr, Age: 9 secs, LinkIn: 2, LinkOut: 2
  Protocol: OSPF(0.0.0.0)
    To: 172.22.220.2-1, Local: 172.22.220.1, Remote: 0.0.0.0
      Local interface index: 0, Remote interface index: 0
      Color: 0x20 gold
      Metric: 1
      Static BW: 1000Mbps
      Reservable BW: 1000Mbps
      Available BW [priority] bps:
          [0] 1000Mbps     [1] 1000Mbps    [2] 1000Mbps    [3] 1000Mbps
          [4] 1000Mbps     [5] 1000Mbps    [6] 1000Mbps    [7] 1000Mbps
      Interface Switching Capability Descriptor(1):
        Switching type: Packet
        Encoding type: Packet
        Maximum LSP BW [priority] bps:
          [0] 1000Mbps     [1] 1000Mbps    [2] 1000Mbps    [3] 1000Mbps
          [4] 1000Mbps     [5] 1000Mbps    [6] 1000Mbps    [7] 1000Mbps
```

Each router maintains network link attributes and topology information in a specialized TED. The TED is used exclusively for calculating explicit paths for the placement of LSPs across the physical topology. A separate database is maintained so that the subsequent traffic engineering computation is independent of the IGP and the IGP's link-state database. Meanwhile, the IGP continues its operation without modification, performing the traditional shortest-path calculation based on information contained in the router's link-state database. There is only one TED, and it can be populated only from the default routing instance.

The TED shows the total number of IS-IS nodes and inet nodes. Each broadcast domain DATA LINK LAYER generates a pseudonode to represent the network. The portion of the TED shown represents a node: the type field indicates `Rtr` (router). It could also indicate `Net` (network) if it were a pseudonode. The node has two input and output links running OSPF Area 0 (only one link is shown in the graphic). One link leads to a router with the IP address of 172.22.220.2.

The TED also includes detailed traffic engineering information for each link. This information includes administrative groups, metrics, static bandwidth, reservable bandwidth, and available bandwidth by priority level.

The `Local:` and `Remote:` fields in the **`show ted database extensive`** command output specify IP address information about the link. Four different combinations of `Local:` and `Remote:` values are possible. If both fields contain nonzero IP addresses, the link is a point-to-point link. If the both fields are `0.0.0.0`, the link represents a pseudonode. If only the `Remote:` value is `0.0.0.0`, the link is a LAN interface. Finally, if only the `Local:` value is `0.0.0.0`, the link is an unnumbered interface.



User-Provided Constraints

- User-defined constraints influence path selection
  - Bandwidth requirements*
  - Hop count limitations (for fast reroute)
  - Administrative groups (colors)
  - Priority (setup and hold)*
  - Explicit route (strict or loose)*

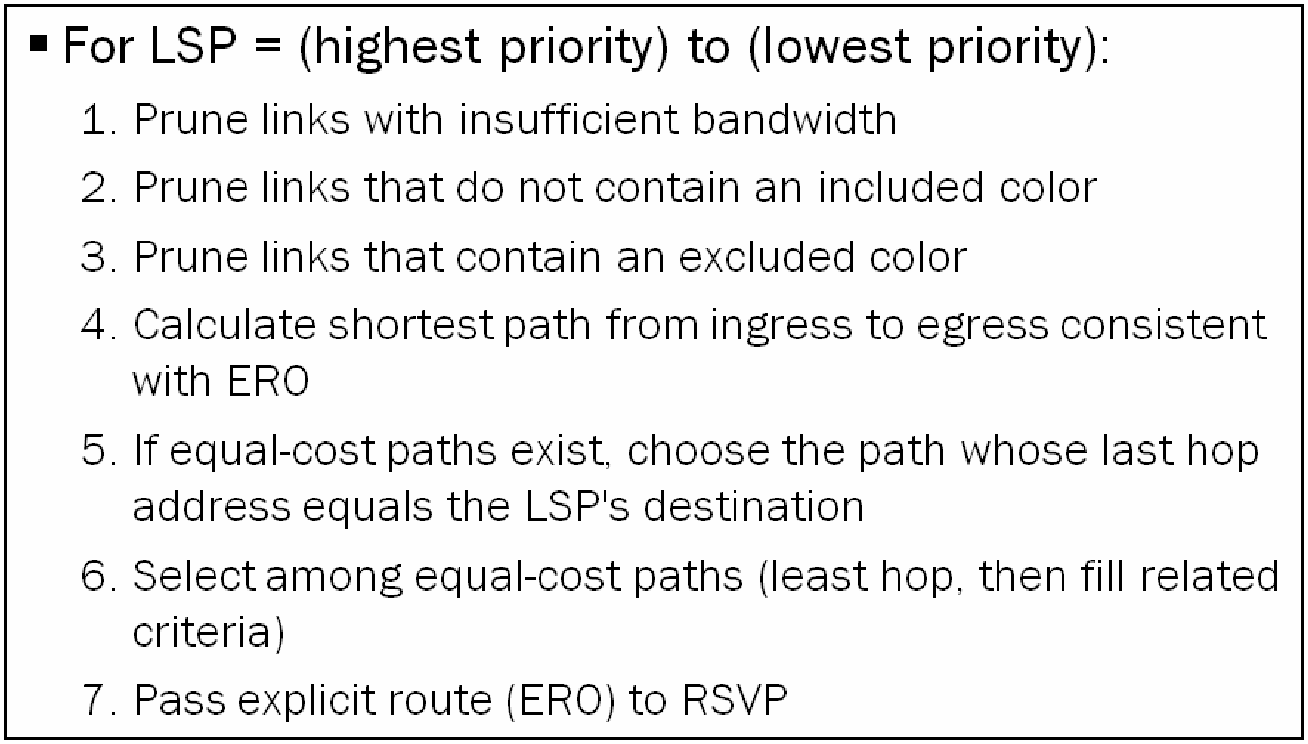
* Can also be specified for non-CSPF-signaled LSPs

User-Provided Constraints

You can influence the outcome of the CSPF path selection process by specifying one of more of the following constraints when defining an RSVP-signaled LSP:

- *Bandwidth*: The bandwidth to reserve for this LSP. The reserved bandwidth is calculated against each link's available bandwidth. The available bandwidth is the bandwidth remaining after the subscription factor is applied to the link and all existing link subscriptions are removed.
- *Hop count*: The maximum number of hops to extend the path to bypass the next downstream node when creating a fast-reroute detour.
- *Link color*: Administrative groups, also known as link coloring or resource class, are manually assigned attributes that describe the *color* of links, such that links with the same color conceptually belong to the same class. You can use administrative groups to implement a variety of policy-based LSP routing controls.
- *Priority*: Specifies the setup and hold priority for the LSP. New setup priorities are compared with existing hold values. When not enough bandwidth is available to satisfy all LSPs concurrently, a given link is considered in the path only when the new LSP's setup priority is stronger than the hold priority of existing LSPs.
- *EROs*: Both CSPF and non-CSPF LSPs can be constrained with one or more ERO routing directives.

How CSPF Selects a Path

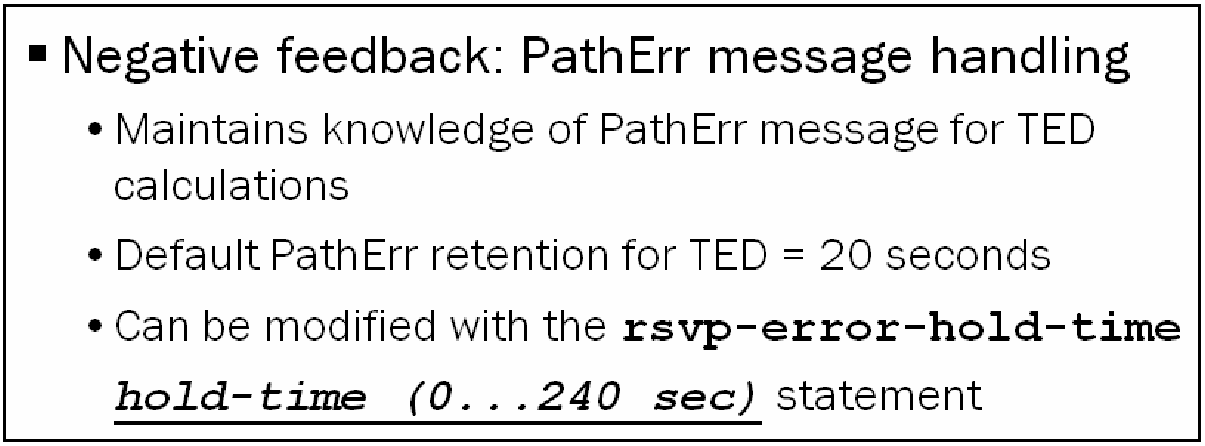


- For LSP = (highest priority) to (lowest priority):
  1. Prune links with insufficient bandwidth
  2. Prune links that do not contain an included color
  3. Prune links that contain an excluded color
  4. Calculate shortest path from ingress to egress consistent with ERO
  5. If equal-cost paths exist, choose the path whose last hop address equals the LSP's destination
  6. Select among equal-cost paths (least hop, then fill related criteria)
  7. Pass explicit route (ERO) to RSVP

The CSPF algorithm computes the path of LSPs one at a time, beginning with the highest-priority LSP (the one with the numerically lowest setup priority value). We cover LSP priority settings in the next chapter. Among LSPs of equal priority, CSPF begins with those that have the highest bandwidth requirement. For each such LSP, the following sequence is executed:

1. Prune the topology database (TED) of all the links that are not full duplex and do not have sufficient reservable bandwidth.
2. If the LSP configuration contains an `include-any` statement, prune all links that do not have at least one of the included colors assigned, including those links with no color assigned. If the LSP configuration contains an `include-all` statement, prune all links that do not have all of the included colors assigned.
3. If the LSP configuration contains an `exclude` statement, prune all links that contain excluded colors; links with no color are not pruned.
4. Find the shortest path towards the LSP's egress router, taking into account ERO constraints. For example, if the path must pass through Router A, two separate SPFs are computed, one from the ingress router to Router A, the other from Router A to the egress router.
5. If several paths have equal cost, choose the one whose last hop address is the same as the LSP's destination.
6. If several equal-cost paths remain, select the one with the fewest number of hops. If equal-cost paths still remain, apply the CSPF load-balancing rule configured on the LSP (least fill, most fill, or random).
7. When a path is chosen, pass the complete ERO list to RSVP for signaling.

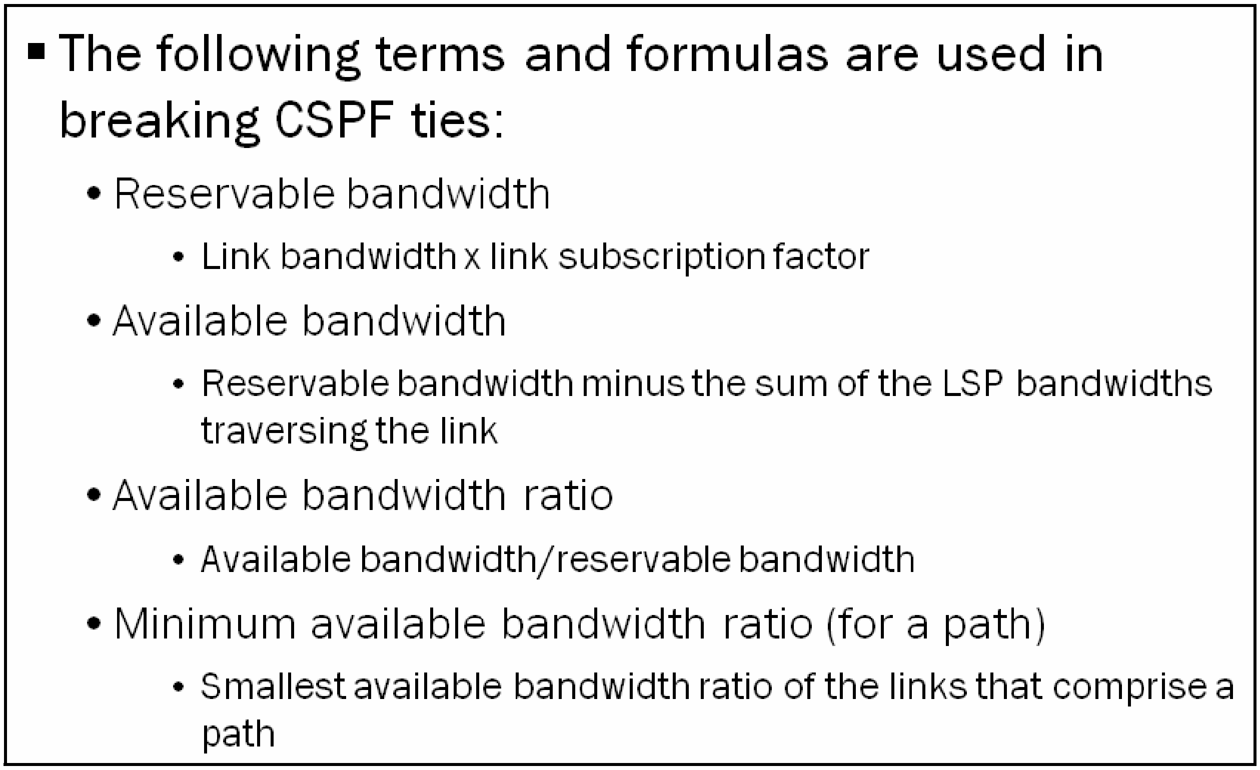
Negative Feedback



- Negative feedback: PathErr message handling
  - Maintains knowledge of PathErr message for TED calculations
  - Default PathErr retention for TED = 20 seconds
  - Can be modified with the **`rsvp-error-hold-time`** ***`hold-time (0...240 sec)`*** statement

Junos OS automatically retains knowledge of RSVP PathErr messages for a short period of time. This knowledge prevents the TED from resignaling in the same direction that caused the original error. By default, the system maintains knowledge of PathErr messages for 20 seconds, configurable from 0 to 240 seconds in the `[edit protocols mpls]` hierarchy with the **`rsvp-error-hold-time`** command.

CSPF Tie-Breaking Terms



- The following terms and formulas are used in breaking CSPF ties:
  - Reservable bandwidth
    - Link bandwidth x link subscription factor
  - Available bandwidth
    - Reservable bandwidth minus the sum of the LSP bandwidths traversing the link
  - Available bandwidth ratio
    - Available bandwidth/reservable bandwidth
  - Minimum available bandwidth ratio (for a path)
    - Smallest available bandwidth ratio of the links that comprise a path

The terms and formulas shown on the graphic are used by the CSPF algorithm to break CSPF ties. You should familiarize yourself with these terms and formulas to understand the various CSPF tie-breaking behaviors available in Junos OS. We explain these terms on subsequent pages.

Random

If more than one path is available after running the CSPF algorithm, a tie-breaking rule is applied to choose the path for the LSP. Three tie-breaking rules are available: *random*, *least fill*, and *most fill*. The actual rule used depends on the specifics of your configuration. The default tie-breaking method is random, which, as you might surmise, chooses one of the qualifying paths at random. This rule tends to place an equal number of LSPs on each link, regardless of the available bandwidth.

Least Fill

The least-fill option chooses the path with the largest minimum available bandwidth ratio. This rule tries to equalize the reservation levels on each link. This form of load balancing might be preferred when the goal is to minimize the total number of LSPs that are disrupted when a link failure occurs.

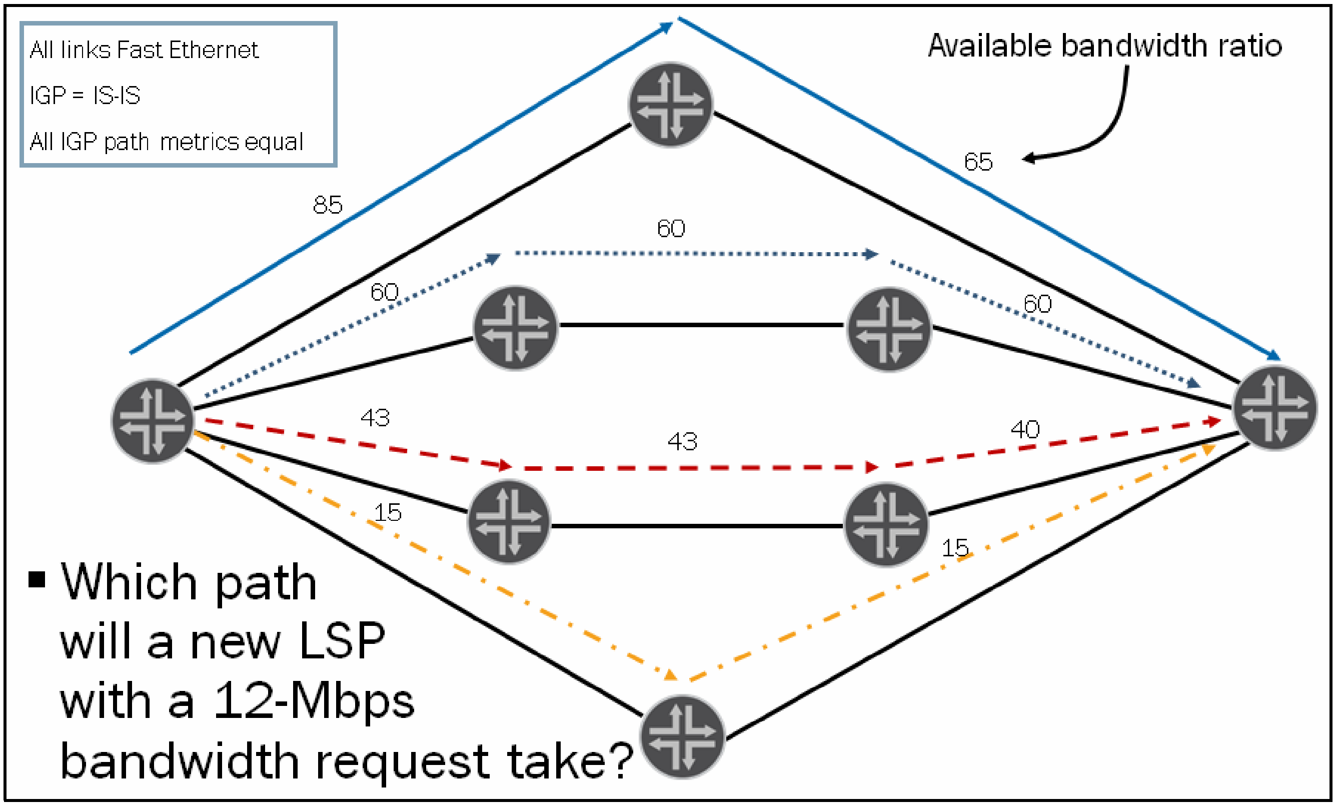
Most Fill

The most-fill option prefers the path with the smallest minimum available bandwidth ratio. This rule tries to fill a link before moving traffic to alternative link and might be preferred in certain usage-based billing environments where bulk discounts are gained by consolidating as much traffic onto as few links as possible. The most-fill option tends to fully pack your lower bandwidth links first, such that your highest bandwidth links remain available for LSPs with large bandwidth requirements, which is another possible motivation for using this type of load balancing.

Configuration

To explicitly configure a tie-breaking behavior, include the **`random`**, **`least-fill`**, or **`most-fill`** statement at the `[edit protocols mpls label-switched-path` *`path-name`*`]` hierarchy level. Note that you do not have to explicitly configure random load balancing as this is the default.

Analysis of Least-Fill Operation


All links Fast Ethernet

IGP = IS-IS

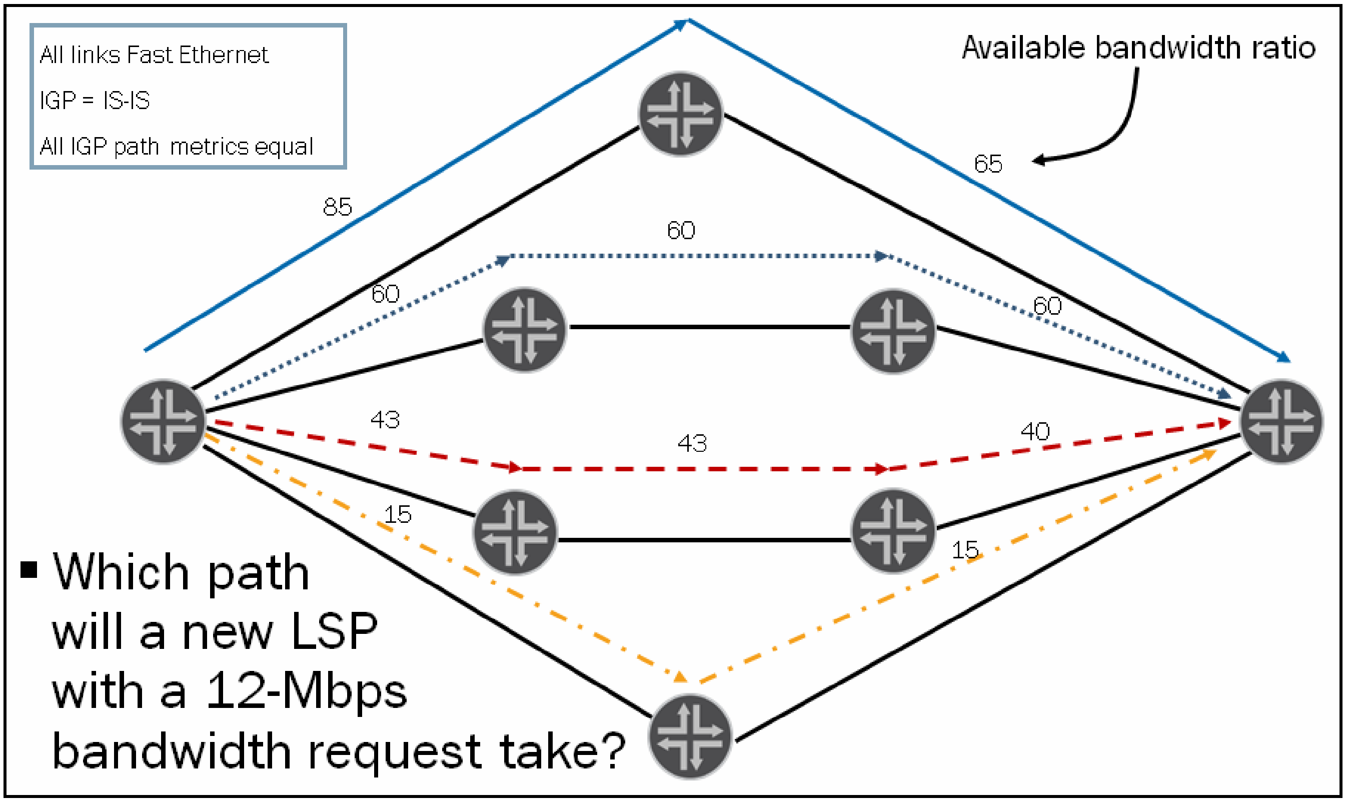
All IGP path metrics equal

- Which path will a new LSP with a 12-Mbps bandwidth request take?

Because all four paths shown in the example on the graphic have equal metric costs (note that the information provided on the graphic indicates that default IS-IS metrics are *not* in use), select the path that has the most available bandwidth. (Remember that you are selecting the path that is least full on a percentage basis; therefore, it has the largest available bandwidth.)

Amongst the two lower hop count links, the top path has the largest minimum available bandwidth ratio. Least fill is desirable when you want to smooth out the overall bandwidth that is available on all your links.

Analysis of Most-Fill Operation

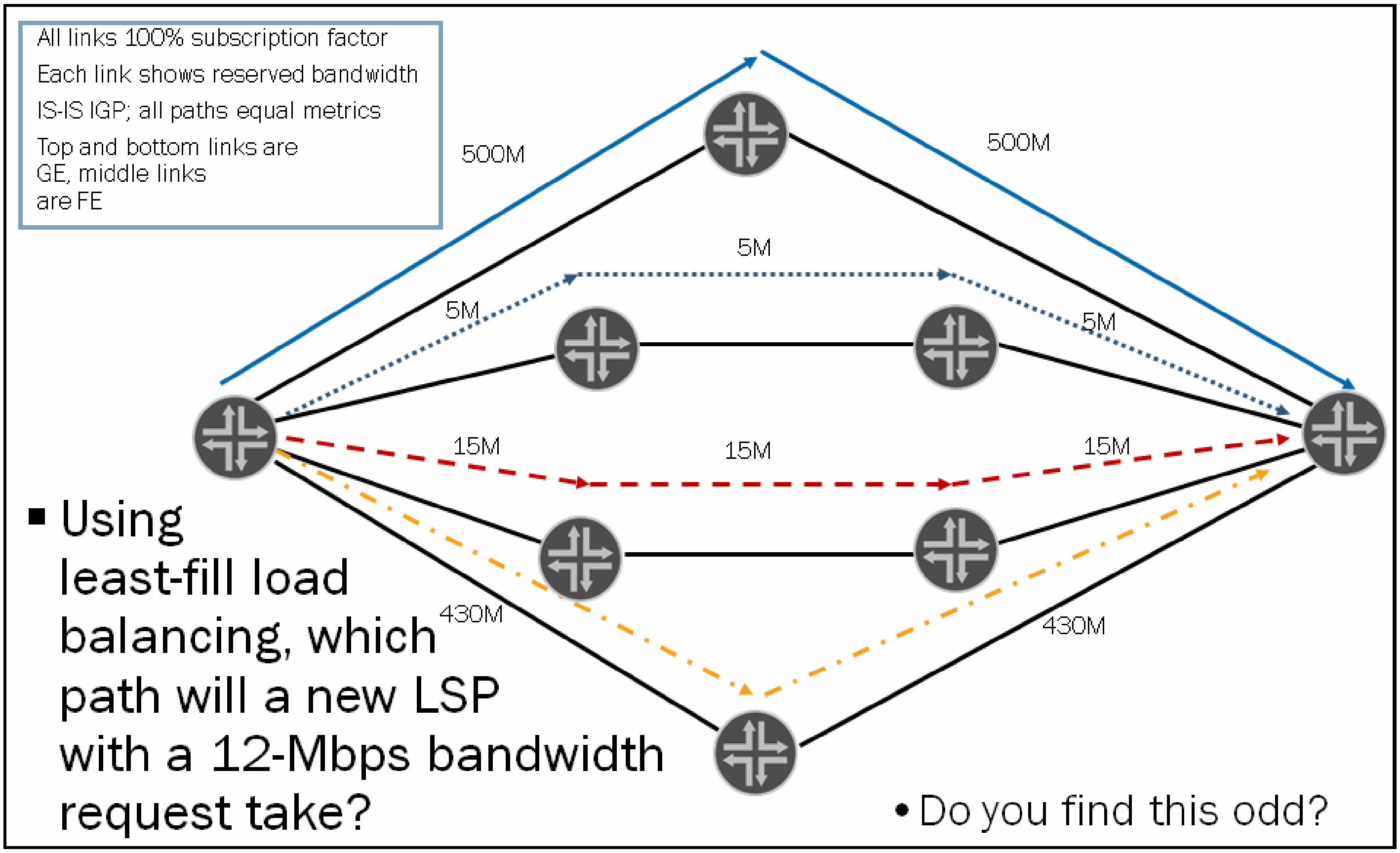


- Which path will a new LSP with a 12-Mbps bandwidth request take?

Because all four paths shown in the example on the graphic have equal metric costs (note again that default IS-IS metrics are *not* in use), select the path that has the least available bandwidth ratio. (Remember that you are selecting the path that is most full on a percentage basis; therefore, it has the smallest available bandwidth.) Amongst the two lower hop count links, the bottom path has the smallest minimum available bandwidth ratio.

You might configure most fill load balancing when you want to fully pack your lower bandwidth links first so that your higher bandwidth links remain available for LSPs with large bandwidth requirements.

An Interesting Question



All links 100% subscription factor
Each link shows reserved bandwidth
IS-IS IGP; all paths equal metrics
Top and bottom links are GE, middle links are FE

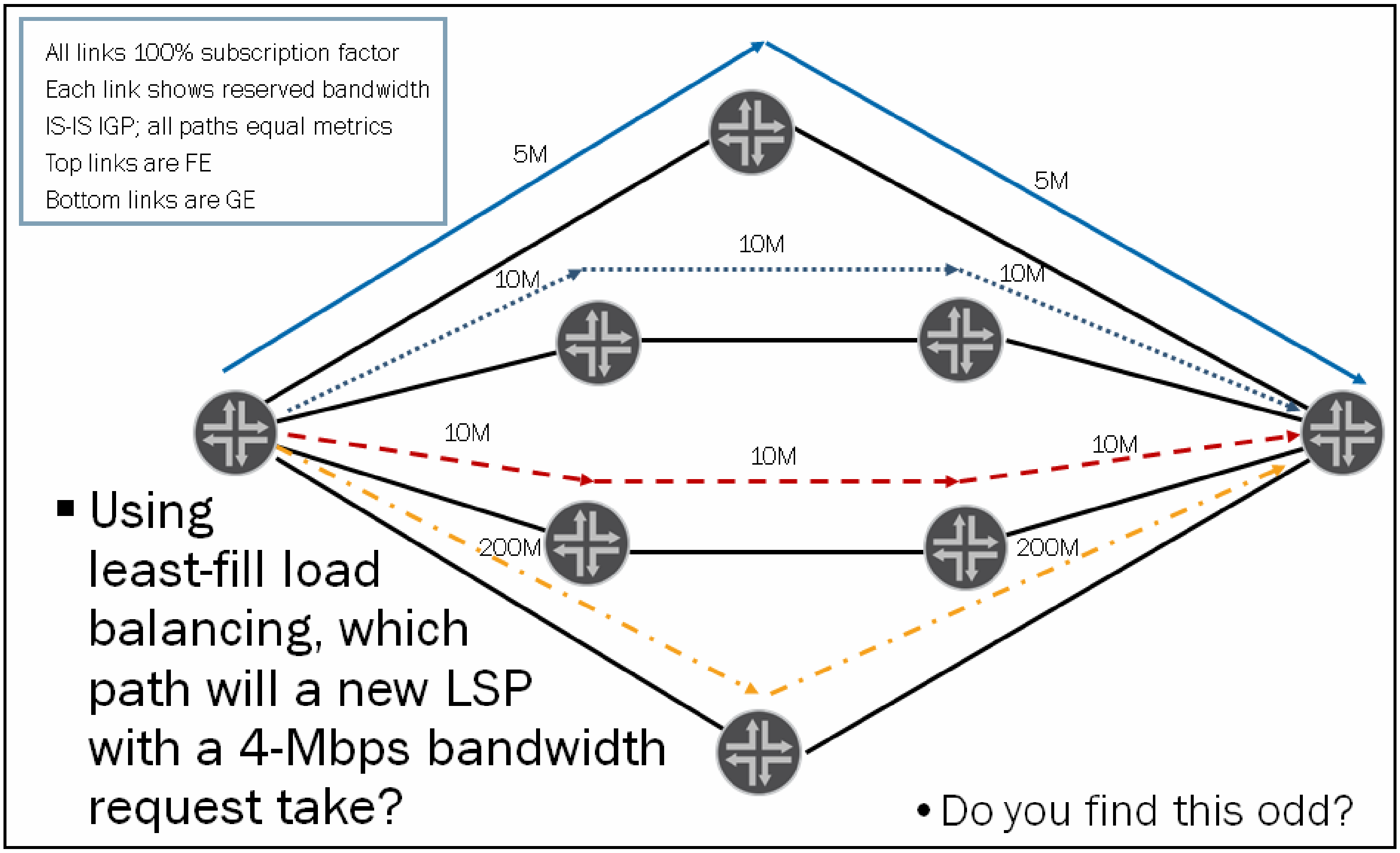
- Using least-fill load balancing, which path will a new LSP with a 12-Mbps bandwidth request take?
- Do you find this odd?

Step 6 of the CSPF algorithm, as explained on a previous page, indicates that if no decision is reached after the router processes the first five steps of the algorithm, the paths with the smaller hop count are selected. When multiple paths remain, the tie-breaking algorithm moves on to consider fill-related criteria.

Because least fill looks for the largest available bandwidth ratio, you might expect the middle links to be selected because they have 95 and 85 available bandwidth ratios. However, these links are not chosen because they have more hops. The two outer paths have their available bandwidth ratios compared because they have lower (and equal) hop counts. Therefore, the bottom path, which has an available bandwidth ratio of 57 ((1000-430)/1000) is selected over the top link, which has an available bandwidth ratio of 50 ((1000-500)/1000).

Note that the same logic is followed regardless of the actual bandwidth available on a link. For example, if the outer paths were Fast Ethernet links with lower available bandwidth ratios than the two middle paths—which could be Gigabit Ethernet with really high available bandwidth ratios—the CSPF hop count rule will still eliminate the middle links due to their higher hop-counts. Even if the Gigabit Ethernet links have huge amounts of percentage or actual bandwidth available, the hop count rule holds sway over what paths are considered *equal*, and therefore subject to a load-balancing decision.

Another Interesting Question



All links 100% subscription factor
Each link shows reserved bandwidth
IS-IS IGP; all paths equal metrics
Top links are FE
Bottom links are GE

- Using least-fill load balancing, which path will a new LSP with a 4-Mbps bandwidth request take?

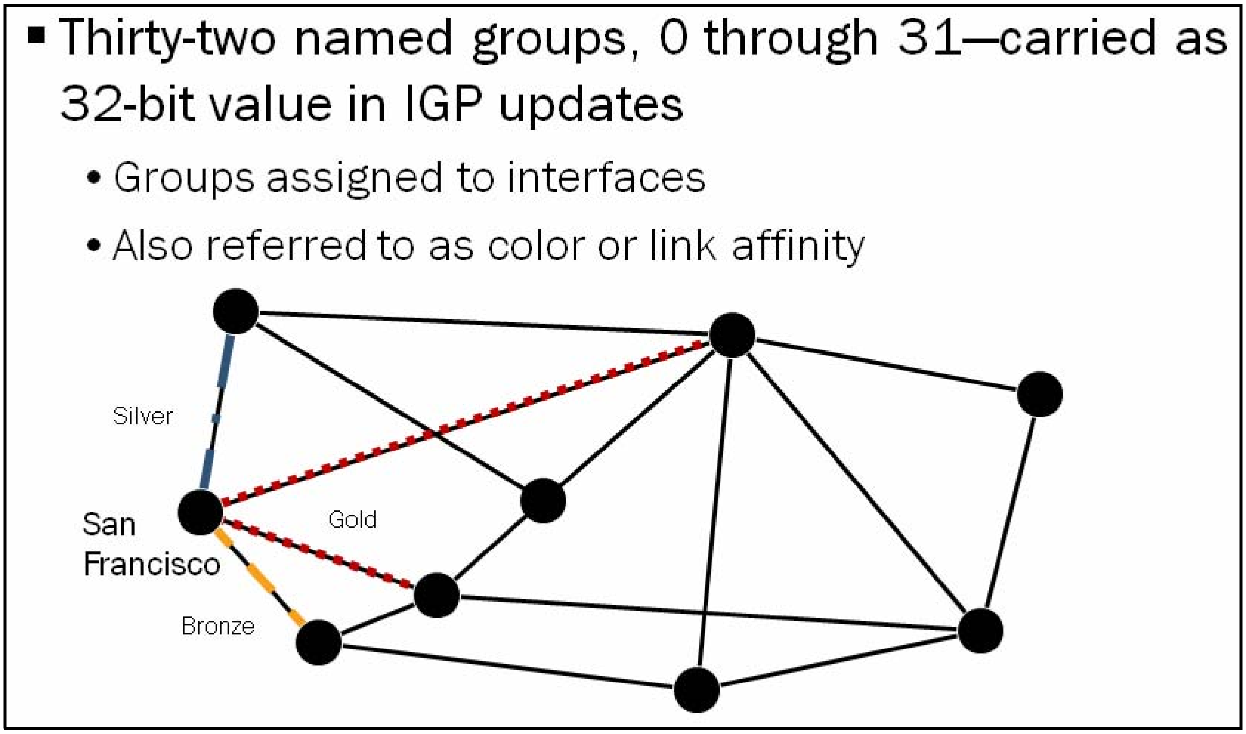
- Do you find this odd?

Because all four paths have equal metric costs in the example on the graphic, you must first look at hop counts, and then available bandwidth ratios, to find the correct answer to this riddle.

The middle two paths have available bandwidth ratios of 90 and 99, the top link has an available bandwidth ratio of 95, and the bottom link has an available bandwidth ratio of 80. In this case, the outer two links have lower hop counts, and therefore, only these links factor into available bandwidth ratio comparisons. The top path has the larger available bandwidth ratio, which causes it to be selected by the least fill algorithm.

The point here is that the Fast Ethernet link has more bandwidth percentage available than the Gigabit Ethernet link in this example. Therefore, even though the total bandwidth available on the Gigabit Ethernet is much greater than that of a Fast Ethernet interface, the LSP will be routed over the Fast Ethernet links based on the bandwidth percentages.

Administrative Group Overview



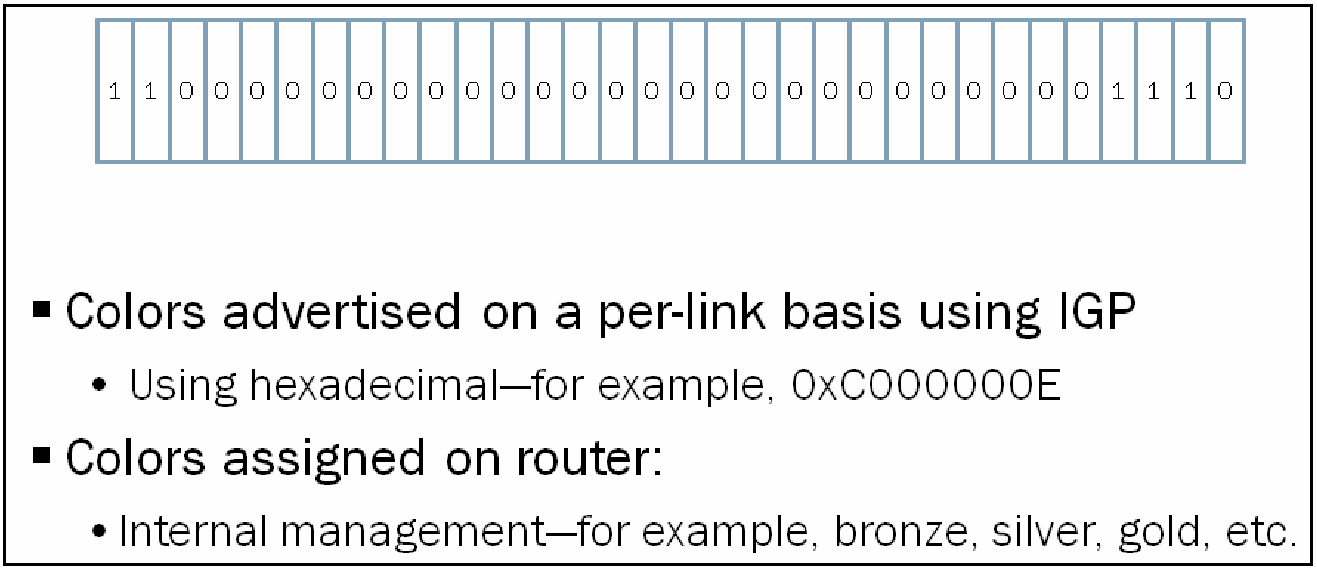
- Thirty-two named groups, 0 through 31—carried as 32-bit value in IGP updates
  - Groups assigned to interfaces
  - Also referred to as color or link affinity

Administrative groups allow you to constrain the routing of an LSP to the set of links that meet the prescribed administrative groupings. Each interface can support 32 different administrative groups. The administrative groups associated with each interface is communicated through the extended IGP for storage in the TED. When the ingress router performs a CSPF computation, it includes or excludes links based on their associated colors, as specified in the LSP's definition. The net result is that the routing of the LSP will be controlled by its need to avoid, or make use of, links with the specified colors.

If you use administrative groups, you must configure them identically on all routers participating in an MPLS domain. Great confusion results when a pair of routers do not agree on the color associated with mutually attached link. You can assign more than one administrative group to each physical link, or you might opt to leave one or more links *uncolored* by not assigning any administrative group values.

IGP Advertisements

A traffic engineering aware IGP communicates the administrative group of each interface as a 32-bit (4 bytes) bit mask. Each of the bit values in 32-bit sequence represents a different administrative group.



1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

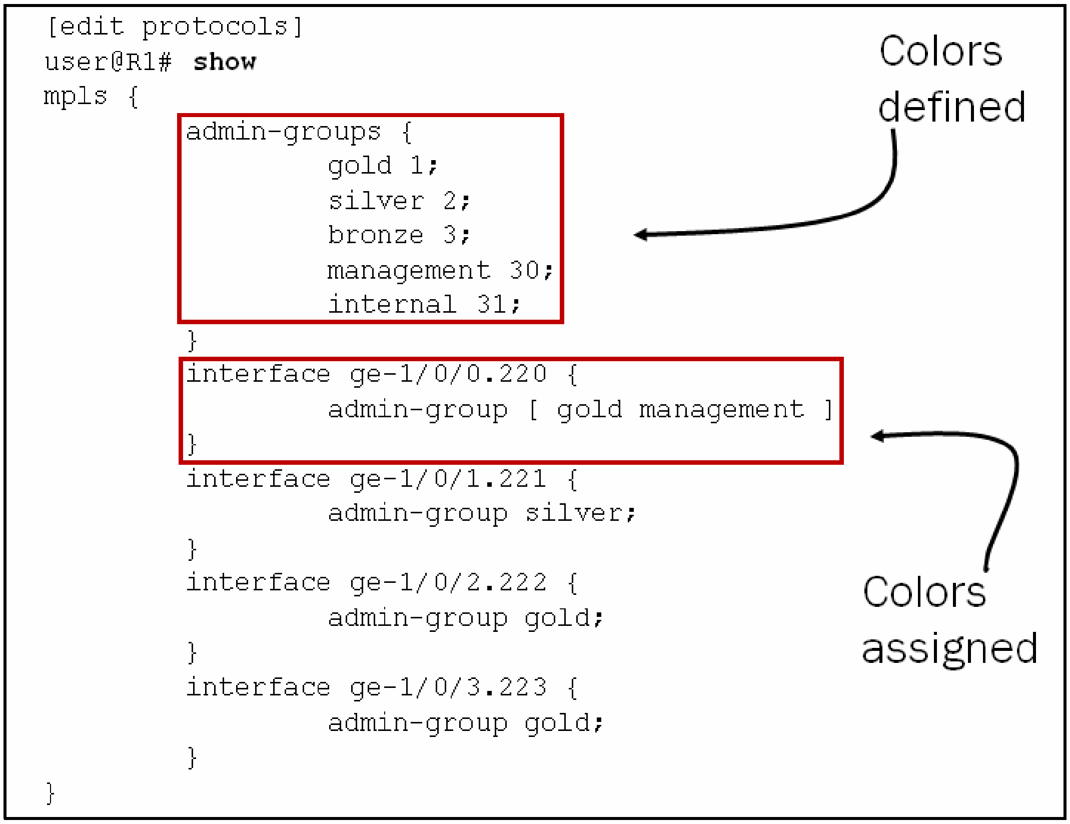
- Colors advertised on a per-link basis using IGP
  - Using hexadecimal—for example, 0xC000000E
- Colors assigned on router:
  - Internal management—for example, bronze, silver, gold, etc.

Color Assignments

Each bit value is correlated through configuration to a human-friendly name within Junos OS. This capability helps to simplify router management, as the name *silver* often means more to the typical human than the hexadecimal value of 0x02, for example.

These names are often assigned as colors, but they do not have to be a color; they can be any descriptive term you want. Each link can have one or more bits enabled, and can therefore be associated with one or more colors simultaneously. The colors advertised by each link are displayed in both hexadecimal and in symbolic form in the output of **`show ted database extensive`** command. When multiple bits are set in the TED, the order of the colors displayed correlates to the order of the bits that are set. When no colors are assigned, the 32 bits default to all zeros (0x00000000).

Configuring Administrative Groups



```
[edit protocols]
user@R1# show
mpls {
        admin-groups {
                gold 1;
                silver 2;
                bronze 3;
                management 30;
                internal 31;
        }
        interface ge-1/0/0.220 {
                admin-group [ gold management ]
        }
        interface ge-1/0/1.221 {
                admin-group silver;
        }
        interface ge-1/0/2.222 {
                admin-group gold;
        }
        interface ge-1/0/3.223 {
                admin-group gold;
        }
}
```

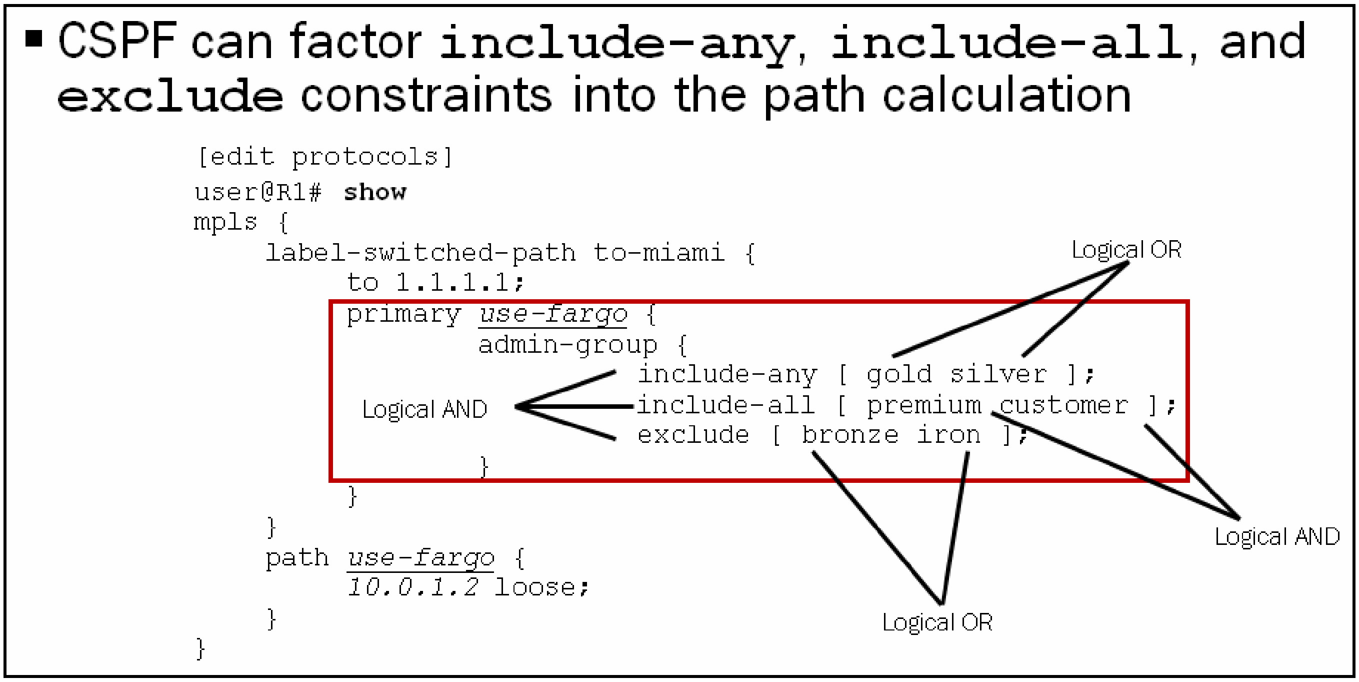
You configure administrative groups under the `[edit protocols mpls]` hierarchy by defining the group names and their associated bit values. The bit values can range from 0 to 31. Note that the actual group name is for local reference only, which means that the exact spelling and case need not be identical between all routers in the traffic engineering domain.

You should configure all defined groups that are in use within the traffic engineering domain on all routers, even though a particular group might not be assigned to any interfaces on every router. This configuration is necessary because undefined administrative groups referenced in a LSP definition prevent your candidate configuration from committing.

After defining all groups, you next associate one or more groups with each router interface. You reference the symbolic name that is associated with each group when assigning groups to your interfaces. Recall that you can configure multiple group names on a single interface.

In this example, the configuration for interface ge-1/0/0.220 is not a typographical error. The two administrative groups assigned to this interface are, in fact, *`gold`* and *`management`*.

Using Administrative Groups



- CSPF can factor **include-any**, **include-all**, and **exclude** constraints into the path calculation

```
[edit protocols]
user@R1# show
mpls {
    label-switched-path to-miami {
        to 1.1.1.1;
        primary use-fargo {
                admin-group {
                        include-any [ gold silver ];
                        include-all [ premium customer ];
                        exclude [ bronze iron ];
                }
        }
    }
    path use-fargo {
        10.0.1.2 loose;
    }
}
```

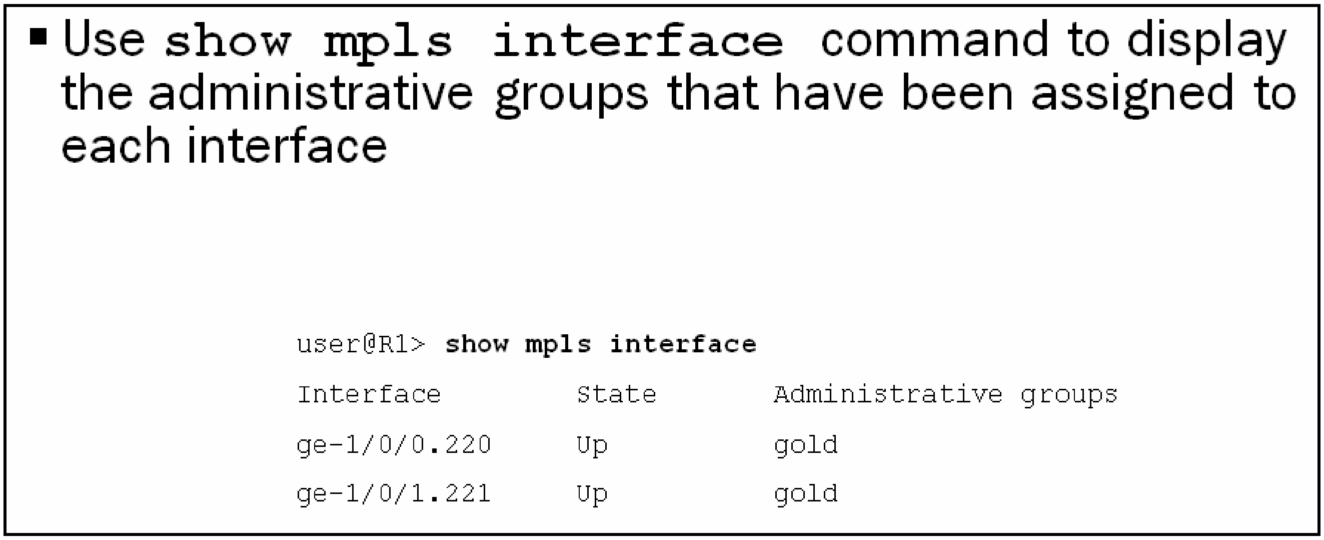
Logical OR
Logical AND
Logical AND
Logical OR

If you omit the include-any, include-all, and exclude statements, the LSP's path calculation proceeds unchanged using the default CSPF path selection criteria. When you configure an include-any list, only links that contain one or more of the specified administrative groups are included in the SPF calculation. In other words, a logical OR is performed on the administrative groups in the include-any statement. When you configure an include-all list, only links that contain all of the specified administrative groups are included in the SPF calculation. In other words, a logical AND is performed on the administrative groups in the include-all statement. Finally, when you configure an exclude list, links that contain any of the specified administrative groups present are automatically excluded from the SPF calculation. In other words, a logical OR is performed on the administrative groups in the exclude statement. Links that do not have an administrative group assigned are automatically disqualified by an include-any or include-all list; such uncolored links can be included in the SPF calculation when only exclude criteria is defined.

When you specify more than one include-any, include-all, and exclude lists for a given LSP, each link considered in the SPF calculation must comply with all lists. This behavior mimics the functionality of a logical AND.

Changing an LSP's administrative group causes an immediate recomputation of its routing, which might result in the LSP being rerouted.

Displaying Administrative Group Assignments

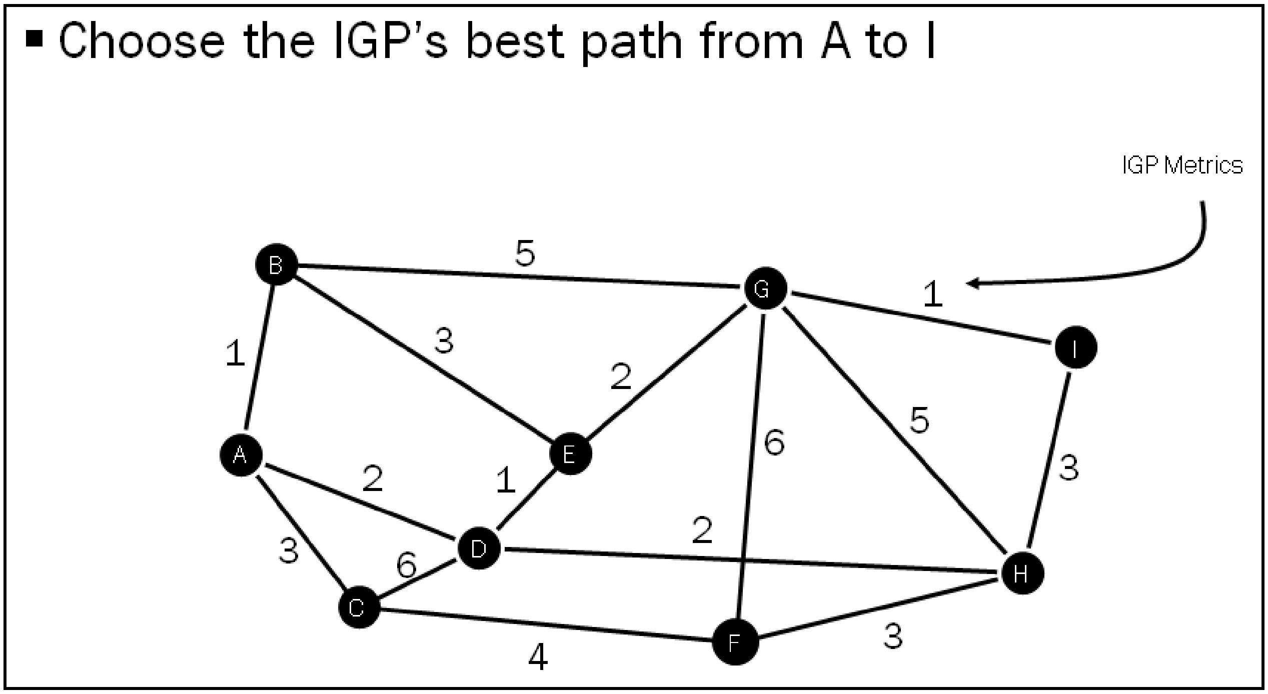


- Use `show mpls interface` command to display the administrative groups that have been assigned to each interface

```
user@R1> show mpls interface
Interface         State       Administrative groups
ge-1/0/0.220      Up          gold
ge-1/0/1.221      Up          gold
```

You can quickly verify the colors assigned to each MPLS-enabled interface using the **`show mpls interface`** command. This command also confirms whether the MPLS family is declared under the correct logical unit in the `[edit interfaces]` hierarchy. If the interface does not show up, the MPLS family is not defined for that interface.

Administrative Groups I: IGP Routing

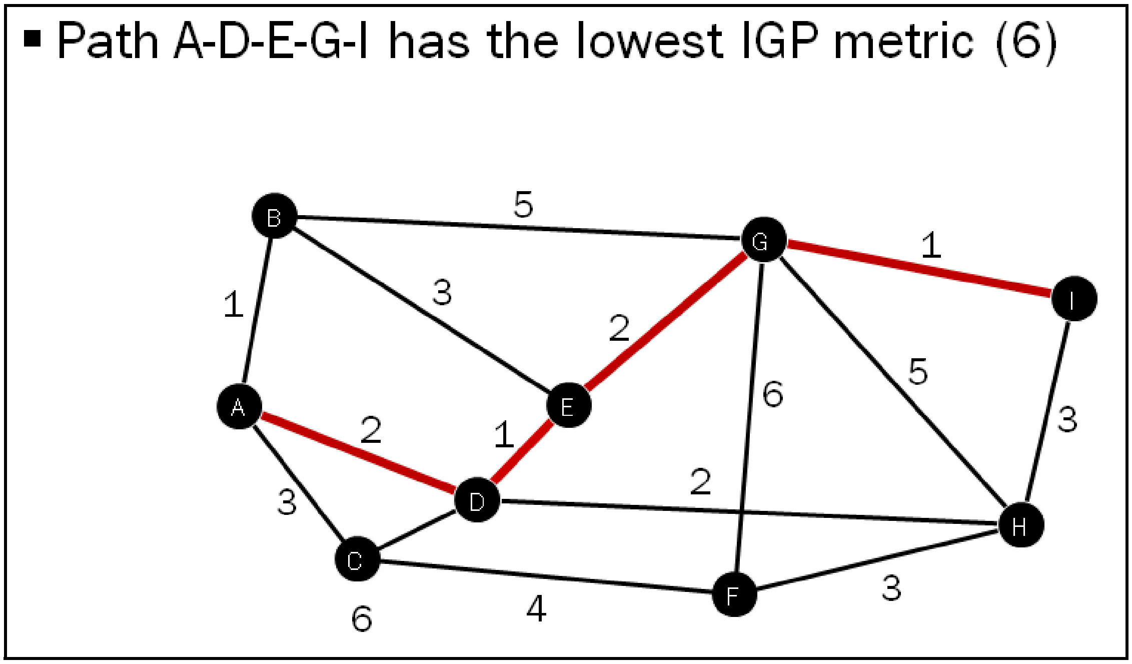


- Choose the IGP's best path from A to I

In this initial example, you must determine the shortest path from A to I according to the perspective of the IGP. Each link displays the associated IGP metric value. It should not take you long to determine that the IGP's shortest path from A to I is path A-D-E-G-I, with a total cost of 6.

This calculation reflects normal IGP processing and therefore, the default routing of an RSVP-signaled LSP. You can influence LSP routing with the inclusion of administrative constraints, as is demonstrated in subsequent pages in this section.

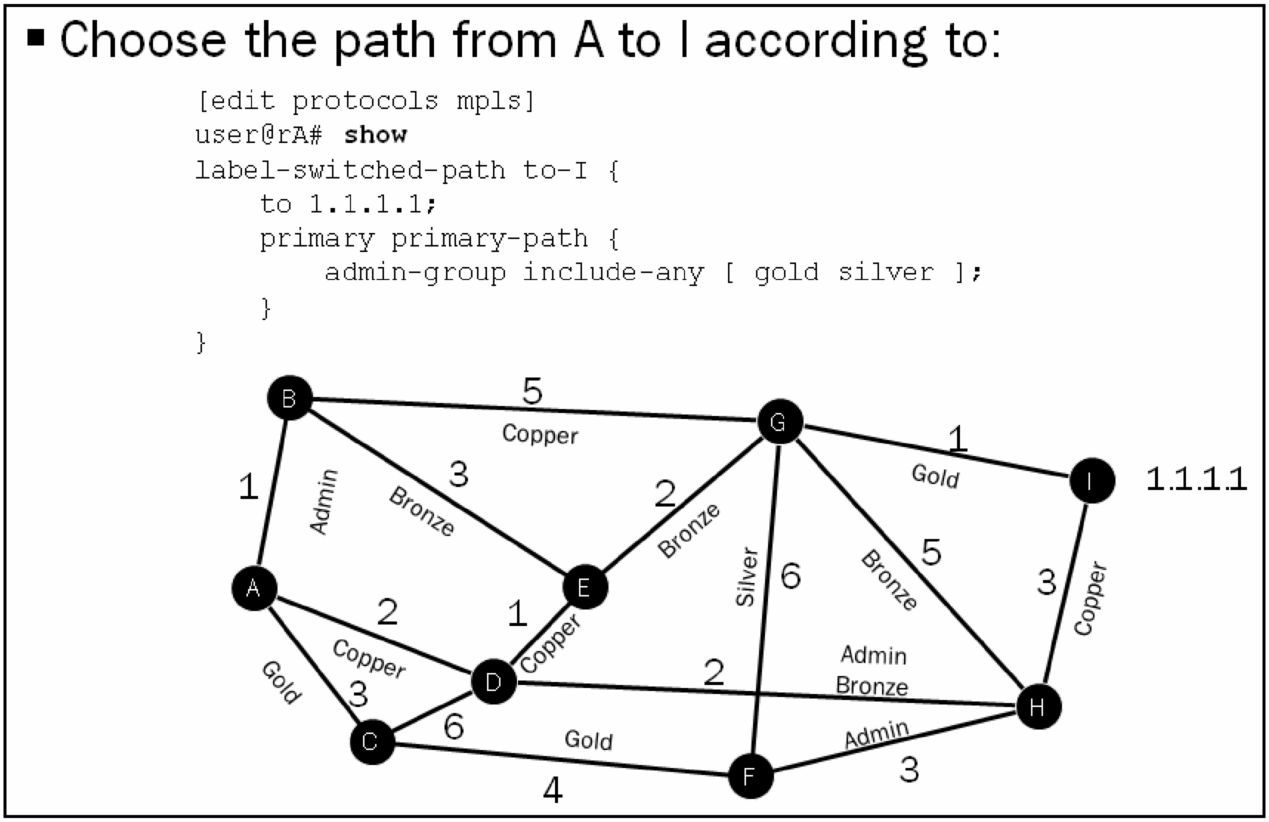
Administrative Groups I: The Solution



- Path A-D-E-G-I has the lowest IGP metric (6)

This graphic displays the solution to the question asked on the previous graphic. In this case, the IGP's shortest path has a metric of 6 and consists of the path A, D, E, G, and I.

Administrative Groups II: Include-Any Constraints



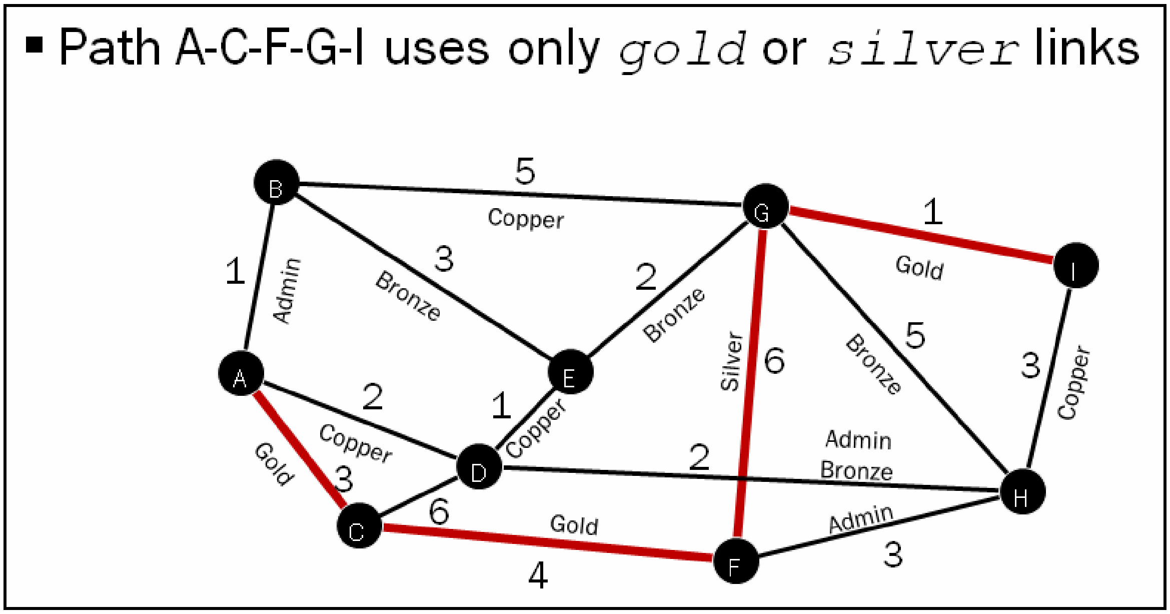
- Choose the path from A to I according to:

```
[edit protocols mpls]
user@rA# show
label-switched-path to-I {
    to 1.1.1.1;
    primary primary-path {
        admin-group include-any [ gold silver ];
    }
}
```

The LSP definition in this example requires that the link include either the color `gold` or the color `silver`. The CSPF algorithm begins by pruning the following links because they do not include the required colors: A-B, A-D, C-D, B-E, B-G, D-E, E-G, D-H, F-H, G-H, or H-I. The links that do comply with the constraints are A-C, C-F, F-G, and G-I.

A shortest path is computed from the links that remain, which in this case yields only one viable path. The only path available, given these constraints, is shown in the next graphic.

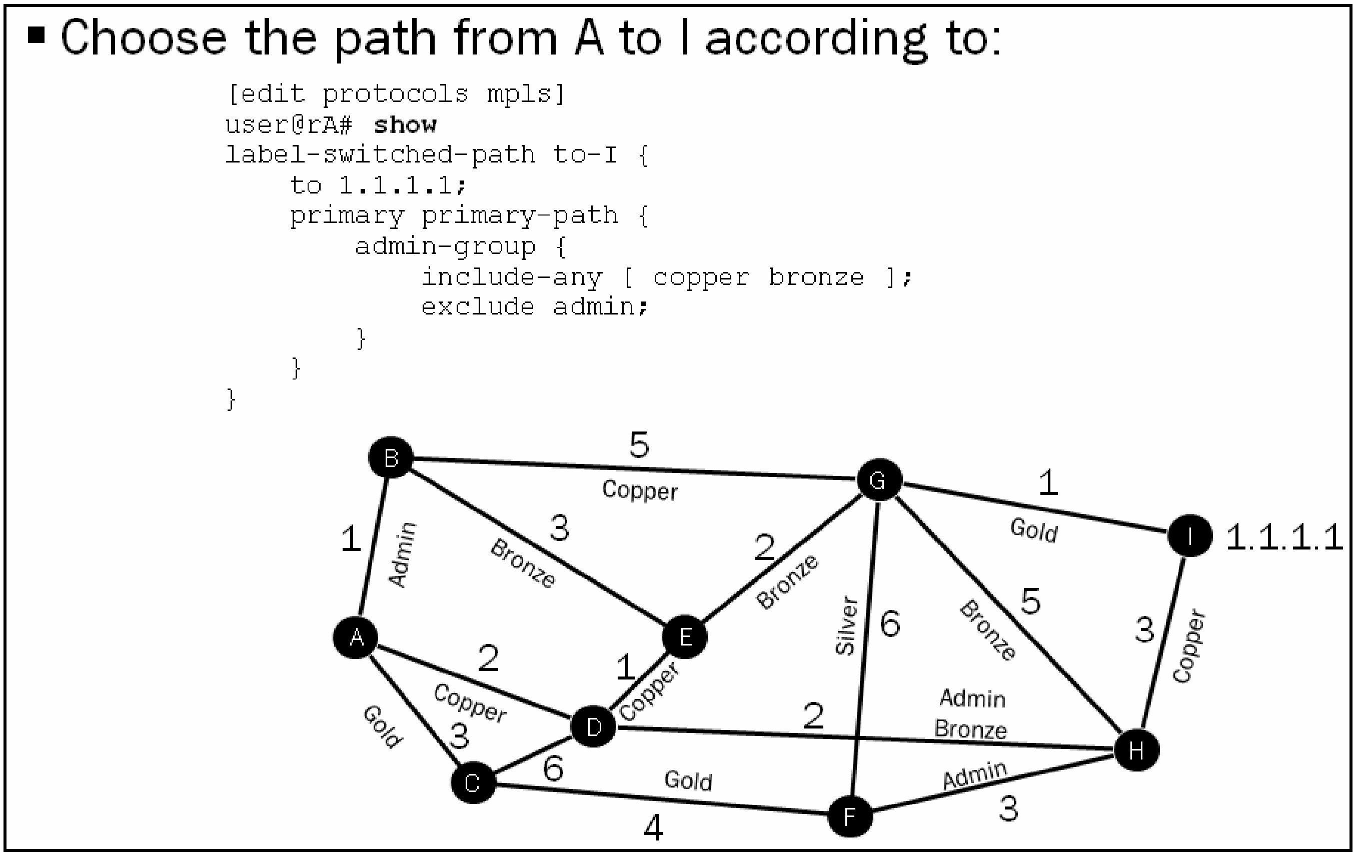
Administrative Groups II: The Solution



- Path A-C-F-G-I uses only `gold` or `silver` links

This graphic displays the solution to the question asked on the previous graphic. In this case, the only path meeting the provided `include-any` constraints consists of the path A, C, F, G, and I.

Administrative Groups III: Include-Any and Exclude Constraints



- Choose the path from A to I according to:

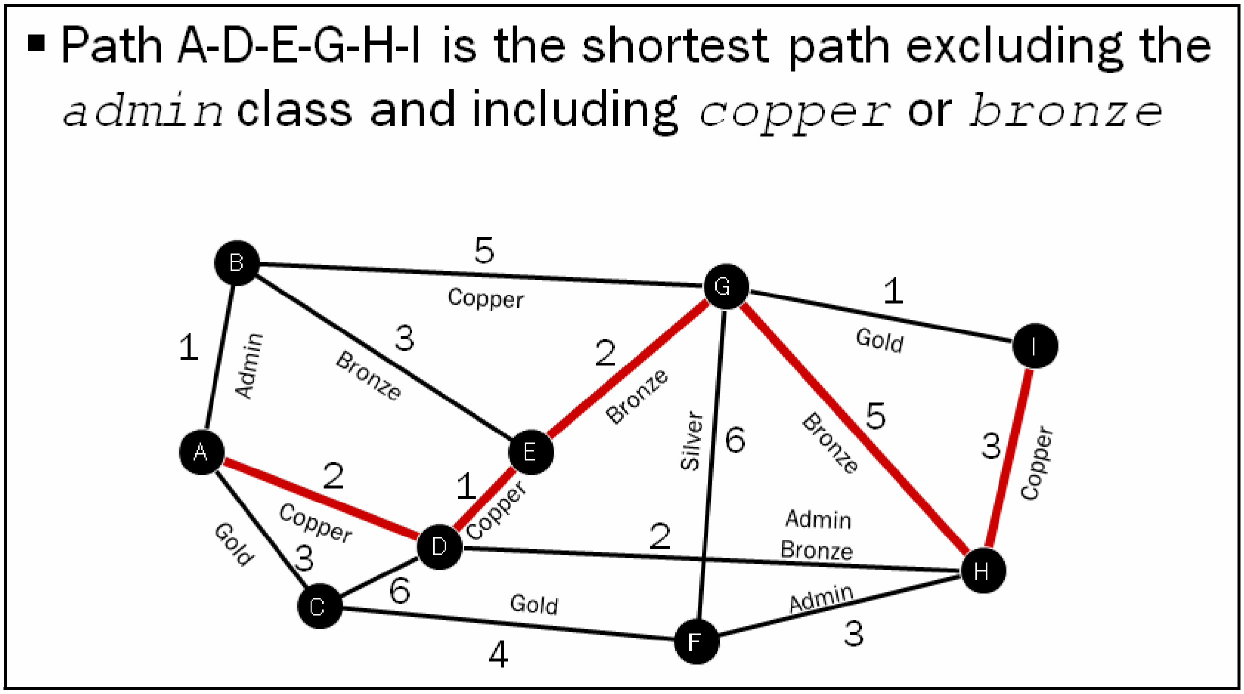
```
[edit protocols mpls]
user@rA# show
label-switched-path to-I {
    to 1.1.1.1;
    primary primary-path {
        admin-group {
            include-any [ copper bronze ];
            exclude admin;
        }
    }
}
```

The LSP definition in this case requires that the link include *either* the *`copper`* or *`bronze`* colors, while *also* excluding the *`admin`* color. Put another way, a qualifying link can include *`copper`* and exclude *`admin`*, or it can include *`bronze`* and exclude *`admin`*. The CSPF algorithm begins by pruning the following links because they do not include the required colors: A-C, A-B, C-F, C-D, F-G, F-H, and G-I.

The CSPF algorithm then prunes the following links because they are associated with an excluded color: D-H. Note that links A-B and F-H are already excluded by virtue of the `include-any` constraint but that link D-H passes the `include-any` constraint, and so it is not pruned until the `exclude` constraint is processed.

The links that pass both sets of constraints are A-D, D-E, E-B, B-G, E-G, G-H, and H-I. A shortest path is now computed from the compliant links. In this case two possible paths exist: A-D-E-B-G-H-I, with a cost of 19, and A-D-E-G-H-I, with a cost of 13. The CSPF algorithm selects the metrically shorter of the two paths, which results in the routing of the LSP over the path A-D-E-G-H-I.

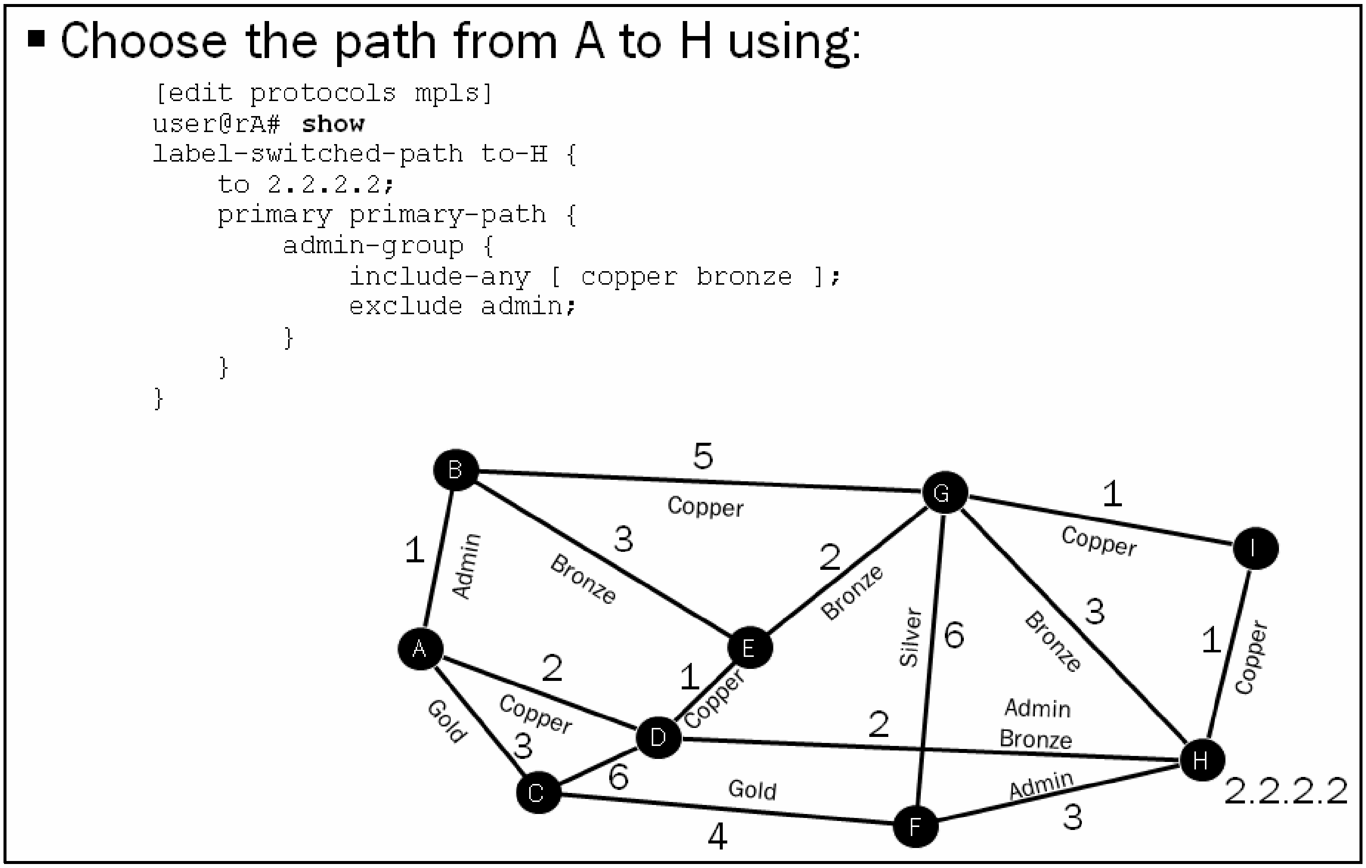
Administrative Groups III: The Solution



- Path A-D-E-G-H-I is the shortest path excluding the *admin* class and including *copper* or *bronze*

This graphic displays the solution to the question asked on the previous graphic. In this case, the metrically shorter of the two paths meeting both the `include-any` and `exclude` constraints consists of the path A, D, E, G, H, and I.

Administrative Groups IV: Include-Any and Exclude Constraints



- Choose the path from A to H using:

```
[edit protocols mpls]
user@rA# show
label-switched-path to-H {
    to 2.2.2.2;
    primary primary-path {
        admin-group {
            include-any [ copper bronze ];
            exclude admin;
        }
    }
}
```

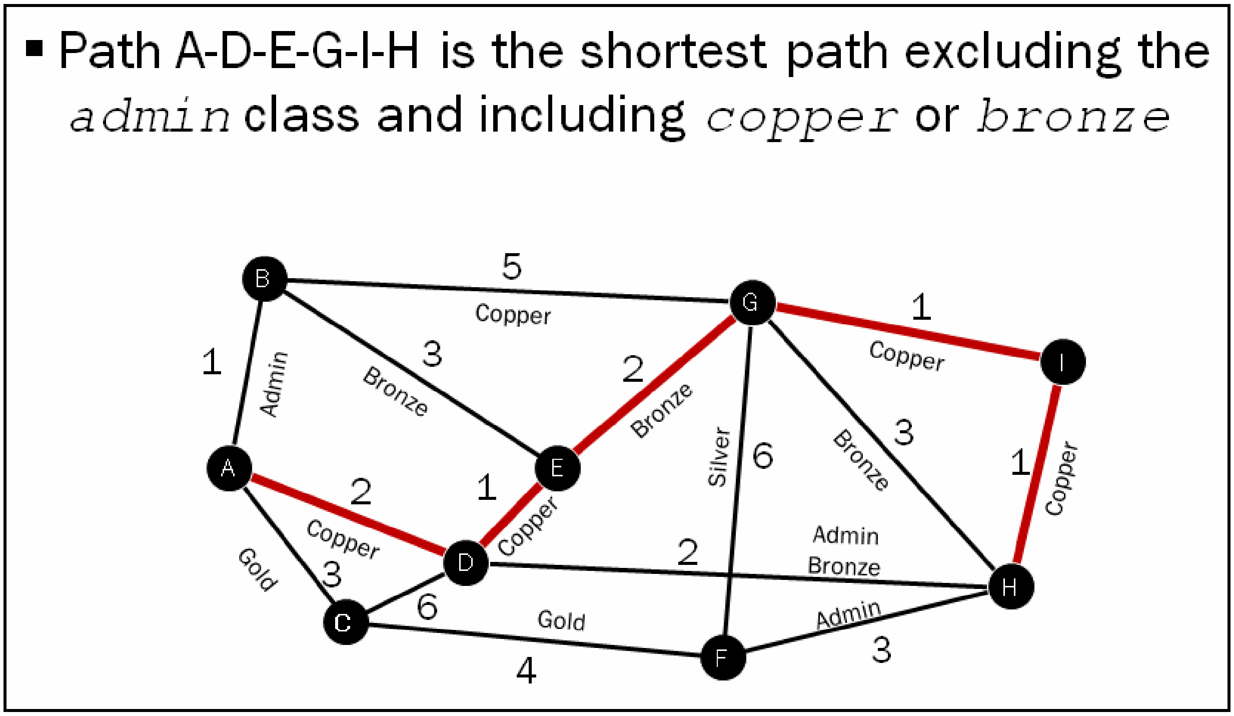
The LSP definition in this case once again requires that the link include *either* the `copper` or `bronze` colors, while *also* excluding the `admin` color. Note that the destination node and the cost of the G-H link has been changed from previous examples.

The CSPF algorithm begins by pruning the following links because they do not include the required colors: A-C, A-B, C-F, C-D, F-G, and F-H. The CSPF algorithm then prunes the following links because they are associated with the excluded color: D-H. Note that links A-B and F-H are already excluded by virtue of the `include-any` constraint but that link D-H passes the `include-any` constraint, and so it is not pruned until the `exclude` constraint is processed.

The links that pass both sets of constraints are A-D, D-E, E-B, B-G, E-G, G-I, G-H, and H-I. A shortest path is now computed from the set of compliant links. In this case, four possible paths exist: A-D-E-B-G-H (cost 14), A-D-E-G-H (cost 8), A-D-E-B-G-I-H (cost 13), and A-D-E-G-I-H (cost 7). The CSPF algorithm selects the metrically shorter of these paths, resulting in the routing of the LSP over the path A-D-E-G-I-H.

Note that the path chosen in this example has the lowest metric but not necessarily the lowest hop count. The metric variation in this example results from the fact that it is metrically closer to traverse both the G-I and I-H links (2) than it is to cross the G-H link directly (3).

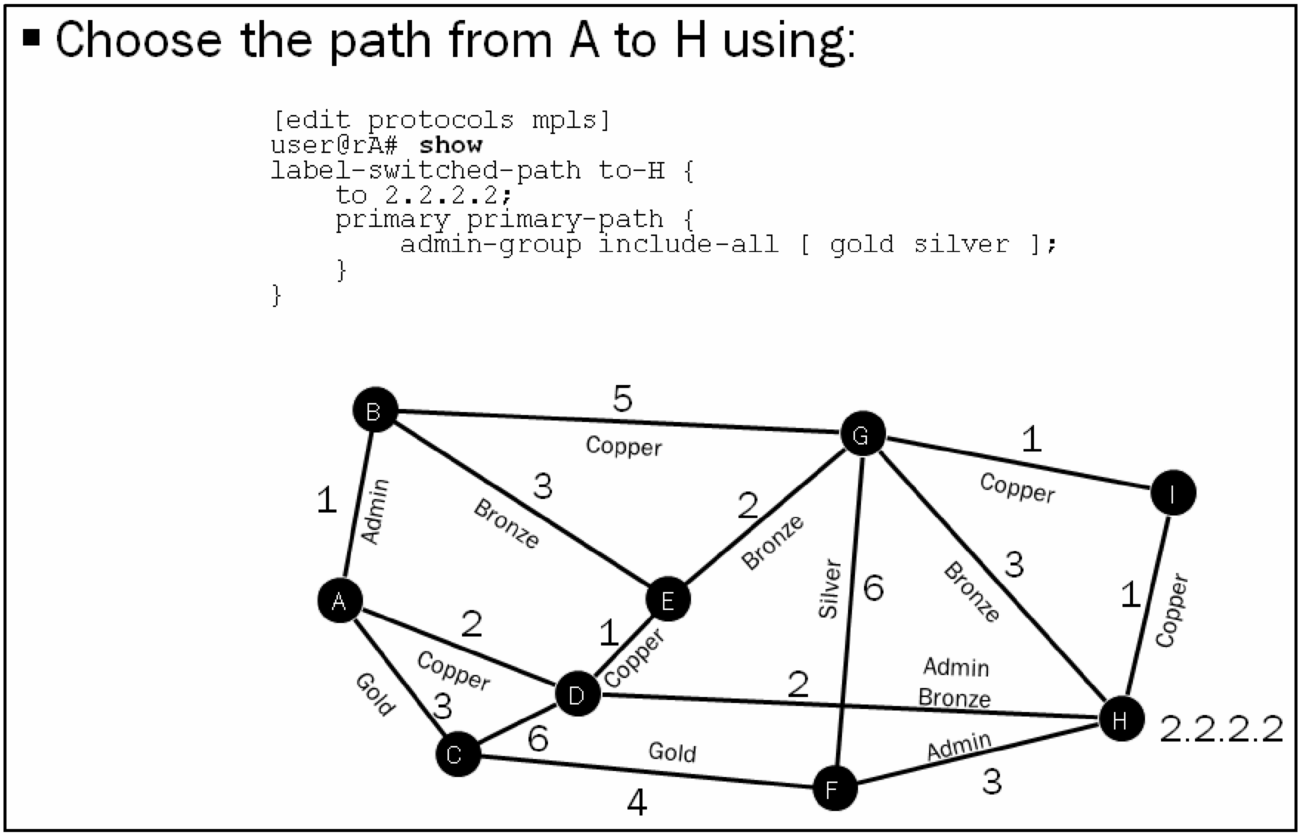
Administrative Groups IV: The Solution



- Path A-D-E-G-I-H is the shortest path excluding the *admin* class and including *copper* or *bronze*

This graphic displays the solution to the question asked on the previous graphic. In this case, the metrically shorter of the paths that meets *both* the `include-any` and `exclude` constraints consists of the path A, D, E, G, I, and H.

Administrative Groups V: Include-All Constraints



- Choose the path from A to H using:

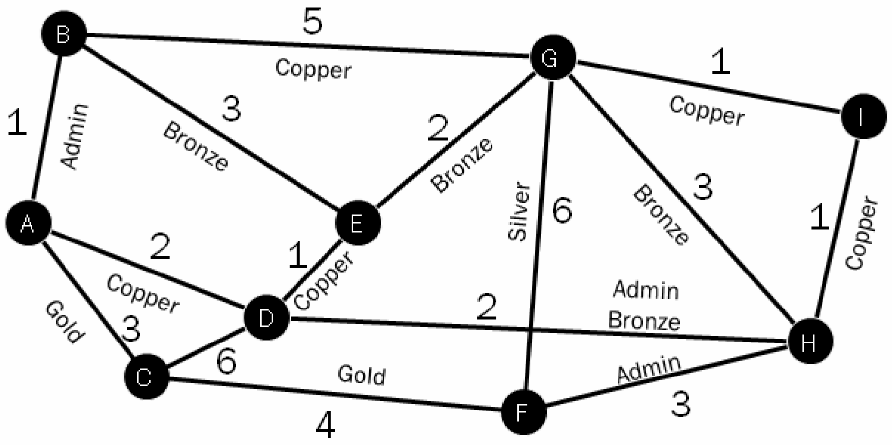
```
[edit protocols mpls]
user@rA# show
label-switched-path to-H {
    to 2.2.2.2;
    primary primary-path {
        admin-group include-all [ gold silver ];
    }
}
```

In this case, the LSP definition requires that the link include *both* the `gold` and `silver` colors.

The CSPF algorithm begins by pruning the links that do not include the required colors. In this example, no link includes *both* the `gold` and `silver` colors. Therefore, all links are pruned from consideration and the LSP setup fails because there is no path that meets the defined constraints.

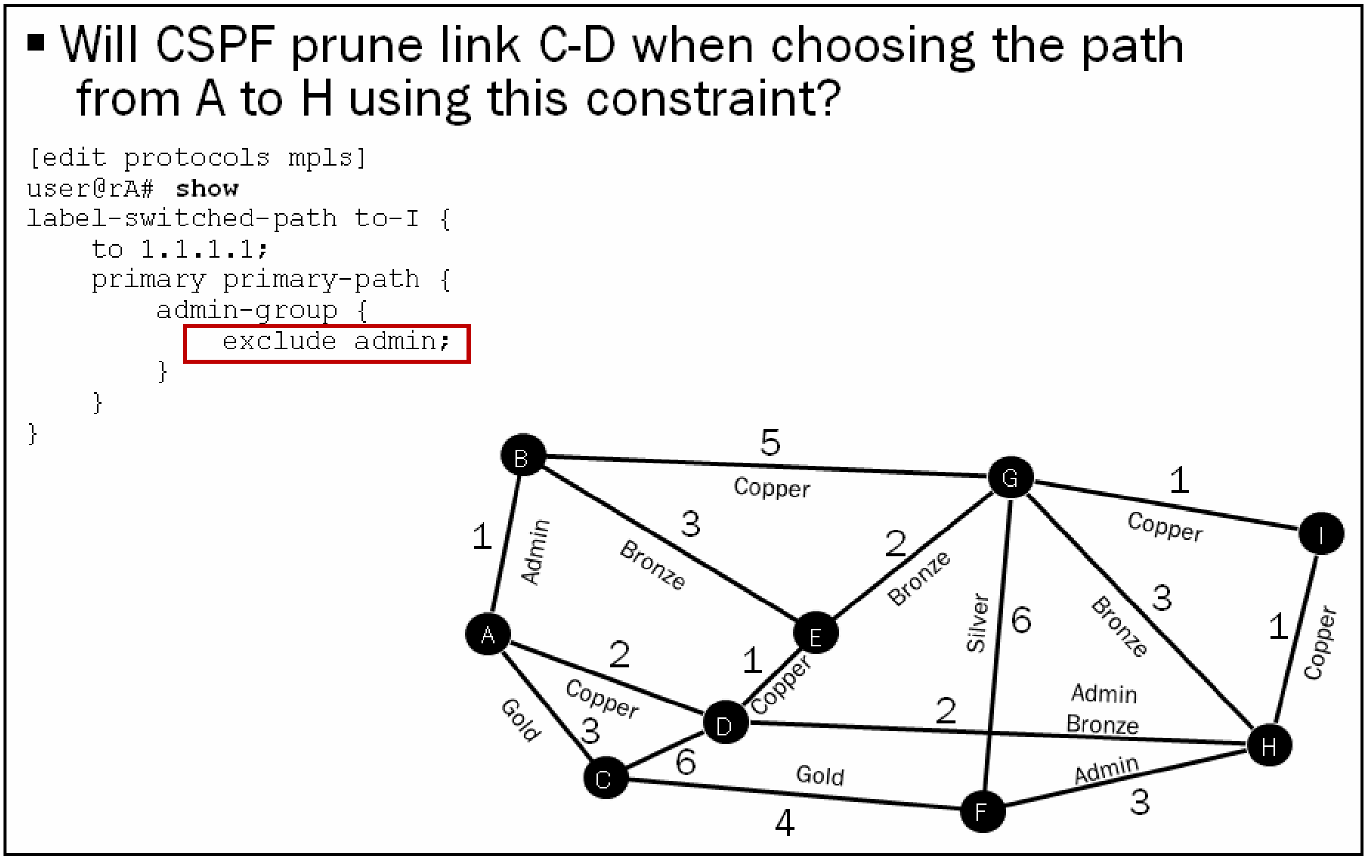
Administrative Groups V: The Solution

- No path between A and H exists that includes both *gold* and *silver* so LSP setup fails



This graphic displays the solution to the question asked on the previous graphic. In this case, there is no path between A and H that meets the defined constraints; LSP setup fails.

Test for Understanding

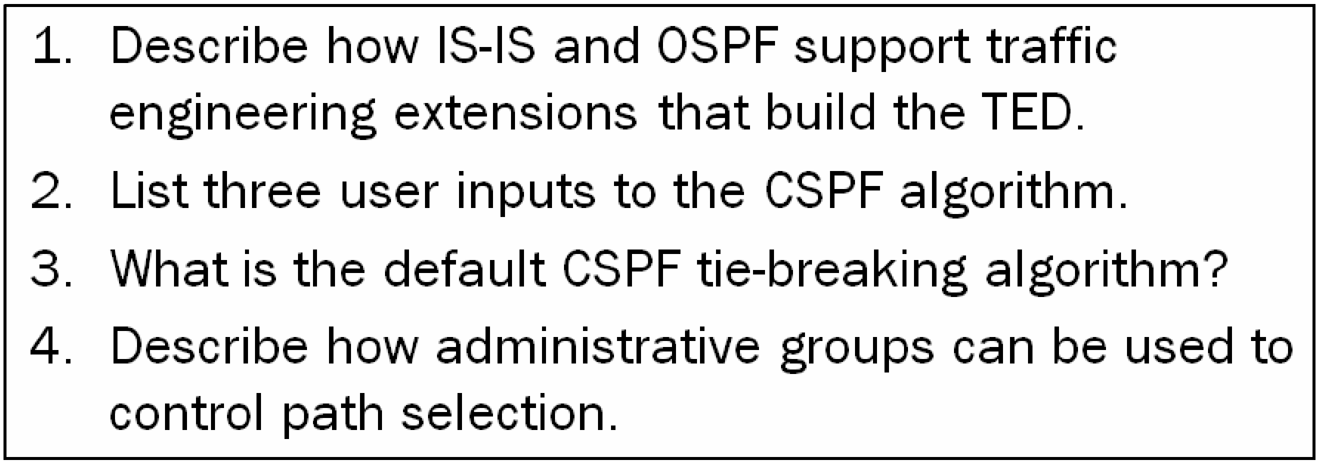


- Will CSPF prune link C-D when choosing the path from A to H using this constraint?

```
[edit protocols mpls]
user@rA# show
label-switched-path to-I {
    to 1.1.1.1;
    primary primary-path {
        admin-group {
            exclude admin;
        }
    }
}
```

The answer to the question on the graphic is *no*. The CSPF algorithm prunes links that do not have the specified `include` colors or that specifically match any specified `exclude` colors. In this example, there are no `include` constraints, and therefore, CSPF does not prune the C-D link. Note that the provided `exclude` color in this case is ***`admin`***; because link C-D has no color, it is considered to meet the constraints provided.

Review Questions



1. Describe how IS-IS and OSPF support traffic engineering extensions that build the TED.
2. List three user inputs to the CSPF algorithm.
3. What is the default CSPF tie-breaking algorithm?
4. Describe how administrative groups can be used to control path selection.

Answers to Review Questions

1.

OSPF supports the flooding of the opaque type 10 LSA. IS-IS supports the flooding of extended TLVs for traffic engineering. Both of the extensions to the protocols support the advertisement of rsvp bandwidth, administrative group, and router ID.

2.

Some possible user inputs are bandwidth requirement, administrative group requirement, explicit route, and priority.

3.

The default CSPF tie-breaking algorithm is random.

4.

When configuring an LSP an administrator can specify which administrative groups the LSP can traverse. The administrator can specify several administrative group constraints for an LSP by using the **`include-any`**, **`include-all`**, and the **`exclude`** statements.

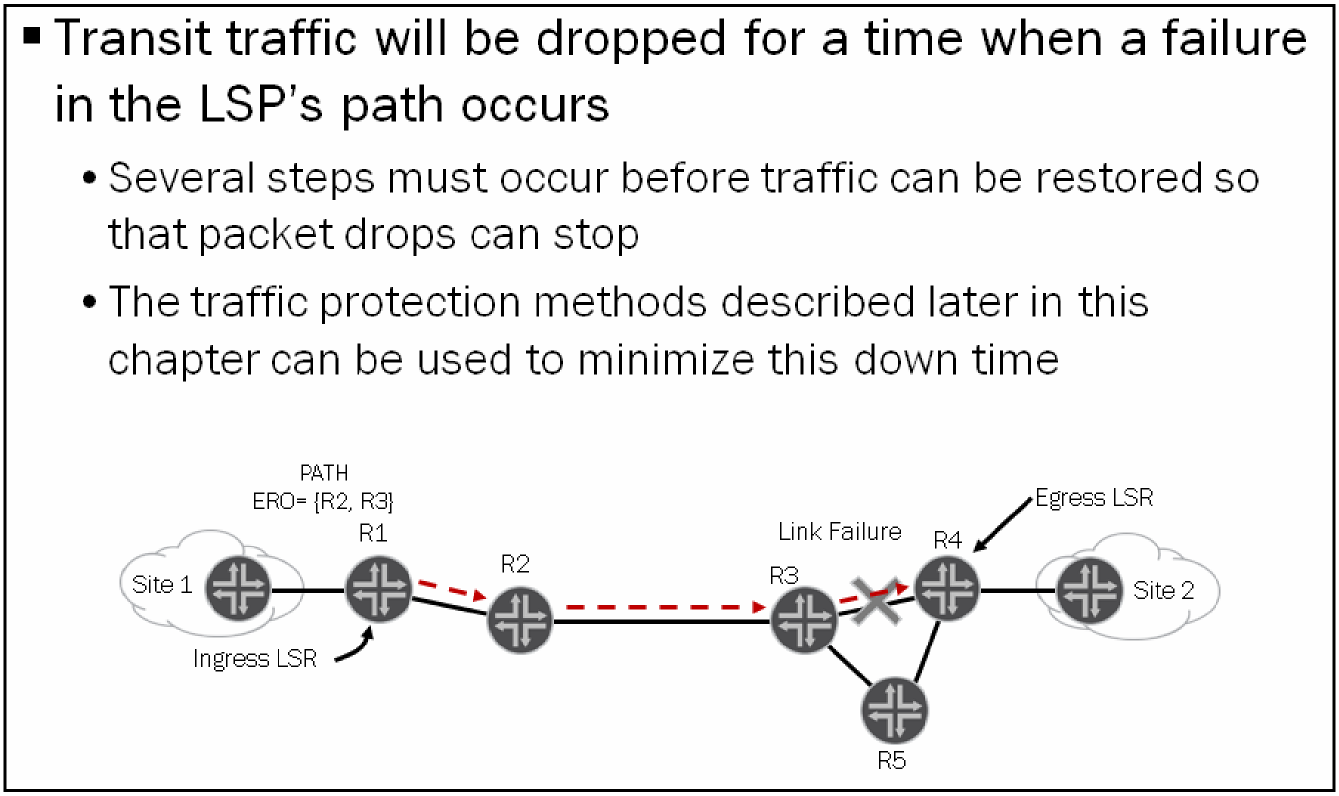
JNCIS-SP Study Guide—Part 3

Chapter 4: Traffic Protection and LSP Optimization

This Chapter Discusses:

- The default traffic protection behavior of RSVP-signaled label-switched paths (LSPs);
- The use of primary and secondary LSPs;
- LSP priority and preemption;
- Operation and configuration of fast reroute;
- Operation and configuration of link and node protection; and
- LSP optimization options.

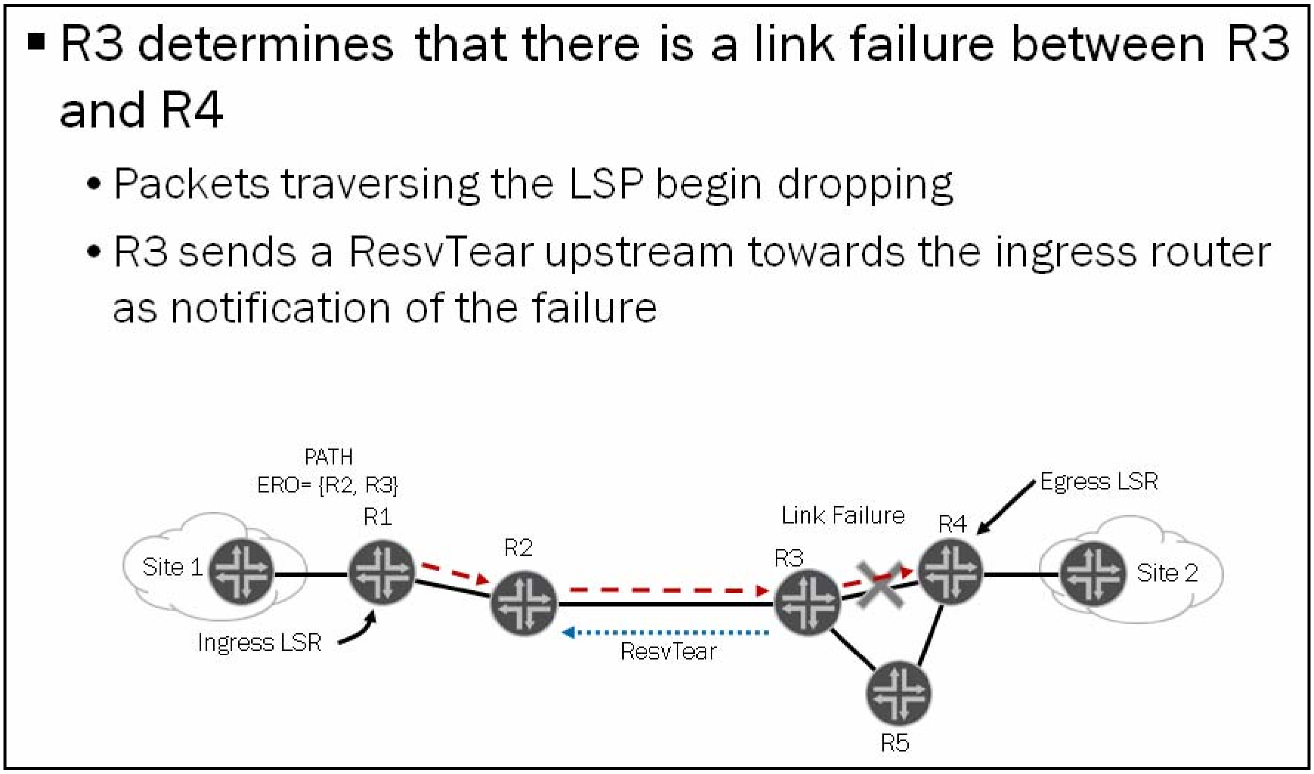
Network Failures



- Transit traffic will be dropped for a time when a failure in the LSP's path occurs
  - Several steps must occur before traffic can be restored so that packet drops can stop
  - The traffic protection methods described later in this chapter can be used to minimize this down time

When a network failure occurs along the path of an RSVP-signaled LSP, traffic that is currently traversing the LSP will be dropped. In the example, at the instant that the link between R3 and R4 fails, traffic that has already encapsulated in an MPLS header by R1 and forwarded downstream will be dropped. Also, until R1 receives PathTear message for the LSP, R1 might continue forwarding traffic using the LSP. That traffic will also be dropped. The time that it takes for traffic flow to be restored depends on the time it takes R1 to be notified of the failure followed as well as resignal a new LSP that will bypass the failed link. There are several features, like fast reroute and link protection which are described later in this chapter, that can significantly reduce down time.

Link Failure Between R3 and R4

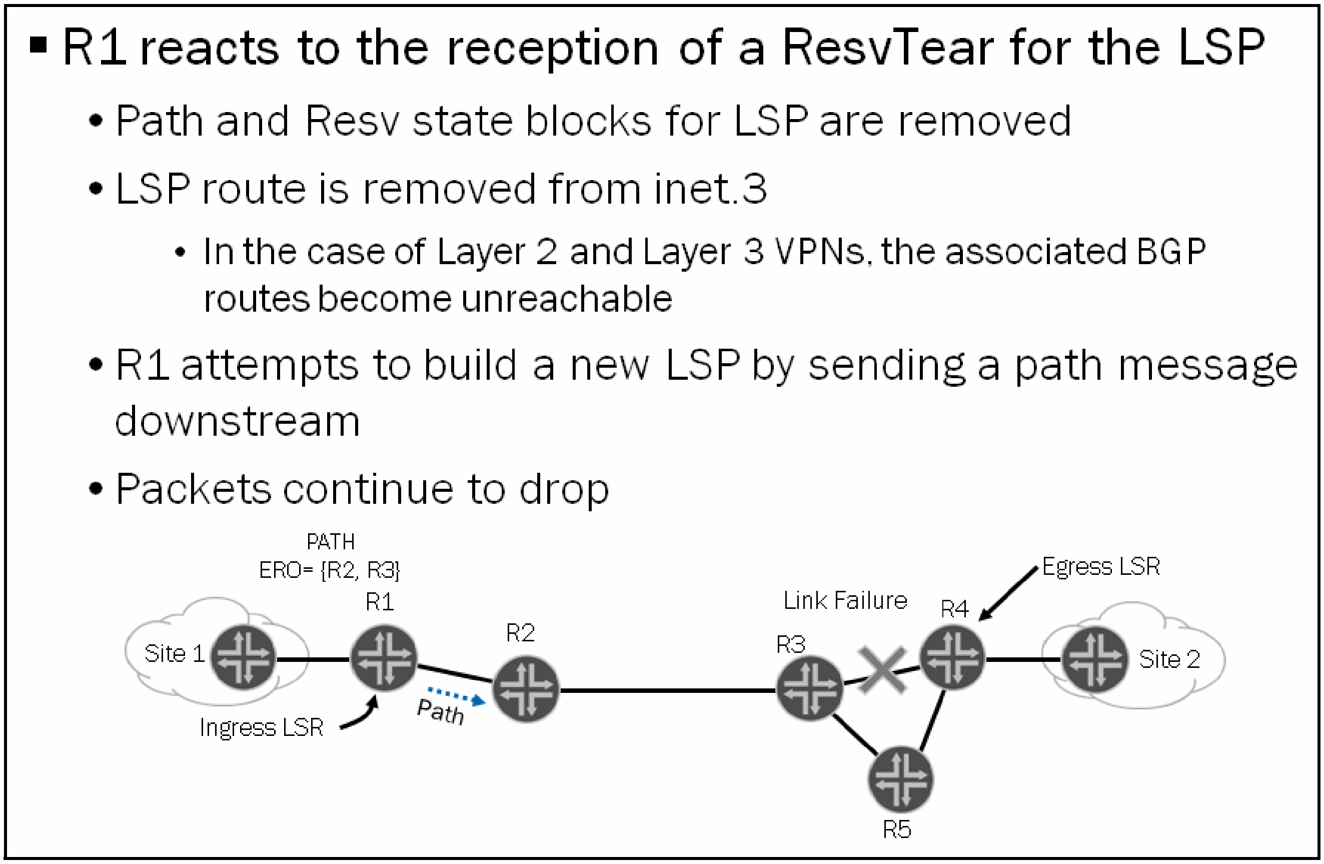


- R3 determines that there is a link failure between R3 and R4
  - Packets traversing the LSP begin dropping
  - R3 sends a ResvTear upstream towards the ingress router as notification of the failure

The following sections illustrate a failure scenario using the default settings of an RSVP-signaled LSP. That is, no traffic protection mechanism has been configured.

Transit packets begin to drop at the instant that the link between R3 and R4 fails. In response to the link failure, R3 will send a ResvTear message upstream to R2. R2 will, in turn, send a ResvTear upstream to R1.

R1 Receives PathTear

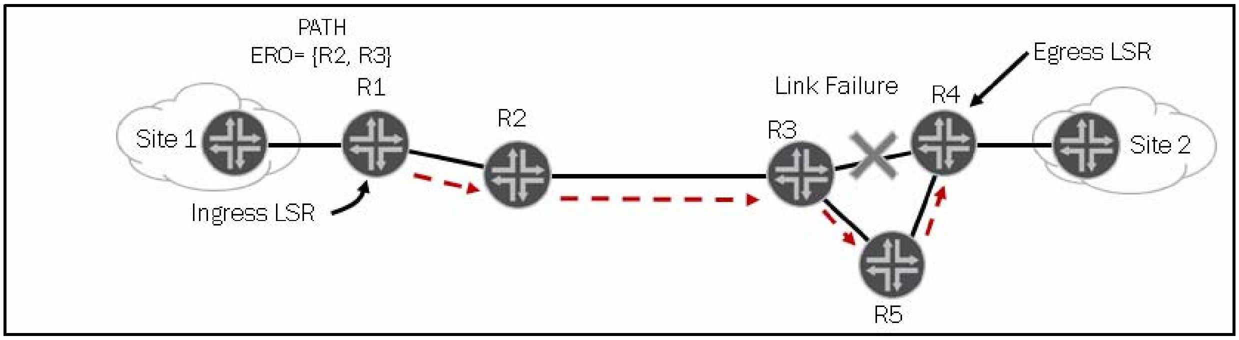


- R1 reacts to the reception of a ResvTear for the LSP
  - Path and Resv state blocks for LSP are removed
  - LSP route is removed from inet.3
    - In the case of Layer 2 and Layer 3 VPNs, the associated BGP routes become unreachable
  - R1 attempts to build a new LSP by sending a path message downstream
  - Packets continue to drop

When R1 receives the PathTear, it considers the LSP down and deletes the Path and Resv state block. The LSP is no longer a valid next-hop for routes in inet.3 (or any other routing table) so the /32 route associate with the LSP in inet3 is removed. Also, any BGP routes that had been using the LSP as a next-hop will need to have their next-hop recalculated. Now at this point, if the LSP was only being used to forward standard IP traffic (non-VPN traffic) packet drops may stop and new packets could be forwarded using next hops learned by using interior gateway protocol (IGP) routes in the inet.0 table. However, in a virtual private network (VPN) scenario as described in future chapters, a route in inet.3 must exist to forward traffic for a VPN between Site 1 and Site 2. Traffic between VPN sites might still continue to be dropped due to the lack of a valid route in the inet.3 table.

Along with the churn that occurs in the routing tables as described above, R1 will also attempt to reestablish the failed LSP by sending a Path message downstream towards the egress router.

A New LSP Is Established

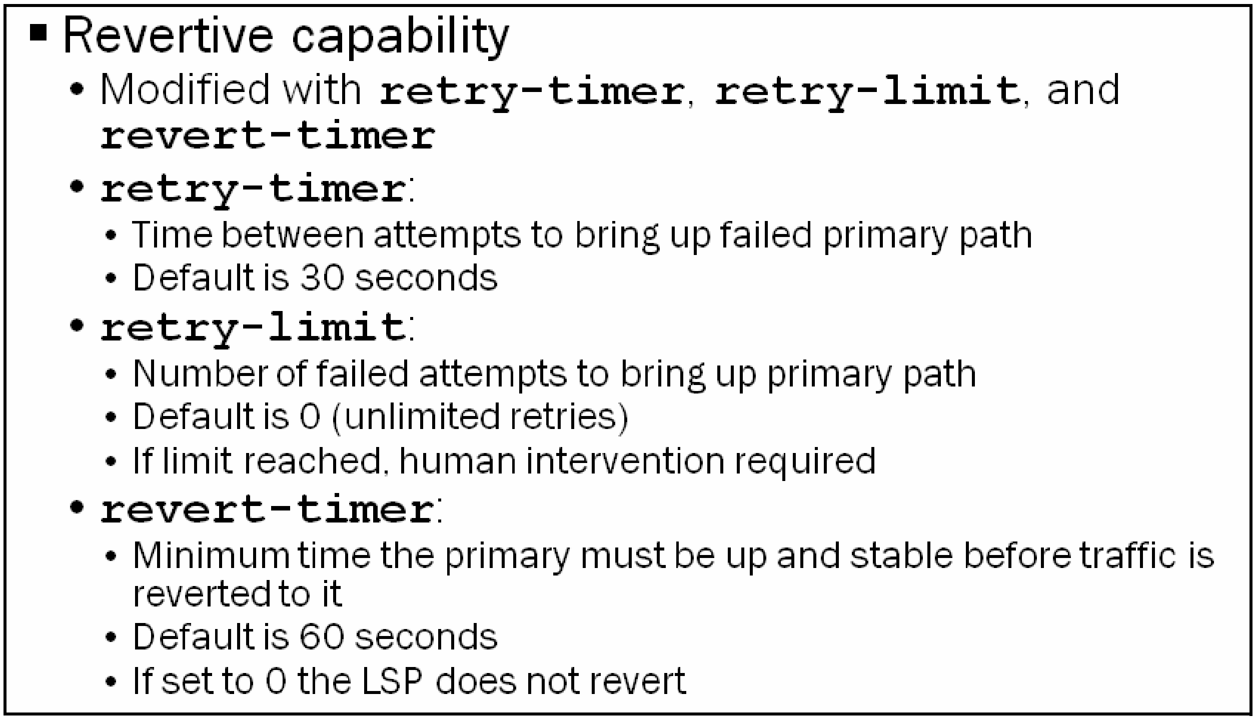


Assuming the link between R3 and R4 remains in a failed state, it is possible in the example network for a new LSP to be established using R5 as a path around the failed link. Once the LSP comes up, the LSP's /32 route is added back into inet.3 and becomes a valid and generally more preferred destination for BGP recursive next hop calculations for routes that were learned from the egress router, R4. At this point, packets between VPN Site 1 and 2 are no longer dropped.

Primary Physical Paths

Primary paths are optional. Only one primary path is permitted per LSP definition. The primary physical path can specify loose or strict Explicit Route Object (ERO) values under the named path hierarchy. Within the primary physical path you can specify parameters, like bandwidth or priority, that affect only the primary physical path. As a side note, the same parameters specified at the `label-switched-path` hierarchy affect both the primary and secondary physical path.

Primary Paths Revert by Default



- Revertive capability
  - Modified with **`retry-timer`**, **`retry-limit`**, and **`revert-timer`**
  - **`retry-timer`**:
    - Time between attempts to bring up failed primary path
    - Default is 30 seconds
  - **`retry-limit`**:
    - Number of failed attempts to bring up primary path
    - Default is 0 (unlimited retries)
    - If limit reached, human intervention required
  - **`revert-timer`**:
    - Minimum time the primary must be up and stable before traffic is reverted to it
    - Default is 60 seconds
    - If set to 0 the LSP does not revert

By default, an LSP fails over to its secondary path if its primary path fails. This failover occurs even if another physical path exists that complies with the primary path's constraints. The LSP still fails over to a secondary path (when such a path is defined) before it attempts to resignal an alternate primary physical path.

The router tries to resignal the primary path according to the number of seconds specified by the `retry-timer`, and it attempts to resignal the primary path LSP the number of times specified by the `retry-limit`. The alternate primary physical path must be up and stable for at least the number of seconds specified by the `revert-timer` before the LSP reverts back to the primary path.

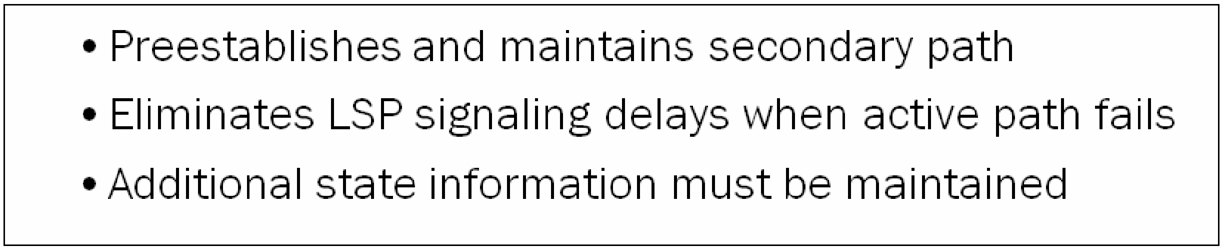
By default, the `retry-timer` is 30 seconds, the `retry-limit` is 0 (unlimited retries), and the `revert-timer` is 60 seconds. Setting the `revert-timer` to 0 means the LSP will not revert. If the `revert-timer` is set to 0 or the `retry-limit` is exceeded, you must manually clear the LSP to restart signaling attempts and move traffic to the primary path.

You configure the `retry-timer` and `retry-limit` values for individual LSPs at the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` configuration hierarchy. You can specify the `revert-timer` for all LSPs at the `[edit protocols mpls]` configuration hierarchy or for an individual LSP at the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` configuration hierarchy. When specified, the per-LSP value overrides the global value.

Secondary Physical Paths

Like primary paths, secondary paths are also optional. By default, a secondary path becomes active when a primary, or another secondary, physical path fails. Secondary paths are signaled in the order they appear in the router configuration when multiple secondary paths are defined.

Standby

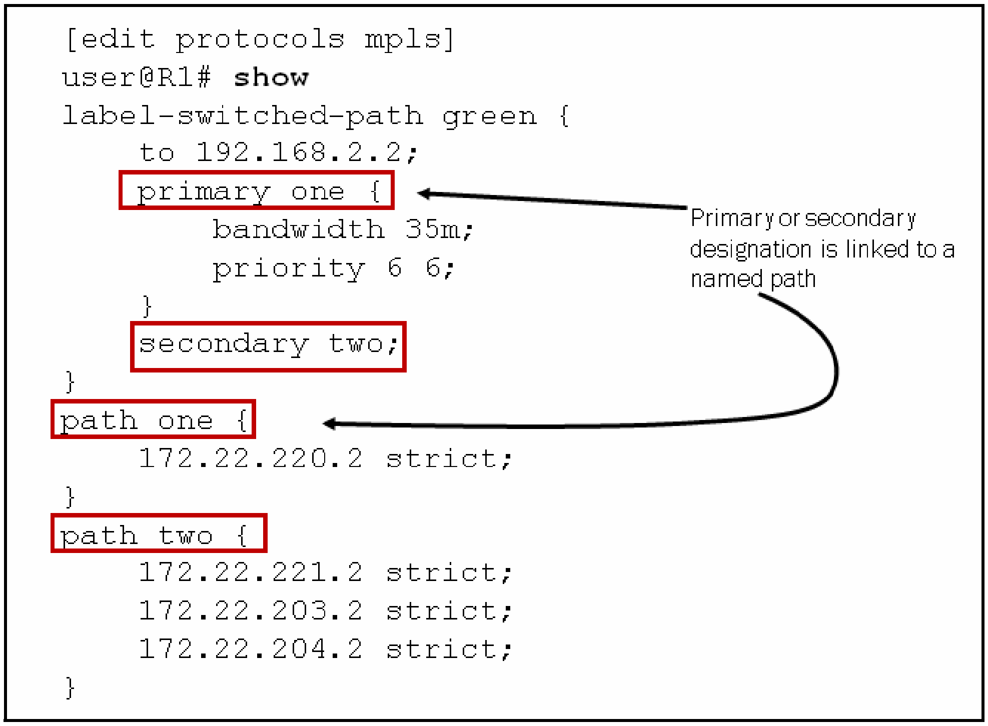


- Preestablishes and maintains secondary path
- Eliminates LSP signaling delays when active path fails
- Additional state information must be maintained

You can specify the **`standby`** command for a secondary path. This command causes the router to signal the secondary path, even though the secondary path is not currently needed, that is, the primary path has not yet failed. Note that standby secondaries result in routers having to maintain additional state in the form of the pre-established standby LSPs. When the standby option is specified at the `label-switched-path` *`lsp-name`* hierarchy, the router maintains standby state for *all* secondary paths. To place only selected secondaries into the standby state, specify the **`standby`** keyword at the secondary name hierarchy, as shown here:

```
[edit]
user@r1# show protocols mpls label-switched-path green
to 192.168.24.1;
primary one {
    bandwidth 35m;
    priority 6 6;
}
secondary two {
    standby;
}
```

Primary and Secondary Configuration

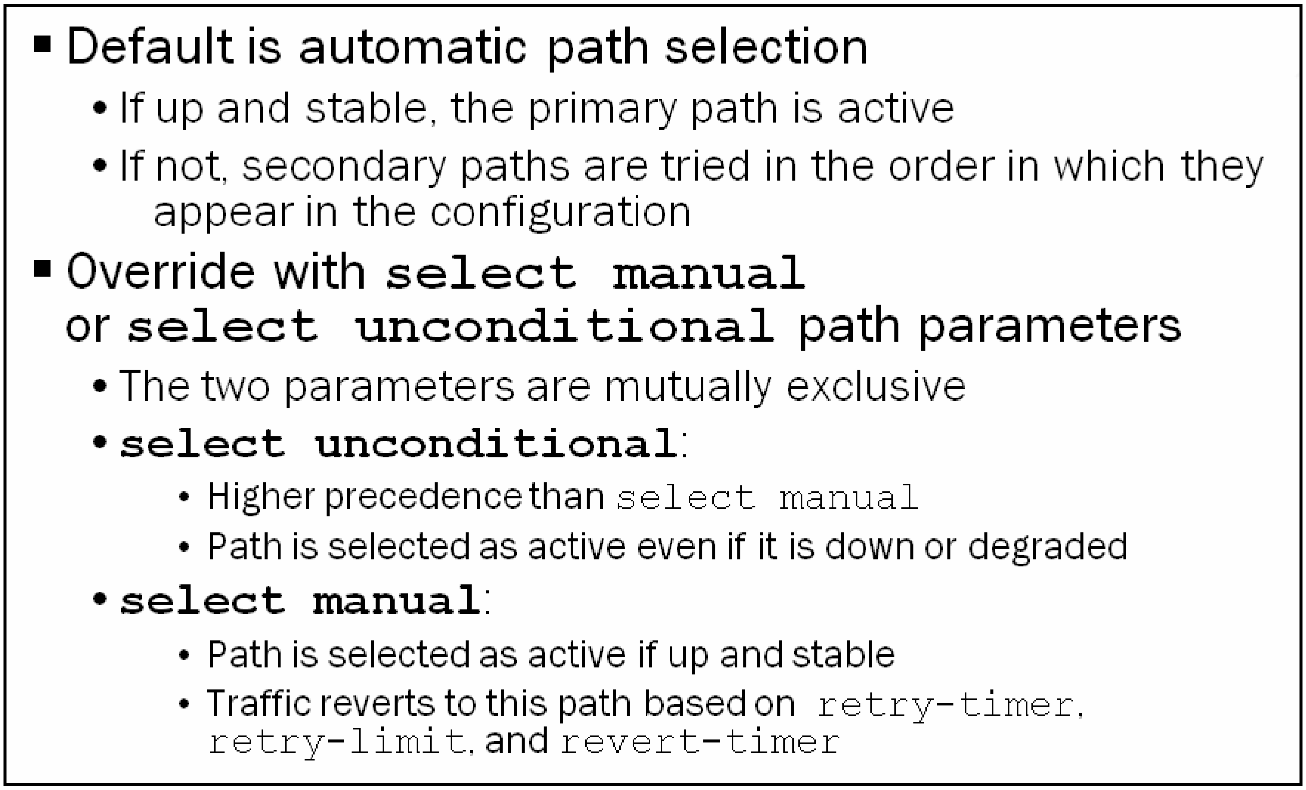


```
[edit protocols mpls]
user@R1# show
label-switched-path green {
    to 192.168.2.2;
    primary one {
        bandwidth 35m;
        priority 6 6;
    }
    secondary two;
}
path one {
    172.22.220.2 strict;
}
path two {
    172.22.221.2 strict;
    172.22.203.2 strict;
    172.22.204.2 strict;
}
```

Primary or secondary designation is linked to a named path

The Junos operating system does not require that a primary and secondary path share the same parameters. You can decide to configure your primary paths with stringent resource requirements while your secondary paths are far more lax in their demands. Such asymmetric settings helps to ensure that your secondary paths can be established during periods of diminished resources. In the example on the graphic, primary path *one* requires 35 Mbps of bandwidth while secondary path two requires only IP reachability.

Automatic Path Selection



- Default is automatic path selection
  - If up and stable, the primary path is active
  - If not, secondary paths are tried in the order in which they appear in the configuration
- Override with **`select manual`** or **`select unconditional`** path parameters
  - The two parameters are mutually exclusive
  - **`select unconditional`**:
    - Higher precedence than `select manual`
    - Path is selected as active even if it is down or degraded
  - **`select manual`**:
    - Path is selected as active if up and stable
    - Traffic reverts to this path based on `retry-timer`, `retry-limit`, and `revert-timer`

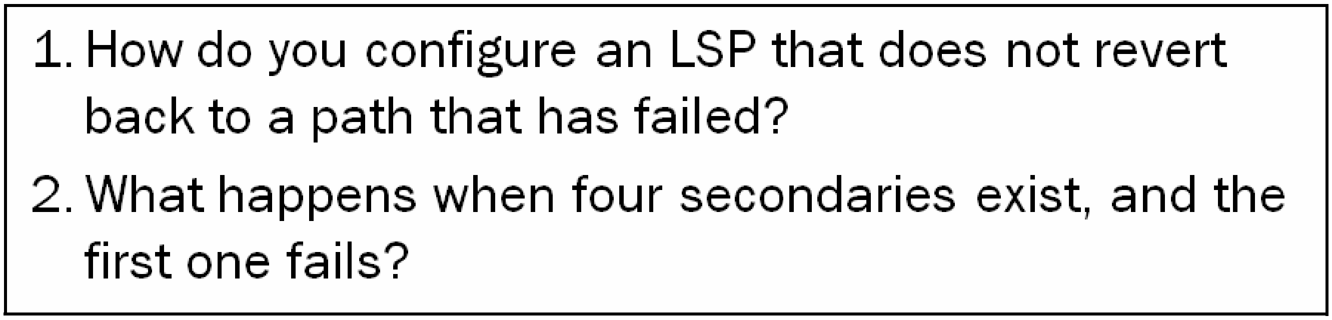
By default, the primary path is selected as the path to actively carry traffic. If the primary path is down or degraded (receiving errors from downstream), the automatic path selection algorithm tries secondary paths in the order in which they appear in the configuration. The first secondary path that is up and stable becomes the active path. Traffic reverts to a recently restored primary path based on the parameters previously discussed.

Overriding Default Behavior

You can override the automatic selection of the active path by specifying either **`select unconditional`** or **`select manual`** at the `[edit protocols mpls label-switched-path` *`lsp-name`* `primary` *`primary-path-name`*`]` or the `[edit protocols mpls label-switched-path` *`lsp-name`* `secondary` *`secondary-path-name`*`]` configuration hierarchy. The two parameters are mutually exclusive, and only one path per LSP can specify each parameter. If one path specifies **`select unconditional`** and another path specifies **`select manual`**, the path with **`select unconditional`** takes precedence.

The **`select unconditional`** parameter forces the path to become active even if it is down or degraded. The **`select manual`** parameter forces the path to become active as long as it is up and stable (and **`select unconditional`** is not configured on another path). If the path with `select manual` is down or degraded, automatic path selection is used to choose the active path. Upon restoration, traffic reverts to the path with the **`select manual`** parameter based on the settings of `retry-timer`, `retry-limit`, and `revert-timer`.

Check Your Knowledge



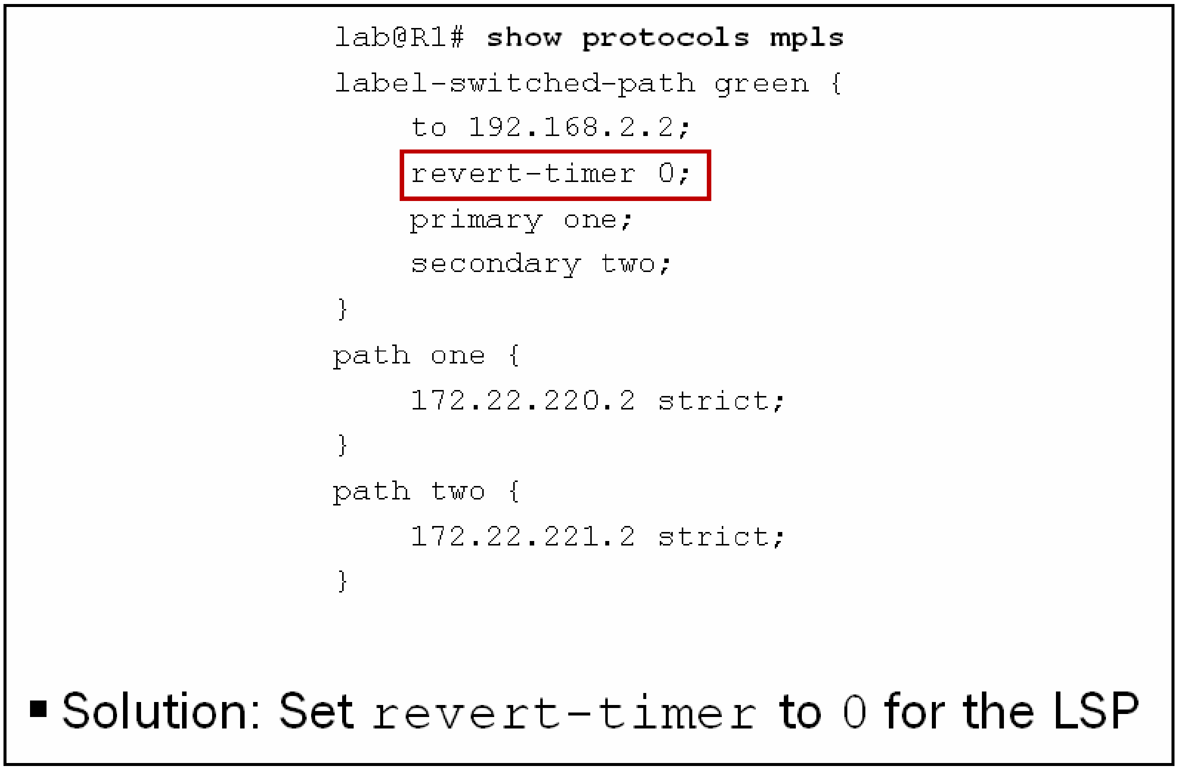
1. How do you configure an LSP that does not revert back to a path that has failed?
2. What happens when four secondaries exist, and the first one fails?

You might want to configure an alternate path through the network in case the active path fails but you do not want the traffic to change its physical path through the network after it has failed over to the alternate path. By default, when a primary path fails, traffic switches over to a secondary physical path, but this traffic reverts back to the primary physical path when it is again deemed operational.

This behavior brings us to the first question on the graphic: *How can you configure alternate LSP paths without chancing reversion to a path that has previously failed?*

The second question is designed to test your understanding of secondary paths and how they are handled in the face of failures.

Check Your Knowledge: Solutions



```
lab@R1# show protocols mpls
label-switched-path green {
    to 192.168.2.2;
    revert-timer 0;
    primary one;
    secondary two;
}
path one {
    172.22.220.2 strict;
}
path two {
    172.22.221.2 strict;
}
```

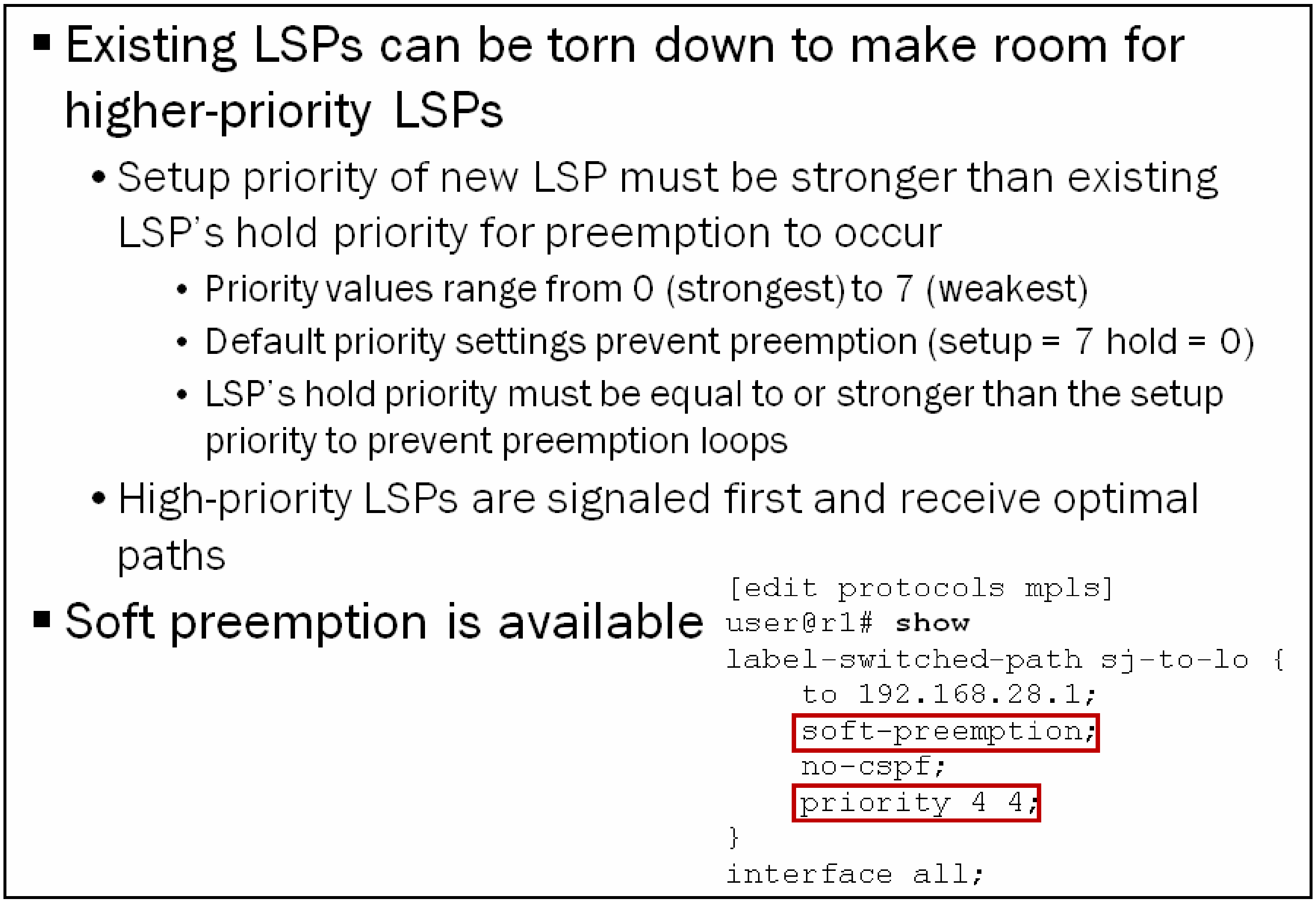
- Solution: Set `revert-timer` to `0` for the LSP

By default, Junos OS will revert back to a defined primary path. You can disable this default behavior by specifying a value of `0` for the **`revert-timer`**. When specified at the LSP level, this value affects only a single LSP. When specified at the MPLS level, the value affects all LSPs.

An alternative solution involves the definition of secondary paths *only*. In this case, Junos OS brings up the second configured secondary LSP when the first secondary path fails. Later, if the first secondary path is capable of being used again, Junos OS continues to use the existing secondary LSP and does not revert to the original secondary path.

The answer to the second question depends on whether or not **`select unconditional`** or **`select manual`** is configured. If neither is configured and the primary path is absent or down, Junos OS attempts to establish an active path by signaling each secondary LSP in the order in which it appears in the configuration. If the first secondary physical path fails or cannot be established, the router attempts to signal the next secondary physical path, and so on. The **`select unconditional`** and **`select manual`** parameters override this behavior.

LSP Priorities and Preemption



- Existing LSPs can be torn down to make room for higher-priority LSPs
  - Setup priority of new LSP must be stronger than existing LSP's hold priority for preemption to occur
    - Priority values range from 0 (strongest) to 7 (weakest)
    - Default priority settings prevent preemption (setup = 7 hold = 0)
    - LSP's hold priority must be equal to or stronger than the setup priority to prevent preemption loops
  - High-priority LSPs are signaled first and receive optimal paths
- Soft preemption is available

```
[edit protocols mpls]
user@r1# show
label-switched-path sj-to-lo {
    to 192.168.28.1;
    soft-preemption;
    no-cspf;
    priority 4 4;
}
interface all;
```

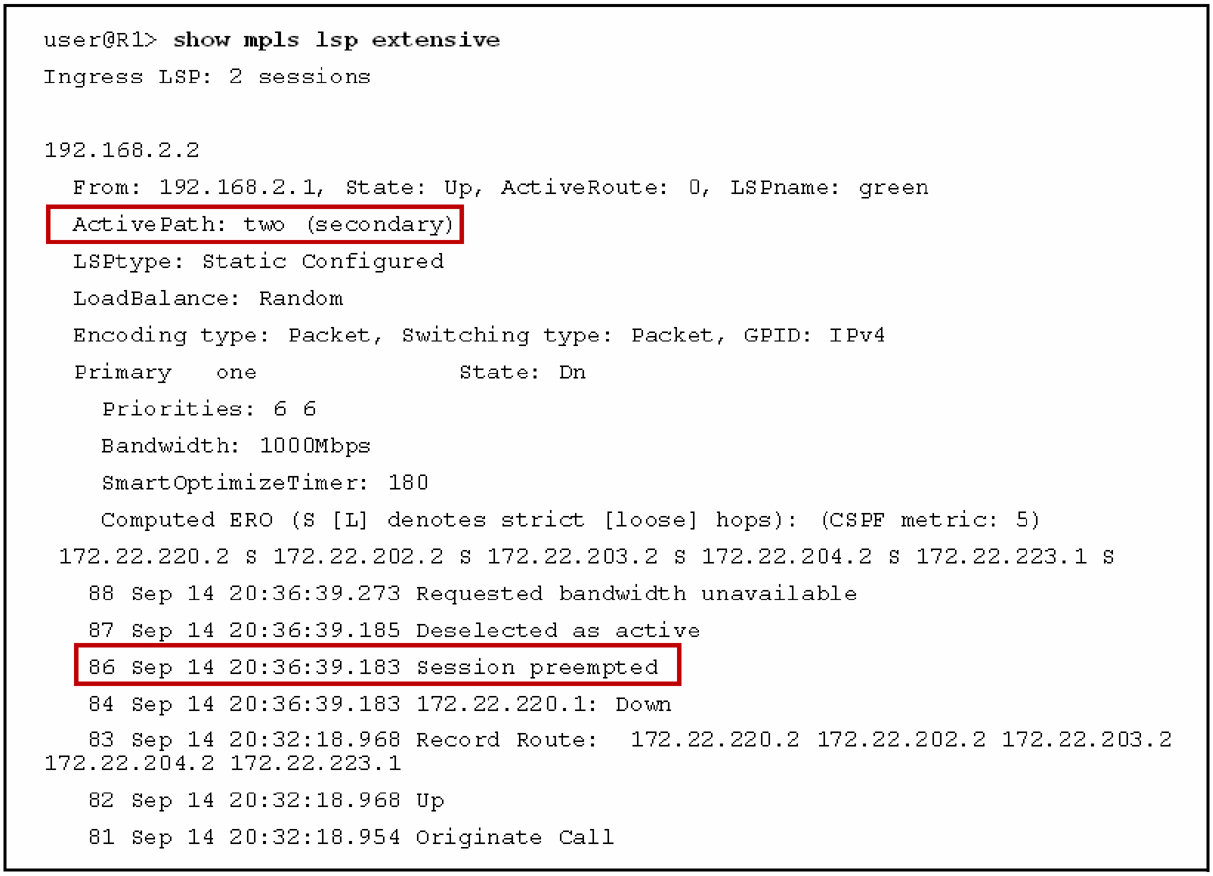
RSVP-signaled LSPs support the notion of LSP setup and hold priorities. These priorities work together to determine the relative priority of a new LSP that must be established versus the hold priority of existing LSPs. When insufficient resources exist to accommodate all LSPs simultaneously, an LSP with a strong setup priority preempts—or causes the teardown—of an existing LSP with a weaker hold priority. At software startup, LSPs are signaled in order from strongest to weakest setup priority; this behavior ensures that high-priority LSPs are established first and are afforded optimal paths.

LSP setup and hold priorities range in value from 0 (the strongest) to 7 (the weakest). The default settings disable preemption by assigning all LSPs the weakest setup priority (7) and the strongest hold priority (0). Note that you cannot commit a configuration in which an LSP's hold priority is less (weaker) than its setup priority because such a configuration can lead to preemption churn. Before the sample LSP shown on the graphic can cause preemption, the default hold priority (0) must be set to a value of 5 or higher on existing LSPs. Modified LSP priority values are displayed in the output of a **`show mpls lsp extensive`** command.

Using Soft Preemption

In normal operation, a preempted LSP is torn down before a new path is located. During this process, traffic associated with the preempted LSP can be lost. To avoid traffic loss the Junos OS can specify soft preemption behavior on a per-LSP basis. When configured, the ingress LSR sets the *soft preemption desired* flag in the record route object (RRO) sub-object of the path message to signal the desire for soft preemption behavior in downstream nodes. This feature is backwards compatible in that LSP establishment succeeds even if one or more nodes does not support this sub-object. To enable soft preemption, add the **`soft-preemption`** keyword at the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` hierarchy. The output of a **`show rsvp session detail`** command displays whether soft preemption is requested for a given LSP.

Monitoring Preemption

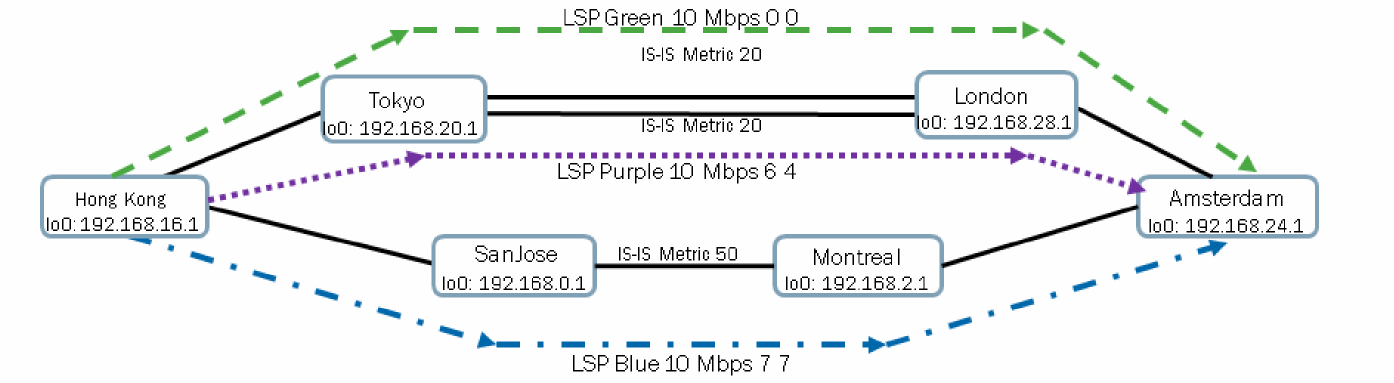


```
user@R1> show mpls lsp extensive
Ingress LSP: 2 sessions

192.168.2.2
  From: 192.168.2.1, State: Up, ActiveRoute: 0, LSPname: green
  ActivePath: two (secondary)
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
  Primary   one              State: Dn
    Priorities: 6 6
    Bandwidth: 1000Mbps
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 5)
 172.22.220.2 S 172.22.202.2 S 172.22.203.2 S 172.22.204.2 S 172.22.223.1 S
   88 Sep 14 20:36:39.273 Requested bandwidth unavailable
   87 Sep 14 20:36:39.185 Deselected as active
   86 Sep 14 20:36:39.183 Session preempted
   84 Sep 14 20:36:39.183 172.22.220.1: Down
   83 Sep 14 20:32:18.968 Record Route:  172.22.220.2 172.22.202.2 172.22.203.2
172.22.204.2 172.22.223.1
   82 Sep 14 20:32:18.968 Up
   81 Sep 14 20:32:18.954 Originate Call
```

The graphic shows that at 20:36:39, the *green* LSP's primary path, path *one*, was preempted and the secondary path, path *two*, became active. This scenario uses equal setup and hold values for each LSP, with the priority values set to 6 for the *green* LSP. In this example, another LSP with a higher priority 0 0 has caused the preemption of the *green* LSP's primary path, path *one*, which in turn results in the establishment of a secondary path, path *two*. The *green* LSP's secondary path is configured with a lower bandwidth requirement to allow it to establish in the event the primary path is preempted. There is no reference to the LSP with 0 0 priority in the screen captures because this is a priori knowledge. The truncated capture below continues the output shown on the graphic. The output confirms that the secondary path, path *two*, became active at the same time the primary path was preempted:

```
*Secondary two              State: Up
    Priorities: 7 0
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 4)
 172.22.221.2 S 172.22.203.2 S 172.22.204.2 S 172.22.223.1 S
Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt
20=Node-ID):
          172.22.221.2 172.22.203.2 172.22.204.2 172.22.223.1
   83 Sep 14 20:36:39.286 Selected as active path
   82 Sep 14 20:36:39.284 Record Route:  172.22.221.2 172.22.203.2 172.22.223.1
   81 Sep 14 20:36:39.284 Up
...
```

Test Understanding



```
label-switched-path Red {
        to 192.168.24.1;
        priority 6 6;
              bandwidth 10M;
        }
```

- Given that all links with existing LSPs have less than 10 M available, which LSPs can be preempted by LSP *Red*?

You can assume in this example that all links with existing LSPs have less than 10 M available. Therefore, the only way to establish LSP *Red* is for it to preempt one of the existing LSPs. Can you determine which LSP will be preempted by LSP *Red*?

The setup priority for *Red* is 6 (the first number). The hold priority (the second number) for LSPs *Green* and *Purple* are both less than 6, which gives these LSPs a stronger hold priority that will prevent their preemption. In contrast, LSP *Blue* has a hold a priority of 7, which is weaker that LSP *Red*'s setup priority. Thus, LSP *Red* can only preempt LSP *Blue*. Note the IS-IS metric has no effect on LSP preemption.

Question: What if LSP *Red* had priority [3 7]?

Answer: This is a trick question because such a priority setting is not allowed. Recall that an LSP cannot have a setup priority that is stronger than its hold setting.

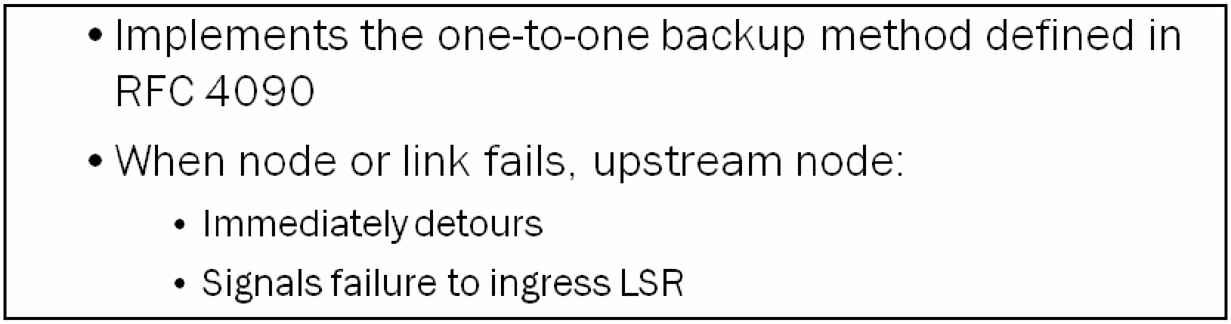
Ask Yourself These Questions

- Is there a way to get quicker failover in the event of primary LSP failure?
- How can I reduce packet loss when I lose my primary LSP?

Fast reroute is a feature that can dramatically reduce packet loss in the event of a primary path failure. If you ever find yourself asking the types of questions posed on the graphic, the answer you seek might very well be fast reroute!

When you define a secondary physical path in the standby state, the router presignals an alternate physical path for the LSP. However, traffic transiting the network is still lost while the network forwards information about failed links (and the failed primary path) to the ingress router. When the ingress router learns that a link is down, it begins using the alternate path immediately, but during this time traffic that is in transit or still being presented to the primary path is lost. Fast reroute provides a way for intermediate LSRs to immediately start forwarding traffic over an alternate route while simultaneously alerting the ingress LSR to the presence of downstream link or node failures.

Fast Reroute Reduces Packet Loss



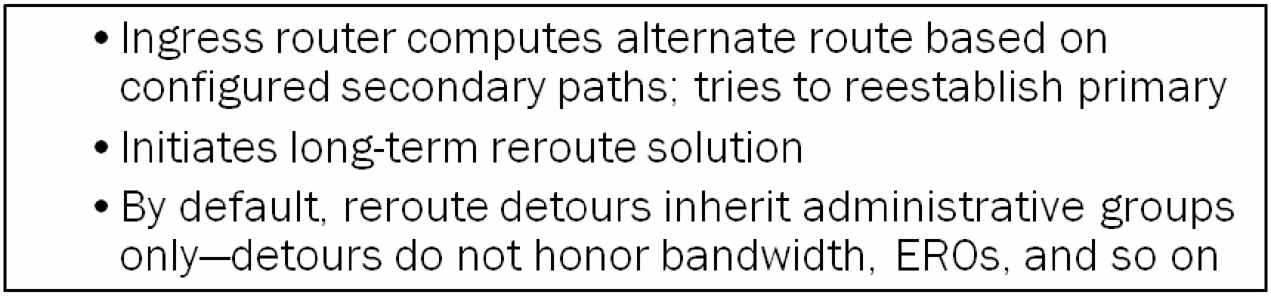
- Implements the one-to-one backup method defined in RFC 4090
- When node or link fails, upstream node:
  - Immediately detours
  - Signals failure to ingress LSR

You configure fast reroute to minimize the effects of a LSP failure. Fast reroute enables a router upstream of the failure to quickly route around the failure while the primary path is torn down and resignaled. The router that detects the primary path failure signals the outage to the ingress router. Fast reroute serves as an interim connectivity mechanism during the establishment of a new primary path. Once the new primary path is signaled, the fast reroute detours associated with the original paths are torn down; fast reroute is a short-term solution.

When fast reroute is enabled, the ingress router adds an object to the RSVP Path messages requesting that downstream routers establish reroute detours. These downstream routers then originate detour Path messages to detour the LSP around that LSRs downstream link and node.

When an active physical path fails and a detour is available, the upstream router sends a PathErr message to the ingress router. This message triggers new CSPF computations and a switchover to an alternate path if available. If a fast reroute detour is not available, the downstream node sends a ResvTear message and begins withdrawing the MPLS labels, which brings down the LSP. A fast reroute path might stay up indefinitely if an alternative primary path is not found.

Only Ingress Knows All Traffic Engineering Constraints



- Ingress router computes alternate route based on configured secondary paths; tries to reestablish primary
- Initiates long-term reroute solution
- By default, reroute detours inherit administrative groups only—detours do not honor bandwidth, EROs, and so on

By default, the fast reroute path only inherits the administrative group settings from the original LSP. It is therefore possible for a fast reroute detour to have substantially less bandwidth than was specified in the original LSP. As soon as the ingress node resignals the LSP, the fast reroute path is torn down. Note that the newly signaled LSP will have the correct traffic parameters, including bandwidth constraints. This behavior tends to classify fast reroute detours as *temporary*. You can configure the following fast reroute parameters if wanted: bandwidth, hop limit, include, and exclude administrative groups. You can also disable the inheritance of include and exclude administrative groups (because fast reroute detours inherit administrative groups by default).

General Characteristics

- Configured on ingress router only
- Detours around node or link failure
  - ≤~100s of ms reroute time after failure detected
- Detour paths immediately available
- Uses TED to calculate detours
  - Does not require a CSPF LSP on ingress node

By default, the router uses the traffic engineering database (TED) to calculate a detour path. These detours can add up to an additional six hops to the LSP path in an attempt to bypass the downstream node. Use the **`hop-count`** parameter to change the default number of hops the router will support when calculating a detour.

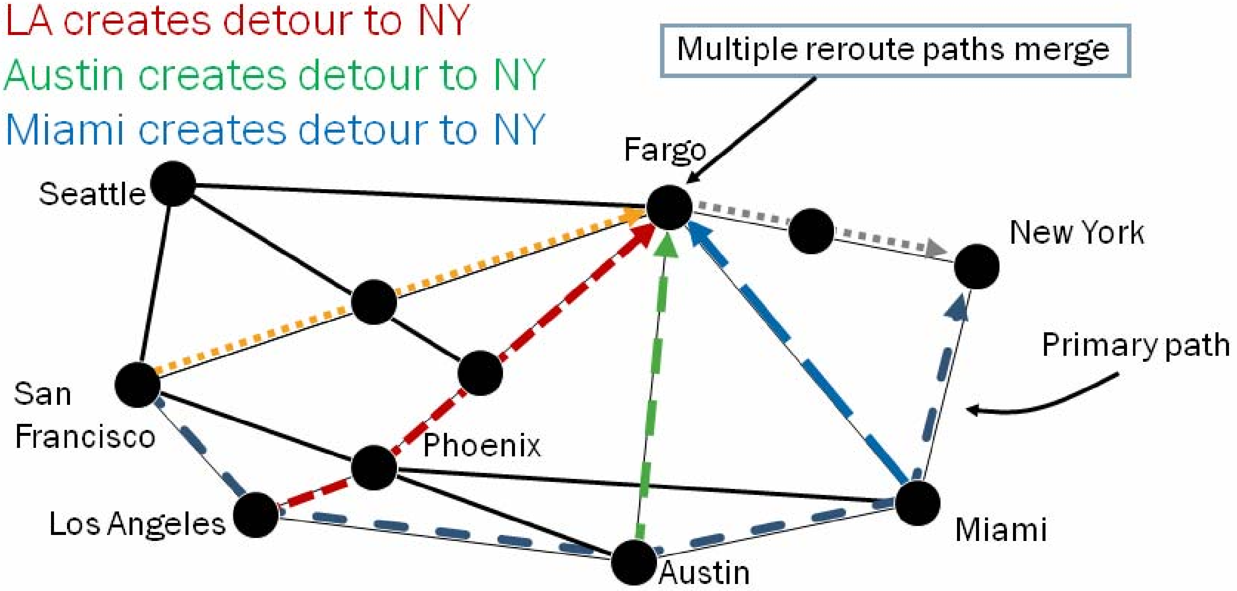
When a router with a fast reroute detour available recognizes a link or node failure, it immediately begins to detour the traffic. The Packet Forwarding Engine (PFE) maintains precomputed fast reroute detours to provide convergence times that, in some cases, rival SONET Automatic Protection Switching (APS)!

Each downstream node originates its own detour path messages. It is possible that a given node will not be able to establish a detour path. The result is that some portions of the LSP might have fast reroute protection while other portions do not. An LSP will never be torn down just because fast reroute detours cannot be established.

You configure fast reroute at the `label-switched-path` *`lsp-name`* hierarchy, which causes all primary and secondary physical paths to signal fast reroute.

LSP from San Francisco to New York

- LSP from San Francisco to New York through LA, Austin, and Miami
- Enable fast reroute on ingress
  - SF creates detour to NY
  - LA creates detour to NY
  - Austin creates detour to NY
  - Miami creates detour to NY



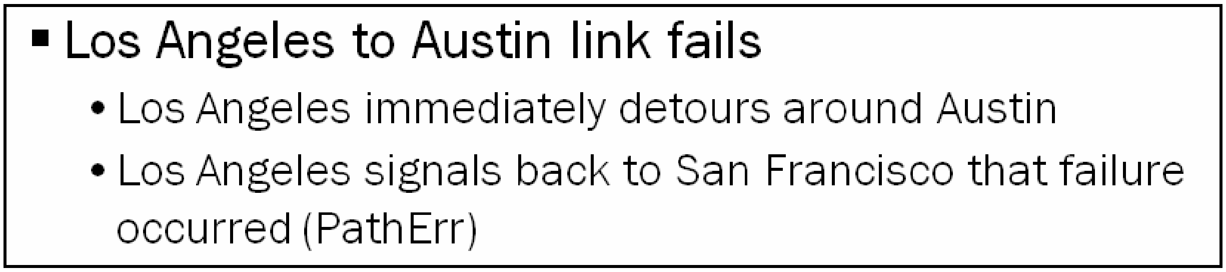
In this fast reroute example we begin with San Francisco acting as the ingress node for a LSP that terminates at New York. The routing of this LSP is via Los Angeles, Austin, and Miami, as shown in the graphic.

Enable Fast Reroute on Ingress

The configuration of the *to-NY* LSP is shown here. Note that the `fast-reroute` keyword is present in this example. As a result, San Francisco determines the next downstream node is Los Angeles, with a follow-on node of Austin. Node San Francisco therefore calculates and signals a fast reroute path around Los Angeles to New York. Los Angeles likewise calculates and signals a path around Austin to New York. Austin calculates and signals a route around Miami to New York. If any link or node fails, the fast reroute path recognizes the failed LSP quickly and immediately begins sending traffic on the fast reroute path.

```
mpls {
  label-switched-path to-NY {
    to 192.168.2.2;
    primary use-austin;
    secondary use-seattle;
    fast-reroute;
  }
  path use-austin {
    192.168.1.2 loose;
  }
  path use-seattle {
    192.168.8.1 loose;
  }
}
```

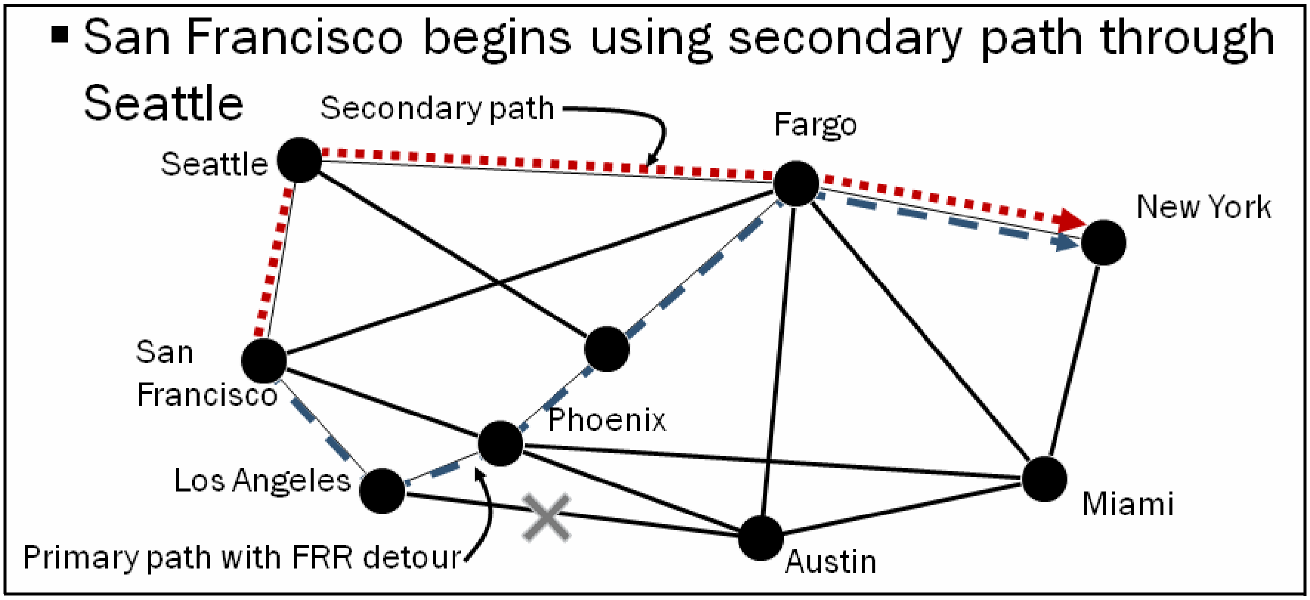
Los Angeles Detects Failure



- Los Angeles to Austin link fails
  - Los Angeles immediately detours around Austin
  - Los Angeles signals back to San Francisco that failure occurred (PathErr)

In the example in the graphic, the link between Los Angeles and Austin failed. Los Angeles recognizes the failure (possibly within milliseconds), and immediately begins to forward the traffic along the fast reroute path to node Phoenix. It also sends a PathErr message to San Francisco so that San Francisco can resignal the LSP.

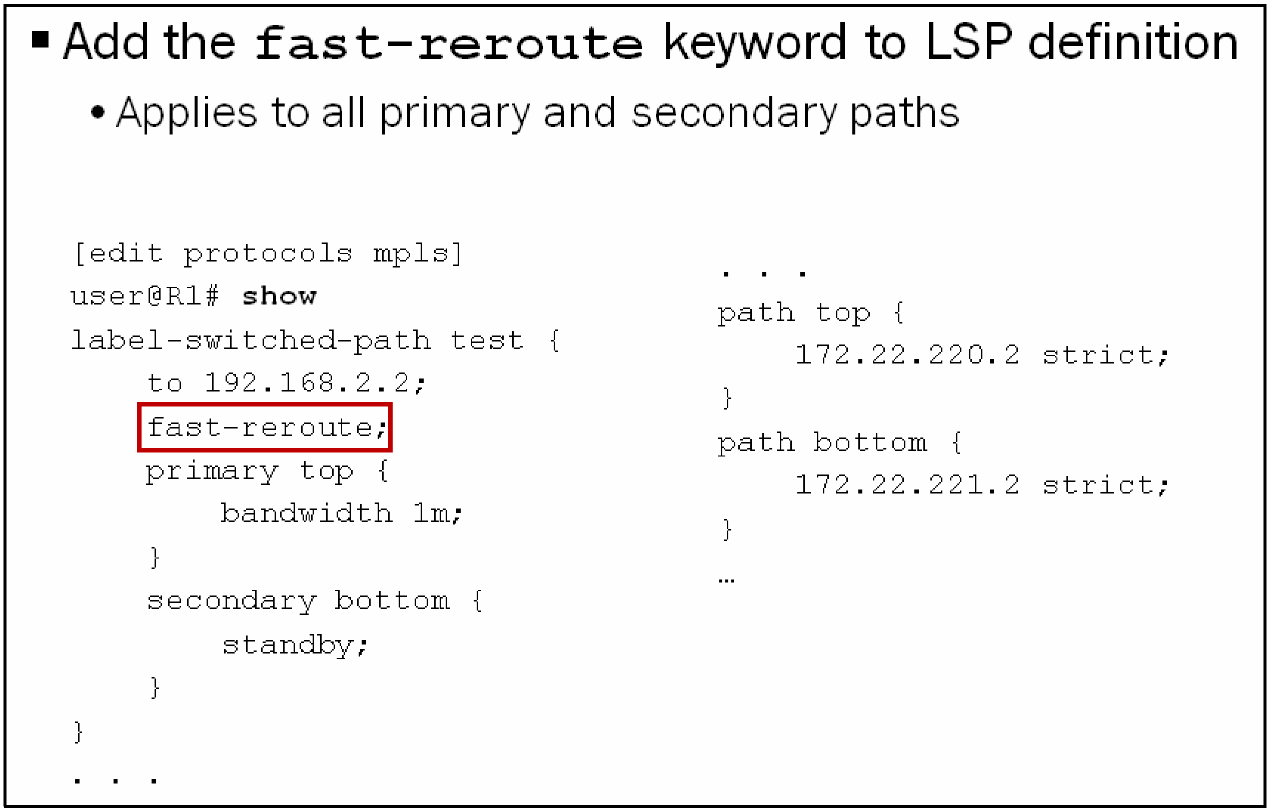
The Final Solution



- San Francisco begins using secondary path through Seattle

Once notified of the primary path failure, node San Francisco signals the secondary path through Seattle to New York. Traffic is then switched from the primary path, which is still using a fast reroute detour, and the remnants of the primary path are torn down. At this point, node San Francisco tries to resignal a new primary path.

Configure Fast Reroute



- Add the **`fast-reroute`** keyword to LSP definition
  - Applies to all primary and secondary paths

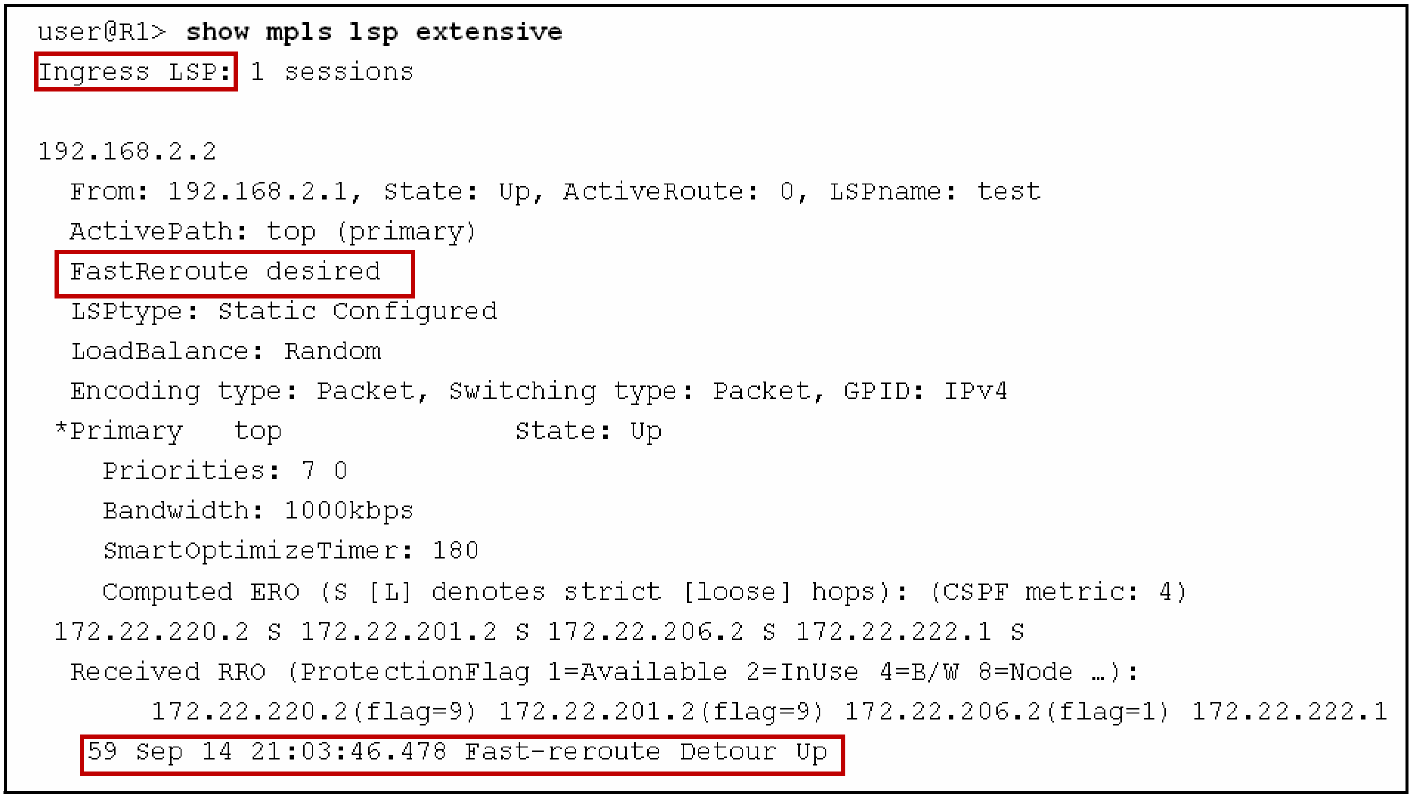
```
[edit protocols mpls]
user@R1# show
label-switched-path test {
    to 192.168.2.2;
    fast-reroute;
    primary top {
        bandwidth 1m;
    }
    secondary bottom {
        standby;
    }
}
. . .
```

```
. . .
path top {
    172.22.220.2 strict;
}
path bottom {
    172.22.221.2 strict;
}
...
```

You configure fast reroute by including the **`fast-reroute`** keyword at the `label-switched-path` *`lsp-name`* hierarchy. This setting applies to all defined primary and secondary paths.

By default, fast reroute has a limit of six hops out of the way to get to the next downstream path. You can configure a larger or smaller number with the **`hop-count`** parameter.

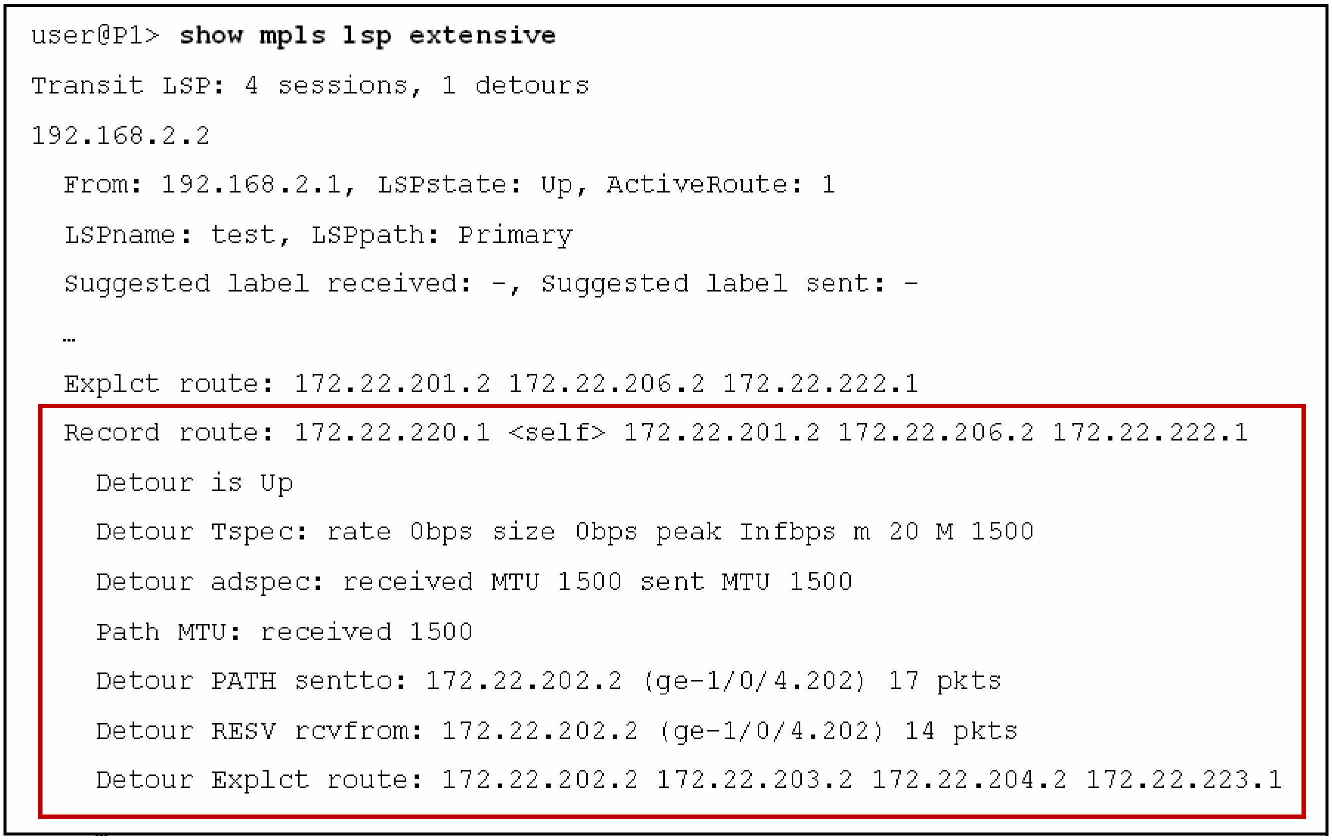
Monitor Fast Reroute: Ingress



```
user@R1> show mpls lsp extensive
Ingress LSP: 1 sessions

192.168.2.2
  From: 192.168.2.1, State: Up, ActiveRoute: 0, LSPname: test
  ActivePath: top (primary)
  FastReroute desired
  LSPtype: Static Configured
  LoadBalance: Random
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary    top              State: Up
    Priorities: 7 0
    Bandwidth: 1000kbps
    SmartOptimizeTimer: 180
    Computed ERO (S [L] denotes strict [loose] hops): (CSPF metric: 4)
 172.22.220.2 S 172.22.201.2 S 172.22.206.2 S 172.22.222.1 S
  Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node …):
       172.22.220.2(flag=9) 172.22.201.2(flag=9) 172.22.206.2(flag=1) 172.22.222.1
   59 Sep 14 21:03:46.478 Fast-reroute Detour Up
```

The output from the **`show mpls lsp extensive`** command indicates that fast reroute was requested. You can also see an indication that the fast reroute path is up—along with a timestamp—within the active path's history.

Confirm Fast Reroute—Transit LSR

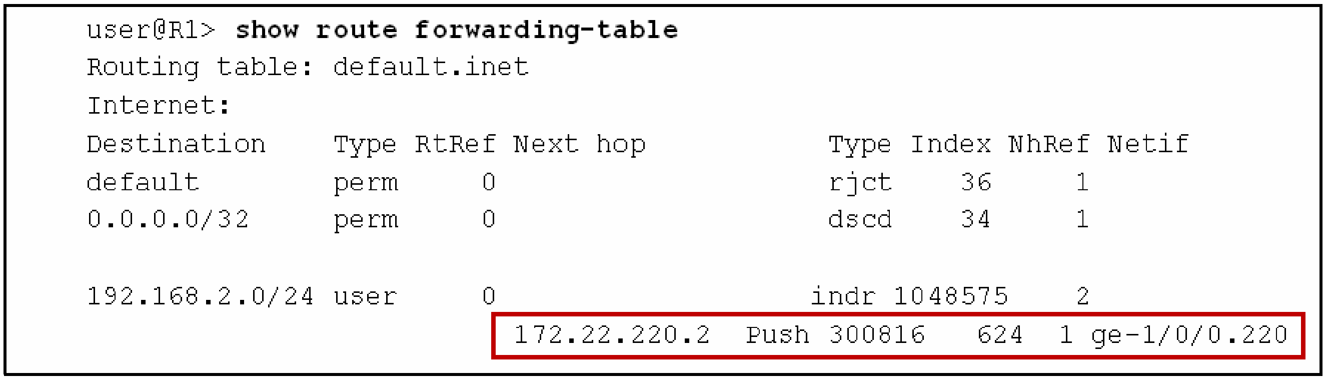


```
user@P1> show mpls lsp extensive
Transit LSP: 4 sessions, 1 detours
192.168.2.2
  From: 192.168.2.1, LSPstate: Up, ActiveRoute: 1
  LSPname: test, LSPpath: Primary
  Suggested label received: -, Suggested label sent: -
  …
  Explct route: 172.22.201.2 172.22.206.2 172.22.222.1
  Record route: 172.22.220.1 <self> 172.22.201.2 172.22.206.2 172.22.222.1
    Detour is Up
    Detour Tspec: rate 0bps size 0bps peak Infbps m 20 M 1500
    Detour adspec: received MTU 1500 sent MTU 1500
    Path MTU: received 1500
    Detour PATH sentto: 172.22.202.2 (ge-1/0/4.202) 17 pkts
    Detour RESV rcvfrom: 172.22.202.2 (ge-1/0/4.202) 14 pkts
    Detour Explct route: 172.22.202.2 172.22.203.2 172.22.204.2 172.22.223.1
    …
```

A previous graphic already showed that fast reroute is enabled for the LSP and that the detour is available at the ingress node. The output on this graphic shows the transit section from **show mpls lsp extensive** for a downstream router that was able to compute a fast reroute detour.

This output shows that a detour branch, used to skip a downstream neighbor is active.

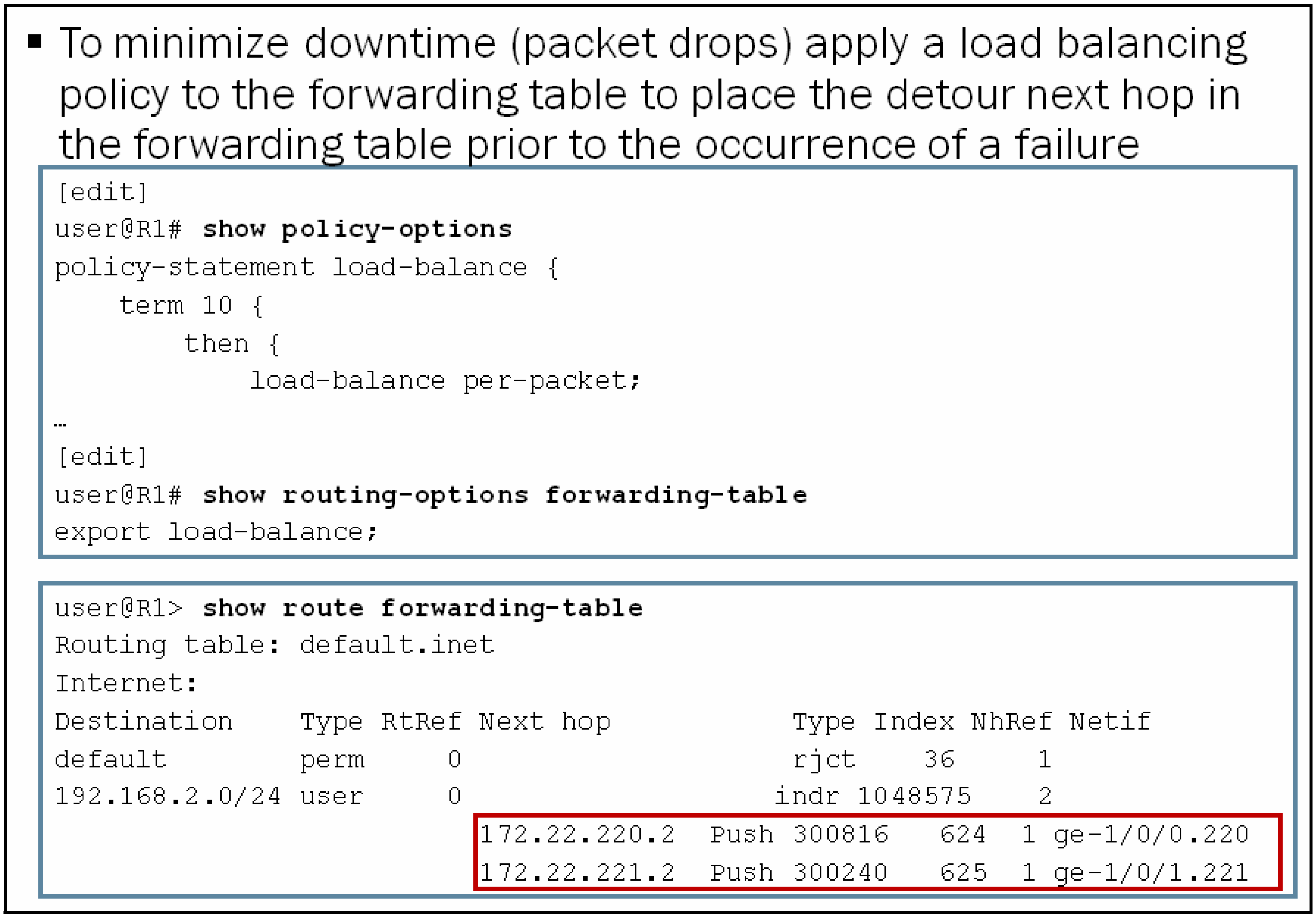
Only Active LSP’s Next Hop Is in the Forwarding Table



```
user@R1> show route forwarding-table
Routing table: default.inet
Internet:
Destination      Type RtRef Next hop            Type Index NhRef Netif
default          perm     0                     rjct    36     1
0.0.0.0/32       perm     0                     dscd    34     1

192.168.2.0/24 user     0                     indr 1048575    2
                          172.22.220.2  Push 300816   624   1 ge-1/0/0.220
```

Even though there is an active fast reroute detour available for an LSP, a router will not install the detour LSP’s next hop in the forwarding table until there is a failure, by default. The output in the graphic shows the single next hop in the forwarding table. When a link failure occurs on ge-1/0/0.220 (on the path of the active and protected LSP) it will take some small time for the routing engine to install the detour next hop in the PFE’s forwarding table.

Minimize Packet Drops



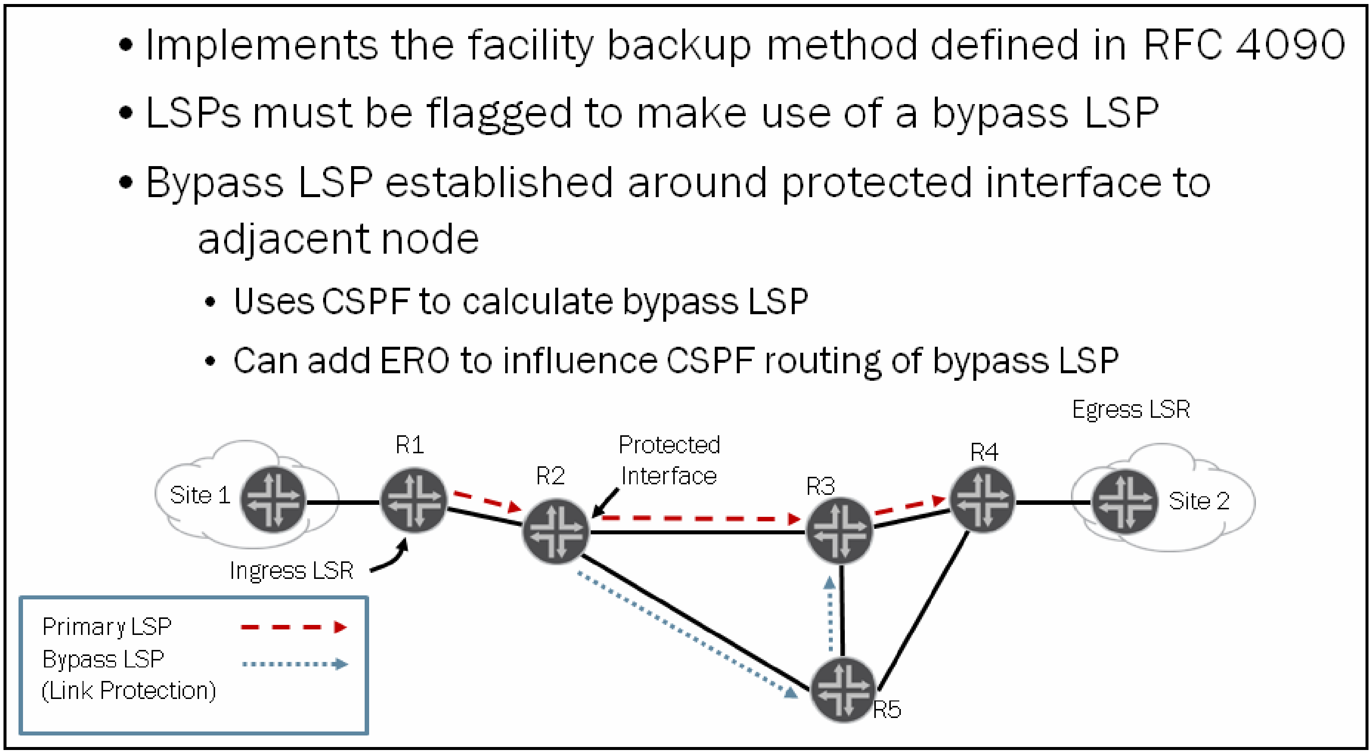
- To minimize downtime (packet drops) apply a load balancing policy to the forwarding table to place the detour next hop in the forwarding table prior to the occurrence of a failure

```
[edit]
user@R1# show policy-options
policy-statement load-balance {
    term 10 {
        then {
            load-balance per-packet;
…
[edit]
user@R1# show routing-options forwarding-table
export load-balance;
```

```
user@R1> show route forwarding-table
Routing table: default.inet
Internet:
Destination    Type RtRef Next hop           Type Index NhRef Netif
default        perm     0                    rjct    36     1
192.168.2.0/24 user     0                   indr 1048575    2
                          172.22.220.2  Push 300816   624   1 ge-1/0/0.220
                          172.22.221.2  Push 300240   625   1 ge-1/0/1.221
```

To override the default behavior of the forwarding table, you can configure a load balancing policy to the forwarding table which will place the fast reroute detour next hop in the PFE's forwarding table prior the occurrence of a failure on the active and protected LSP. The output in the graphic shows that after applying the load balancing policy to the forwarding table, two next hops are available for use by the router's PFE.

Protects Interfaces

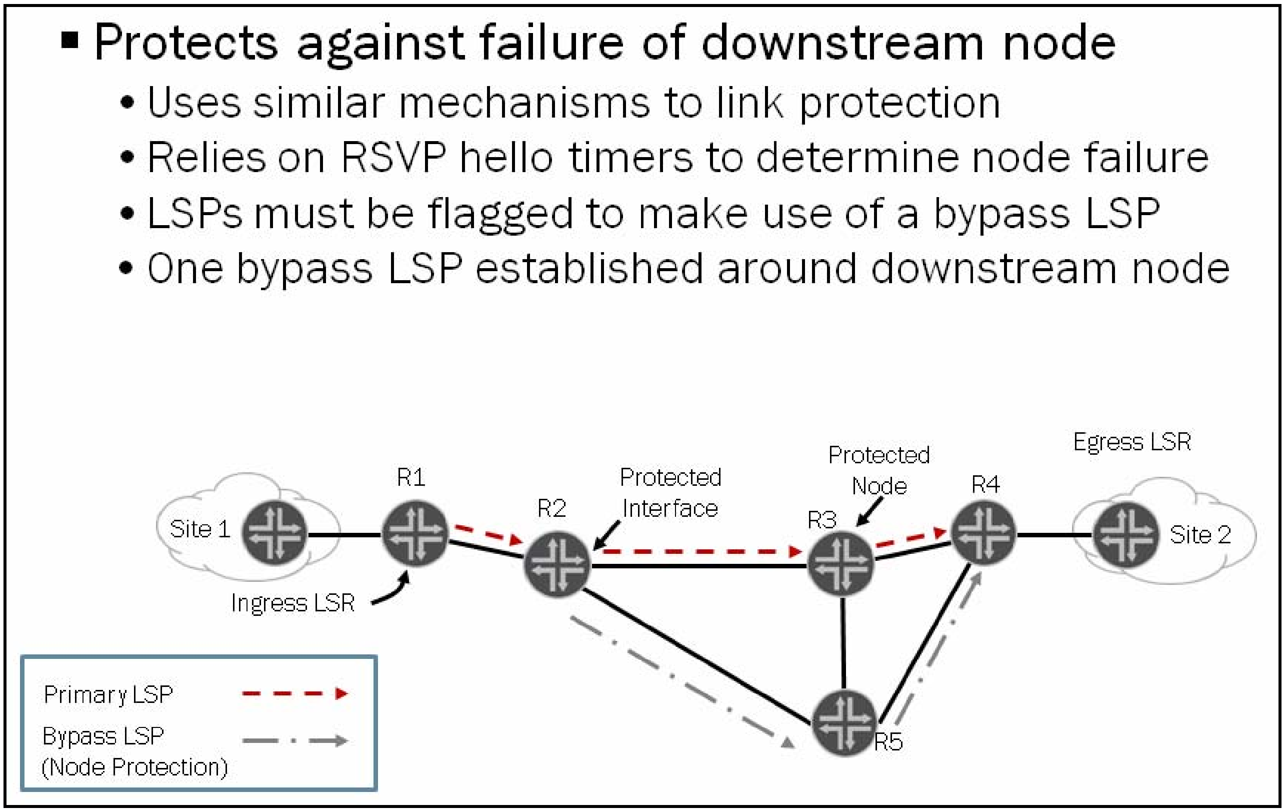


- Implements the facility backup method defined in RFC 4090
- LSPs must be flagged to make use of a bypass LSP
- Bypass LSP established around protected interface to adjacent node
  - Uses CSPF to calculate bypass LSP
  - Can add ERO to influence CSPF routing of bypass LSP

Link protection is the Junos OS nomenclature for the facility backup feature defined in RFC 4090. The link protection feature is interface based, rather than LSP based. The graphic shows how the R2 node is protecting its interface and link to R3 through a bypass LSP that is calculated using CSPF and the node's TED.

While fast reroute attempts to protect the entire path of a given LSP, you can apply link protection on a per-interface basis as needed. LSPs must be tagged for them to make use of a bypass LSP, and you can provide an ERO list to influence the CSPF-based routing of the bypass LSP. Note that a bypass LSP must terminate on the adjacent downstream node, but the bypass LSP can transit other nodes as shown on the graphic.

Node Protection

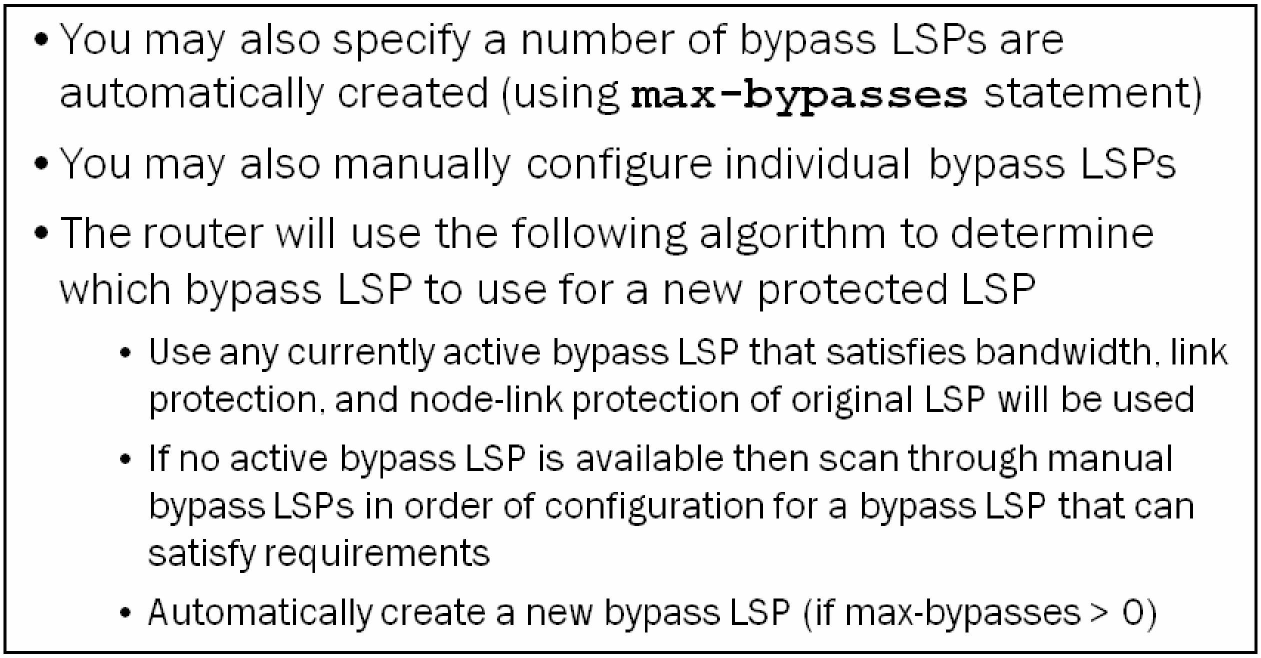


- Protects against failure of downstream node
  - Uses similar mechanisms to link protection
  - Relies on RSVP hello timers to determine node failure
  - LSPs must be flagged to make use of a bypass LSP
  - One bypass LSP established around downstream node

Node protection is the Junos OS nomenclature for the facility backup feature defined in RFC 4090. Node protection uses the same messaging as link protection. The graphic shows that R2 is protecting against the complete failure of R3 through a bypass LSP that is calculated using CSPF.

LSPs must be tagged for node-link protection to make use of the bypass LSPs, and you can provide an ERO list to influence the CSPF-based routing of the bypass LSP.

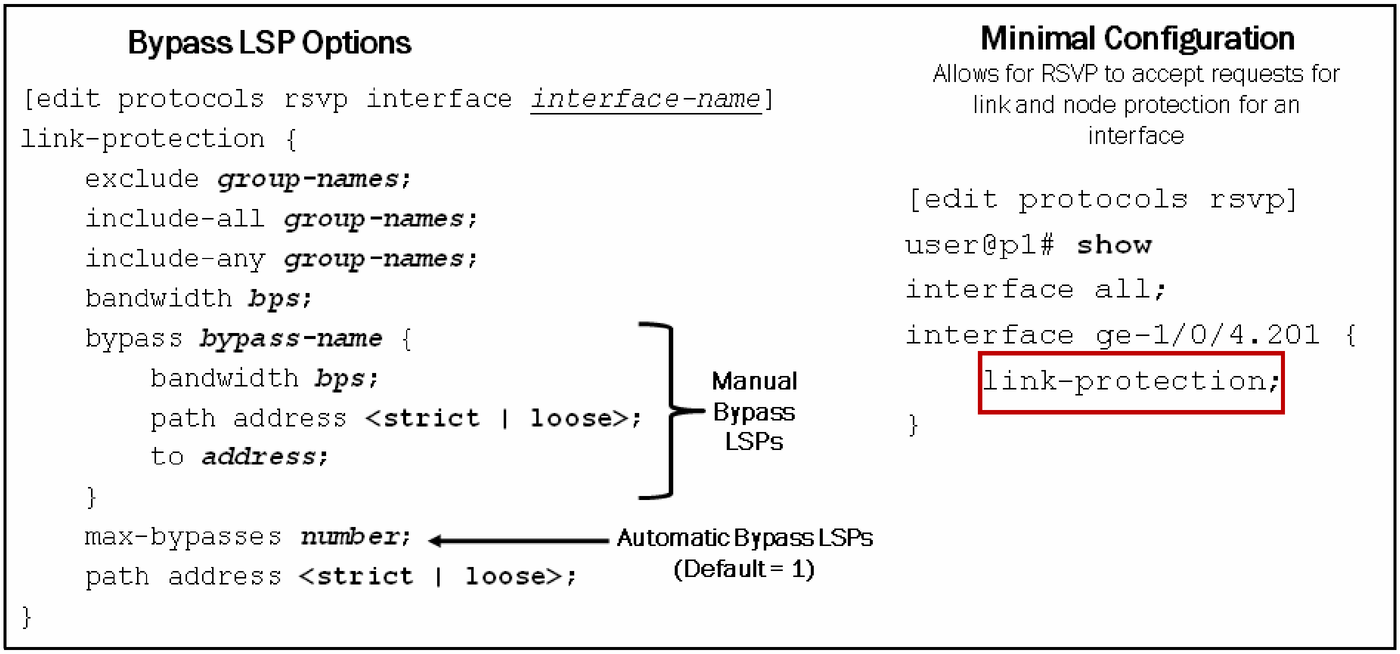
Single Bypass LSPs Are Automatically Created for Protection



- You may also specify a number of bypass LSPs are automatically created (using **`max-bypasses`** statement)
- You may also manually configure individual bypass LSPs
- The router will use the following algorithm to determine which bypass LSP to use for a new protected LSP
  - Use any currently active bypass LSP that satisfies bandwidth, link protection, and node-link protection of original LSP will be used
  - If no active bypass LSP is available then scan through manual bypass LSPs in order of configuration for a bypass LSP that can satisfy requirements
  - Automatically create a new bypass LSP (if max-bypasses > 0)

When an LSP is configured for link protection it will signal to downstream routers that it requires that protection in the Path message. If the LSP will traverse a downstream link that is also configured for link-protection (under `[edit protocols` `rsvp interface` *`interface-name`*`]`) the attached upstream router to the protected link will automatically create a bypass LSP. You can also specify that the router can automatically create more than one bypass LSP. Finally, you can also manually configure a number of bypass LSPs. The graphic shows the algorithm that is used to determine which bypass LSP will be used to protect a new LSP that is signaling that link protection is necessary.

Configuring Bypass LSP



Bypass LSP Options

```
[edit protocols rsvp interface interface-name]
link-protection {
    exclude group-names;
    include-all group-names;
    include-any group-names;
    bandwidth bps;
    bypass bypass-name {
        bandwidth bps;
        path address <strict | loose>;
        to address;
    }
    max-bypasses number;
    path address <strict | loose>;
}
```

Manual
Bypass
LSPs
Automatic Bypass LSPs
(Default = 1)

Minimal Configuration

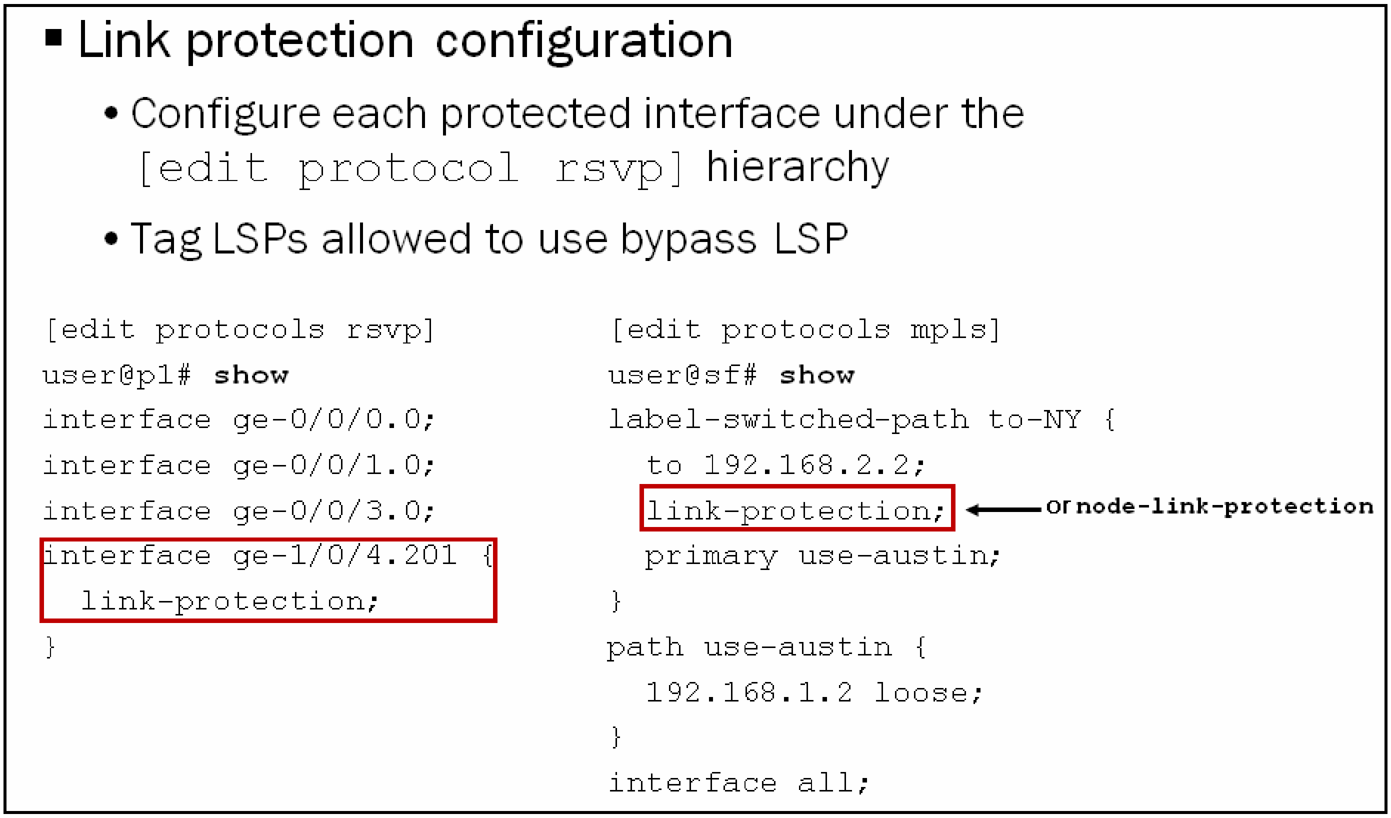
Allows for RSVP to accept requests for link and node protection for an interface

```
[edit protocols rsvp]
user@p1# show
interface all;
interface ge-1/0/4.201 {
    link-protection;
}
```

Bypass LSPs are configured at the `[edit protocols rsvp interface` *`interface-name`*`]` level of the hierarchy. You can specify manual bypass LSP to be used for protection by specifying a bypass LSP by name along with its associate parameters. Also, to give the router the ability to automatically create more the one bypass LSP (the default value) simply use the **`max-bypasses`** statement using a value greater than 1.

The graphic also shows the minimum configuration for a router to support both link or node-link protection on a particular interface.

Link Protection Configuration



- Link protection configuration
  - Configure each protected interface under the `[edit protocol rsvp]` hierarchy
  - Tag LSPs allowed to use bypass LSP

```
[edit protocols rsvp]
user@p1# show
interface ge-0/0/0.0;
interface ge-0/0/1.0;
interface ge-0/0/3.0;
interface ge-1/0/4.201 {
  link-protection;
}
```

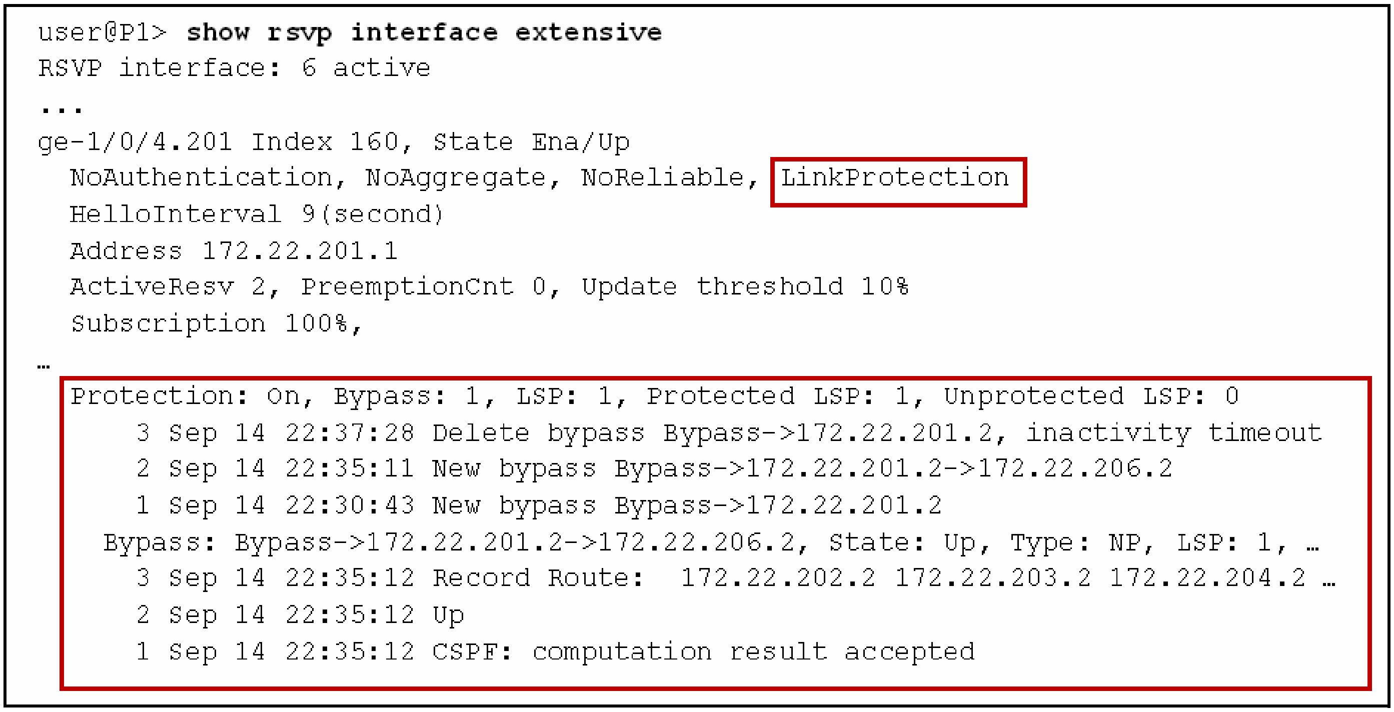
```
[edit protocols mpls]
user@sf# show
label-switched-path to-NY {
  to 192.168.2.2;
  link-protection;
  primary use-austin;
}
path use-austin {
  192.168.1.2 loose;
}
interface all;
```

or **node-link-protection**

You configure the interfaces to be protected under the `[edit protocols rsvp]` hierarchy as shown on the left of the graphic. This configuration allows for but does not cause a bypass LSP to be signaled. Instead, it is a request in the form of a Path message for an LSP requesting protection that causes a bypass LSP to be create. A bypass LSP will only then be created if the node (`p1`, in this case) has the TED entries that are needed to compute the bypass LSP's route.

Note that the mere presence of a bypass LSP does not, in itself, provide protection to the LSPs that might happen to egress the protected interface. You must add the **`link-protection`** keyword to the ingress node's LSP definitions for all LSPs that are expected to benefit from the existence of bypass LSPs.

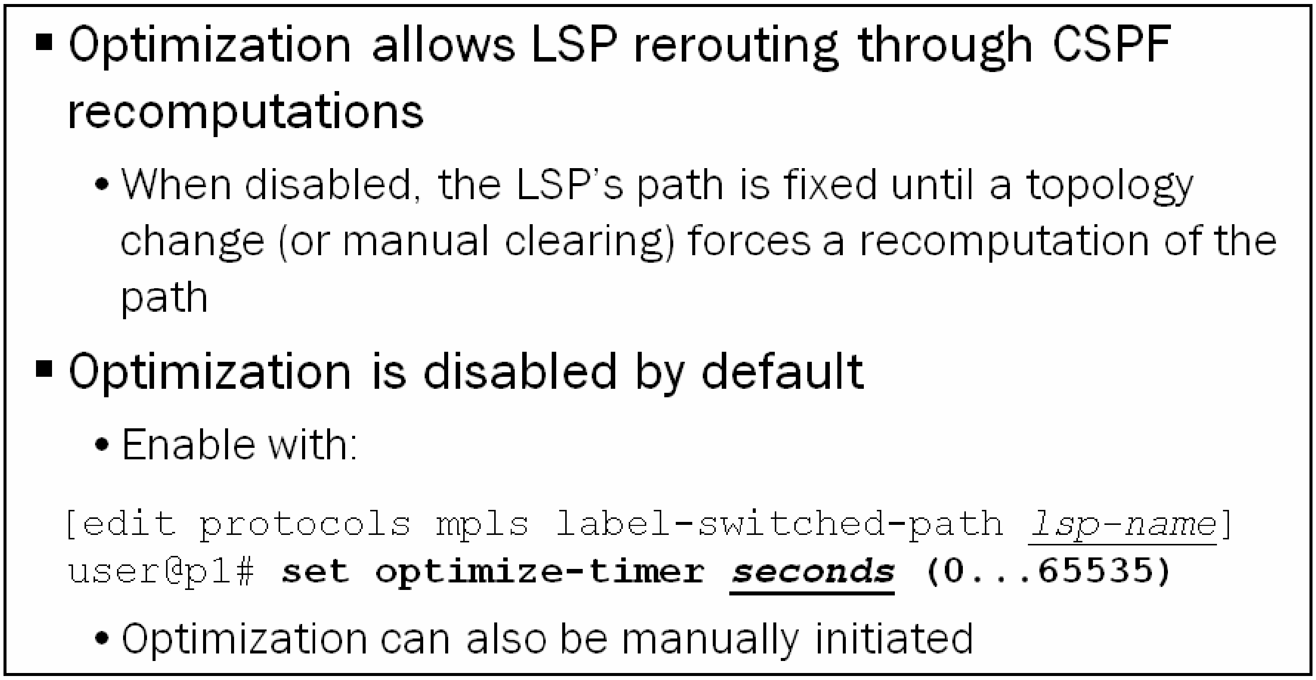
Monitoring Link and Node Protection



```
user@P1> show rsvp interface extensive
RSVP interface: 6 active
...
ge-1/0/4.201 Index 160, State Ena/Up
  NoAuthentication, NoAggregate, NoReliable, LinkProtection
  HelloInterval 9(second)
  Address 172.22.201.1
  ActiveResv 2, PreemptionCnt 0, Update threshold 10%
  Subscription 100%,
…
  Protection: On, Bypass: 1, LSP: 1, Protected LSP: 1, Unprotected LSP: 0
      3 Sep 14 22:37:28 Delete bypass Bypass->172.22.201.2, inactivity timeout
      2 Sep 14 22:35:11 New bypass Bypass->172.22.201.2->172.22.206.2
      1 Sep 14 22:30:43 New bypass Bypass->172.22.201.2
    Bypass: Bypass->172.22.201.2->172.22.206.2, State: Up, Type: NP, LSP: 1, …
      3 Sep 14 22:35:12 Record Route:  172.22.202.2 172.22.203.2 172.22.204.2 …
      2 Sep 14 22:35:12 Up
      1 Sep 14 22:35:12 CSPF: computation result accepted
```

The graphic shows how the output of a **show rsvp interface extensive** command indicates the presence of a bypass LSP at the transit node. Note that link protection is an RSVP feature, and as a result, the resulting bypass LSPs are not listed in the output of **show mpls lsp** commands.

LSP Rerouting



- Optimization allows LSP rerouting through CSPF recomputations
  - When disabled, the LSP's path is fixed until a topology change (or manual clearing) forces a recomputation of the path
- Optimization is disabled by default
  - Enable with:

```
[edit protocols mpls label-switched-path lsp-name]
user@p1# set optimize-timer seconds (0...65535)
```

  - Optimization can also be manually initiated

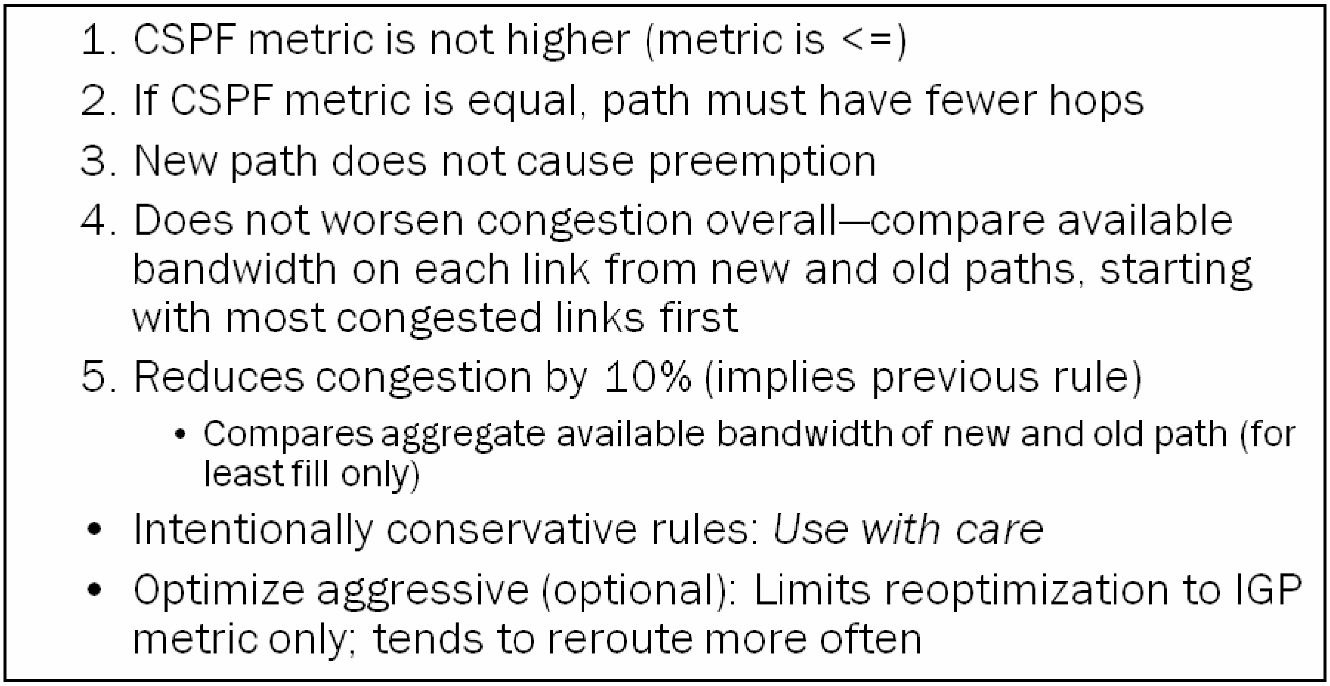
Once an LSP is established, changes in topology or available resources might result in the existing path becoming suboptimal. A subsequent CSPF recomputation might result in the determination that a better path is now available. When optimization is enabled, LSPs can be rerouted as a result of periodic CSPF recomputations. Without optimization the LSP has a fixed path and cannot take advantage of newly available network resources, at least until the next topology change or operator induced clearing breaks the LSP and forces recomputation of a new path. Note that optimization is not related to failover; a new path is always computed when topology failures occur that disrupt an established path.

Enable Optimization

Because of the potential system overhead involved, you should carefully consider the frequency at which routers perform optimization runs. By default, the `optimize-timer` is set to 0 (that is, it is disabled). LSP optimization is only meaningful for CSPF LSPs. Due to statistical characteristics that arise in large topologies, a network can effectively *synchronize* and end up trying to recalculate all LSPs at the same time when all reoptimization timers are set the same. To prevent this behavior, the LSP reoptimization timer is modified to include a randomization factor when recalculating LSPs. The randomization factor is fixed and cannot be modified.

Note that you can manually trigger optimization with the operational mode **`clear mpls lsp optimize`** command.

Optimization Rules



1. CSPF metric is not higher (metric is <=)
2. If CSPF metric is equal, path must have fewer hops
3. New path does not cause preemption
4. Does not worsen congestion overall—compare available bandwidth on each link from new and old paths, starting with most congested links first
5. Reduces congestion by 10% (implies previous rule)
    - Compares aggregate available bandwidth of new and old path (for least fill only)

- Intentionally conservative rules: *Use with care*
- Optimize aggressive (optional): Limits reoptimization to IGP metric only; tends to reroute more often

By default, an LSP can only have its path optimized when all of the following criteria are met. These rules are intentionally conservative as stability is better than being optimal in many cases:

1. The new path is not higher in CSPF metric. (The metric for the old path is updated during computation, so if a recent link metric changed somewhere along the old path, it is accounted for.)
2. If the new path has the same CSPF metric, it must not have more hops.
3. The new path does not cause preemption. (This is to reduce the ripple effect of one preemption causing yet more preemption.)
4. The new path does not worsen congestion overall. This is determined by comparing the percentage of available bandwidth on each link traversed by the new and old paths, starting from the most congested links.

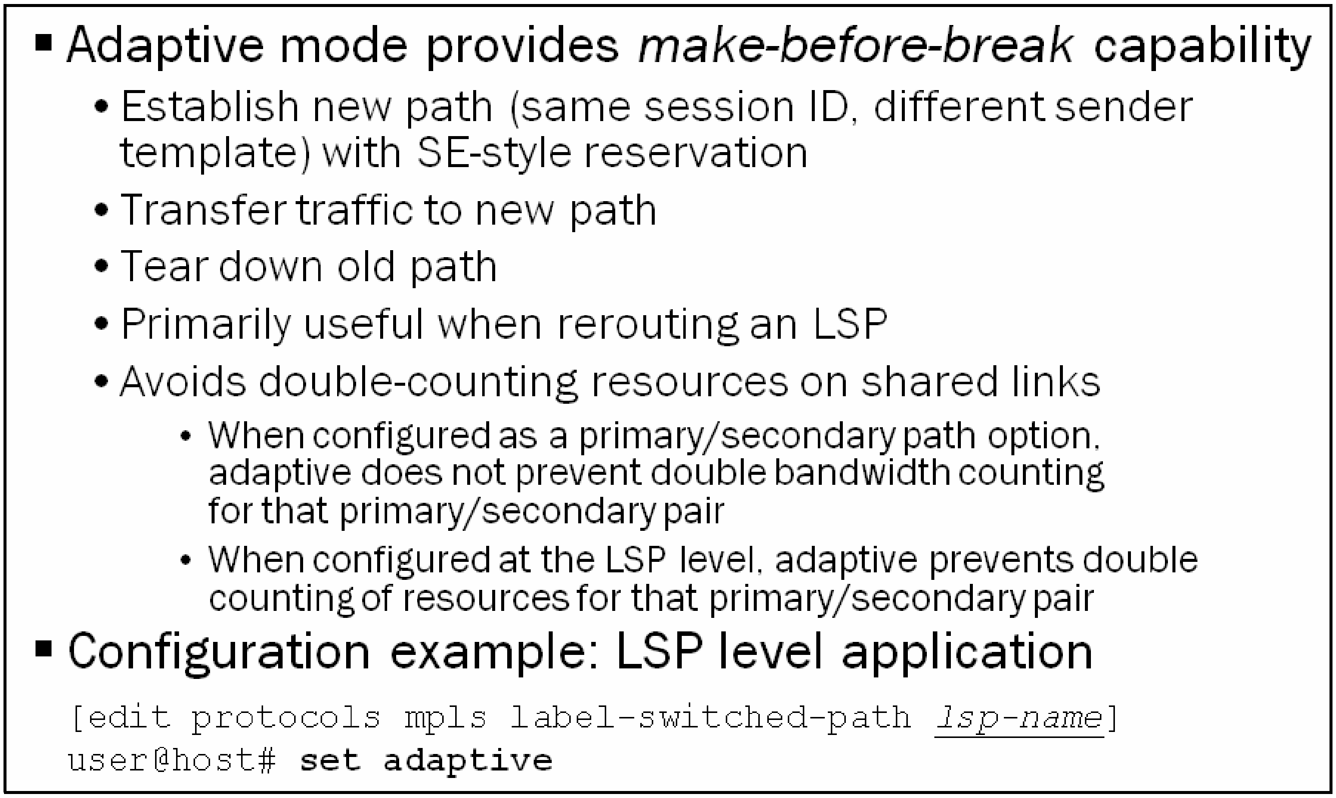
When all the above conditions are met, then if the new path has a lower CSPF metric, it is accepted. If the new path has an equal CSPF metric and lower hop count, it is accepted. If you choose least fill as a load-balancing algorithm and if the new path reduces congestion by at least 10 percent aggregated over all links it traversed, it is accepted. For random or most-fill algorithms, this rule does not apply. Otherwise, the new path is rejected.

Here is a sample calculation to help explain how links are compared. You compare the percentage of available bandwidth on each link traversed by the new and old paths, starting from the most congested links. Assume that Path 1 (active) has four hops with availability: hop 1: 10%; hop 2: 15%; hop 3: 25%; hop 4:15%. The *new candidate* path has a lower IGP metric, will not cause preemption, and is three hops away with availability as follows if the new LSP was implemented, and the old LSP removed: hop 1: 10%; hop 2: 50%; hop 3: 15%. The active path will be sorted as 10, 15, 15, 25. The candidate path will be sorted as 10, 15, 50, 100. The two paths will be compared, on an item by item basis. 100>25, 50>15, 15=15 10=10. The candidate path does not worsen congestion. Every single link in the candidate path must have available bandwidth >= those in the active. For example, all four candidate links were >= available bandwidth of the original path (do not forget that a pseudo 100 availability was used for the final link).

What if only three out of four links had better availability? In this case, the congestion is considered *worsened* so the reoptimization is not accepted.

To force the reoptimization to be based upon IGP metric only, enable the **`optimize-aggressive`** keyword. This setting negates the tests outlined in Steps 2, 3, and 4 on the previous page. You can manually trigger aggressive optimization with a **`clear mpls optimize-aggressive`** command. The LSP must still comply with the original CSPF constraints when optimized aggressively, but no attention is paid to available bandwidth ratios, as explained in the sample calculation above, so you will tend to see more LSP rerouting when operating in aggressive mode.

Adaptive Provides Make Before Break



- Adaptive mode provides *make-before-break* capability
  - Establish new path (same session ID, different sender template) with SE-style reservation
  - Transfer traffic to new path
  - Tear down old path
  - Primarily useful when rerouting an LSP
  - Avoids double-counting resources on shared links
    - When configured as a primary/secondary path option, adaptive does not prevent double bandwidth counting for that primary/secondary pair
    - When configured at the LSP level, adaptive prevents double counting of resources for that primary/secondary pair
- Configuration example: LSP level application

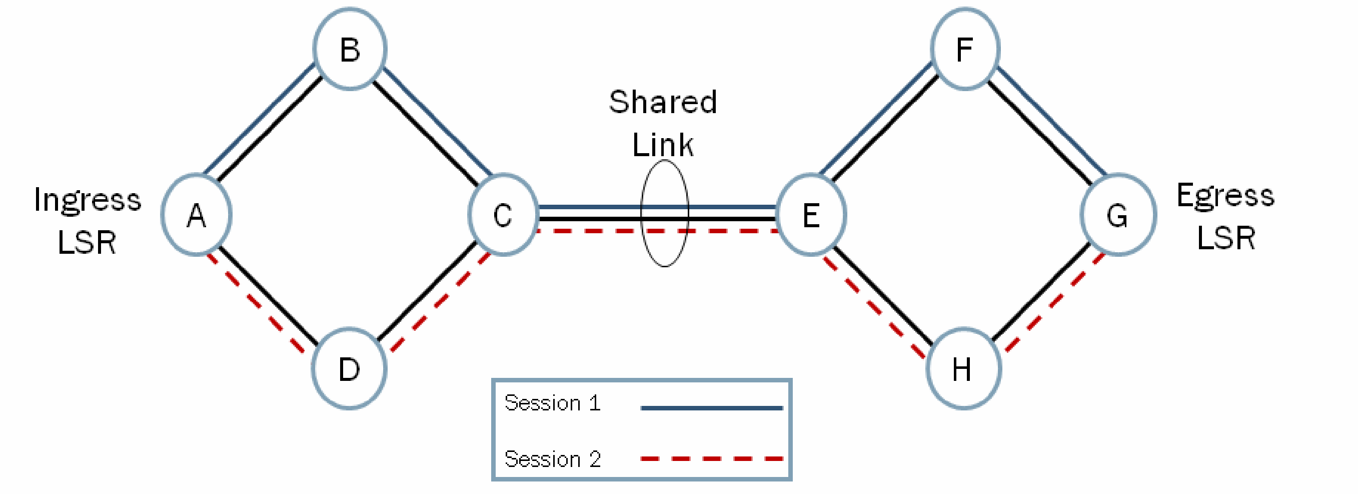
```
[edit protocols mpls label-switched-path lsp-name]
user@host# set adaptive
```

You can configure an LSP to use a shared explicit (SE) style reservation by setting it to be adaptive. While any LSP can be established with an SE-style reservation, this capability is most useful when attempting to reroute an LSP. When an LSP is adaptive, it holds onto existing resources until the new path is successfully established and traffic is cut over to the new path. To retain its resources, an adaptive LSP does the following: 1) Maintains existing paths and allocated bandwidths (which ensures that the existing path is not torn down prematurely and allows the current traffic to continue flowing while the new path is being set up), and 2) Avoids double-counting for links that share the new and old paths. Double-counting occurs when an intermediate router does not recognize that the new and old paths belong to the same LSP and counts them as two separate LSPs, requiring separate bandwidth allocations. If some links are close to saturation, double-counting might cause the setup of the new path to fail. By default, adaptive behavior is disabled.

Configuration

To define an adaptive LSP, include the **`adaptive`** statement when defining the LSP, as shown on the graphic. When **`adaptive`** is specified at the `label-switched-path` *`lsp-name`* hierarchy and sufficient resources exist to establish both LSPs, the primary and all secondary paths share the same bandwidth reservation (the higher of all bandwidths defined). When **`adaptive`** is included at the `primary` or `secondary` hierarchy level, the SE-style reservation behavior is enabled only for the path (primary or secondary) that is so configured. When specified at the primary and secondary level, the corresponding primary and secondary paths are considered as separate adaptive settings, and therefore, their resources are double-counted.

RSVP Reservation Styles



FF Reservation Style: (default)

- Each session/sender has its own identity
- Each session has its own bandwidth reservation

SE Reservation Style: (adaptive)

- Each session/sender has its own identity
- Sessions share a single bandwidth reservation

Fixed filter (FF): The FF-style reservation is commonly used for applications where traffic from each sender is likely to be concurrent and independent. Each of the individual senders is identified by an IP address and an internal identification number—an LSP ID.

When used with MPLS, the FF style allows the establishment of multiple, parallel, unicast, point-to-point LSPs to support load balancing. If the LSPs share a common link, the total amount of reserved bandwidth for the shared link is the sum of the reservations for the individual senders. By default, Junos OS uses the FF style.

Shared Explicit (SE): SE reservations share the bandwidth of the largest request across any shared links. The SE-style reservation is critical for supporting the ability to reroute an LSP with the make-before-break capability because on shared links, if reservations are counted twice, the router's admission control function could reject the new LSP due to a lack of resources. The SE reservation style permits the old and new LSPs to share resources over shared links. You can configure SE-style reservations with the **`adaptive`** keyword under the LSP or primary/secondary path configuration hierarchy.

It is extremely important that the flow of subscriber traffic is not disrupted when an LSP is rerouted. A smooth transition requires support for a concept called *make before break*—the new LSP tunnel must be established and the traffic transferred to it before the old LSP tunnel is torn down. One of the benefits of RSVP signaling is that the legacy SE reservation style provides an elegant solution to this challenging problem.

Establishing the Initial LSP Tunnel: In the initial Path message, the ingress LSR:

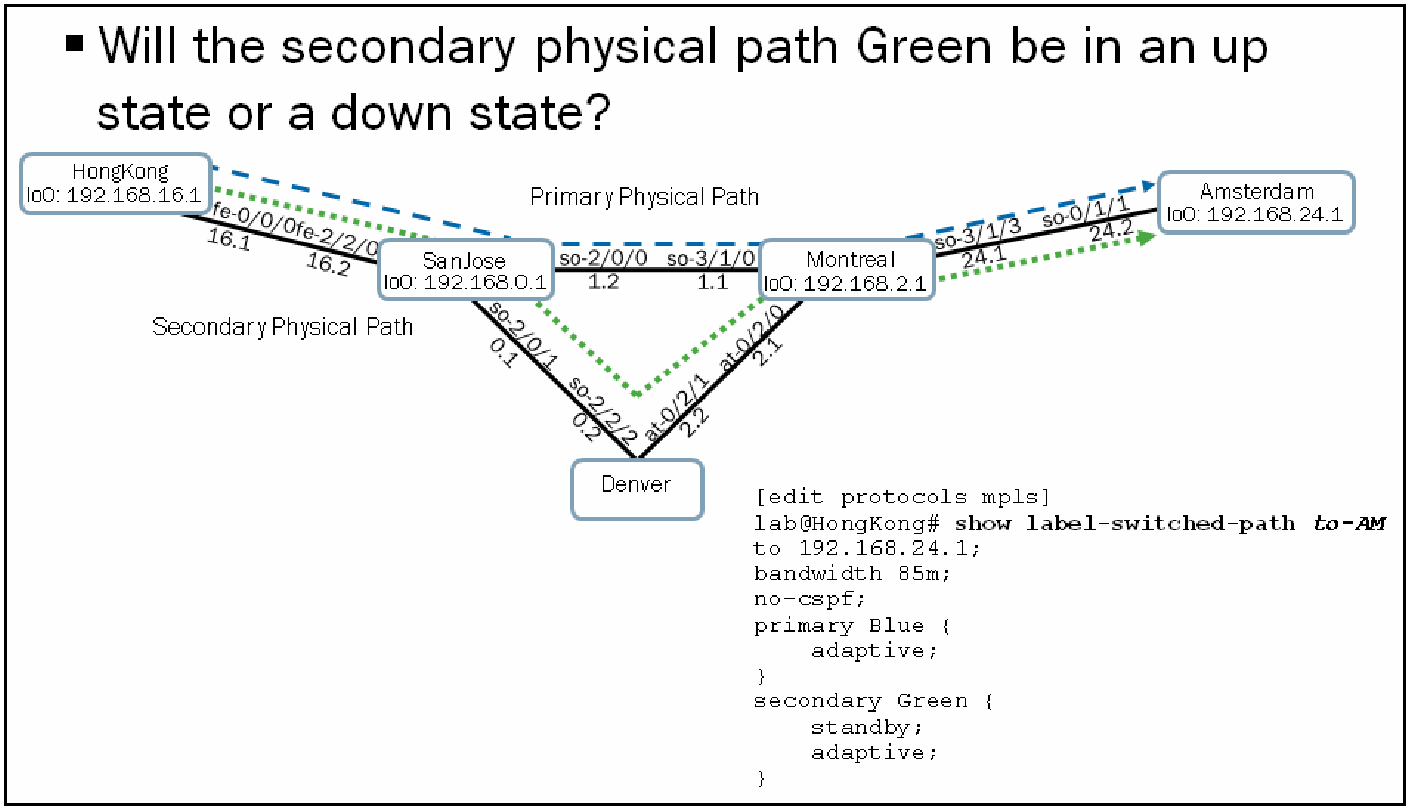
1. Forms a LSP_TUNNEL_IPv4 SESSION object that uniquely identifies the LSP tunnel. The LSP_TUNNEL_IPv4 SESSION object contains: a) IP version 4 (IPv4) address of the egress node for the LSP tunnel, b) Tunnel_ID that remains constant for the life of the LSP tunnel between the ingress and egress LSRs, and c) the Extended_Tunnel_ID that identifies the ingress node of the tunnel (that is, the ingress router's IPv4 address).
2. Sets the *ingress node might reroute bit* of the SESSION_ATTRIBUTE object to request that the egress LSR use the SE reservation style.
3. Forms a SENDER_TEMPLATE object that contains: a) The IPv4 address of the sender (ingress) node, and b) an LSP_ID that can be changed in the future to allow the ingress LSR to appear as a different sender so it can share resources with itself if the LSP needs to be rerouted (see the LSP_ID field of the LSP_TUNNEL_IPv4 C-type extension for the SENDER_TEMPLATE and FILTER_SPEC objects).
4. Upon receipt of the Path message, the egress LSR sends a Resv message with a SE reservation style toward the ingress node. When the ingress LSR receives the Resv message, the initial LSP tunnel is established with an SE reservation style.

Establishing the Rerouted LSP Tunnel: When the ingress LSR wants to increase the bandwidth or change the path for an existing LSP, it transmits a new Path message. During the reroute operation, the ingress LSR must appear as two different senders to the RSVP session. This is achieved by including a new LSP_ID in the SENDER_TEMPLATE and the FILTER_SPEC objects. In the new Path message, the ingress LSR:

1. Creates an EXPLICIT_ROUTE (ERO) object for the new LSP tunnel.
2. Uses the existing LSP_TUNNEL_IPv4 SESSION object to identify the LSP that will be rerouted.
3. Picks a new LSP_ID and creates a new SENDER_TEMPLATE. By selecting a new LSP_ID for the SENDER_TEMPLATE, the ingress LSR appears as a different sender to the RSVP session.
4. The ingress LSR transmits the new Path message toward the egress LSR. (However, the ingress LSR continues to use the old LSP tunnel to forward traffic and continues to refresh the original Path message.)
5. The egress LSR responds to the receipt of the new Path message with a Resv message that contains a number of RSVP objects, including: a) A LABEL object to support the upstream on demand label distribution process, and b) an SE reservation style object.

On links not shared by the old and new LSP tunnels, the new Path/Resv message pair is treated as a new conventional LSP setup. However, on links that are traversed by both the old and the new LSP tunnels, the LSP_TUNNEL_IPv4 SESSION object and SE reservation style allow the new LSP tunnel to be established so that it shares resources with the old LSP tunnel. This eliminates the double counting problem on shared links. After the ingress LSR receives the Resv message for the new LSP, it can begin using the new LSP tunnel to forward traffic. The ingress LSR should send a PathTear message for the old LSP tunnel to remove its state from intermediate LSRs.

Check Your Knowledge: Adaptive



- Will the secondary physical path Green be in an up state or a down state?

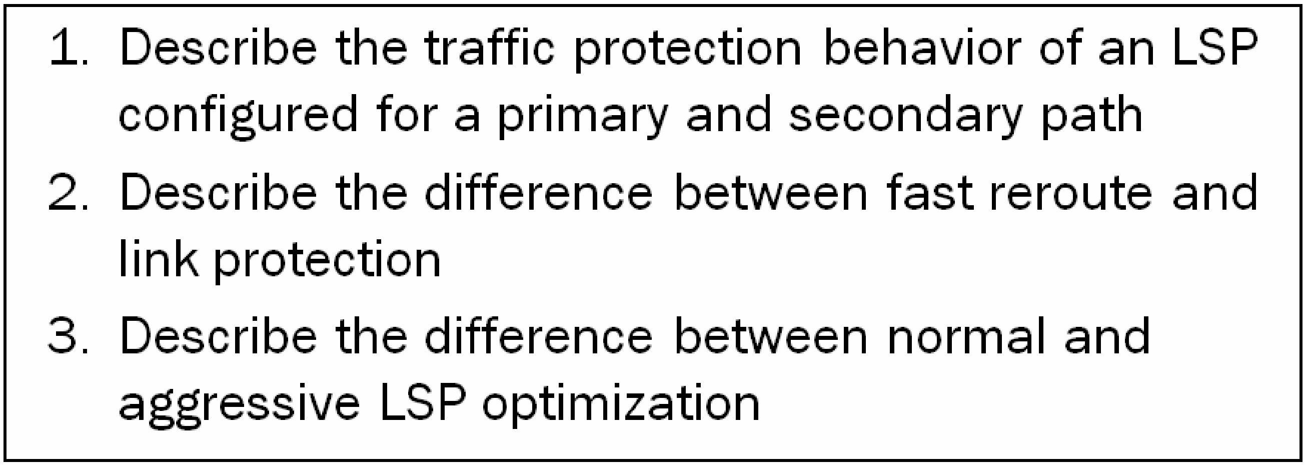
```
[edit protocols mpls]
lab@HongKong# show label-switched-path to-AM
to 192.168.24.1;
bandwidth 85m;
no-cspf;
primary Blue {
    adaptive;
}
secondary Green {
    standby;
    adaptive;
}
```

In the example on the graphic, the secondary physical path Green will be in a down state. Although the `adaptive` keyword indicates resources should not be double-counted, this behavior only applies to LSPs that are considered to belong to a *common* session.

Because the `adaptive` keyword was specified at both the `primary` and `secondary` levels, the result is two independent sessions that both signal an SE-style reservation. The fact that the two sessions are seen as being independent means that, despite the SE-style reservations, the bandwidth requirements for the `primary` and `secondary` paths will be double-counted. In this specific topology shown on the graphic, this causes the secondary path to fail. Note that application of the **`adaptive`** keyword at the LSP level, as shown here, allows the establishment of both primary and secondary paths with a single session ID that does not incur double bandwidth counting.

```
[edit protocols mpls]
user@HongKong# show label-switched-path to-AM
to 192.168.24.1;
bandwidth 85m;
no-cspf;
adaptive;
primary Blue;
secondary Green {
    standby;
    adaptive;
}
```

Review Questions



1. Describe the traffic protection behavior of an LSP configured for a primary and secondary path
2. Describe the difference between fast reroute and link protection
3. Describe the difference between normal and aggressive LSP optimization

Answers to Review Questions

1.

When a primary is active and there is a failure along the path the ingress router will signal the secondary LSP to provide protection for traffic while the primary is down.

2.

Fast reroute protects an entire LSP from failures along the path. Link protection provides protection for the failure of a single link.

3.

Aggressive optimization only takes IGP metric into consideration. Normal optimization also takes into consideration number of hops, congestion, and preemption.

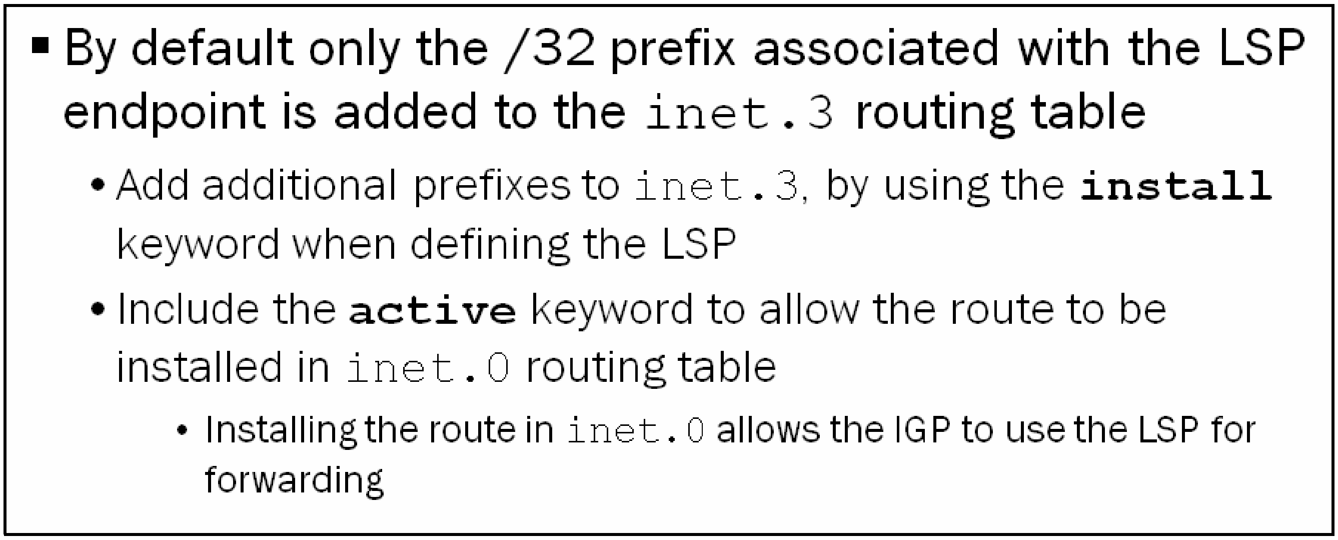
JNCIS-SP Study Guide—Part 3

Chapter 5: Miscellaneous MPLS Features

This Chapter Discusses:

- The purpose of several miscellaneous MPLS features; and
- The features that will meet given design requirements.

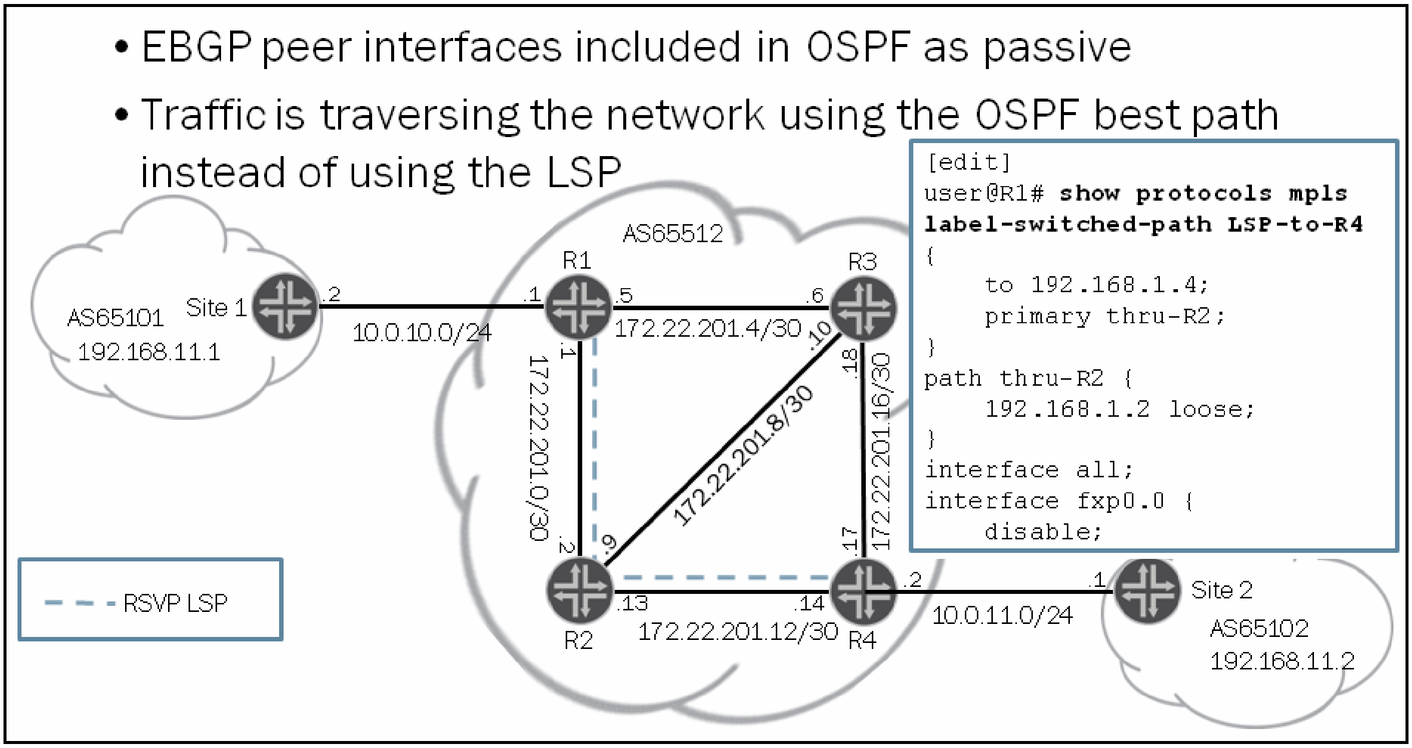
Default Routing Table Behavior



- By default only the /32 prefix associated with the LSP endpoint is added to the `inet.3` routing table
  - Add additional prefixes to `inet.3`, by using the **`install`** keyword when defining the LSP
  - Include the **`active`** keyword to allow the route to be installed in `inet.0` routing table
    - Installing the route in `inet.0` allows the IGP to use the LSP for forwarding

By default only the /32 prefix associated with the egress point of the label-switched path (LSP) is installed in the inet.3 routing table. You can add additional prefixes to the `inet.3` routing table by using the **`install <`*`prefix`*`>`** option under the LSP you are configuring. You can also add this prefix to the `inet.0` routing table by including the **`active`** tag. Adding the prefix to `inet.0` allows the LSP to be used by BGP as well as the interior gateway protocol (IGP). We will discuss these options in more detail in the following section.

Adding Prefixes Example

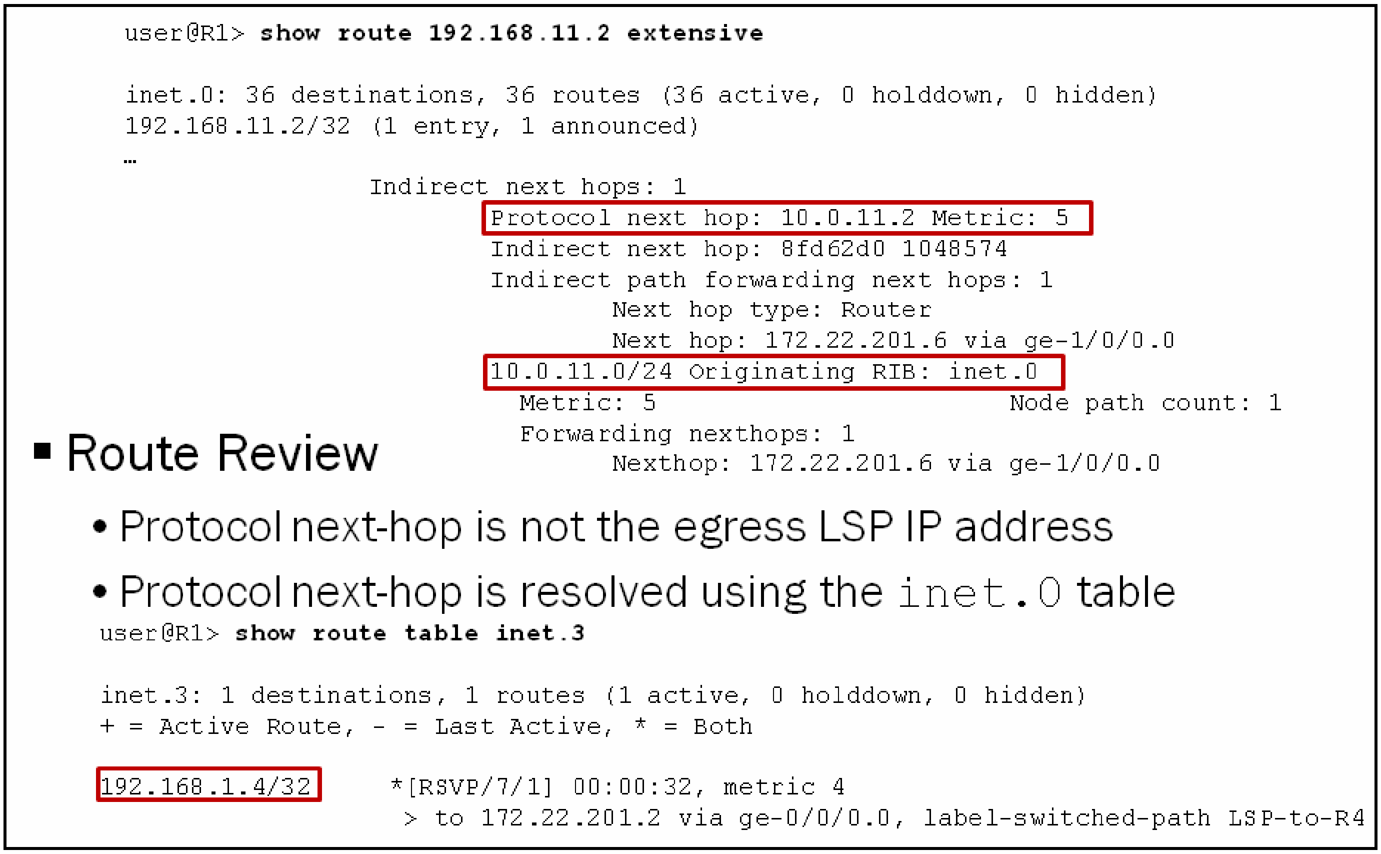


- EBGP peer interfaces included in OSPF as passive
- Traffic is traversing the network using the OSPF best path instead of using the LSP

```
[edit]
user@R1# show protocols mpls
label-switched-path LSP-to-R4
{
    to 192.168.1.4;
    primary thru-R2;
}
path thru-R2 {
    192.168.1.2 loose;
}
interface all;
interface fxp0.0 {
    disable;
```

We will be using the example to demonstrate a circumstance where you may need to utilize the **install *prefix*** option. In the topology reflected on the graphic we are connecting site 1 with site 2. There is a LSP that has been signaled from R1 to R4 that must traverse R2 because of the loose Explicit Route Object (ERO) that has been configured. The external facing interfaces on R1 and R4 have been included in the IGP as passive interfaces so that internal BGP (IBGP) could resolve the next hop for EBGP routes learned from site 1 and site 2. The traffic that is sent from site 1 to site 2 is using the IGP best path and not the LSP as expected.

Route Review



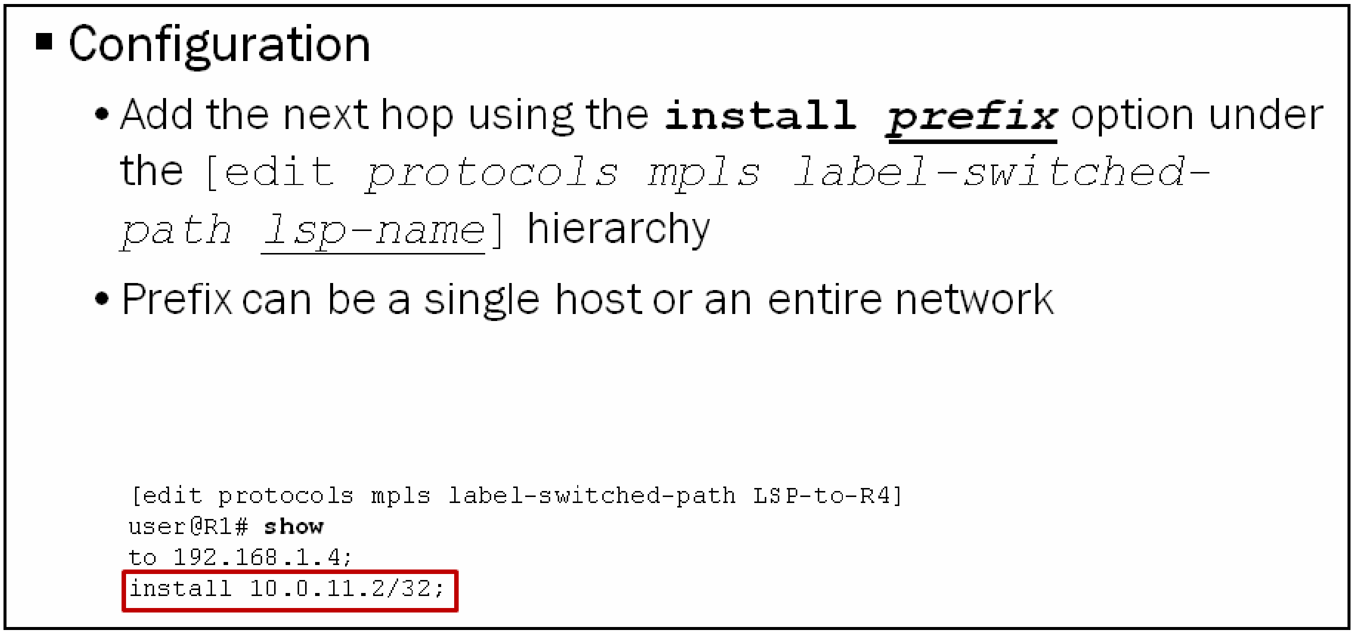
```
user@R1> show route 192.168.11.2 extensive

inet.0: 36 destinations, 36 routes (36 active, 0 holddown, 0 hidden)
192.168.11.2/32 (1 entry, 1 announced)
…
                Indirect next hops: 1
                        Protocol next hop: 10.0.11.2 Metric: 5
                        Indirect next hop: 8fd62d0 1048574
                        Indirect path forwarding next hops: 1
                                Next hop type: Router
                                Next hop: 172.22.201.6 via ge-1/0/0.0
                        10.0.11.0/24 Originating RIB: inet.0
                          Metric: 5                     Node path count: 1
                          Forwarding nexthops: 1
                                Nexthop: 172.22.201.6 via ge-1/0/0.0
```

- Route Review
  - Protocol next-hop is not the egress LSP IP address
  - Protocol next-hop is resolved using the `inet.0` table

```
user@R1> show route table inet.3

inet.3: 1 destinations, 1 routes (1 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.4/32     *[RSVP/7/1] 00:00:32, metric 4
                    > to 172.22.201.2 via ge-0/0/0.0, label-switched-path LSP-to-R4
```

Start by looking at your EBGP routes to determine what the next hop is. As displayed in the graphic, we do have an active route. The protocol next hop for this route is 10.0.11.2 and you can tell that the next hop was resolved in the `inet.0` table. You can also see that there is not an entry in the inet.3 routing table. This is because the LSP terminates at the loopback address of R4. Because BGP will first try to resolve the next hop in the `inet.3` table, we need to add this prefix in order to route the traffic through the LSP.

Configuring the Install Option

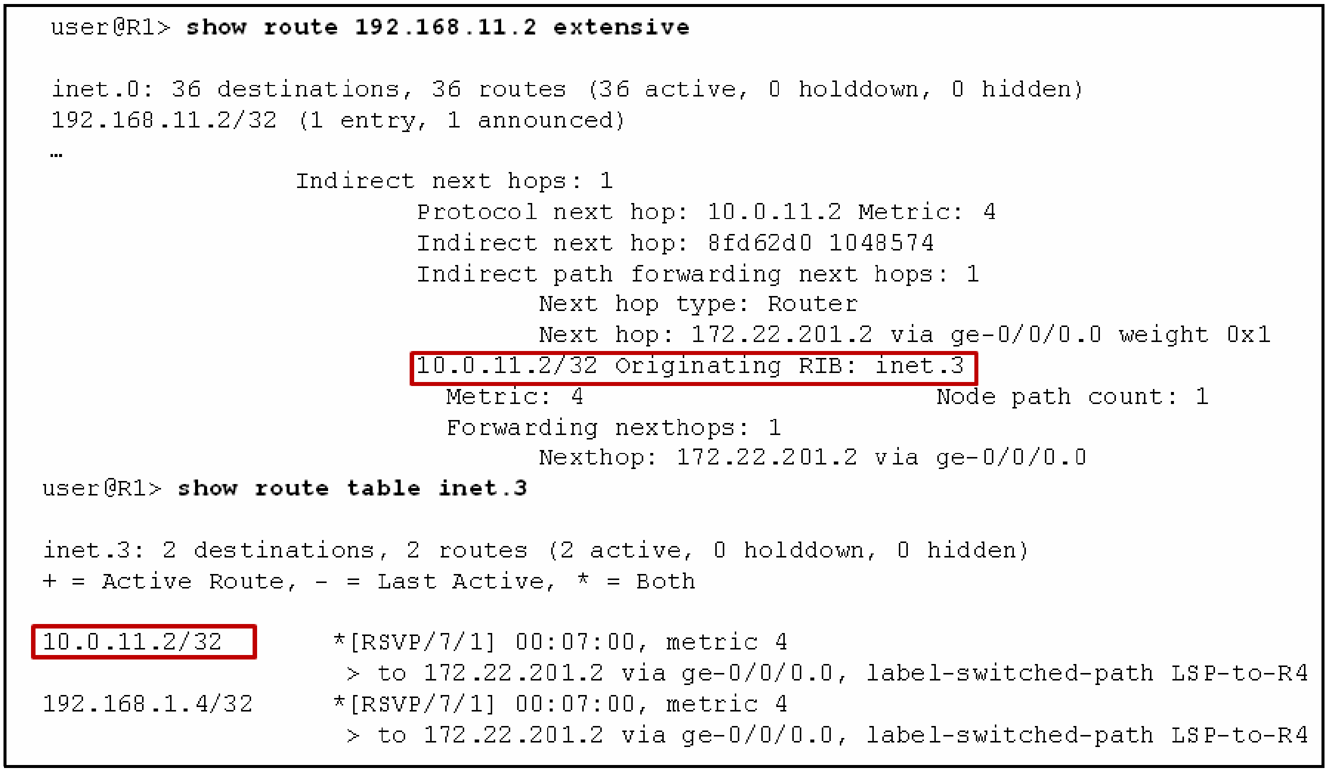


- Configuration
  - Add the next hop using the **`install`** ***`prefix`*** option under the `[edit` *`protocols mpls label-switched-path lsp-name`*`]` hierarchy
  - Prefix can be a single host or an entire network

```
[edit protocols mpls label-switched-path LSP-to-R4]
user@R1# show
to 192.168.1.4;
install 10.0.11.2/32;
```

The **`install`** option is configured under the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` hierarchy. By specifying the **`install`** ***`prefix`*** statement under the specific LSP, the Junos OS knows what LSP to associate the prefix with. With the **install** option you can indicate and single host with a /32 or you can include an entire network. These routes will be installed into the `inet.3` routing table. When BGP needs to resolve a next hop to the address you installed it will use this route over and IGP route that may exist. The sample configuration on the graphic shows that the 10.0.11.2/32 prefix is now installed in the `inet.3` routing table. Remember from the previous graphic that 10.0.11.2/32 is the BGP protocol next hop for our routes to site 2.

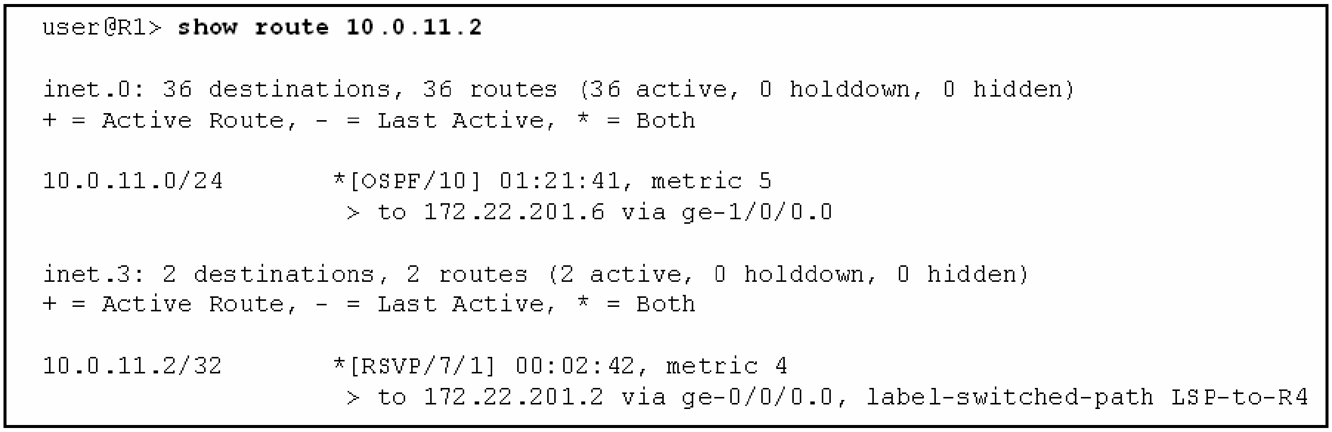
Verify the Route Changes



```
 user@R1> show route 192.168.11.2 extensive

 inet.0: 36 destinations, 36 routes (36 active, 0 holddown, 0 hidden)
 192.168.11.2/32 (1 entry, 1 announced)
 …
                 Indirect next hops: 1
                         Protocol next hop: 10.0.11.2 Metric: 4
                         Indirect next hop: 8fd62d0 1048574
                         Indirect path forwarding next hops: 1
                                 Next hop type: Router
                                 Next hop: 172.22.201.2 via ge-0/0/0.0 weight 0x1
                         10.0.11.2/32 Originating RIB: inet.3
                           Metric: 4                       Node path count: 1
                           Forwarding nexthops: 1
                                 Nexthop: 172.22.201.2 via ge-0/0/0.0
user@R1> show route table inet.3

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.11.2/32       *[RSVP/7/1] 00:07:00, metric 4
                    > to 172.22.201.2 via ge-0/0/0.0, label-switched-path LSP-to-R4
192.168.1.4/32     *[RSVP/7/1] 00:07:00, metric 4
                    > to 172.22.201.2 via ge-0/0/0.0, label-switched-path LSP-to-R4
```

After making the appropriate configuration changes it is important to verify the results. As the graphic shows, the protocol next hop for the BGP route to site 2 is being resolved in the `inet.3` routing table. The graphic also show the 10.0.11.2/32 route is installed in the `inet.3` routing table. The traffic from site 1 to site 2 is going to traverse the network using the LSP.

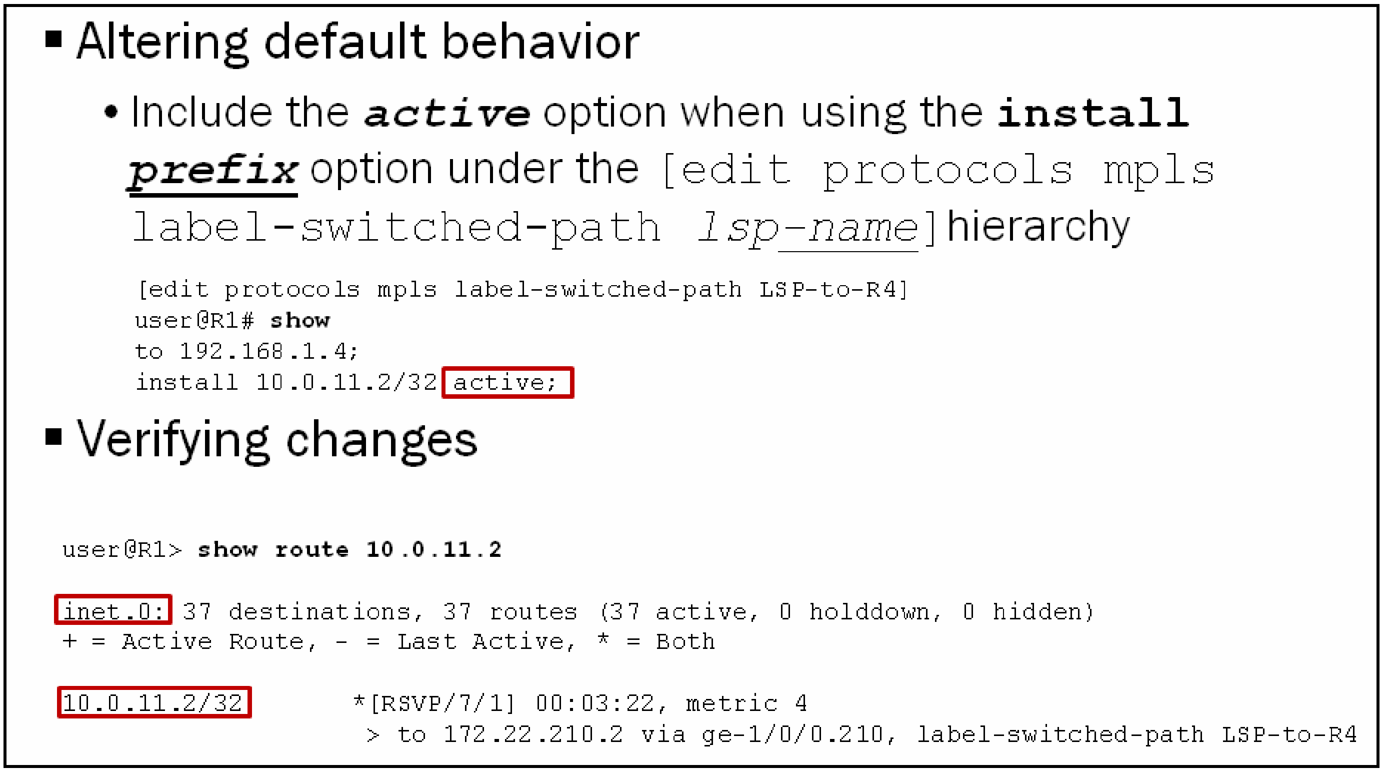
Default Routing Table Behavior



```
user@R1> show route 10.0.11.2

inet.0: 36 destinations, 36 routes (36 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.11.0/24       *[OSPF/10] 01:21:41, metric 5
                    > to 172.22.201.6 via ge-1/0/0.0

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.11.2/32       *[RSVP/7/1] 00:02:42, metric 4
                    > to 172.22.201.2 via ge-0/0/0.0, label-switched-path LSP-to-R4
```

The `inet.3` routing table is where LDP and RSVP signaled routes are stored. By default only BGP pays attention to the entries stored in `inet.3` and only then when it is resolving a BGP next hop. The LSPs are hidden from the main IP routing table, which allows non-BGP traffic to continue to use the IGP forwarding path. This behavior can be altered so that non-BGP traffic can also use the LSP. The output on the graphic shows an active OSPF route used to route IGP traffic and a RSVP route used by BGP traffic when the protocol next hop is the 10.0.11.2 prefix. We are going to discuss altering the default behavior next.

Altering the Default Routing Behavior



- Altering default behavior
  - Include the ***active*** option when using the **install** ***prefix*** option under the `[edit protocols mpls label-switched-path` *`lsp-name`*`]` hierarchy

```
[edit protocols mpls label-switched-path LSP-to-R4]
user@R1# show
to 192.168.1.4;
install 10.0.11.2/32 active;
```

- Verifying changes

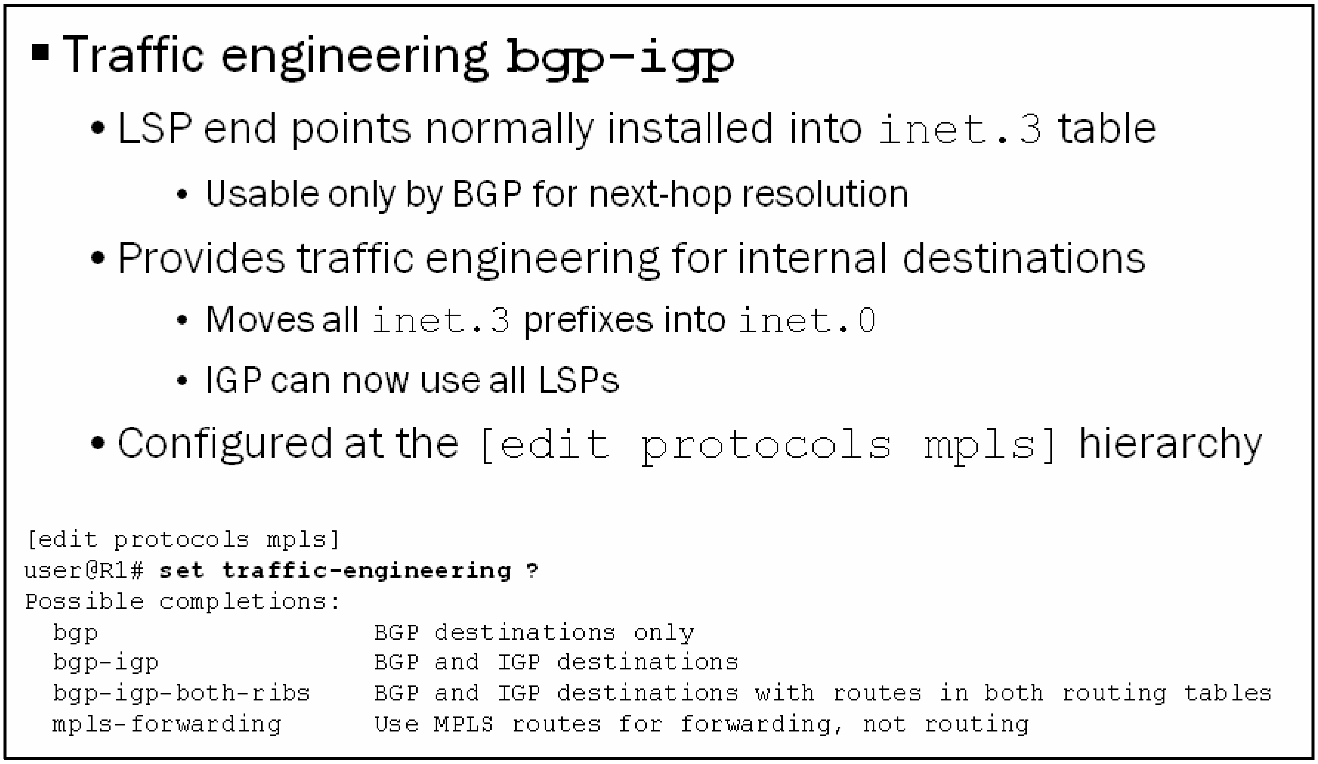
```
user@R1> show route 10.0.11.2

inet.0: 37 destinations, 37 routes (37 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.11.2/32       *[RSVP/7/1] 00:03:22, metric 4
                    > to 172.22.210.2 via ge-1/0/0.210, label-switched-path LSP-to-R4
```

To allow prefixes you have installed in the `inet.3` routing table to be installed and usable by the IGP you need to include the **`active`** tag when configuring the **`install`** ***`prefix`*** statement. The result is a route that is installed in the `inet.0` table any time the LSP is established, which means you can ping or trace the route. Use this option with care, because this type of prefix is very similar to a static route. This is especially useful when you need to push all traffic (internal and external) destined for a specific network through the LSP. In the graphic example we installed a specific /32 prefix and included the **`active`** option. Any BGP traffic with a protocol next hop of 10.0.11.2 will use the LSP as well as any IGP traffic that is destined to the 10.0.11.2 address.

Verifying the Changes

You can verify this easily by looking at the route for the prefix you installed using the command **`show route`** ***`prefix`***. You should see that the route has been moved from the `inet.3` routing table into the `inet.0` routing table.

Traffic Engineering for BGP and IGP



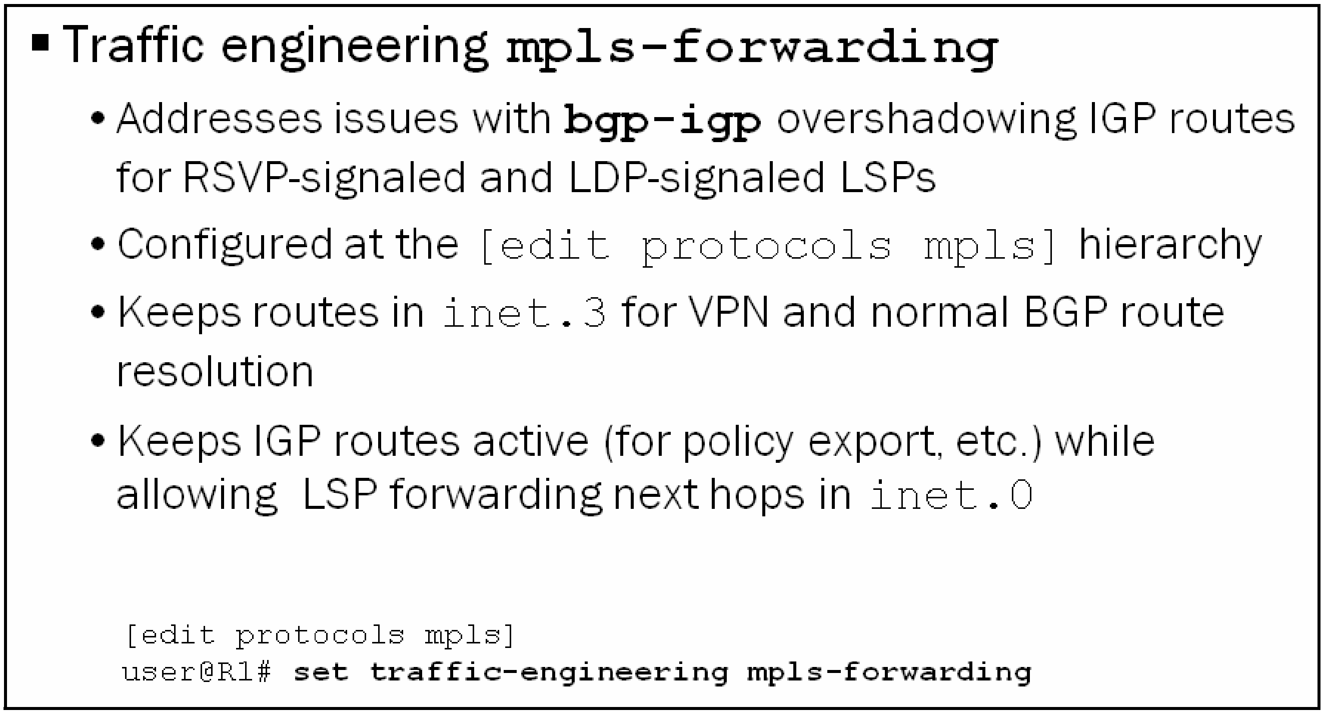
- Traffic engineering `bgp-igp`
  - LSP end points normally installed into `inet.3` table
    - Usable only by BGP for next-hop resolution
  - Provides traffic engineering for internal destinations
    - Moves all `inet.3` prefixes into `inet.0`
    - IGP can now use all LSPs
  - Configured at the `[edit protocols mpls]` hierarchy

```
[edit protocols mpls]
user@R1# set traffic-engineering ?
Possible completions:
  bgp                    BGP destinations only
  bgp-igp                BGP and IGP destinations
  bgp-igp-both-ribs      BGP and IGP destinations with routes in both routing tables
  mpls-forwarding        Use MPLS routes for forwarding, not routing
```

If traffic engineering for BGP and IGP is enabled, the router moves the routes from the `inet.3` routing table into the main routing table, `inet.0`. This move merges all routes together and at the same time empties the `inet.3` table. The number of routes in `inet.0` will be exactly the same as before, but they will now have the potential to be reachable via LSPs as next hops. The next hops for any given route can point to a physical interface, an LSP, or both if the metrics are equal. The **bgp** option restores the default behavior, which installs the LSP endpoints into `inet.3` only.

The **`bgp-igp-both-ribs`** option allows the routes to stay in `inet.0` and `inet.3`. This option helps resolve hidden route issues when running virtual private networks (VPNs), which is in keeping with normal VPN route resolution behavior that makes use of the `inet.3` routing table. The LDP **`no-forwarding`** option maintains LDP routes in `inet.3`, even when **`bgp-igp`** is configured. This option is discussed next.

MPLS Forwarding



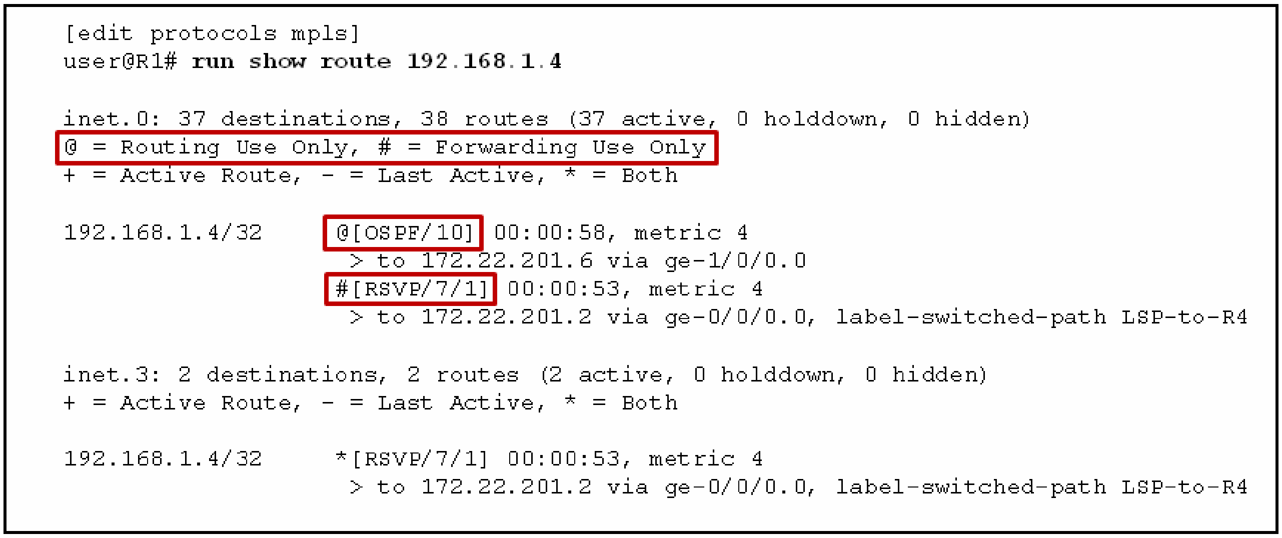
- Traffic engineering **mpls-forwarding**
  - Addresses issues with **bgp-igp** overshadowing IGP routes for RSVP-signaled and LDP-signaled LSPs
  - Configured at the `[edit protocols mpls]` hierarchy
  - Keeps routes in `inet.3` for VPN and normal BGP route resolution
  - Keeps IGP routes active (for policy export, etc.) while allowing LSP forwarding next hops in `inet.0`

```
[edit protocols mpls]
user@R1# set traffic-engineering mpls-forwarding
```

Another option for traffic engineering is **`mpls-forwarding`**. The **`mpls-forwarding`** option is designed to overcome some of the problems associated with the use of **`traffic-engineering bgp-igp`**. Specifically, the option is designed to prevent the overshadowing of IGP routes in the `inet.0` routing table when RSVP or LDP-signaled LSPs are copied from `inet.3` into `inet.0` so that LSPs can be used when forwarding to internal destinations.

By keeping the IGP routes active, your export policies continue to operate as expected, even though forwarding might occur over an LSP next hop. Unlike the **`bgp-igp`** option, **`mpls-forwarding`** maintains copies of the LSPs in the `inet.3` routing table where they can still be used for normal VPN or BGP next-hop resolution.

MPLS Forwarding: Operational Results



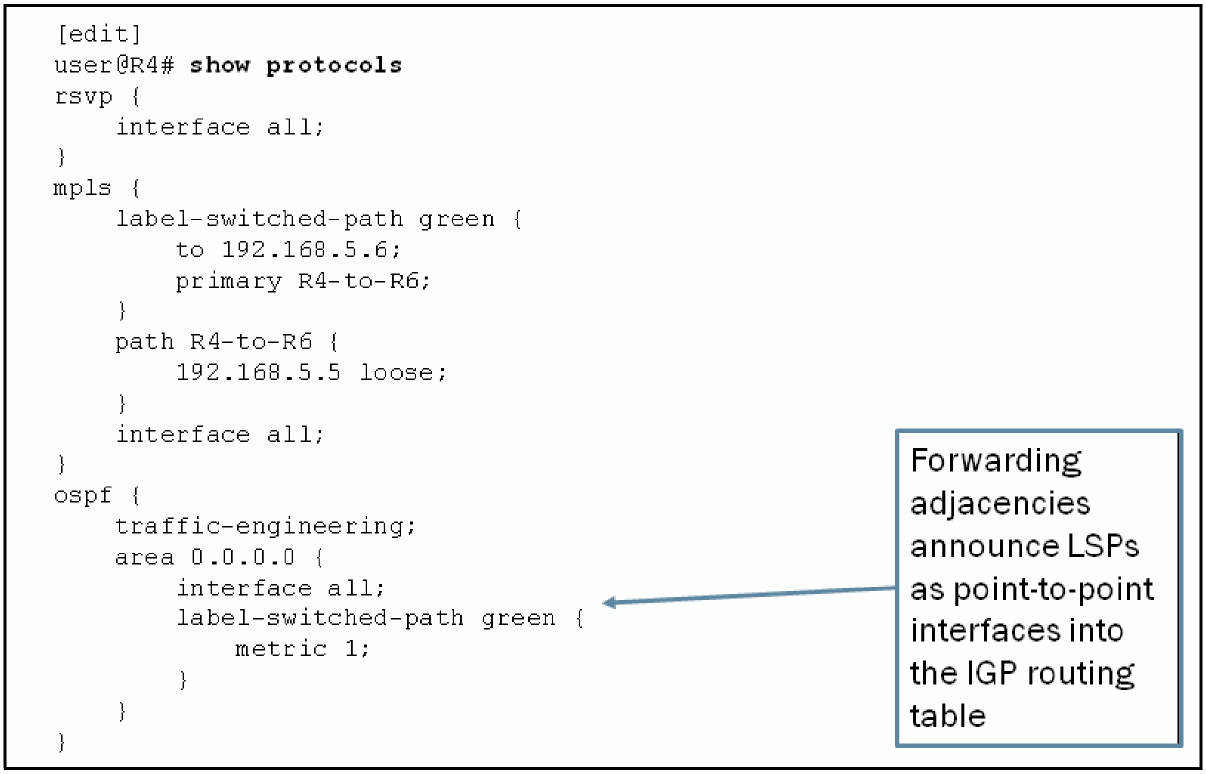
```
[edit protocols mpls]
user@R1# run show route 192.168.1.4

inet.0: 37 destinations, 38 routes (37 active, 0 holddown, 0 hidden)
@ = Routing Use Only, # = Forwarding Use Only
+ = Active Route, - = Last Active, * = Both

192.168.1.4/32      @[OSPF/10] 00:00:58, metric 4
                     > to 172.22.201.6 via ge-1/0/0.0
                    #[RSVP/7/1] 00:00:53, metric 4
                     > to 172.22.201.2 via ge-0/0/0.0, label-switched-path LSP-to-R4

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.1.4/32      *[RSVP/7/1] 00:00:53, metric 4
                     > to 172.22.201.2 via ge-0/0/0.0, label-switched-path LSP-to-R4
```

This graphic demonstrates the effects of adding the **`mpls-forwarding`** statement to an RSVP-based configuration. In this case, the 192.168.1.4 route is present in both the `inet.0` and `inet.3` routing tables as an RSVP-signaled LSP. Note that the route is also present in the `inet.0` table as an OSPF route.

When the **`mpls-forwarding`** option is enabled, new symbols are used to indicate the status of a route. The @ symbol is used to indicate a route that is active for routing use only, that is, active from the perspective of an export policy. The corresponding forwarding entry is identified with a # symbol.

You would normally use the **`mpls-forwarding`** option as a substitute for **`bgp-igp`** when you want to engineer traffic to both IGP and BGP destinations without having to concern yourself with the effects of having LDP or RSVP signaled LSPs overshadow existing routes in the `inet.0` table.

Forwarding Adjacency Overview



```
[edit]
user@R4# show protocols
rsvp {
    interface all;
}
mpls {
    label-switched-path green {
        to 192.168.5.6;
        primary R4-to-R6;
    }
    path R4-to-R6 {
        192.168.5.5 loose;
    }
    interface all;
}
ospf {
    traffic-engineering;
    area 0.0.0.0 {
        interface all;
        label-switched-path green {
            metric 1;
        }
    }
}
```

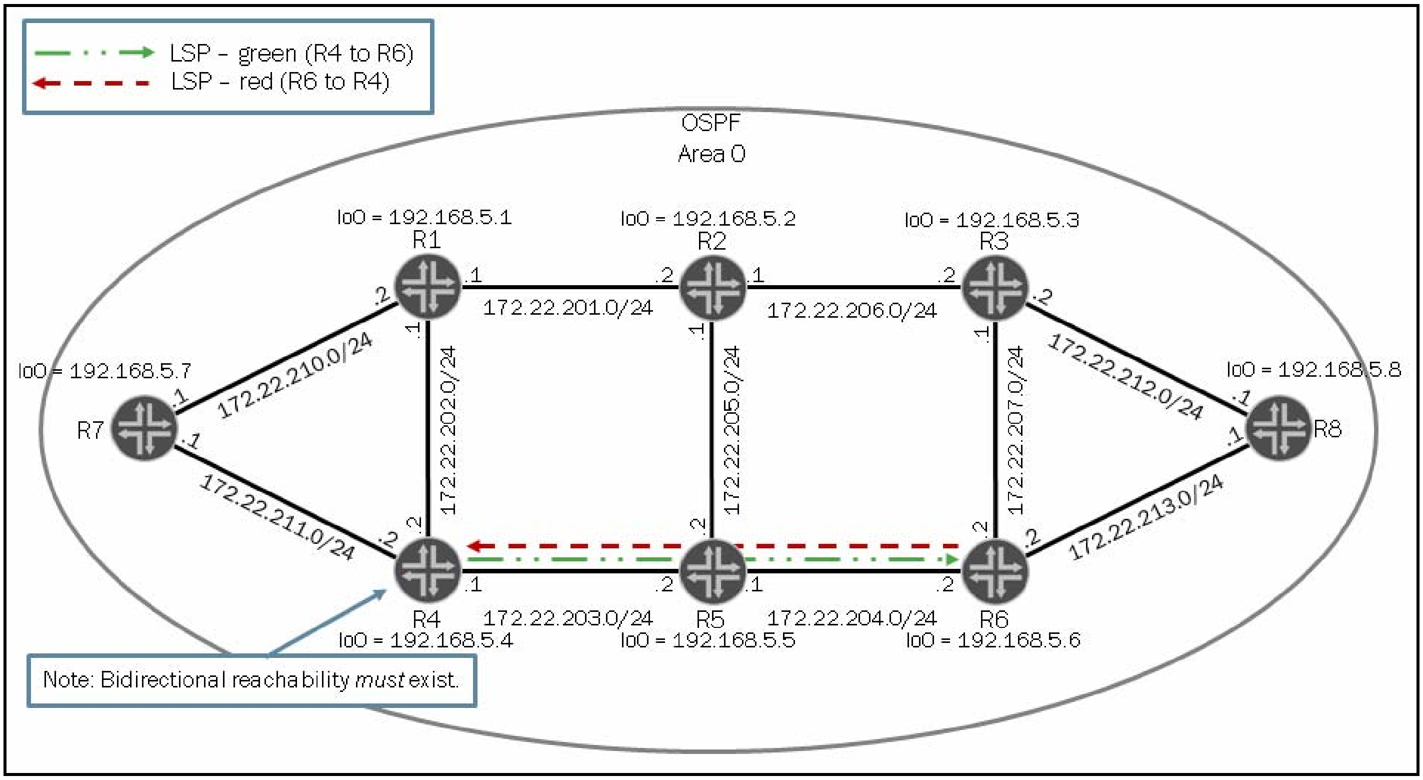
Forwarding adjacencies announce LSPs as point-to-point interfaces into the IGP routing table

Forwarding adjacencies allow the advertisement of LSPs as point-to-point interfaces within a link-state routing protocol's link-state advertisements. This behavior allows nodes that are upstream of the LSP ingress to factor the LSP as part of their shortest-path-first (SPF) calculations.

Forwarding adjacencies might be useful in a network that has a full-mesh of RSVP traffic-engineered LSPs between core routers. Forwarding adjacencies allow edge routers to utilize traffic-engineered LSPs in the network core without the complexity and scaling issues involved with extending the full-mesh of RSVP traffic-engineered LSPs to all routers.

The graphic illustrates how the OSPF protocol is configured to advertise the *`green`* LSP into area 0 with a metric of 1. Note that you must enable traffic engineering for OSPF. Remember traffic engineering is enabled by default for IS-IS.

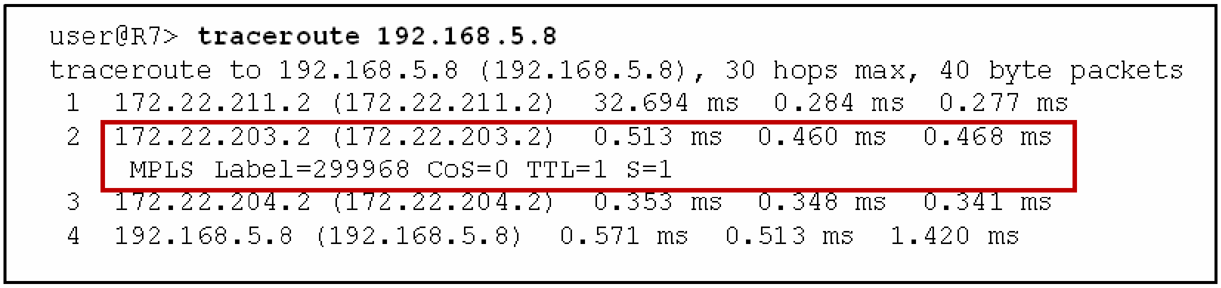
Forwarding Adjacency: Operation



You should only use Constrained Shortest Path First (CSPF) LSPs for forwarding adjacency applications to ensure that the LSP is injected into the IGP for IGP calculations, but not injected into the traffic engineering database (TED). A CSPF LSP is not placed into the TED, and therefore, other CSPF LSPs will not try to form over the LSP that is now being advertised into the IGP. If you use non-CSPF LSPs, it is possible that a new LSP will attempt to establish itself over an existing LSP (because you are not using the TED in this case), which causes an RSVP error.

Remember that LSPs are unidirectional. IS-IS requires that an LSP have a corresponding LSP in the reverse direction before advertising the forwarding adjacency in link-state advertisements. OSPF only requires the reverse direction to have IP-level reachability (by means of an LSP or a routed path) before advertising the forwarding adjacency in link-state advertisements. In both IS-IS and OSPF the LSP is advertised into the IGP, but no hellos or routing updates occur over the LSP—only user traffic is sent over the LSP. IS-IS and OSPF use the local copy of the link-state database to verify their bidirectional reachability requirements.

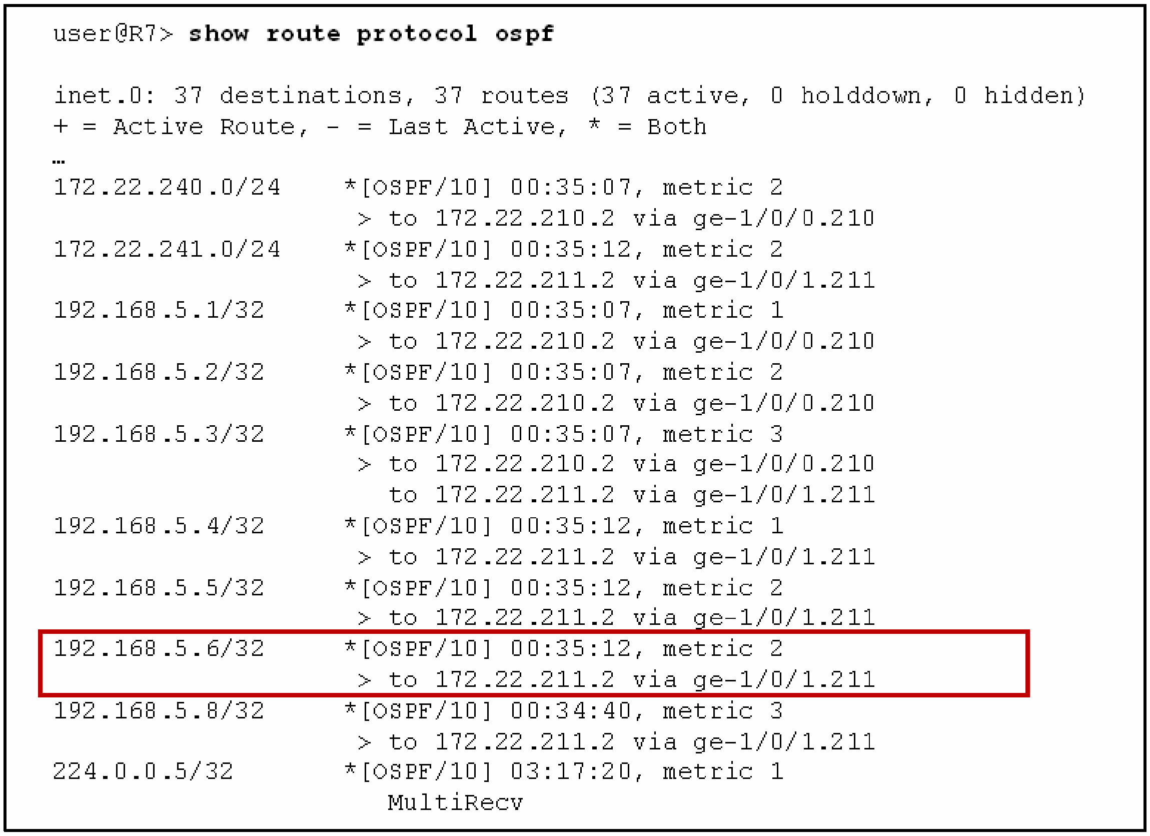
The Forwarding Adjacency Data Plane



```
user@R7> traceroute 192.168.5.8
traceroute to 192.168.5.8 (192.168.5.8), 30 hops max, 40 byte packets
 1  172.22.211.2 (172.22.211.2)  32.694 ms  0.284 ms  0.277 ms
 2  172.22.203.2 (172.22.203.2)  0.513 ms  0.460 ms  0.468 ms
     MPLS Label=299968 CoS=0 TTL=1 S=1
 3  172.22.204.2 (172.22.204.2)  0.353 ms  0.348 ms  0.341 ms
 4  192.168.5.8 (192.168.5.8)  0.571 ms  0.513 ms  1.420 ms
```

In the screen capture, a traceroute shows that packets are traversing an LSP that was provided in the middle of the network. This screen capture correlates to the diagram in the previous section.

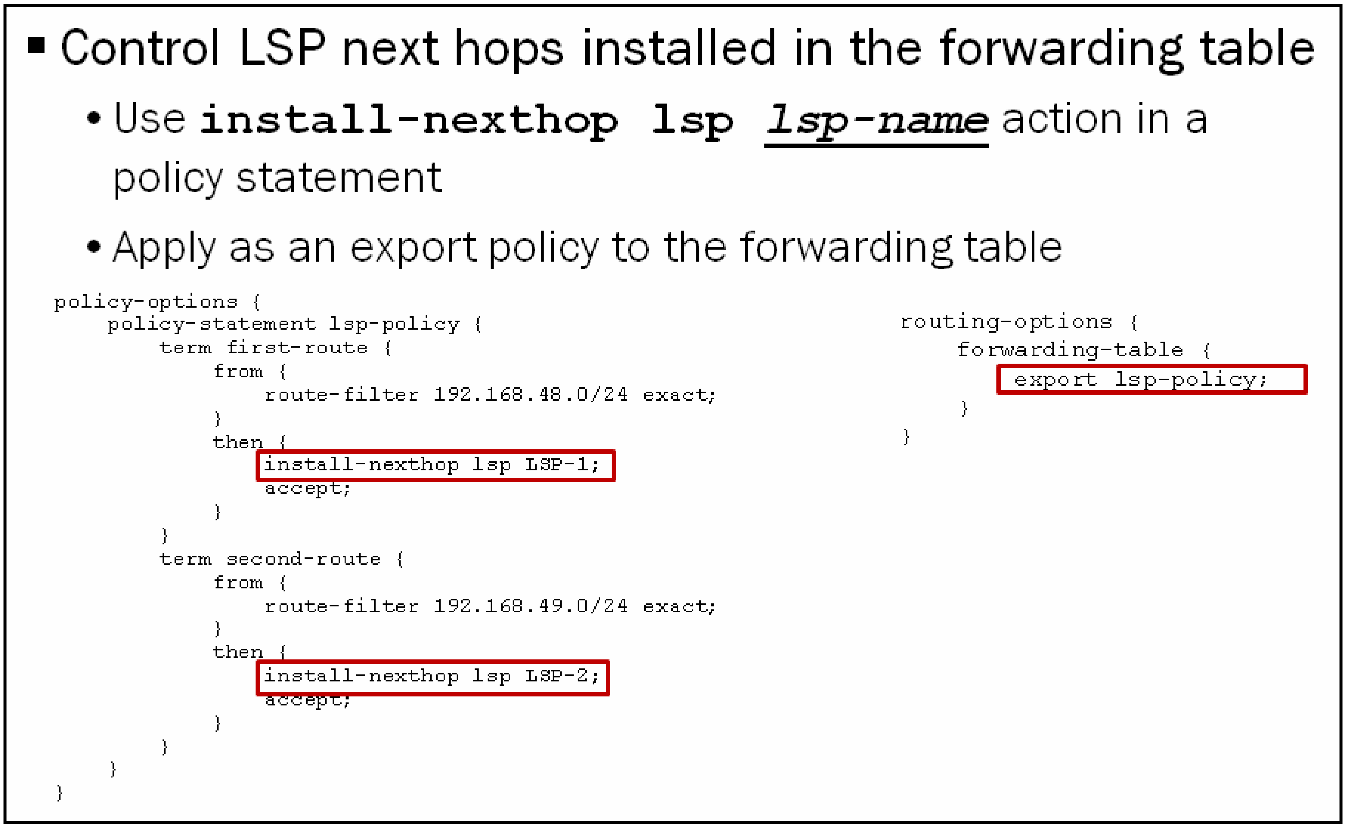
The Forwarding Adjacency Control Plane



```
user@R7> show route protocol ospf

inet.0: 37 destinations, 37 routes (37 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
…
172.22.240.0/24    *[OSPF/10] 00:35:07, metric 2
                    > to 172.22.210.2 via ge-1/0/0.210
172.22.241.0/24    *[OSPF/10] 00:35:12, metric 2
                    > to 172.22.211.2 via ge-1/0/1.211
192.168.5.1/32     *[OSPF/10] 00:35:07, metric 1
                    > to 172.22.210.2 via ge-1/0/0.210
192.168.5.2/32     *[OSPF/10] 00:35:07, metric 2
                    > to 172.22.210.2 via ge-1/0/0.210
192.168.5.3/32     *[OSPF/10] 00:35:07, metric 3
                    > to 172.22.210.2 via ge-1/0/0.210
                      to 172.22.211.2 via ge-1/0/1.211
192.168.5.4/32     *[OSPF/10] 00:35:12, metric 1
                    > to 172.22.211.2 via ge-1/0/1.211
192.168.5.5/32     *[OSPF/10] 00:35:12, metric 2
                    > to 172.22.211.2 via ge-1/0/1.211
192.168.5.6/32     *[OSPF/10] 00:35:12, metric 2
                    > to 172.22.211.2 via ge-1/0/1.211
192.168.5.8/32     *[OSPF/10] 00:34:40, metric 3
                    > to 172.22.211.2 via ge-1/0/1.211
224.0.0.5/32       *[OSPF/10] 03:17:20, metric 1
                      MultiRecv
```

Checking the IGP routes on Router R7 clearly shows that OSPF is including the LSP, with an associated cost of 1, in its best route calculations for a total metric of 2. This capture correlates to the diagram and the configuration shown in the preceding sections.

Selecting an LSP Next Hop



- Control LSP next hops installed in the forwarding table
  - Use **install-nexthop lsp** ***lsp-name*** action in a policy statement
  - Apply as an export policy to the forwarding table

```
policy-options {
    policy-statement lsp-policy {
        term first-route {
            from {
                route-filter 192.168.48.0/24 exact;
            }
            then {
                install-nexthop lsp LSP-1;
                accept;
            }
        }
        term second-route {
            from {
                route-filter 192.168.49.0/24 exact;
            }
            then {
                install-nexthop lsp LSP-2;
                accept;
            }
        }
    }
}
```

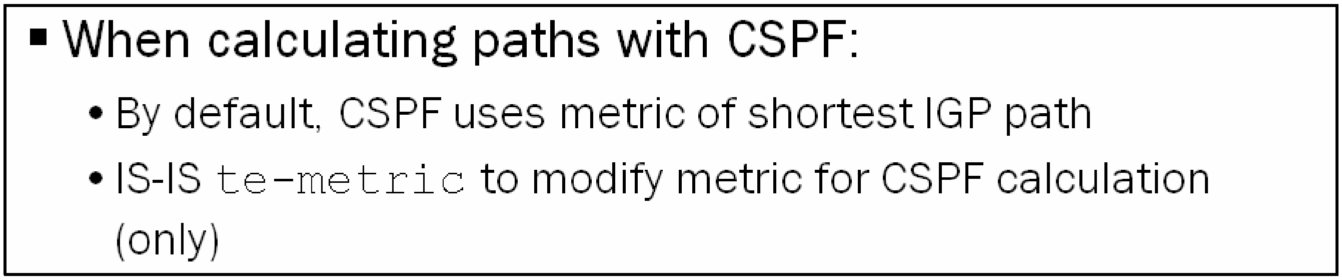
```
routing-options {
    forwarding-table {
        export lsp-policy;
    }
}
```

When multiple equal-cost LSPs to a destination exist, you can use policy to control which LSP gets installed in the forwarding table. This control provides fine-grained engineering of traffic flows across equal-cost LSPs.

Use the **install-nexthop lsp** ***lsp-name*** command as the action in a policy statement, and then apply the export policy to the forwarding table. The configuration on the graphic will install the *LSP-1* LSP as the next hop for the 192.168.48.0/24 prefix and the *LSP-2* LSP as the next hop for the 192.168.49.0/24 prefix. You can use the **show route** command to confirm the desired operation as shown in this capture:

```
user@R7> show route 192.168.48.0/24 exact

inet.0: 47 destinations, 47 routes (47 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.48.0/24    *[BGP/170] 2d 06:23:21, MED 0, localpref 100, from 192.168.1.4
                      AS path: I
                    > to 172.22.211.2 via ge-1/0/1.211, label-switched-path LSP-1

user@R7> show route 192.168.49.0/24 exact

inet.0: 47 destinations, 47 routes (47 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.49.0/24    *[BGP/170] 00:20:29, MED 0, localpref 100, from 192.168.1.4
                      AS path: I
                    > to 172.22.210.2 via ge-1/0/0.210, label-switched-path LSP-2
```

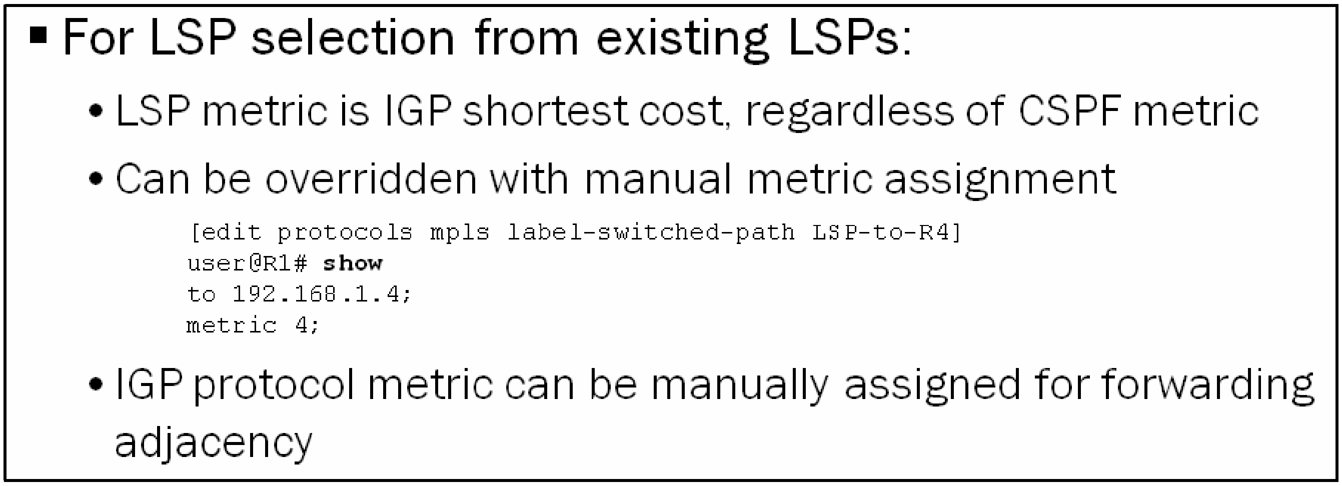
CSPF Path Metrics



- When calculating paths with CSPF:
  - By default, CSPF uses metric of shortest IGP path
  - IS-IS `te-metric` to modify metric for CSPF calculation (only)

You can assign LSP metrics to several different locations. By default, CSPF uses the default IGP metric for the links that comprise the shortest path. For IS-IS, you can assign a `te-metric` on each interface that is only used for CSPF while the IS-IS link-state database continues to utilize the standard IS-IS link metric. Basically, CSPF will use either the IGP metric (the default) or the `te-metric` value when so configured, to compute a shortest path for an LSP. In essence, the `te-metric` option allows you to have an IS-IS shortest path topology that differs from the TEDs view of the shortest path through your network.

Path Selection with LSP Metrics



- For LSP selection from existing LSPs:
  - LSP metric is IGP shortest cost, regardless of CSPF metric
  - Can be overridden with manual metric assignment

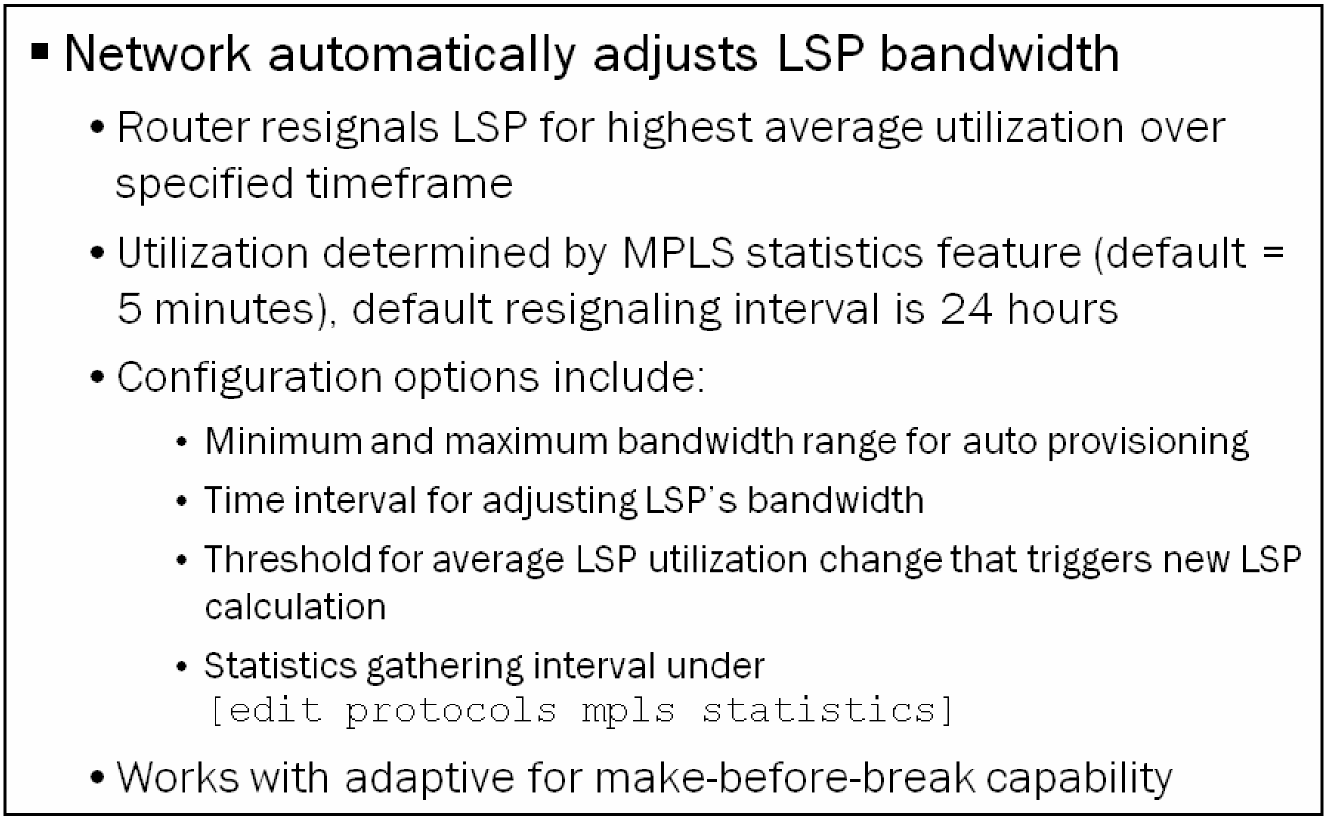
```
[edit protocols mpls label-switched-path LSP-to-R4]
user@R1# show
to 192.168.1.4;
metric 4;
```

  - IGP protocol metric can be manually assigned for forwarding adjacency

By default, each LSP inherits the IGP's shortest-path metric, regardless of whether or not the LSP actually follows the shortest path. You can override the value derived from the TED (based on default inheritance of the IGP's path metric or values specified with the **`te-metric`** option) by explicitly specifying an LSP metric value within the LSP's definition using the **`metric`** keyword.

When using forwarding adjacencies, you can also explicitly specify an LSP metric along with the LSP's declaration in the corresponding IGP stanza. To avoid awkward forwarding situations, you should only explicitly assign a metric in the MPLS definition OR in the LSP's reference within the IGP; we recommend that you do not assign a metric in both places.

Automatic Bandwidth Provisioning



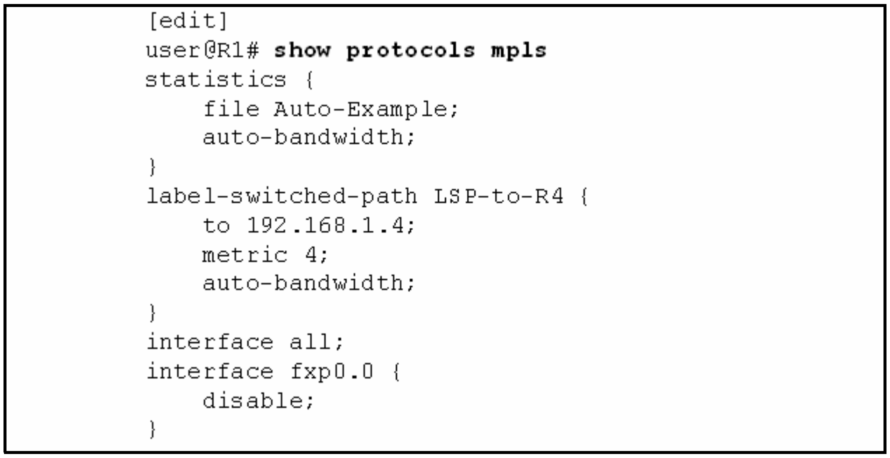
- Network automatically adjusts LSP bandwidth
  - Router resignals LSP for highest average utilization over specified timeframe
  - Utilization determined by MPLS statistics feature (default = 5 minutes), default resignaling interval is 24 hours
  - Configuration options include:
    - Minimum and maximum bandwidth range for auto provisioning
    - Time interval for adjusting LSP's bandwidth
    - Threshold for average LSP utilization change that triggers new LSP calculation
    - Statistics gathering interval under `[edit protocols mpls statistics]`
  - Works with adaptive for make-before-break capability

Auto-bandwidth provisioning allows the router to monitor actual traffic usage on each LSP and reconfigure the bandwidth of a given LSP to support observed traffic levels.

The MPLS statistics feature gathers statistics that are used to support automatic bandwidth calculations. The router monitors the highest utilization levels for the LSP over a predefined time period (24 hours by default). At the end of the time period, the existing LSP is resignaled, using make-before-break and shared explicit (SE)-style reservations to provision a new bandwidth reservation of the LSP.

Configure Automatic Bandwidth Provisioning

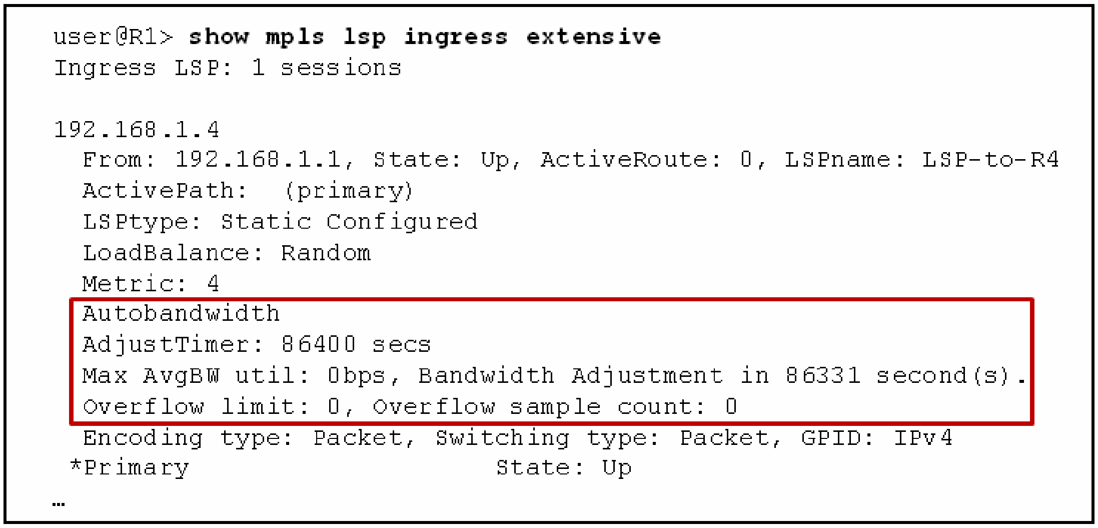
This graphic provides a sample configuration in support of automatic bandwidth provisioning. Note that MPLS statistics must be enabled to allow the router to monitor traffic usage. By default, the router monitors usage every 300 seconds. If the average utilization over the adjustment interval exceeds a certain value, the router resignals the LSP. The following options are available for automatic bandwidth provisioning:



```
[edit]
user@R1# show protocols mpls
statistics {
    file Auto-Example;
    auto-bandwidth;
}
label-switched-path LSP-to-R4 {
    to 192.168.1.4;
    metric 4;
    auto-bandwidth;
}
interface all;
interface fxp0.0 {
    disable;
}
```

Option	Meaning
`adjust-interval`	Time to adjust LSP bandwidth (`300..4294967295 seconds`)
`adjust-threshold`	Change in average LSP utilization to trigger auto-adjustment
`maximum-bandwidth`	Maximum LSP bandwidth (bps)
`minimum-bandwidth`	Minimum LSP bandwidth (bps)
`monitor-bandwidth`	Monitor LSP bandwidth without adjustments

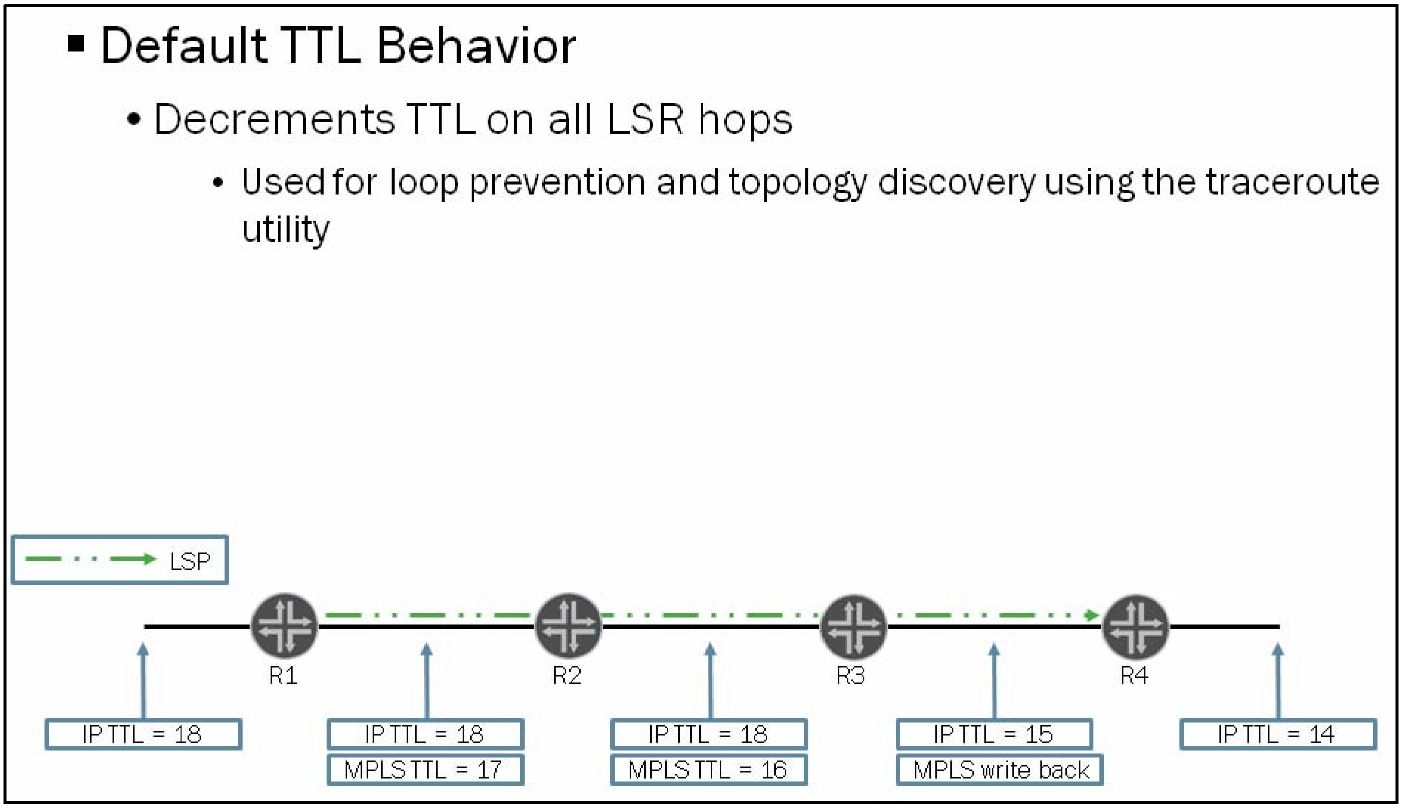
Monitor Automatic Bandwidth Provisioning



```
user@R1> show mpls lsp ingress extensive
Ingress LSP: 1 sessions

192.168.1.4
  From: 192.168.1.1, State: Up, ActiveRoute: 0, LSPname: LSP-to-R4
  ActivePath:  (primary)
  LSPtype: Static Configured
  LoadBalance: Random
  Metric: 4
  Autobandwidth
  AdjustTimer: 86400 secs
  Max AvgBW util: 0bps, Bandwidth Adjustment in 86331 second(s).
  Overflow limit: 0, Overflow sample count: 0
  Encoding type: Packet, Switching type: Packet, GPID: IPv4
 *Primary                    State: Up
…
```

The operational aspects of automatic bandwidth provision is displayed in the output of a **`show mpls lsp extensive`** command, as shown on this graphic.

Default TTL Behavior

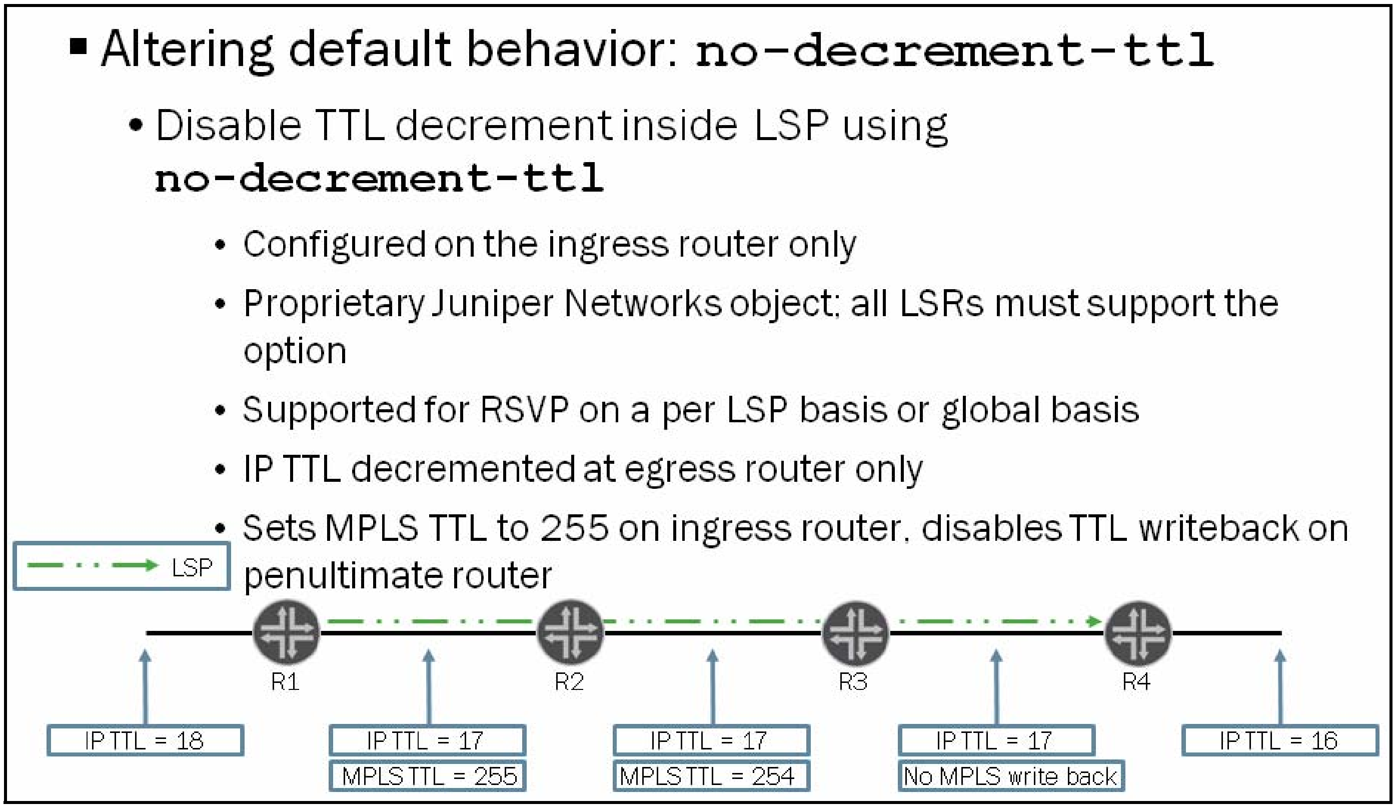


- Default TTL Behavior
  - Decrements TTL on all LSR hops
    - Used for loop prevention and topology discovery using the traceroute utility

By default, the time-to-live (TTL) value in the packet header is decremented by 1 for every hop the packet traverses in the LSP, thereby preventing loops and allowing topology discovery. If the TTL field value reaches 0, packets are dropped and an Internet Control Message Protocol (ICMP) error packet can be sent to the originating router. You might want to disable normal TTL decrementing to make the MPLS cloud appear transparent, thereby hiding the network topology.

The normal TTL handing behavior maps the IP packet's TTL value into the MPLS TTL field on the ingress router. When the MPLS packet leaves the router, it is decremented by one, as shown in the graphic. Each transit label-switching router (LSR) decrements the TTL field by one until the packet reaches the penultimate hop. At the penultimate hop, the penultimate router strips off the top label and writes the MPLS TTL value back into the IP TTL value. The egress router decrements the IP TTL by one. The TTL values are indicated for every hop in the path on the graphic.

MPLS TTL Handling: No Decrement



- Altering default behavior: `no-decrement-ttl`
  - Disable TTL decrement inside LSP using `no-decrement-ttl`
    - Configured on the ingress router only
    - Proprietary Juniper Networks object; all LSRs must support the option
    - Supported for RSVP on a per LSP basis or global basis
    - IP TTL decremented at egress router only
    - Sets MPLS TTL to 255 on ingress router, disables TTL writeback on penultimate router

On the ingress of the LSP, if you include the **`no-decrement-ttl`** statement at the `[edit protocols mpls label-switched-path lsp-name]` hierarchy, the ingress router negotiates with all downstream routers using a proprietary RSVP object to ensure all routers are in agreement. This command can also be typed within the primary or secondary path hierarchy. If negotiation succeeds, the whole LSP appears as two hops for transit IP traffic.

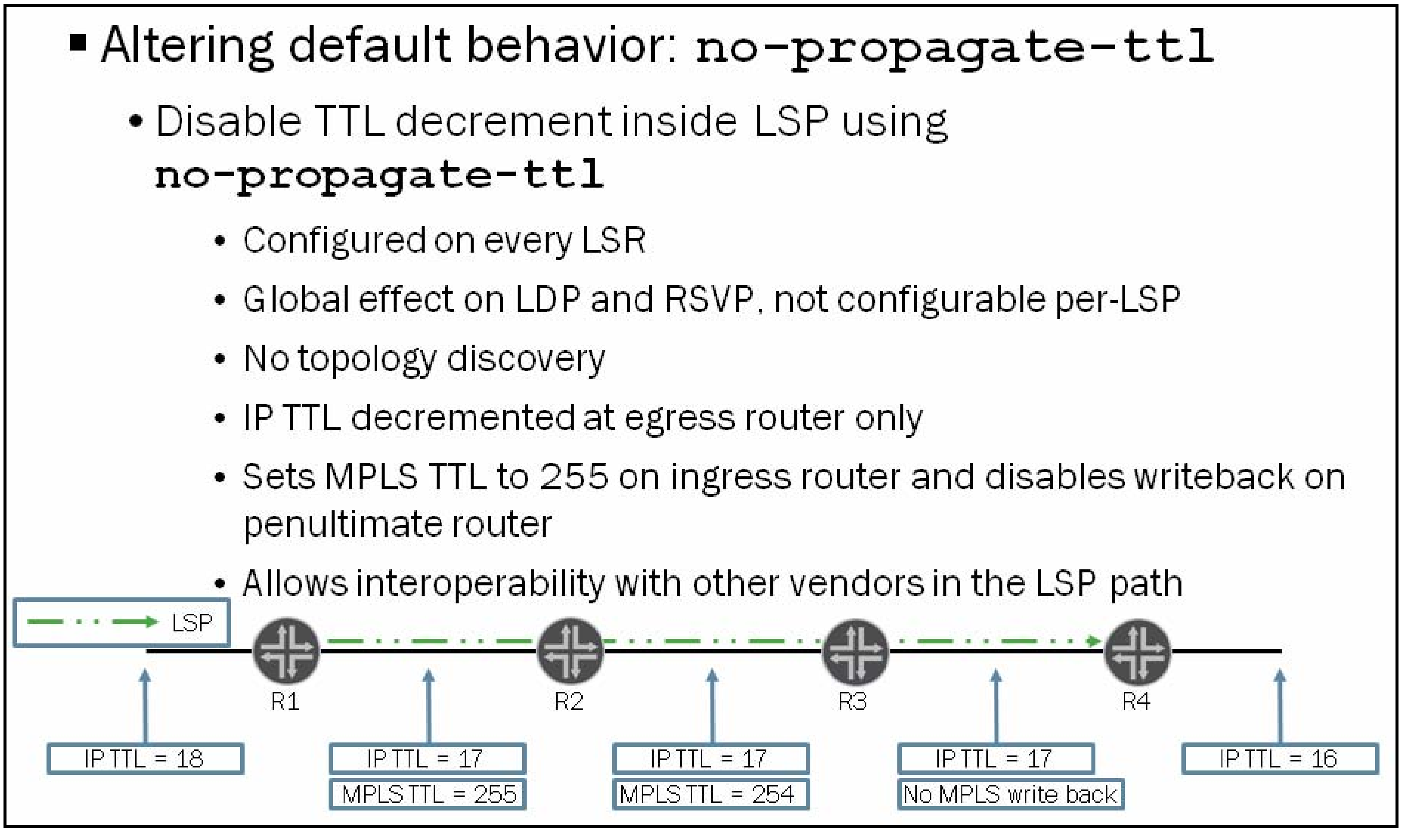
```
[edit protocols mpls label-switched-path lsp-name]
user@R1# set no-decrement-ttl
```

Note that the RSVP object is proprietary to the Junos OS and might not work with other vendors. Further, this potential incompatibility applies only to RSVP-signaled LSPs, not LDP-signaled LSPs. Also note that you can apply **`no-decrement-ttl`** on a per-LSP basis or globally under the `[edit protocols mpls]` hierarchy.

If normal TTL decrement is disabled, the TTL field of IP packets entering LSPs are decremented by 1 upon transiting the LSP, making the LSP appear as a two-hop router to diagnostic tools like traceroute. This function is performed by the ingress router, which pushes a label on IP packets with the TTL field in the label initialized to 255. The label's TTL field value is decremented by 1 for every hop the MPLS packet traverses in the LSP. On the penultimate hop of the LSP, the router pops the label but does not write the label's TTL field value to the IP packet's TTL field. Instead, when the IP packet reaches the egress router, the IP packet's TTL field value is decremented by 1.

When you use traceroute to diagnose problems with an LSP, traceroute sees the ingress router, although the egress router performs the TTL decrement. Note that this assumes that traceroute is initiated outside of the LSP. The behavior of traceroute is different if it is initiated from the ingress router of the LSP. In this case, the egress router would be the first router to respond to traceroute.

MPLS TTL Handling: No Propagate



- Altering default behavior: `no-propagate-ttl`
  - Disable TTL decrement inside LSP using **`no-propagate-ttl`**
    - Configured on every LSR
    - Global effect on LDP and RSVP, not configurable per-LSP
    - No topology discovery
    - IP TTL decremented at egress router only
    - Sets MPLS TTL to 255 on ingress router and disables writeback on penultimate router
    - Allows interoperability with other vendors in the LSP path

You must include the **`no-propagate-ttl`** statement at the `[edit protocols mpls]` hierarchy level of all routers in the path of the LSP for proper operation, which is in contrast to the ingress based setting for the **`no-decrement-ttl`** option. Note that this statement applies to all LSPs in a global manner, regardless of whether they are RSVP or LDP signaled. Once set, all future LSPs traversing through this router behave as a single hop to IP packets. LSPs established before this statement is committed are not affected. Note that this option affects RSVP-signaled LSPs, despite its being configured under the `[edit protocol mpls]` hierarchy:

```
[edit protocols mpls]
user@R1# set no-propagate-ttl
```

Make sure all routers are configured consistently within an MPLS domain when using **`no-propagate-ttl`**; failing to do so might cause the IP packet TTL to increase while in transit within LSPs. This can happen, for example, when the ingress router has **`no-propagate-ttl`** configured but the penultimate router does not, which results in the penultimate router writing the MPLS TTL value (which starts from the ingress router as 255) back into the IP packet.

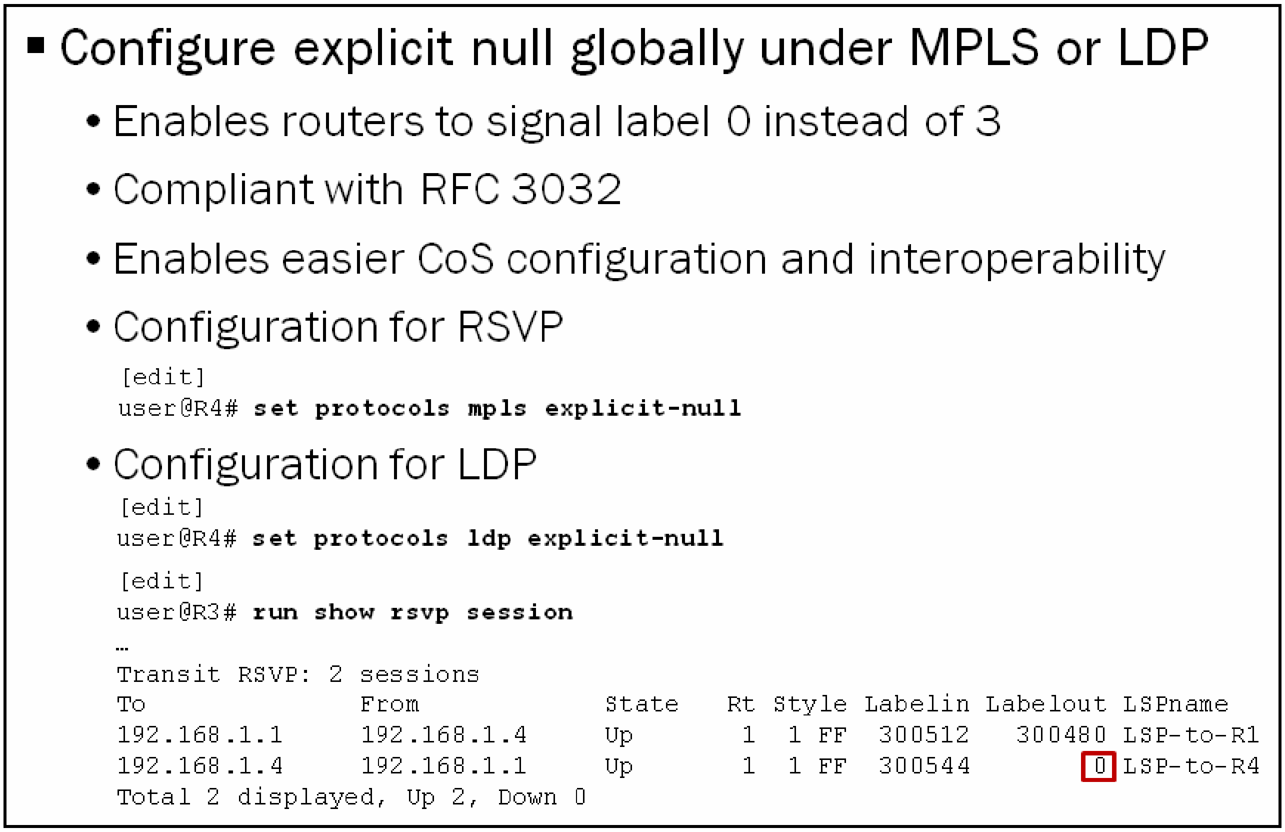
The **`no-propagate-ttl`** option is designed to be interoperable with equipment made by other vendors. However, you must ensure all routers are configured identically.

The **`no-propagate-ttl`** option also causes the MPLS cloud to show up as two hops from the perspective of IP packets transiting the LSP.

The penultimate router pops the label and forwards the IP packet, but does not copy the MPLS TTL value back into the IP packet's TTL field. The egress router then decrements the IP packet, thereby making the cloud appear as if it consisted of only two hops.

Note that with either option (**`no-propagate-ttl`** and **`no-decrement-ttl`**), the ingress router decrements the IP packet's TTL by one prior to placing the MPLS shim label on incoming packets. This performance is to prevent the possibility of an endless routing loop (formed when two LSPs have a routing loop pointing at each other). If the IP TTL were not decremented by one on ingress, the egress router would encapsulate the IP packet with a new MPLS header without decrementing the IP TTL. If the two routers have a routing loop, the packet would loop to infinity.

Configuring Explicit Null



- Configure explicit null globally under MPLS or LDP
  - Enables routers to signal label 0 instead of 3
  - Compliant with RFC 3032
  - Enables easier CoS configuration and interoperability
  - Configuration for RSVP

```
[edit]
user@R4# set protocols mpls explicit-null
```

  - Configuration for LDP

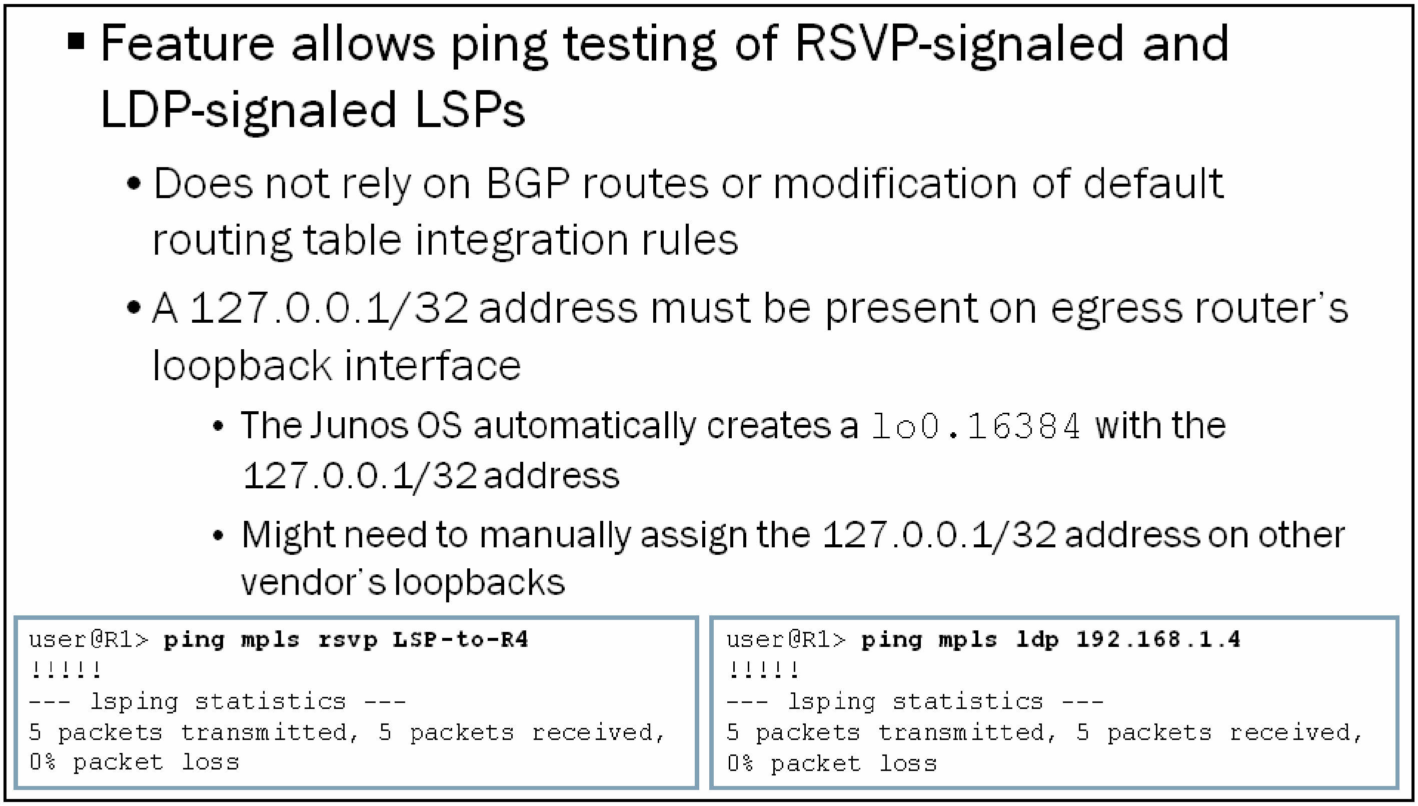
```
[edit]
user@R4# set protocols ldp explicit-null

[edit]
user@R3# run show rsvp session
…
Transit RSVP: 2 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.168.1.1     192.168.1.4     Up       1  1 FF  300512   300480 LSP-to-R1
192.168.1.4     192.168.1.1     Up       1  1 FF  300544        0 LSP-to-R4
Total 2 displayed, Up 2, Down 0
```

Because class-of-service (CoS) configurations with penultimate hop popping (PHP) require that the egress router classify and queue packets based on IP-related parameters as opposed to MPLS shim header values, end-to-end CoS designs can be made complex with PHP.

With the Junos OS you can negate the default PHP behavior to effect the receipt of labeled packets at the egress node. In operation, the egress router will signal label 0 upstream instead of label 3. As a result, the same CoS configuration used at transit LSRs can now also be used for the egress router.

This feature is configured globally for either MPLS or LDP. The implementation is compliant with RFC 3032, “MPLS Label Stack Encoding”.

MPLS Ping Utility



- Feature allows ping testing of RSVP-signaled and LDP-signaled LSPs
  - Does not rely on BGP routes or modification of default routing table integration rules
  - A 127.0.0.1/32 address must be present on egress router's loopback interface
    - The Junos OS automatically creates a `lo0.16384` with the 127.0.0.1/32 address
    - Might need to manually assign the 127.0.0.1/32 address on other vendor's loopbacks

```
user@R1> ping mpls rsvp LSP-to-R4
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received,
0% packet loss
```

```
user@R1> ping mpls ldp 192.168.1.4
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received,
0% packet loss
```

This graphic illustrates the operation of the MPLS ping capability. By adding the **mpls** switch to a standard **ping** command, you can now verify the forwarding plane of RSVP-signaled or LDP-signaled LSPs. In the past, ping testing of an LSP required the presence of BGP routes, or the modification of the default routing table integration rules to permit traffic engineering for internal destinations, that is, the egress node's router ID. For RSVP-signaled LSPs, you specify the LSP name as the target for the MPLS ping.

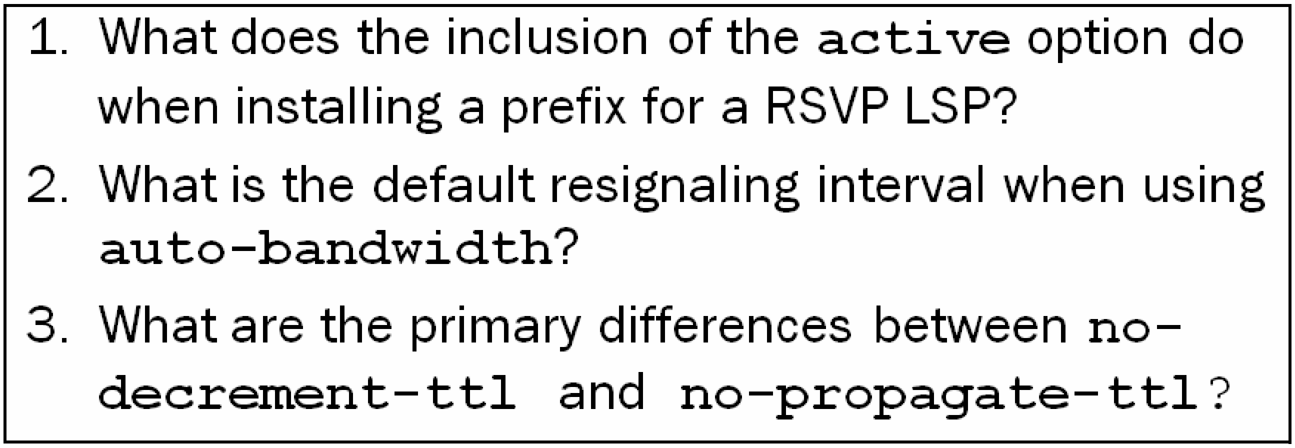
Note that the target address of an MPLS ping is hard-coded to 127.0.0.1; the LSP's egress node must have a 127.0.0.1 address assigned to its loopback interface for the MPLS ping to succeed.The Junos OS automatically creates a `lo0.16384` with the 127.0.0.1/32 address assigned. You might however, need to manually assign the 127.0.0.1/32 address if the egress router is not running the Junos OS. You can verify the loopback address is present on the egress router by issuing the **show interfaces terse | match lo0** operational mode command on the egress router.

```
user@R4> show interfaces terse | match lo0
lo0                     up    up
lo0.0                   up    up   inet     192.168.1.4         --> 0/0
lo0.16384               up    up   inet     127.0.0.1           --> 0/0
lo0.16385               up    up   inet
```

The detail switch provides additional output:

```
user@R1> ping mpls rsvp test count 1 detail
Request for seq 1, to interface 9, labels <100096, 0, 0>
Reply for seq 1, return code: Egress-ok

--- lsping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
```

Review Questions



1. What does the inclusion of the `active` option do when installing a prefix for a RSVP LSP?
2. What is the default resignaling interval when using `auto-bandwidth`?
3. What are the primary differences between `no-decrement-ttl` and `no-propagate-ttl`?

Answers to Review Questions

1.

Including the **`active`** option when installing a prefix for a RSVP LSP installs the route in the `inet.0` routing table and allows both the IGP and BGP to use the LSP.

2.

The default resignaling interval is 24 hours when using the `auto-bandwidth` feature.

3.

First `no-decrement-ttl` is only configured on the ingress router and `no-propagate-ttl` must be configured on all LSRs in the path. Second, using `no-decrement-ttl` allows you to change default behavior on a per LSP basis while `no-propagate-ttl` is only allowed at the global level and applies to all LSPs.

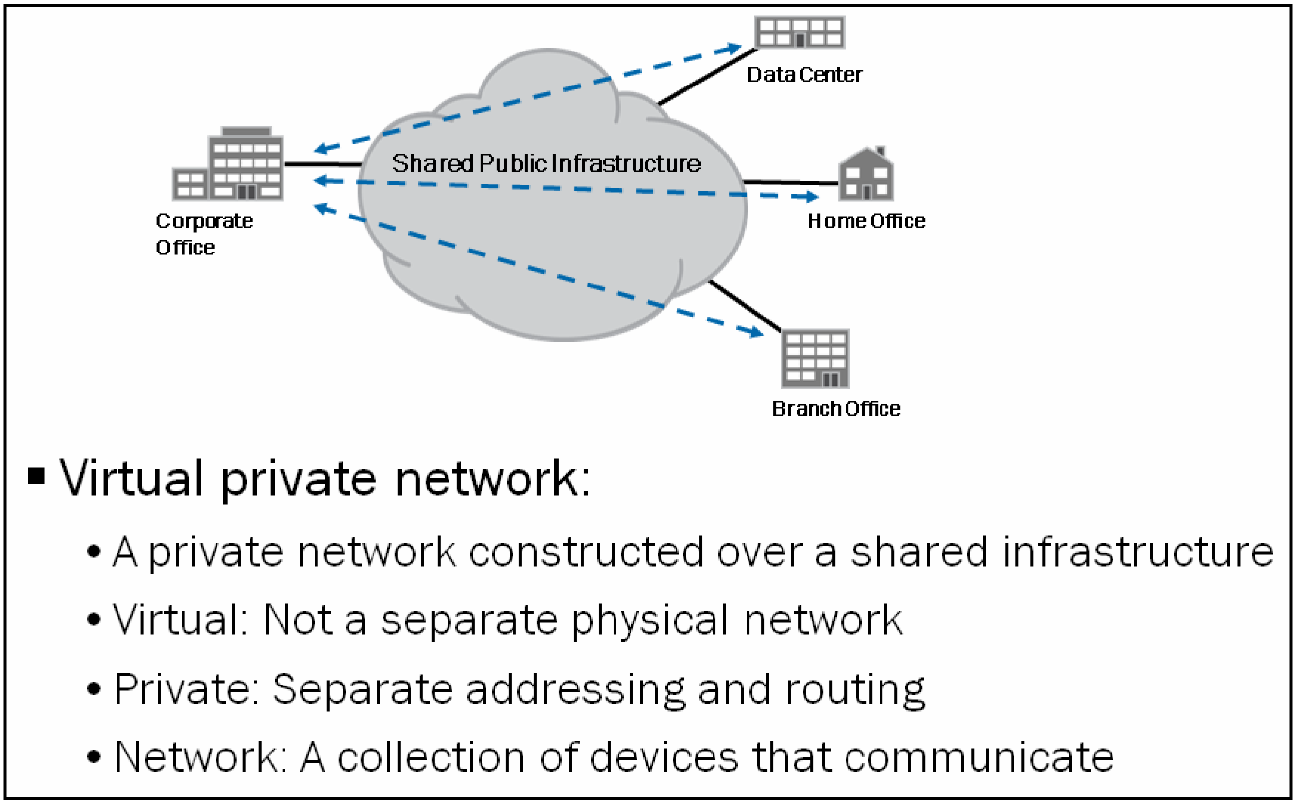
JNCIS-SP Study Guide—Part 3

Chapter 6: VPN Review

This Chapter Discusses:

- The definition of the term virtual private network (VPN);
- Differences between provider-provisioned and customer-provisioned VPNs;
- Differences between Layer 2 and Layer 3 VPNs; and
- The features of provider-provisioned VPNs supported by the Junos operating system.

Virtual Private Network



- Virtual private network:
  - A private network constructed over a shared infrastructure
  - Virtual: Not a separate physical network
  - Private: Separate addressing and routing
  - Network: A collection of devices that communicate

A VPN is a private network that is constructed over a shared, public infrastructure such as Frame Relay, an Asynchronous Transfer Mode (ATM) network, or the Internet. It is considered virtual because it does not require a separate physical network, but instead it is a logical network, one of possibly many logical networks, that uses a single physical network. It is considered a private network because a VPN can have its own separate addressing and routing scheme to interconnect devices that must communicate.

A VPN is designed so that only devices intended to communicate with each other can do so. For instance, as shown on the graphic, a VPN can be the network infrastructure that provides communication between the corporate headquarters, branch offices, mobile users, data centers, suppliers, and customers, while ensuring that unwanted devices cannot gain access to this private network.

Uses IP Infrastructure

Most companies today provide their employees with access to the Internet for e-mail and web browsing services. The Internet has become part of everyday life in today's society. By utilizing the Internet as the public infrastructure for building VPNs, companies can use their existing equipment to reduce costs.

Increasing Importance of IP/MPLS (Not ATM/Frame Relay)

With more people having access to the Internet, it makes sense to use the IP/MPLS network as a building block for VPNs. MPLS is now being used by many Internet service provider (ISP). This allows these providers to offer VPN services to its customers using an IP solution.

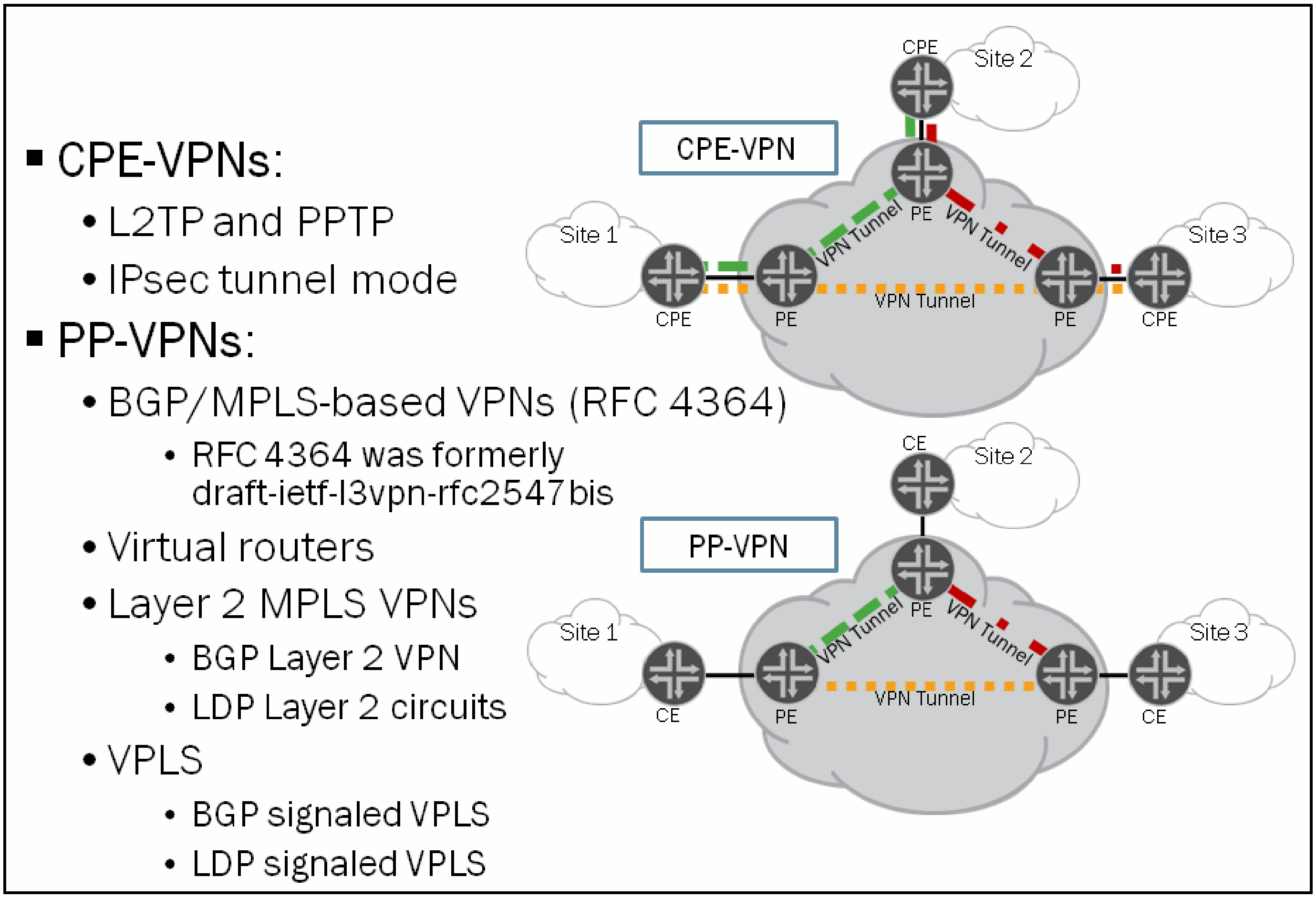
Subscriber Benefits

VPNs deployed over the Internet can lower operational expenses for companies by making it possible to use a single network connection to provide multiple services. A company no longer needs a Frame Relay network to provide VPN services and Internet connectivity for e-mail services; it can all be done using one Internet connection.

Provider Benefits

VPNs can also create an additional source of revenue for the provider. ISPs can now offer not only Internet service but also value-added VPN services. Everybody wins!

Customer Premises VPN Solutions



- CPE-VPNs:
  - L2TP and PPTP
  - IPsec tunnel mode
- PP-VPNs:
  - BGP/MPLS-based VPNs (RFC 4364)
    - RFC 4364 was formerly draft-ietf-l3vpn-rfc2547bis
  - Virtual routers
  - Layer 2 MPLS VPNs
    - BGP Layer 2 VPN
    - LDP Layer 2 circuits
  - VPLS
    - BGP signaled VPLS
    - LDP signaled VPLS

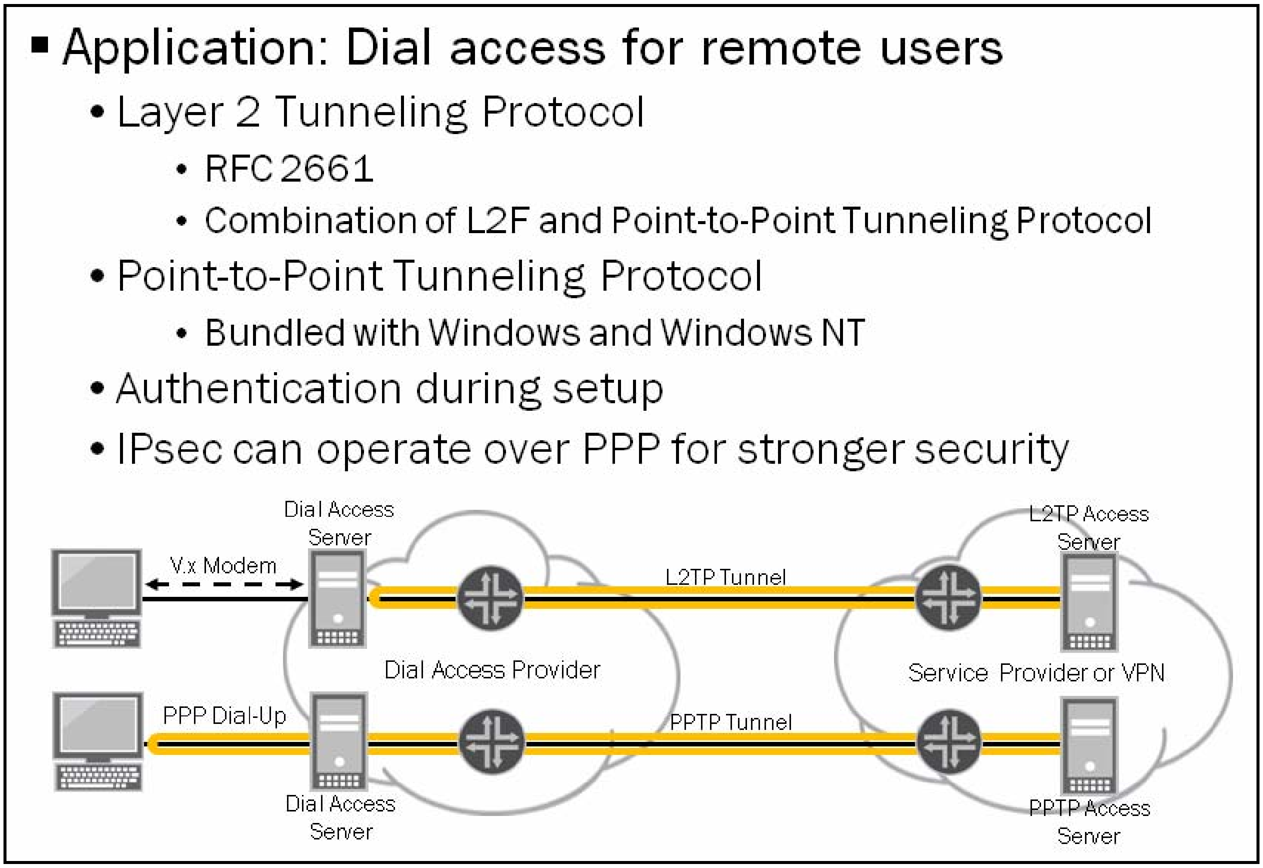
A customer premises equipment (CPE) VPN solution is a VPN that relies only on the customer's equipment to create and manage tunnels for the private IP traffic. Layer 2 Tunneling Protocol (L2TP), Point-to-Point Tunneling Protocol (PPTP), and

IP Security (IPsec) tunnel mode are protocols used by customer premises equipment for this purpose. When the ISP receives IP packets from the customer, they are treated as normal IP packets and are routed accordingly.

Provider-Provisioned VPN Solutions

A provider-provisioned VPN solution is a VPN that relies on the provider's equipment to create and manage tunnels for the private traffic using MPLS as the enabling technology. Examples of provider-provisioned VPNs include BGP/MPLS-based VPNs, such as Layer 3 VPNs (defined in RFC 4364), Layer 2 MPLS-based VPNs, including BGP Layer 2 VPNs (defined in draft-kompella), and LDP Layer 2 circuits (defined in RFC 4447), as well as virtual routers and the virtual private LAN service (VPLS) approach, which includes BGP signaled VPLS (defined in RFC 4761) and LDP signaled VPLS (defined in RFC 4762).

Application: Dial Access for Remote Users



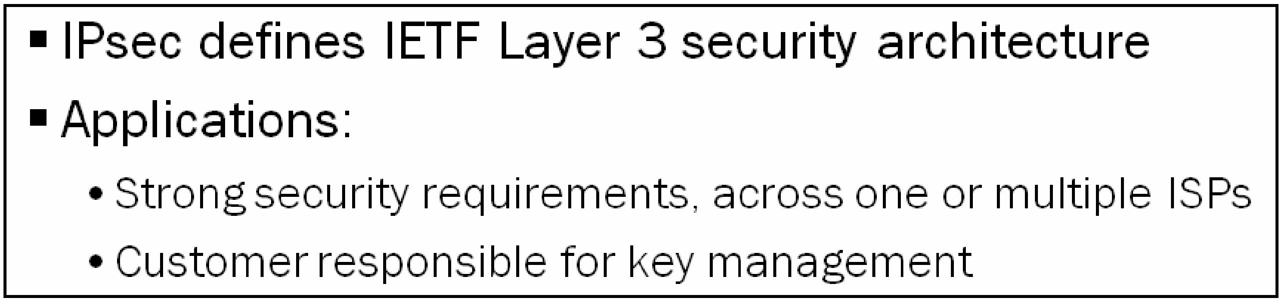
- Application: Dial access for remote users
  - Layer 2 Tunneling Protocol
    - RFC 2661
    - Combination of L2F and Point-to-Point Tunneling Protocol
  - Point-to-Point Tunneling Protocol
    - Bundled with Windows and Windows NT
  - Authentication during setup
  - IPsec can operate over PPP for stronger security

Several protocols that provide dial access for remote users to their corporate sites are in use today. L2TP and PPTP are the most common methods used for tunneling Point-to-Point Protocol (PPP) traffic over an IP network. L2TP is a defined in RFC 2661. It combines Cisco's Layer 2 Forwarding (L2F) protocol and Microsoft's PPTP and uses User Datagram Protocol (UDP) for transport.

PPTP, which is typically bundled with Windows and Windows NT, uses Transmission Control Protocol (TCP) to transport PPP. PPTP and PPP use a system of authentication during the setup of the tunnels. Also, both make use of the PPP authentication protocols—the Challenge Handshake Authentication Protocol (CHAP) and the Password Authentication Protocol (PAP)—to provide access authentication.

IPsec, another tunneling protocol, is used to tunnel private IP traffic over an IP backbone. L2TP and PPTP are used to tunnel PPP traffic (Layer 2) using UDP or TCP as the transport protocol. IPsec tunnels IP traffic (Layer 3) using IP as the delivery protocol.

IPsec Defines IETF Layer 3 Security Architecture



- IPsec defines IETF Layer 3 security architecture
- Applications:
  - Strong security requirements, across one or multiple ISPs
  - Customer responsible for key management

RFCs 4301, 4302, 4303, and 4305 contain the definition of IPsec.

Applications

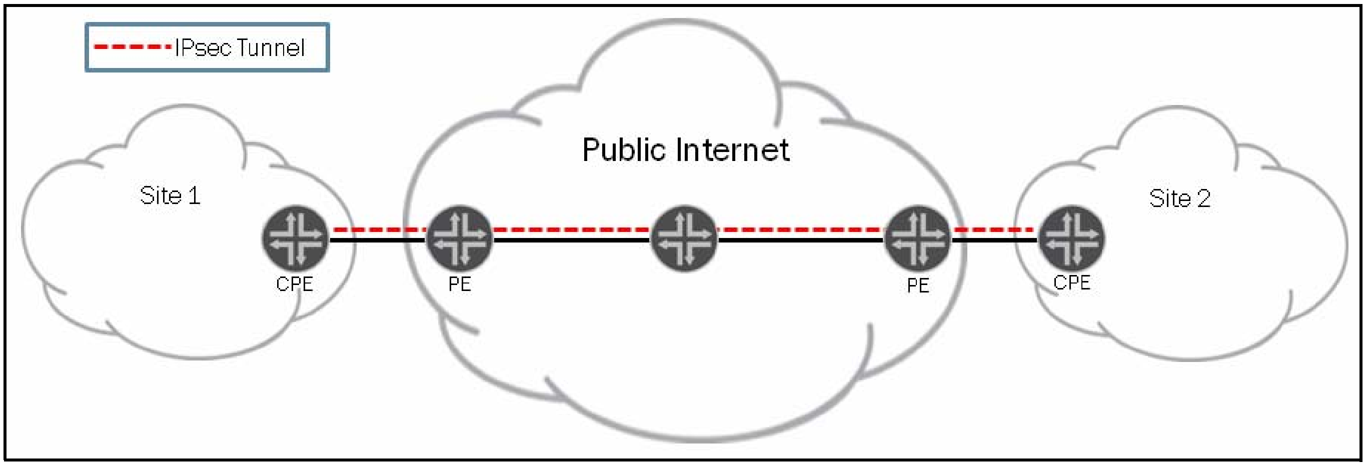
For a customer with a strong security requirement, IPsec is a perfect fit. However, key management and routing between sites are the customer's responsibility.

Security Services

Security services include the following:

- Access control;
- Data origin authentication;
- Replay protection;
- Data integrity;
- Data privacy (encryption); and
- Key management.

Routing Performed at CPE



In the example on the graphic, the customer provides the routing for its internal network. The tunneled traffic is forwarded across the public Internet as a normal IP packet.

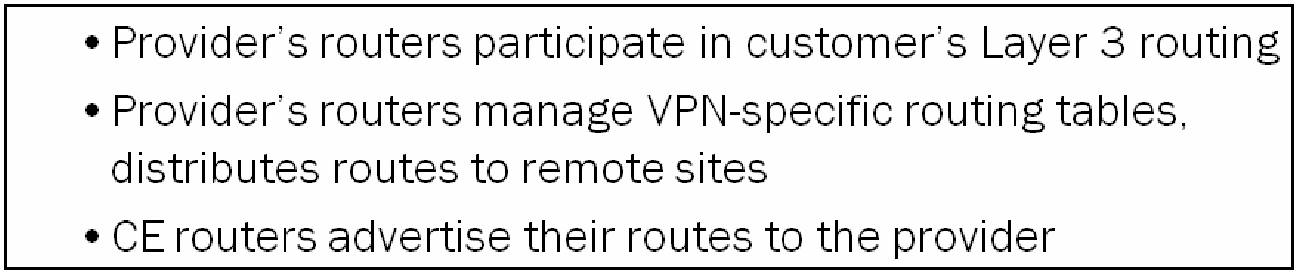
Tunnels Terminate on Subscriber Premises

Private IP traffic from the site 1 destined for site 2 is encapsulated using IPsec by the CPE. This traffic is then forwarded across the public Internet to the destination CPE. The branch office CPE then de-encapsulates the private IP traffic and forwards it to the destination host.

Only CPE Must Support IPsec

Typically, the customer’s edge devices are IPsec capable and create and maintain the tunnels between themselves. The ISP is only responsible for providing IP connectivity between the sites.

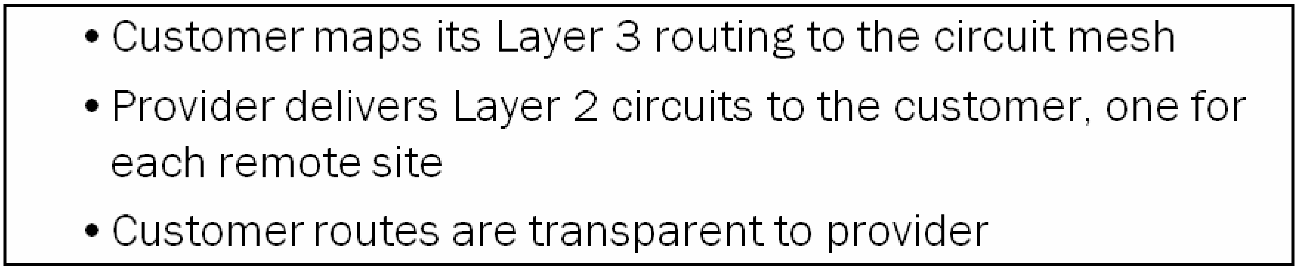
Provider-Provisioned Layer 3 VPN Characteristics



- Provider’s routers participate in customer’s Layer 3 routing
- Provider’s routers manage VPN-specific routing tables, distributes routes to remote sites
- CE routers advertise their routes to the provider

For Layer 3 VPNs, the provider’s routers participate in the customer’s Layer 3 routing. Thus, the customer’s routing protocol is terminated by the provider’s router. It is the responsibility of the provider’s router to manage VPN-specific routing tables and to distribute those VPN-specific routes to the customer’s remote sites.

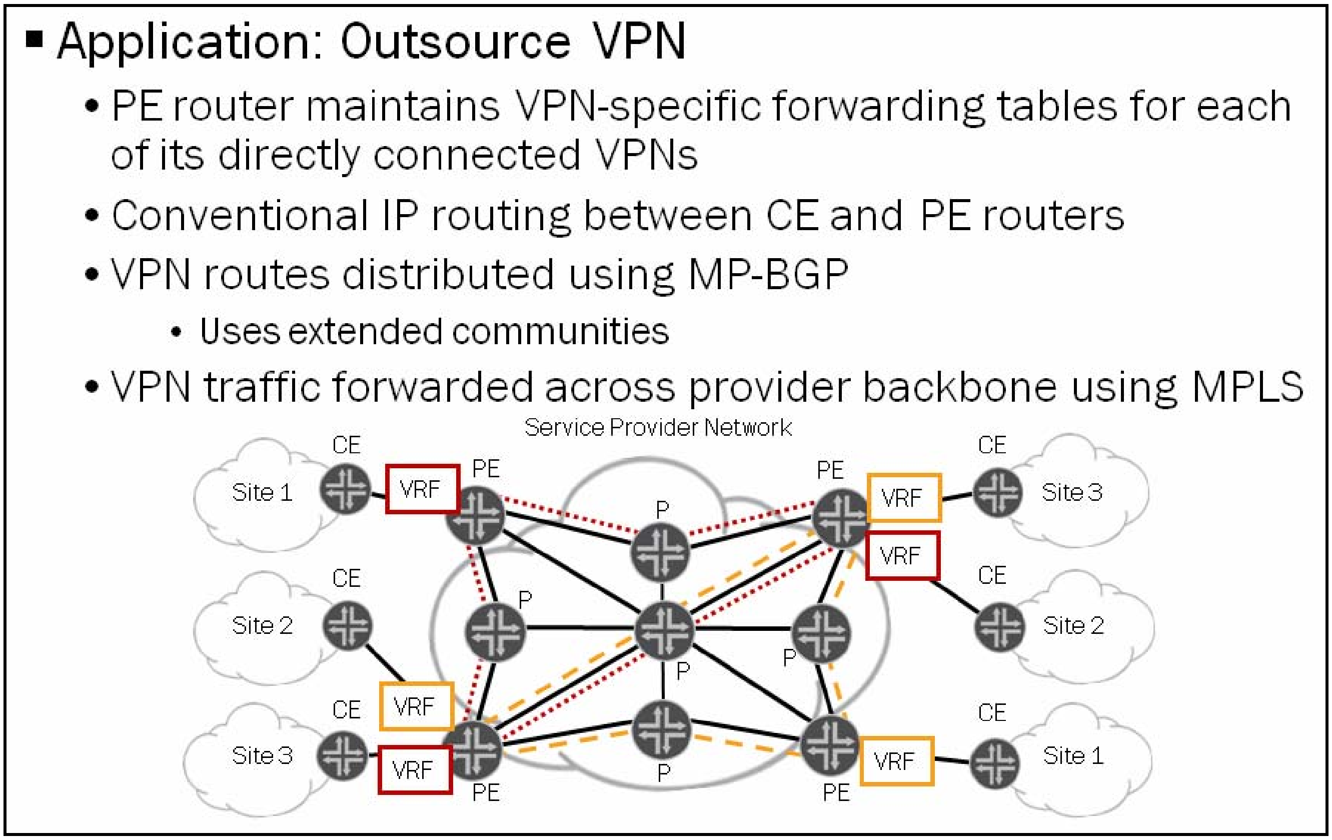
Provider-Provisioned Layer 2 VPN Characteristics



- Customer maps its Layer 3 routing to the circuit mesh
- Provider delivers Layer 2 circuits to the customer, one for each remote site
- Customer routes are transparent to provider

For Layer 2 VPNs, as with Frame Relay, a Layer 2 VPN customer maps its Layer 3 routing to the Layer 2 circuit mesh. In this situation, the provider delivers Layer 2 circuits to the customer, one for each remote site. The provider does not participate in the routing of the customer’s private IP traffic, so the routing protocol used by the customer edge (CE) device is terminated by the remote CE device.

Outsourced VPNs



- Application: Outsource VPN
  - PE router maintains VPN-specific forwarding tables for each of its directly connected VPNs
  - Conventional IP routing between CE and PE routers
  - VPN routes distributed using MP-BGP
    - Uses extended communities
  - VPN traffic forwarded across provider backbone using MPLS

MPLS-based VPNs make it possible for a service provider to offer value-added services to new and existing customers using its existing network infrastructure.

The Junos OS supports Layer 3 provider-provisioned VPNs based on RFC 4364. In this model, the provider edge (PE) routers maintain VPN-specific routing tables called VPN route and forwarding (VRF) tables for each of their directly connected VPNs. To populate these forwarding tables, the CE routers advertise routes to the PE routers using conventional routing protocols like RIP, OSPF and EBGP.

The PE routers then advertise these routes to other PE routers with Multiprotocol Border Gateway Protocol (MP-BGP) using extended communities to differentiate traffic from different VPN sites. Traffic forwarded from one VPN site to another is tunneled across the service provider's network using MPLS. The MPLS-based forwarding component allows support for overlapping address space and private addressing.

Label Distribution Protocol for LSPs

Setting up and maintaining label-switched paths (LSPs) between PE routers requires a label distribution protocol. Options include the LDP or RSVP.

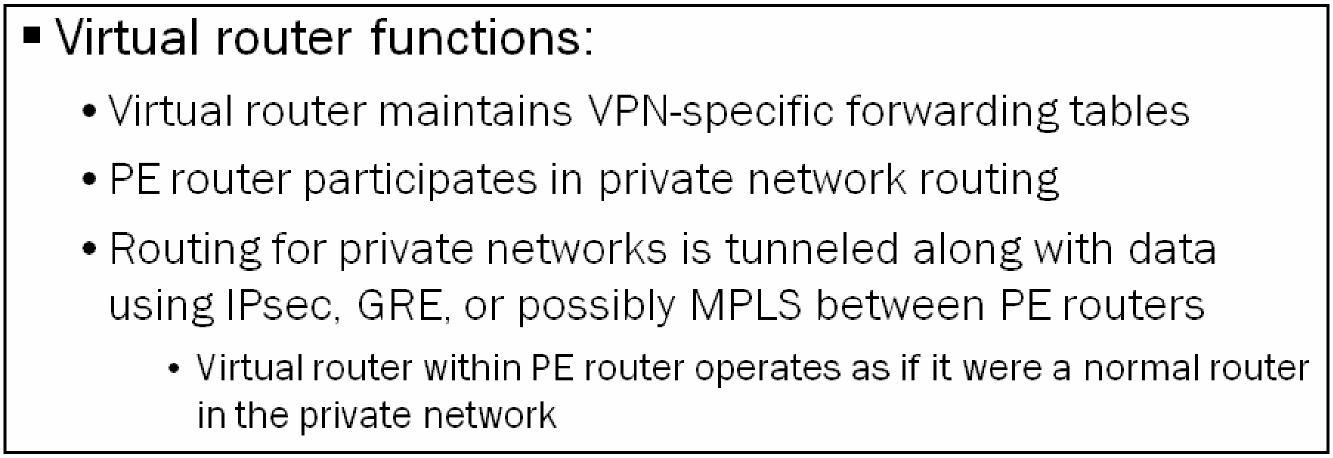
MP-BGP Distributes VPN Information

MP-BGP is used to distribute information about the VPNs. These communications include routing and reachability information as well as the MPLS labels that map traffic to a particular VPN forwarding table and interface.

Provider Constrains Connectivity by Route Filtering

To ensure that routing information about a particular VPN is only made available to sites participating in that VPN, the provider must constrain advertisements using routing policy (for example, route filtering).

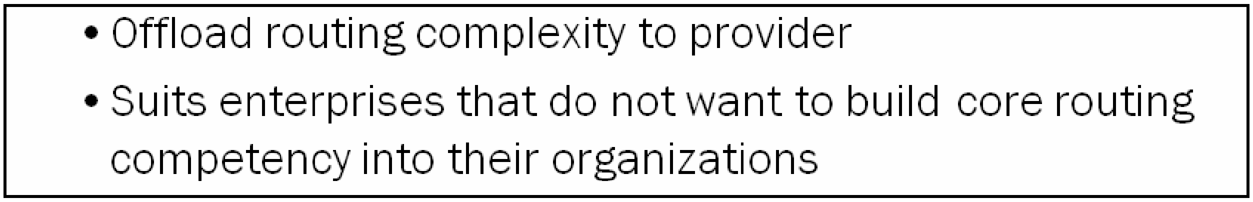
Virtual Routers



- Virtual router functions:
  - Virtual router maintains VPN-specific forwarding tables
  - PE router participates in private network routing
  - Routing for private networks is tunneled along with data using IPsec, GRE, or possibly MPLS between PE routers
    - Virtual router within PE router operates as if it were a normal router in the private network

A virtual router functions much like an RFC 4364 PE router in that it maintains site-specific routing instances and tables for use in the forwarding of IP-based VPN traffic. A significant difference, however, is that in the virtual router approach, the PE router does not terminate the routing protocol used by the CE device. In effect, the two PE routers create a *sham* link representing the connection between the PE routers for use in the flooding of OSPF LSAs across the provider's backbone.

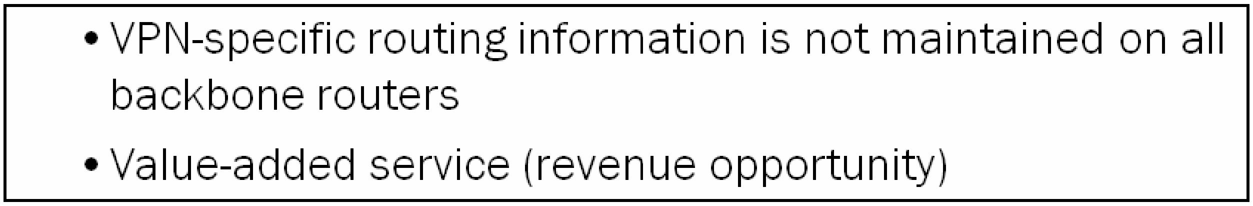
Advantages for the Subscriber



- Offload routing complexity to provider
- Suits enterprises that do not want to build core routing competency into their organizations

With Layer 3 provider-provisioned VPNs, the subscriber can offload its routing responsibilities to the provider, thus allowing the customer to focus on its core competencies.

Advantages for the Provider



- VPN-specific routing information is not maintained on all backbone routers
- Value-added service (revenue opportunity)

The provider can offer a value-added (revenue producing) service to its customers using a scalable IP-centric-based backbone technology.

Limitations of Provider-Provisioned VPNs

Layer 3 provider-provisioned VPNs do have some drawbacks. The configuration and maintenance of an RFC 4364 solution can represent a significant increase in the provider's administrative burden. This is especially true during situations where adds, moves, and changes to the VPNs are required. The use of automated provisioning tools can simplify day-to-day operations in the network greatly.

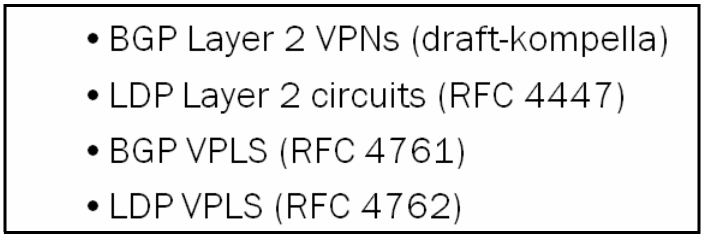
VPN provisioning mistakes can be costly, especially when considering that the provider could become liable for the security of the customer's networks.

Resistance to Outsourced Routing

It might be difficult to convince some customers to outsource their routing to the provider. For these customers, a Layer 2 VPN can be an ideal fit, as it allows them to control all aspects of their routing.

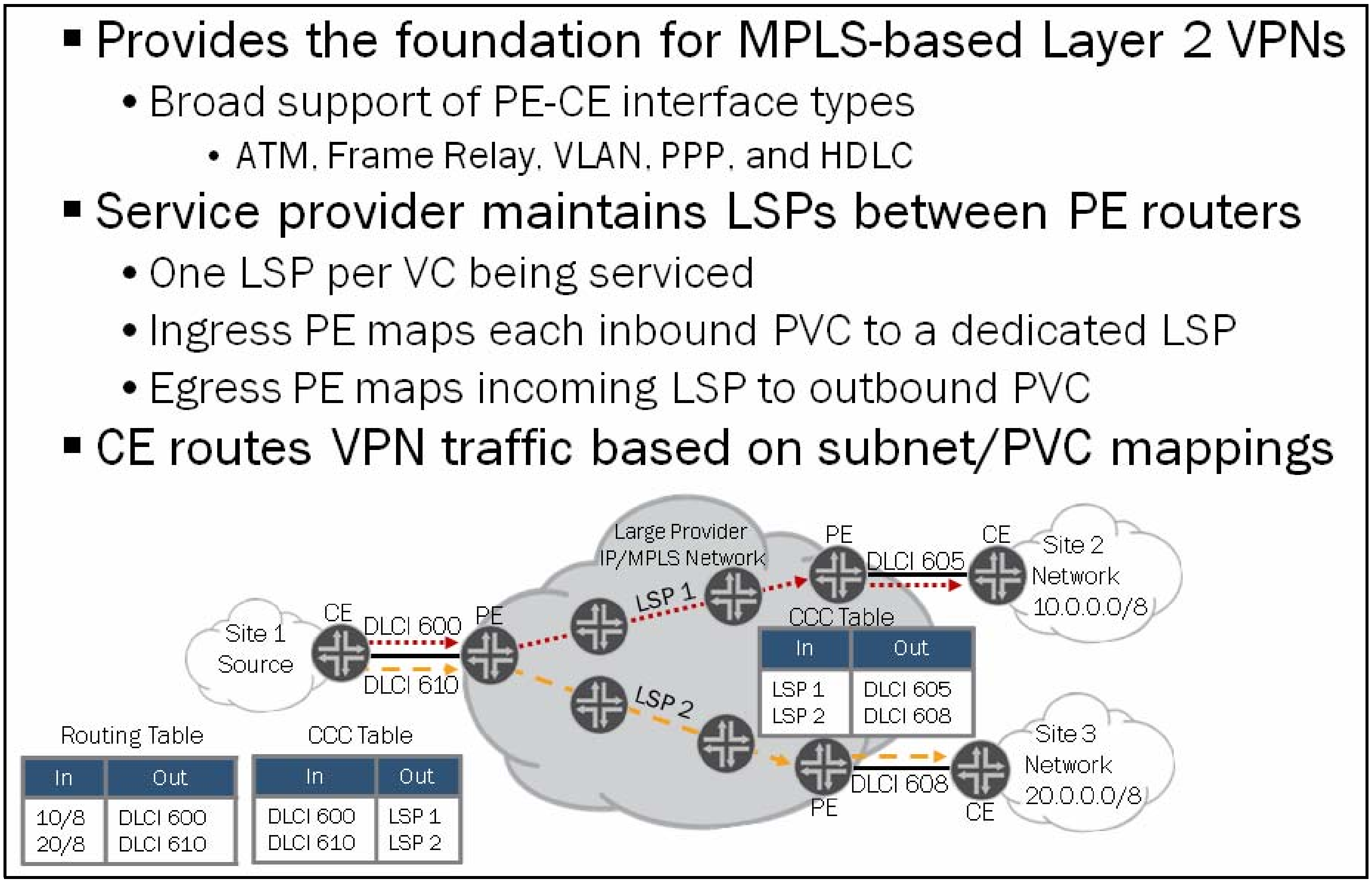
Layer 2 MPLS-Based VPNs

The following pages discuss circuit cross-connect (CCC), BGP Layer 2 VPNs, LDP Layer 2 VPNs, BGP signaled VPLS, and LDP signaled VPLS.



- BGP Layer 2 VPNs (draft-kompella)
- LDP Layer 2 circuits (RFC 4447)
- BGP VPLS (RFC 4761)
- LDP VPLS (RFC 4762)

CCC: The Foundation of Layer 2 VPNs



- Provides the foundation for MPLS-based Layer 2 VPNs
  - Broad support of PE-CE interface types
    - ATM, Frame Relay, VLAN, PPP, and HDLC
- Service provider maintains LSPs between PE routers
  - One LSP per VC being serviced
  - Ingress PE maps each inbound PVC to a dedicated LSP
  - Egress PE maps incoming LSP to outbound PVC
- CE routes VPN traffic based on subnet/PVC mappings

Large Provider IP/MPLS Network
Site 1 Source
CE
DLCI 600
DLCI 610
PE
LSP 1
LSP 2
PE
DLCI 605
CE
Site 2 Network 10.0.0.0/8
PE
DLCI 608
CE
Site 3 Network 20.0.0.0/8

Routing Table

In	Out
10/8 20/8	DLCI 600 DLCI 610

CCC Table

In	Out
DLCI 600 DLCI 610	LSP 1 LSP 2

CCC Table

In	Out
LSP 1 LSP 2	DLCI 605 DLCI 608

CCC provides the foundation for MPLS-based Layer 2 VPNs by providing support for the tunneling of Layer 2 frames over MPLS LSPs. CCC supports a variety of Layer 2 protocols including ATM, Frame Relay, virtual LANs (VLANs), PPP, and High-Speed Data Link Control (HDLC).

Provider Maintains LSP and Connection Mesh

The CCC application does not support label stacking. As a result, the provider must configure one LSP, per direction, per virtual circuit being serviced. The provider must also define each connection by manually mapping local connection identifiers to LSPs.

CE Maps Connections to Remote Sites

Customers route traffic based on subnet/permanent virtual connection (PVC) mappings, as they would with any conventional Frame Relay, ATM, or private line solution.

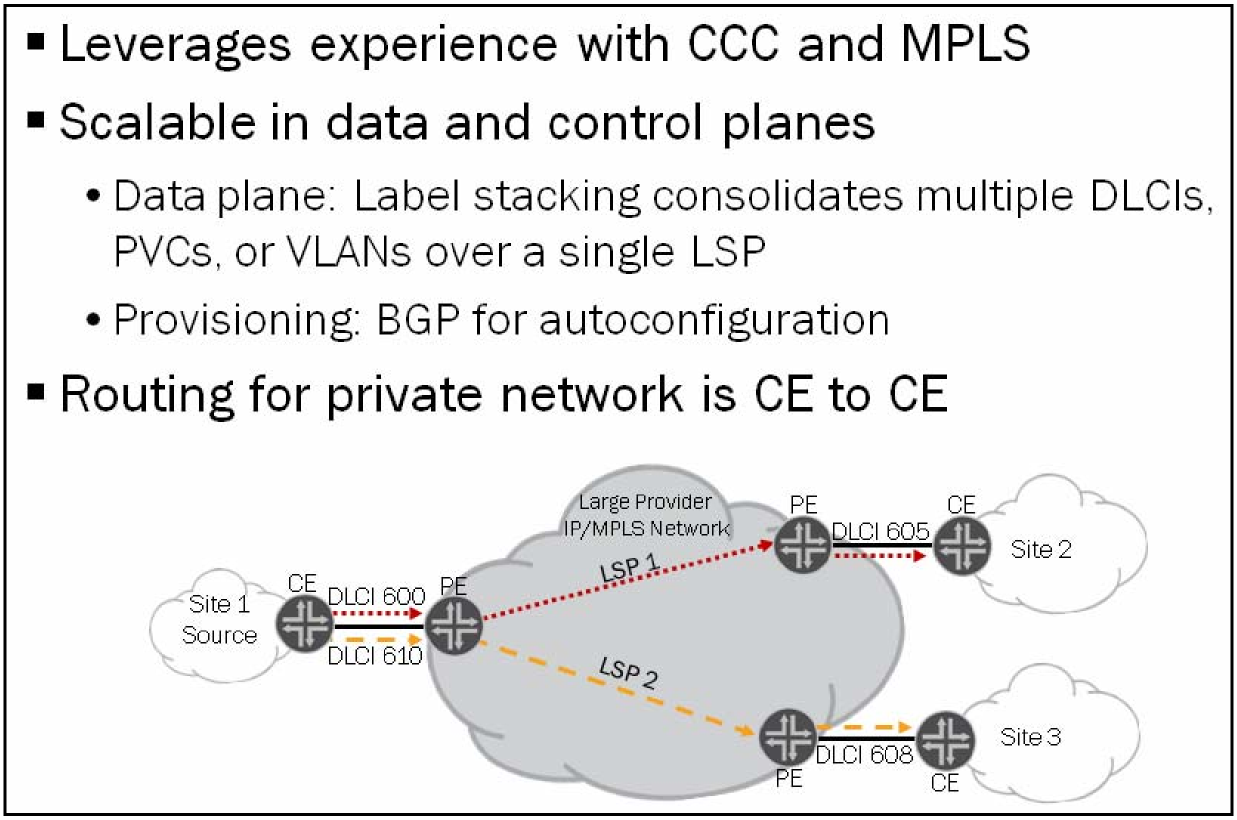
CCC Drawbacks

The following list details some of the drawbacks of CCC:

- CE and PE router configuration can be complex, especially during adds, moves, and changes. The customer must coordinate with the service provider.
- Each data-link connection identifier (DLCI)/PVC requires a dedicated set of MPLS LSPs. There can be no sharing of the LSP when using CCC.

- CCC as a Layer 2 VPN solution is only appropriate for small numbers of individual private connections.
- Interface types must be the same at all CE device locations. For instance, if Frame Relay is used at one VPN site then Frame Relay must be used at all other sites. However, the Junos OS has a feature called translational cross-connect (TCC) that can be used when there are different interfaces types at the CE locations.

Leverages Experience with CCC and MPLS



- Leverages experience with CCC and MPLS
- Scalable in data and control planes
  - Data plane: Label stacking consolidates multiple DLCIs, PVCs, or VLANs over a single LSP
  - Provisioning: BGP for autoconfiguration
- Routing for private network is CE to CE

With BGP Layer 2 VPNs the VPNs are created using bidirectional MPLS LSPs, similar to CCC. However, instead of mapping the LSPs to an interface on the PE routers, the LSPs are automatically mapped to Layer 2 circuits. Data is forwarded using a two-level label stack that permits the sharing of the LSP by numerous Layer 2 connections. The outer label delivers the data across the core of the network from the ingress PE router to the egress PE router. The egress PE router then uses the inner label to map the Layer 2 data to a particular VPN-specific table.

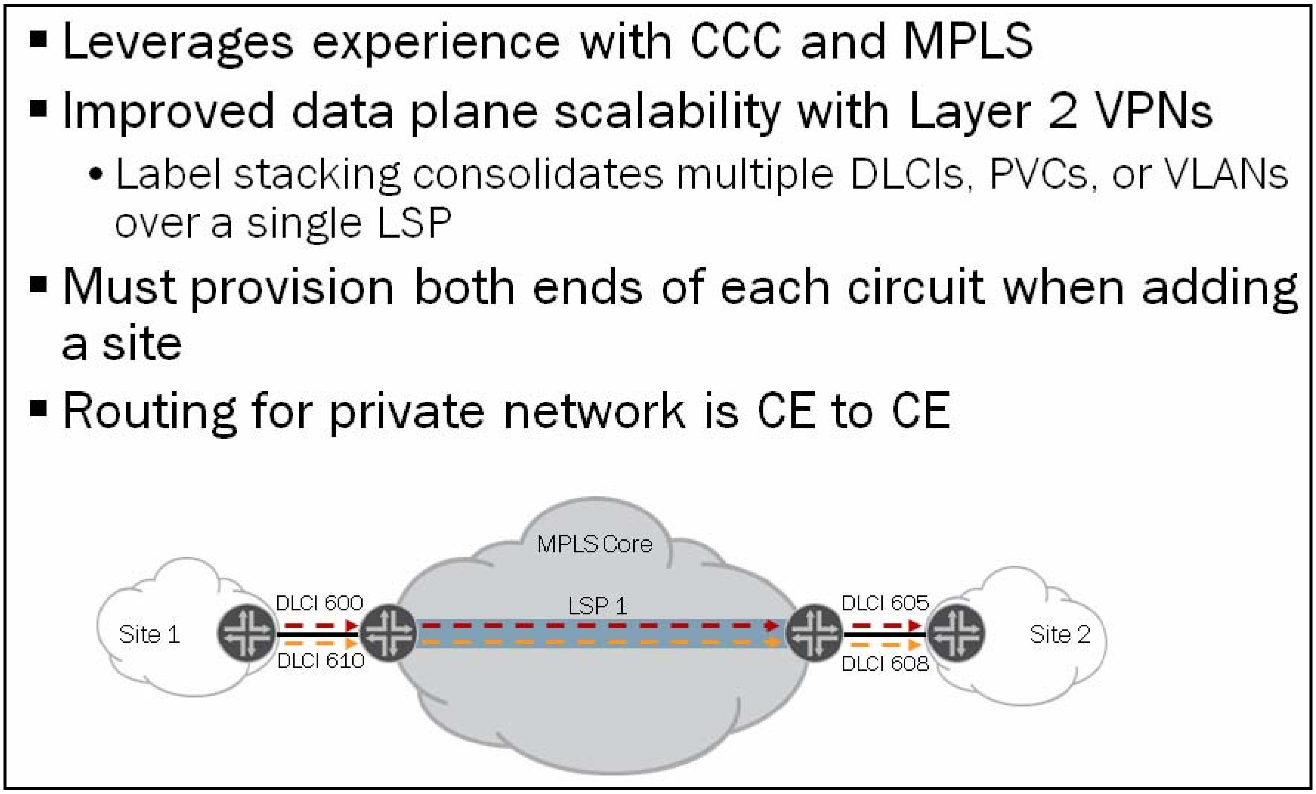
Scalable in the Data and Control Planes

The use of label stacking improves scalability, as now multiple VPNs can share a single set of LSPs for the forwarding of traffic. Also, by over-provisioning Layer 2 circuits on the PE device (described in a later chapter), adds and changes are simplified, as only the new site's PE router requires configuration. This automatic connection to LSP mapping greatly simplifies operations when compared to the CCC approach.

Routing Is CE to CE

Because the provider delivers Layer 2 circuits to the customer, the routing for the customer's private network is entirely in the hands of the customer. From the perspective of the attached CE device, there is no operational difference between a Frame Relay service, CCC, and a BGP Layer 2 VPN solution.

Leverages Experience with CCC and MPLS



- Leverages experience with CCC and MPLS
- Improved data plane scalability with Layer 2 VPNs
  - Label stacking consolidates multiple DLCIs, PVCs, or VLANs over a single LSP
- Must provision both ends of each circuit when adding a site
- Routing for private network is CE to CE

MPLS Core
Site 1
DLCI 600
DLCI 610
LSP 1
DLCI 605
DLCI 608
Site 2

With LDP Layer 2 circuits, the circuits are created using bidirectional MPLS LSPs, like CCC. However, instead of mapping the LSPs to an interface on the PE routers, the LSPs are mapped to a VPN-specific forwarding table (similar to BGP Layer 2 VPNs). This table then maps the data to a Layer 2 circuit. The LDP Layer 2 circuit approach also makes use of stacked headers for improved scalability.

Data Plane Scalability

Label stacking means that PE devices can use a single set of MPLS LSPs between them to support many VPNs. The LDP Layer 2 circuit signaling approach does not support the auto-provisioning of Layer 2 connections, and it relies on LDP for signaling.

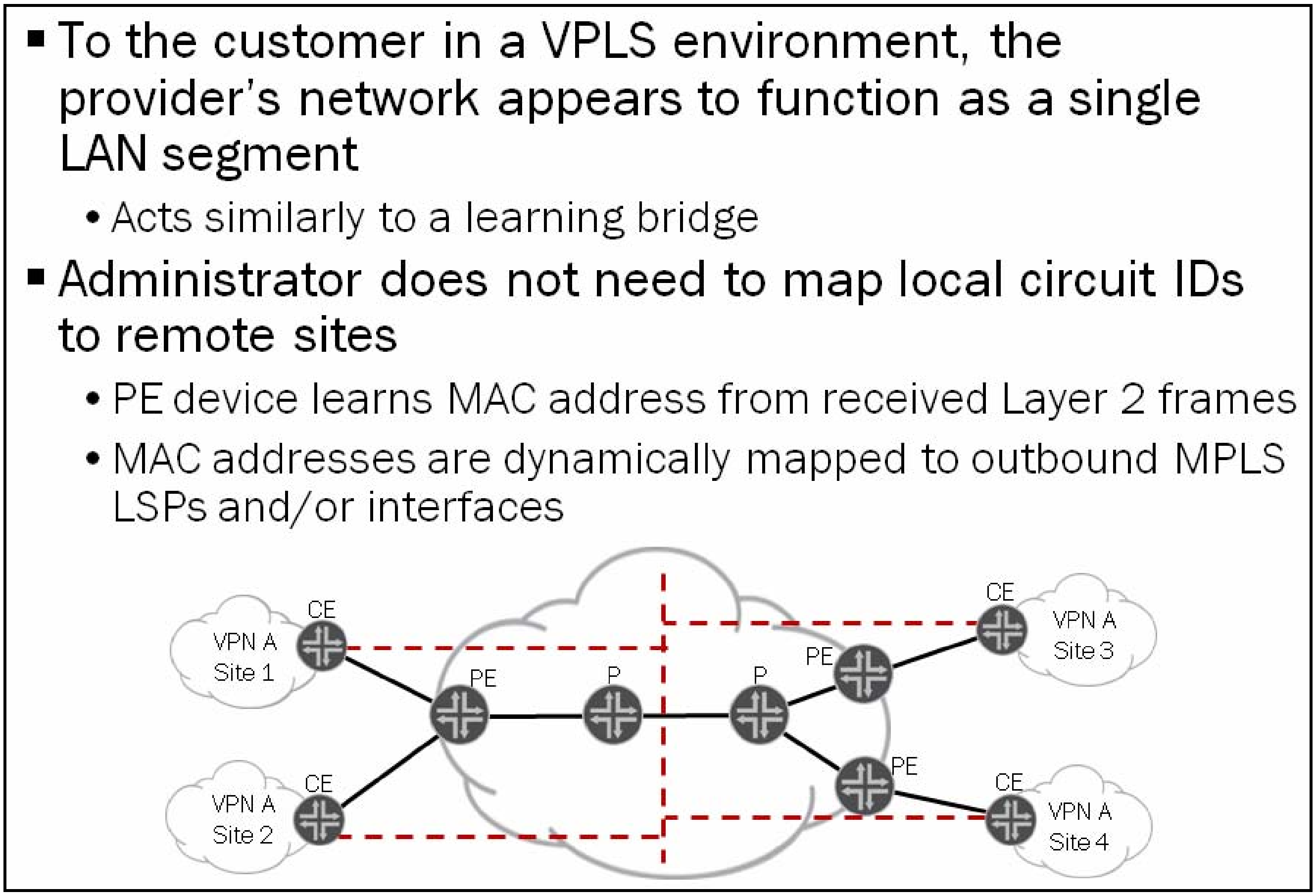
Manual Provisioning for Moves and Changes

The LDP Layer 2 circuit approach requires configuration on all PE routers involved in the VPN when moves, adds, and changes occur. Draft-kompella support for MP-BGP-based signaling and its automatic connection mapping features make it far simpler to deploy and maintain a Layer 2 VPN service.

Routing for Private Network Is CE to CE

Because the provider delivers Layer 2 circuits to the customer, the routing for the customer’s private network is entirely in the hands the customer.

Provider's Network Appear to Be a Single LAN Segment



- To the customer in a VPLS environment, the provider's network appears to function as a single LAN segment
  - Acts similarly to a learning bridge
- Administrator does not need to map local circuit IDs to remote sites
  - PE device learns MAC address from received Layer 2 frames
  - MAC addresses are dynamically mapped to outbound MPLS LSPs and/or interfaces

VPN A Site 1
CE
PE
P
P
PE
CE
VPN A Site 3
VPN A Site 2
CE
PE
CE
VPN A Site 4

A newer service that can be provided to the customer is VPLS. To the customer, VPLS appears to be a single LAN segment. In fact, it appears to act similarly to a learning bridge. That is, when the destination media access control (MAC) address is not known, an Ethernet frame is sent to all remote sites. If the destination MAC address is known, it is sent directly to the site that owns it. The Junos OS supports two variations of VPLS, BGP signaled VPLS and LDP signaled VPLS. VPLS is covered in more detail in later chapters.

No Need to Map Local Circuit to Remote Sites

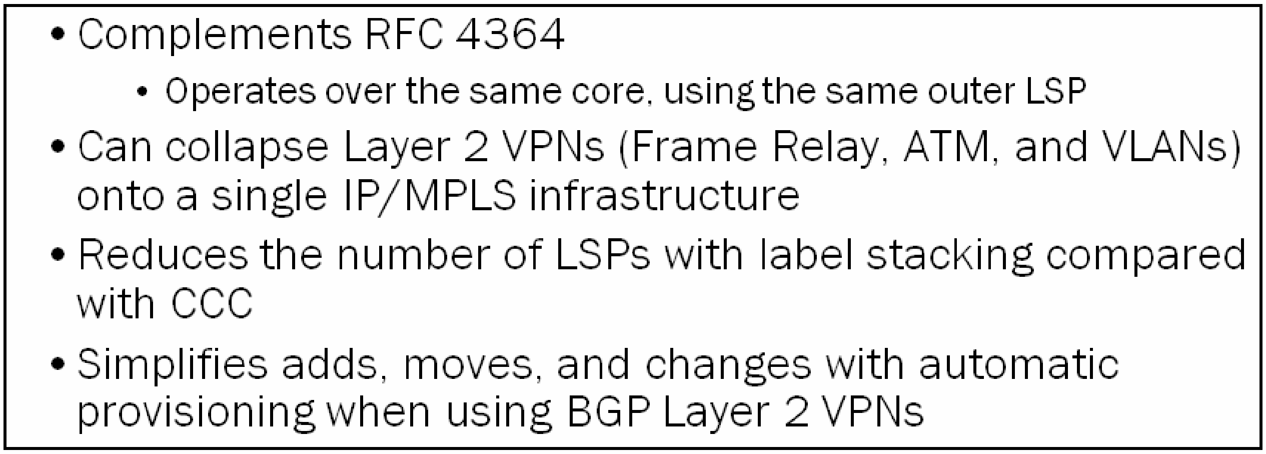
In VPLS, PE devices learn MAC addresses from the frames that it receives. They use the source and destination addresses to dynamically create a forwarding table (*`vpn-name`*`.vpls`) for Ethernet frames. Based on this table, frames are forwarded out of directly connected interfaces or over an MPLS LSP across the provider core. This behavior allows an administrator to not have to manually map Layer 2 circuits to remote sites.

Subscriber Advantages

- Can outsource circuits
- Maintains control of routing
- Uses any Layer 3 protocol

With Layer 2 VPNs the customer can outsource Layer 2 circuits to the provider over an existing Internet access circuit while maintaining control over the routing of its traffic. Also, because the provider is encapsulating Layer 2 traffic for transport using MPLS, the customer can use any Layer 3 protocol—not only IP-based protocols.

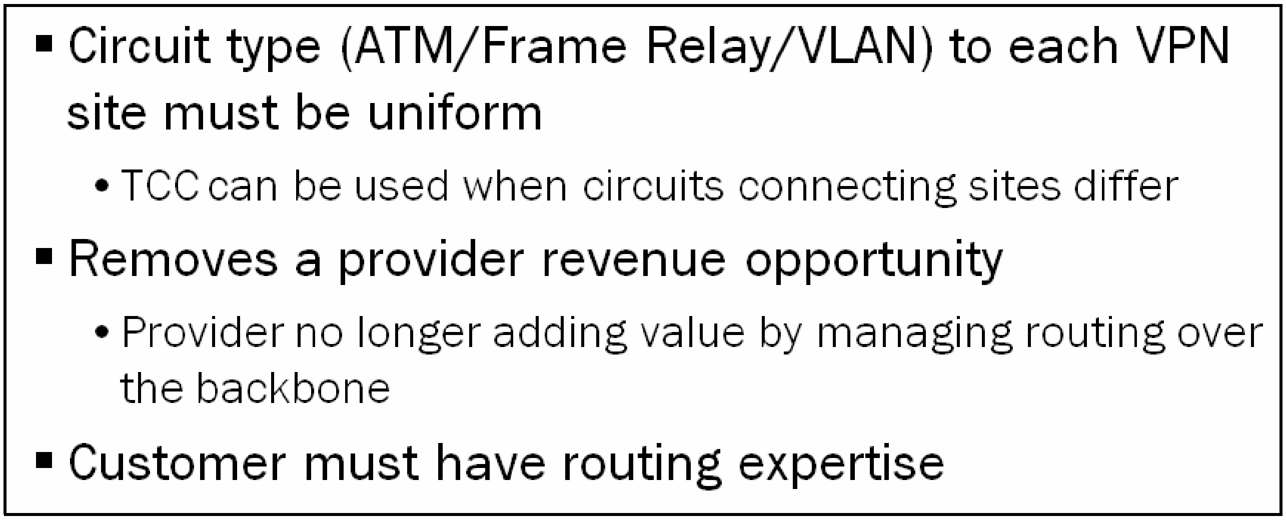
Provider Advantages



- Complements RFC 4364
  - Operates over the same core, using the same outer LSP
- Can collapse Layer 2 VPNs (Frame Relay, ATM, and VLANs) onto a single IP/MPLS infrastructure
- Reduces the number of LSPs with label stacking compared with CCC
- Simplifies adds, moves, and changes with automatic provisioning when using BGP Layer 2 VPNs

Layer 2 and Layer 3 VPNs can coexist by using the same MPLS transport and signaling protocols. The provider can now sell Frame Relay or ATM circuits to different customers using its existing IP core. Automatic provisioning of Layer 2 circuits simplifies the processes of adds, moves, and changes. Also, the use of label stacking greatly improves efficiency and scalability.

Circuit Types Must Be the Same



- Circuit type (ATM/Frame Relay/VLAN) to each VPN site must be uniform
  - TCC can be used when circuits connecting sites differ
- Removes a provider revenue opportunity
  - Provider no longer adding value by managing routing over the backbone
- Customer must have routing expertise

The circuit type (ATM/Frame Relay/VLAN) to each VPN site must be uniform. TCC can be used to connect sites with different circuit types. TCC removes the Layer 2 frame and replaces it with a new one used for the outgoing circuit.

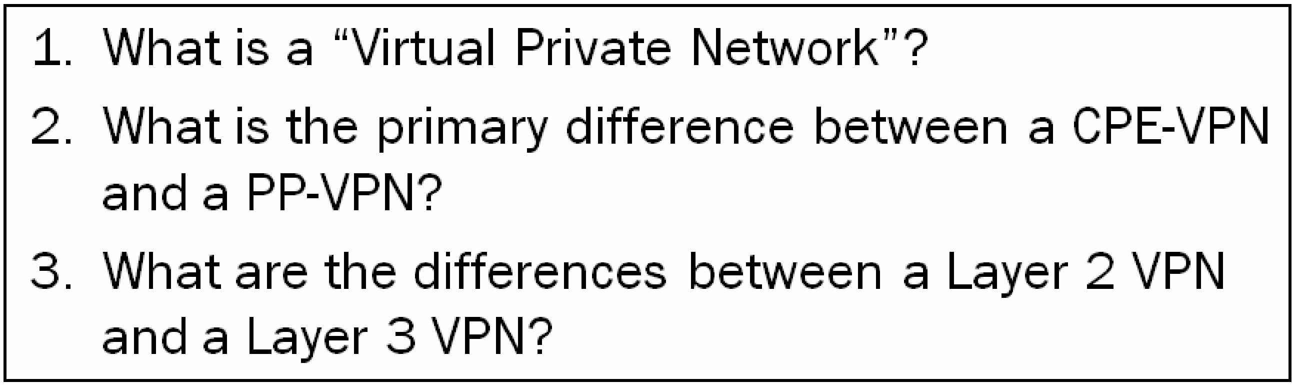
Removes Revenue Opportunity

Because the customer is responsible for the routing of traffic between its sites, the provider can no longer add value by providing outsourced routing services, as is the case with Layer 3 VPNs.

Customer Routing Experience

The customer must have routing expertise and the necessary staffing to configure and maintain its backbone routing.

Review Questions



1. What is a “Virtual Private Network”?
2. What is the primary difference between a CPE-VPN and a PP-VPN?
3. What are the differences between a Layer 2 VPN and a Layer 3 VPN?

Answers to Review Questions

1.

A VPN is a private network that is constructed over a shared, public infrastructure such as Frame Relay, an ATM network, or the Internet.

2.

The primary difference is that the CPE VPN requires the customer equipment to create and manage tunnels for the private IP traffic. A provider-provisioned VPN solution is a VPN that relies on the provider’s equipment to create and manage tunnels for the private traffic using MPLS as the enabling technology.

3.

The first and most notable difference is with a Layer 2 VPN solution, the backbone routing is the responsibility of the customer. With a Layer 3 VPN solution, the backbone routing is handled by the provider. Another difference is that with a Layer 2 VPN solution, non-IP traffic can be passed from site to site. With a Layer 3 VPN solution the traffic has to be IP.

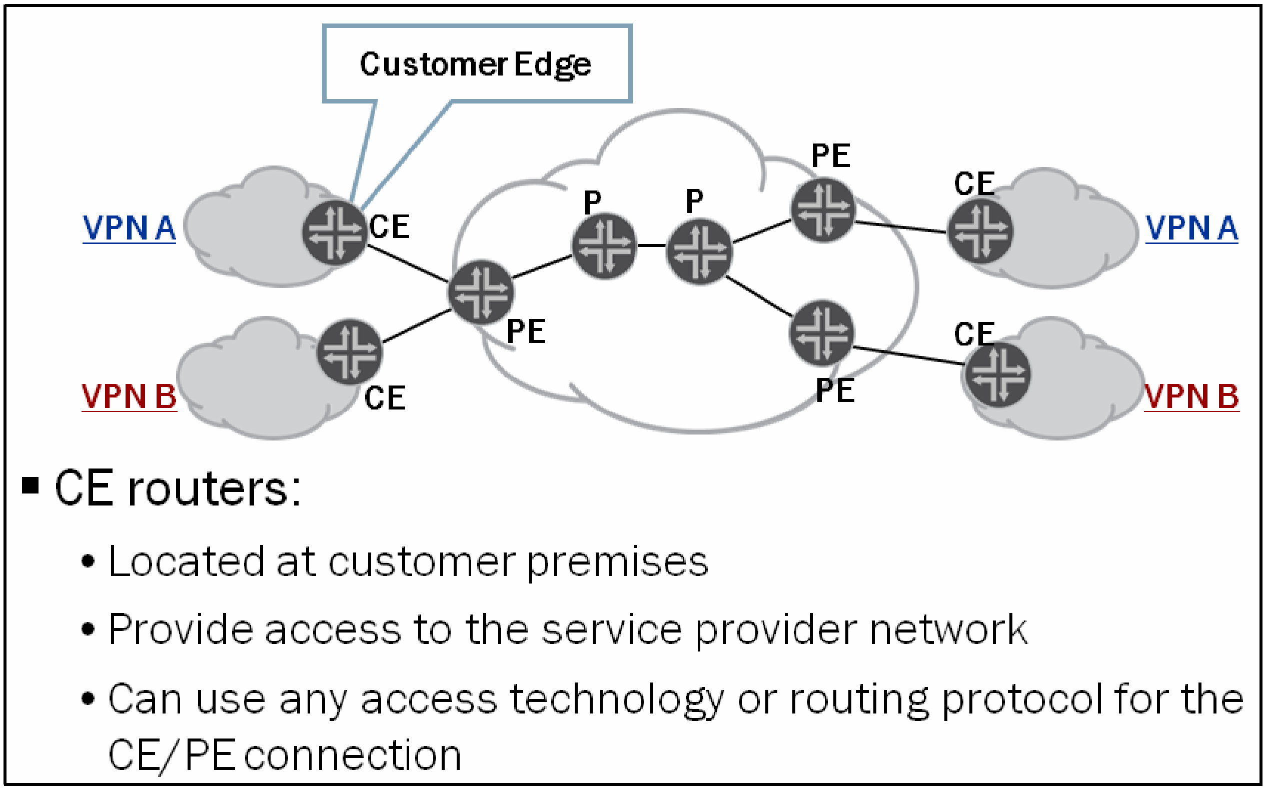
JNCIS-SP Study Guide—Part 3

Chapter 7: Layer 3 VPNs

This Chapter Discusses:

- The roles of provider (P), provider edge (PE), and customer edge (CE) routers;
- Virtual private network (VPN) IP version 4 (IPv4) address formats;
- Route distinguisher use and formats;
- RFC 4364 control flow; and
- RFC 4364 data flow.

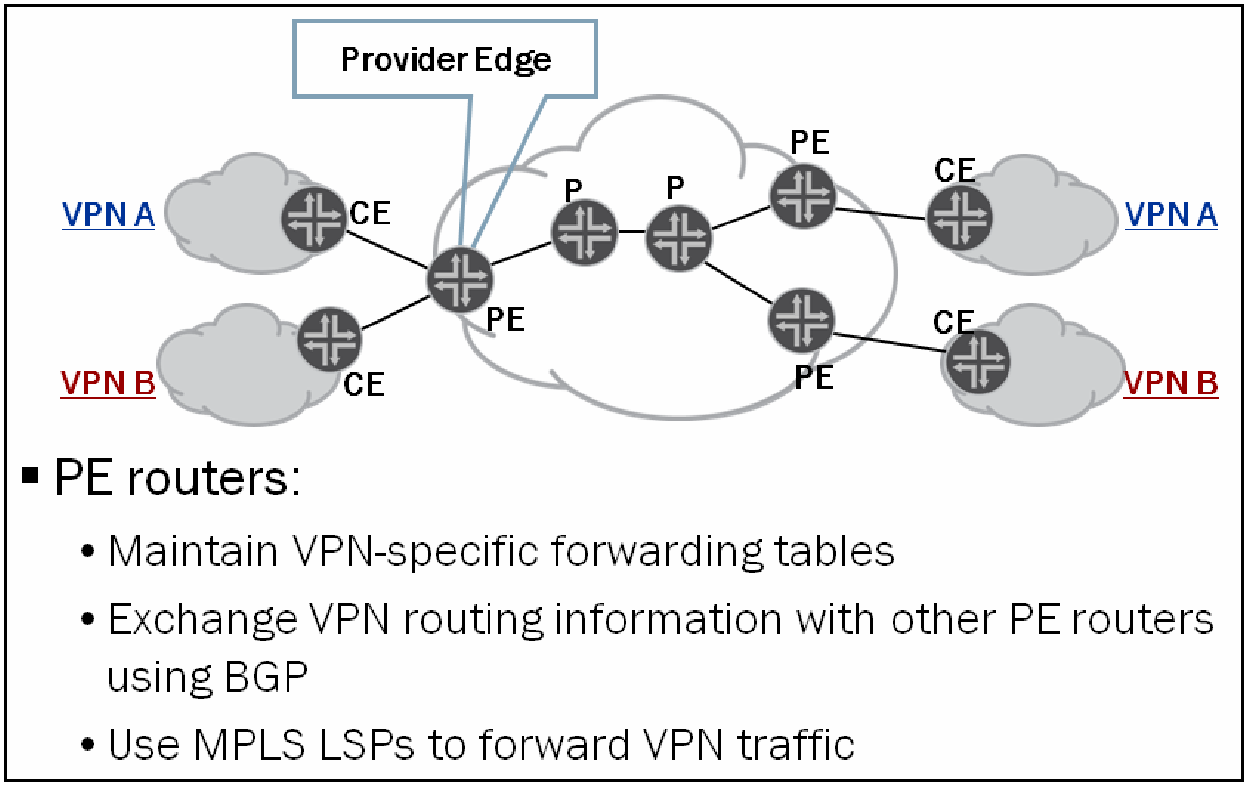
Customer Edge Routers



- CE routers:
  - Located at customer premises
  - Provide access to the service provider network
  - Can use any access technology or routing protocol for the CE/PE connection

CE routers are located at the customer location and provide access to the provider-provisioned VPN (PP-VPN) service. CE routers can interface to provider PE routers using virtually any Layer 2 technology and routing protocol.

Provider Edge Routers



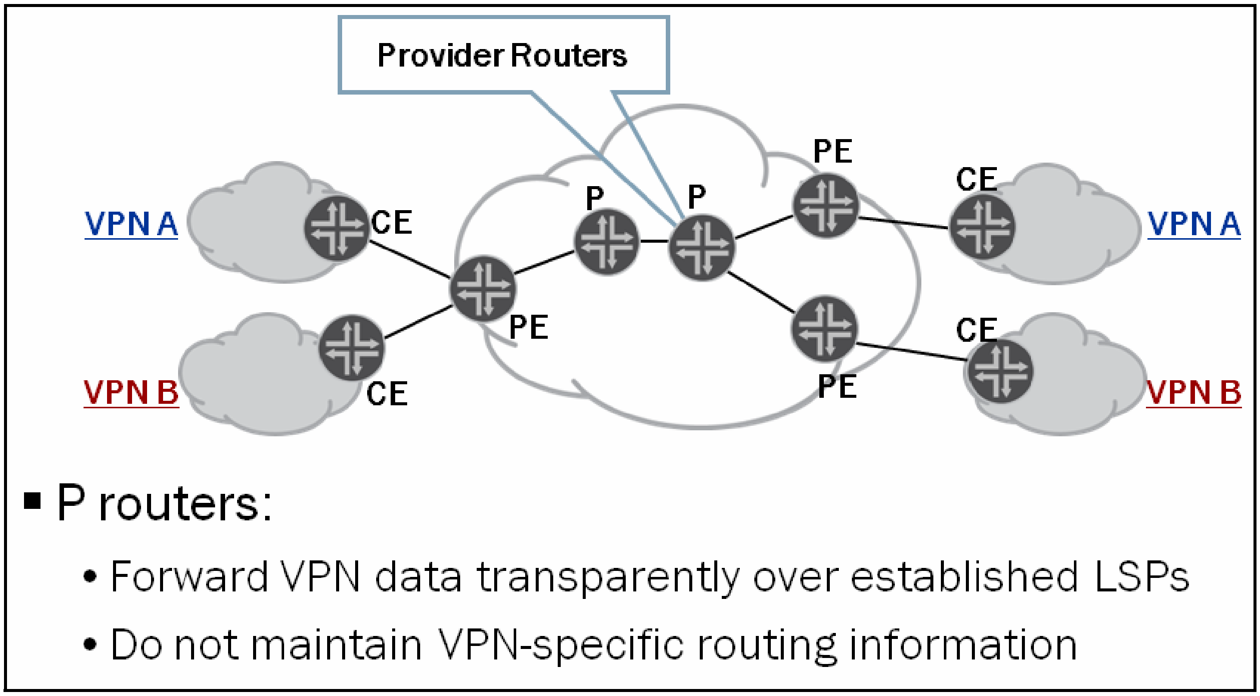
- PE routers:
  - Maintain VPN-specific forwarding tables
  - Exchange VPN routing information with other PE routers using BGP
  - Use MPLS LSPs to forward VPN traffic

PE routers are located at the edge of the provider's network. They interface to the CE routers on one side and to the provider's core routers on the other. PE routers maintain site-specific VPN route and forwarding (VRF) tables. The PE and CE routers function as routing peers, with the PE router terminating the routing exchange between customer sites and the provider's core.

Routes learned from the CE routers (and stored in the PE router's VRF table) are sent to remote PE routers using Multiprotocol Border Gateway Protocol (MP-BGP).

PE routers use MPLS LSPs when forwarding customer VPN traffic between sites. LSP tunnels in the provider's network separate VPN traffic in the same fashion as PVCs in a legacy ATM or Frame Relay network.

Provider Routers

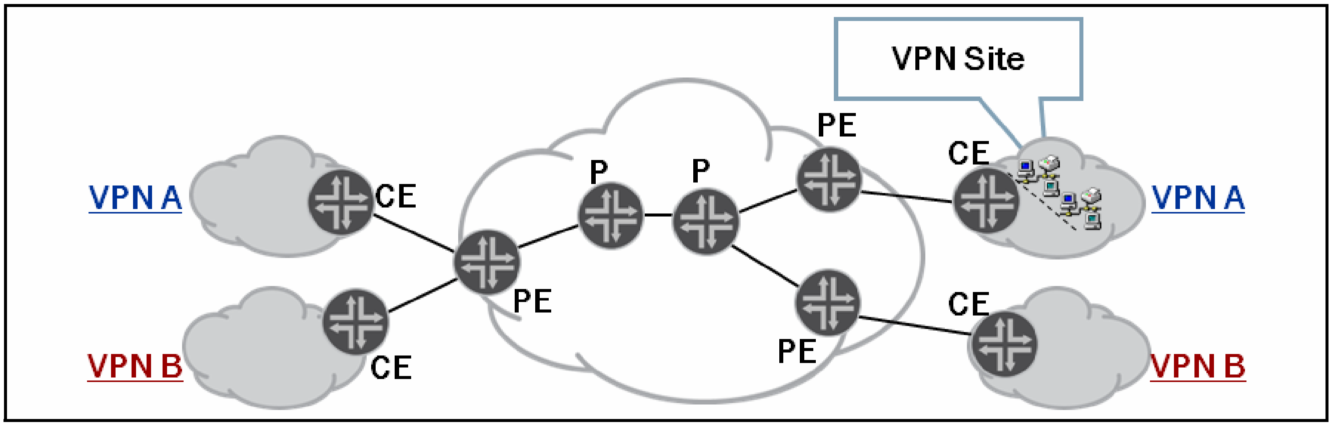


- P routers:
  - Forward VPN data transparently over established LSPs
  - Do not maintain VPN-specific routing information

Provider (P) routers are located in the provider's core. These routers do not carry VPN customer routes, nor do they interface in the VPN control and signaling planes. This is a key aspect of the RFC 4364 scalability model; only PE routers are aware of VPN customer routes, and no single PE router must hold all VPN customer state information.

P routers are involved in the VPN forwarding plane where they act as label-switching routers (LSRs) performing label swapping (and popping) operations.

VPN Sites

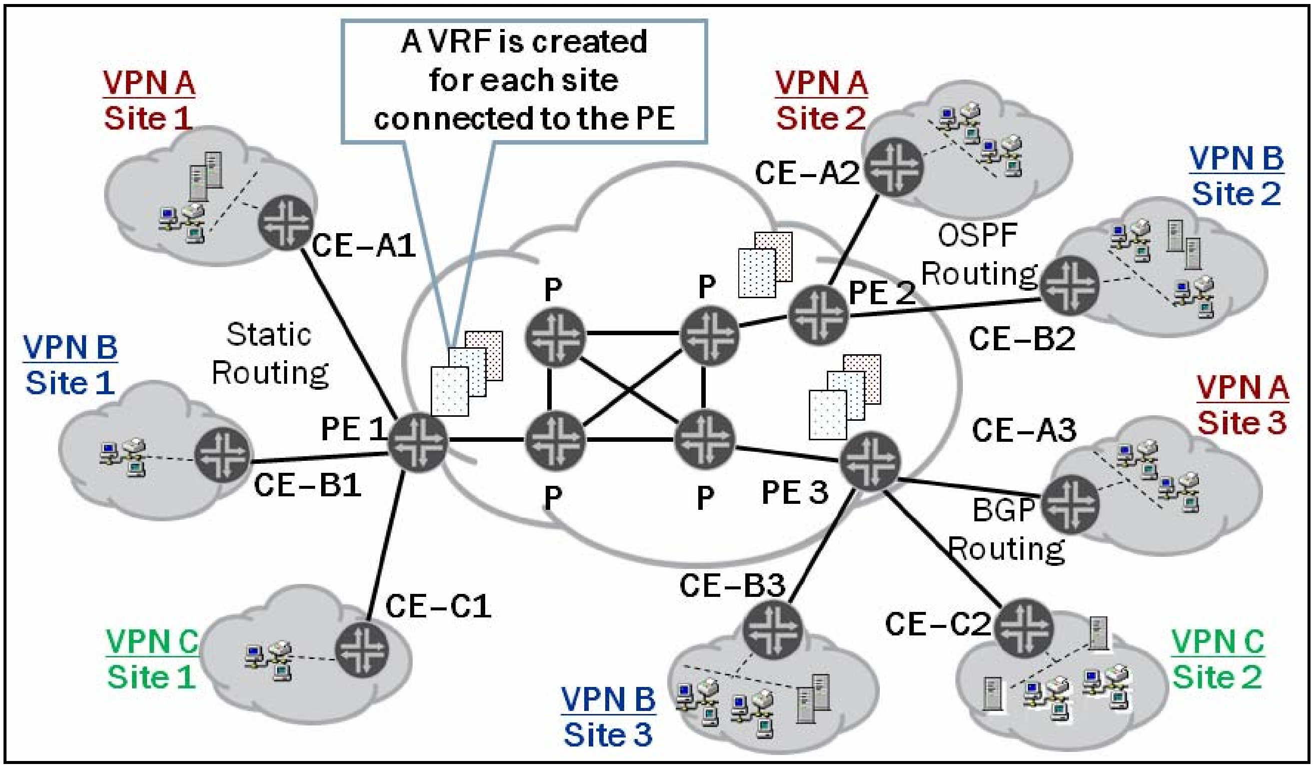


A VPN site is a collection of devices that can communicate with each other without the need to transit the provider's backbone. A site can range from a single location with one router to a network consisting of many geographically diverse routers.

Mapped to a VRF

Each VPN site is attached to at least one PE router and can be dual-homed with multiple connections to different PE routers. Each site is associated with a site-specific VRF table in the PE routers. It is here that the PE router maintains the routes specific to that site and, based on policy, the routes for remote sites to which this location can communicate.

Virtual Private Network Routing and Forwarding Tables



In the Layer 3 VPN model, site-specific VRF tables house each site's routes. This separation of routes allows VPN customers to use private addresses that can overlap with addresses used by other VPN customers.

In this graphic, PE 1 has three VRF tables—one for each of its attached VPN sites. The VRF tables store routes learned from the attached site, as well as routes learned through MP-BGP interaction with remote PE routers. In the latter case, VPN policy determines which routes are copied into which VRF tables based on the presence of a VPN-IPv4 route attribute known as a route target.

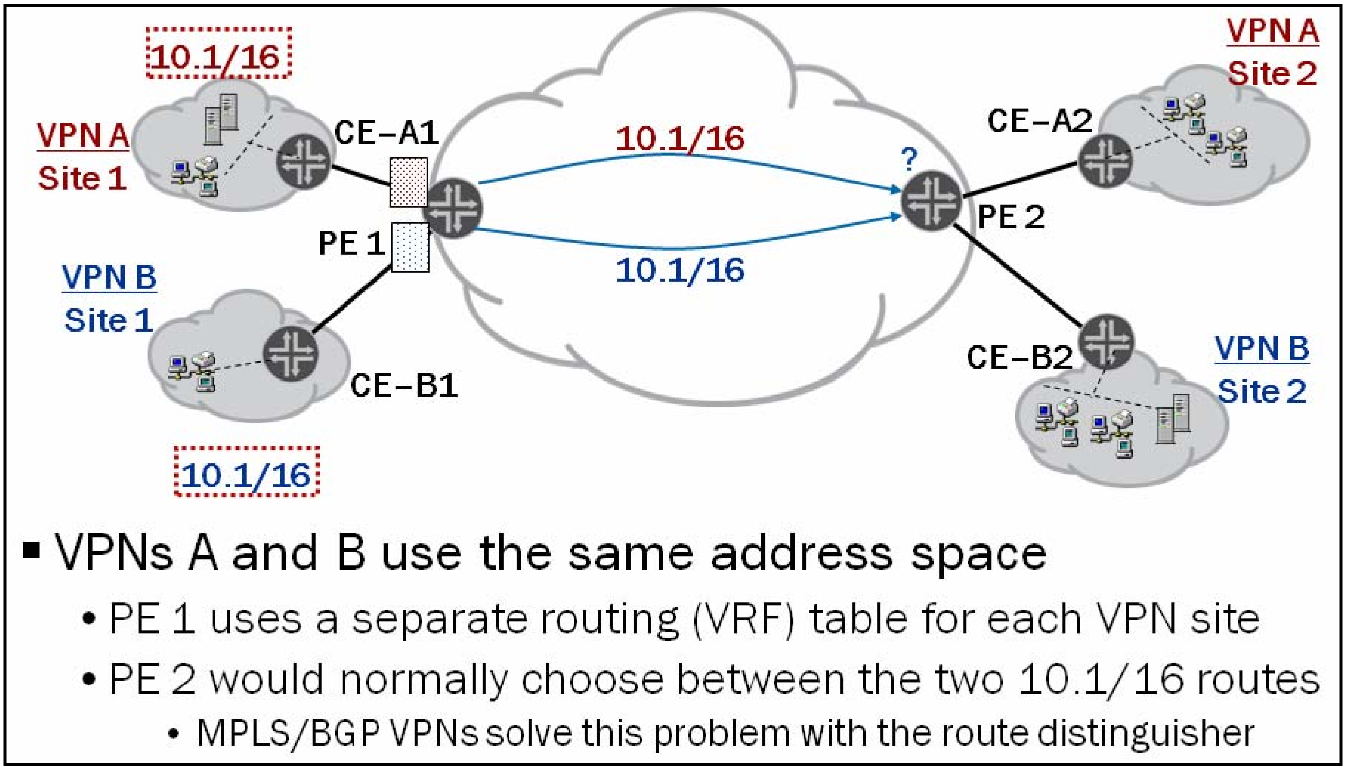
VRF Table Population

As mentioned previously, each PE router maintains site-specific VRF tables that house routes learned from the local CE device, as well as routes learned from remote PE routers having matching route attributes.

Site Separation

When a packet is received from a given site, the PE router performs a longest-match Layer 3 lookup against only the entries housed in that site's VRF table. This separation permits duplicate addressing among VPN customers with no chance of routing ambiguity.

Duplicate Addresses Welcome!



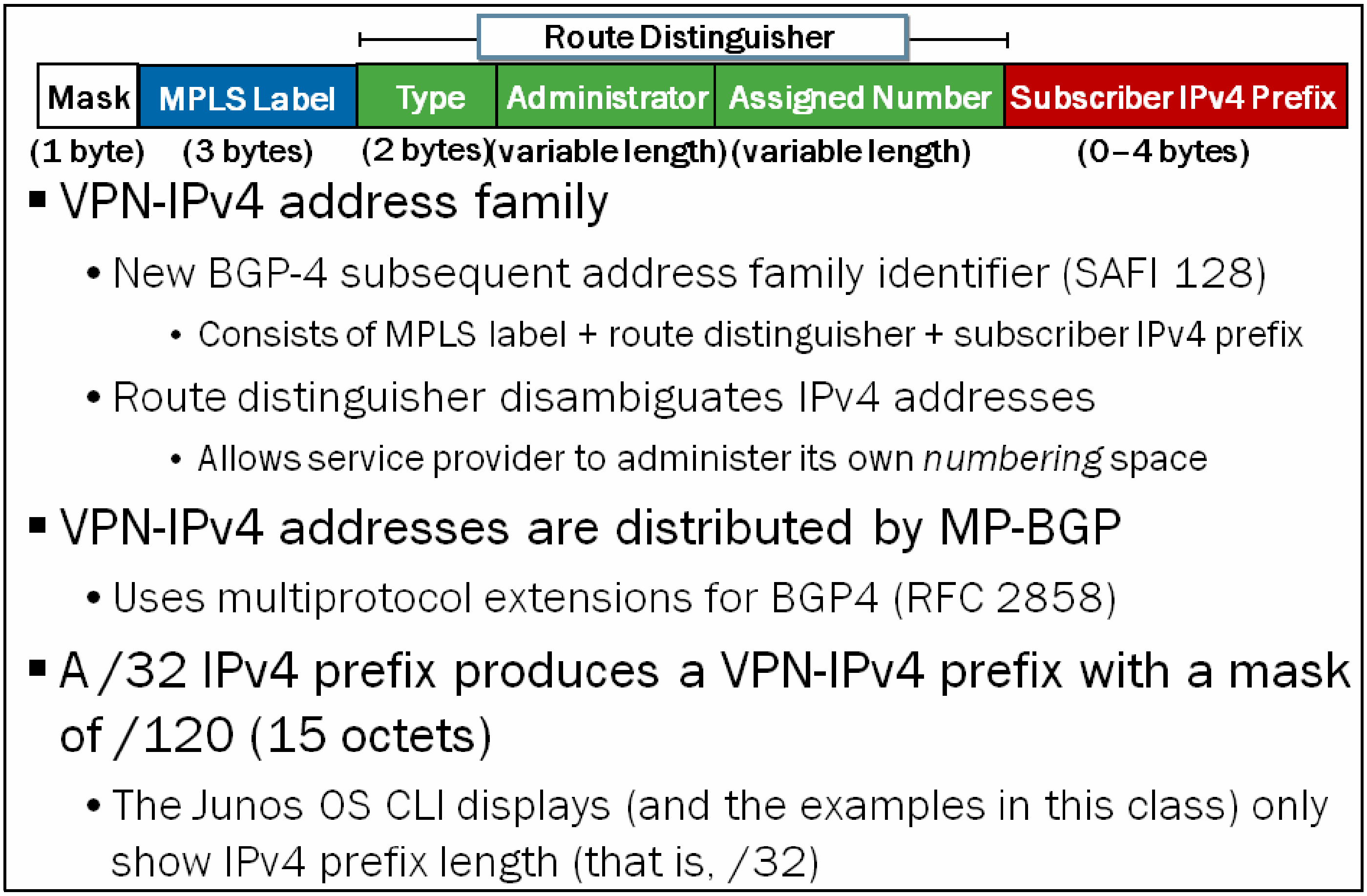
- VPNs A and B use the same address space
  - PE 1 uses a separate routing (VRF) table for each VPN site
  - PE 2 would normally choose between the two 10.1/16 routes
    - MPLS/BGP VPNs solve this problem with the route distinguisher

This graphic stresses that two VPN customers can use overlapping address space with no issues due to the separation of their routes in site-specific VRF tables.

In this example, VPN Site A is using the 10.1/16 addresses space, which is also being used by VPN customer B. However, housing these overlapping routes in separate VRF tables on PE routers is only half of the solution. A mechanism is needed to allow the PE routers to exchange these routes with remote PE routers without any chance of one address *stepping* on the other.

For example, when PE 1 advertises routes from its two VRF tables to PE 2, they arrive over a common MP-BGP connection that is not inherently associated with a particular VRF table. How can we assure that PE 2 interprets these routes as being independent and unrelated? The answer lies in the structure of a VPN-IPv4 address containing a route distinguisher designed to fix the very problem posed here.

VPN-IPv4 Address Family



		Route Distinguisher			
Mask	MPLS Label	Type	Administrator	Assigned Number	Subscriber IPv4 Prefix
(1 byte)	(3 bytes)	(2 bytes)	(variable length)	(variable length)	(0–4 bytes)

- VPN-IPv4 address family
  - New BGP-4 subsequent address family identifier (SAFI 128)
    - Consists of MPLS label + route distinguisher + subscriber IPv4 prefix
  - Route distinguisher disambiguates IPv4 addresses
    - Allows service provider to administer its own *numbering* space
- VPN-IPv4 addresses are distributed by MP-BGP
  - Uses multiprotocol extensions for BGP4 (RFC 2858)
- A /32 IPv4 prefix produces a VPN-IPv4 prefix with a mask of /120 (15 octets)
  - The Junos OS CLI displays (and the examples in this class) only show IPv4 prefix length (that is, /32)

The graphic shows the structure of a VPN-IPv4 address. VPN addresses use a new MP-BGP subsequent address family identifier (SAFI). Because they are, in the end, IPv4 addresses, they use the same family identifier (1) as conventional IPv4 routes.

VPN network layer reachability information (NLRI) contains a 24-bit MPLS label, which is sometimes called a VRF label because the label's function is to associate packets with a particular VRF instance in the receiving PE router. VPN addresses also contain a route distinguisher field, which is used to disambiguate VPN routes. In other words, two identical IP prefixes are considered as different, and therefore incomparable, when they carry different route distinguisher values.

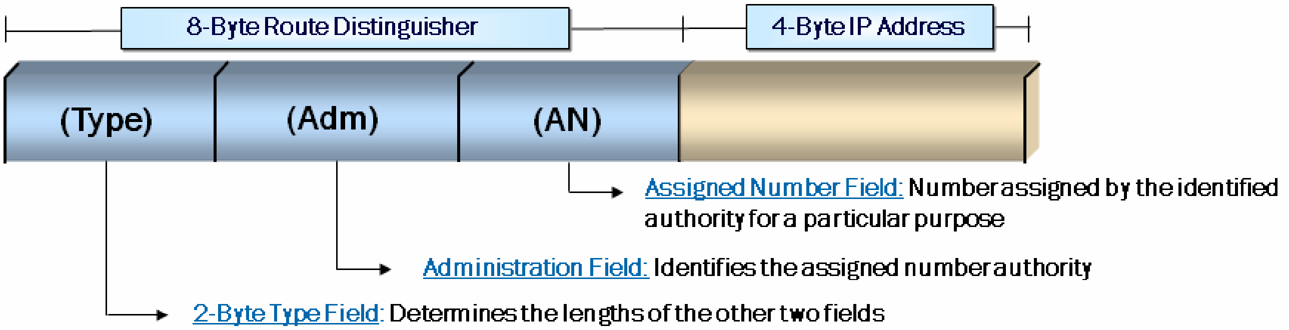
Distributed by MP-BGP

Labeled VPN routes are exchanged over the MP-BGP sessions, which terminate on the PE routers.

VPN Route Masks

A 32-bit IPv4 prefix combined with the other fields in a VPN-IPv4 address produce a VPN-IPv4 prefix with a mask of 120 bits. The Junos operating system only displays the mask for the IPv4 prefix portion of the prefix. Thus, in this case, the operation would see a VPN-IPv4 prefix with a mask length of /32.

Two Route Distinguisher Formats Defined



Assigned Number Field: Number assigned by the identified authority for a particular purpose

Administration Field: Identifies the assigned number authority

2-Byte Type Field: Determines the lengths of the other two fields

- Two values are defined for type field: 0 and 1
  - Type 0: adm field = 2 bytes, AN field = 4 bytes
    - Adm field should contain an ASN from IANA
    - AN field is a number assigned by service provider
  - Type 1: adm field = 4 bytes, AN field = 2 bytes
    - Administration field should contain an IP address assigned by IANA
    - Assigned number field is a number assigned by service provider
  - Examples: 10458:22:10.1.0.0/16 or 1.1.1.1:33:10.1.0.0/16

The route distinguisher can be formatted two ways:

- *Type 0*: This format uses a 2-byte administration field that codes the provider's autonomous system number, followed by a 4-byte assigned number field. The assigned number field is administered by the provider and should be unique across the autonomous system.
- *Type 1*: This format uses a 4-byte administration field that is normally coded with the router ID (RID) of the advertising PE router, followed by a 2-byte assigned number field that caries a unique value for each VRF table supported by the PE router.

The examples in the graphic show both the Type 0 and Type 1 route distinguisher formats. The first example shows the 2-byte administration field with the 4-byte assigned number field (Type 0).

Disambiguates IPv4 Addresses

As mentioned on the previous page, the route distinguisher allows the router to disambiguate two identical IP prefixes.

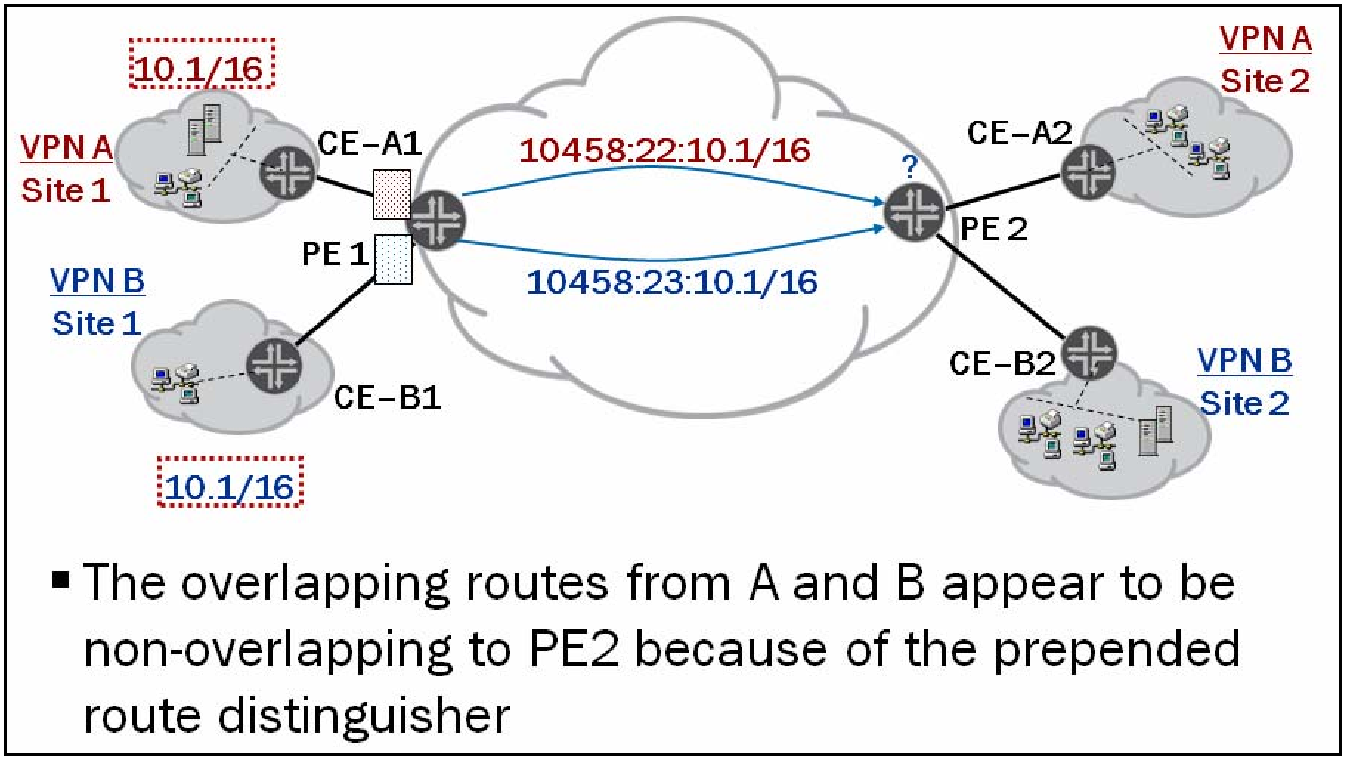
VPN-IPv4 Routes

The ingress PE router adds (or prepends) the route distinguisher to the IPv4 prefix of routes received from each CE router. These VPN-IPv4 routes are then exchanged between PE routers using MP-BGP. The egress router converts the VPN-IPv4 routes back into IPv4 routes before inserting them into the site's routing table.

Used Only in the Control Plane

The VPN address family exists only in the signaling or control plane between PE routers. Routes that match VPN policy, and are therefore installed into a particular VRF table, have the 8-byte route distinguisher (and MPLS label) removed so that they appear as conventional IPv4 routes in the VRF table. Because the site-specific VRF tables provide route isolation, there is no need for the route distinguisher once a route is safely stored away in a VRF table. Only signaling exchanges between PE routers use the VPN-IPv4 address format.

Overlapping Routes Revisited

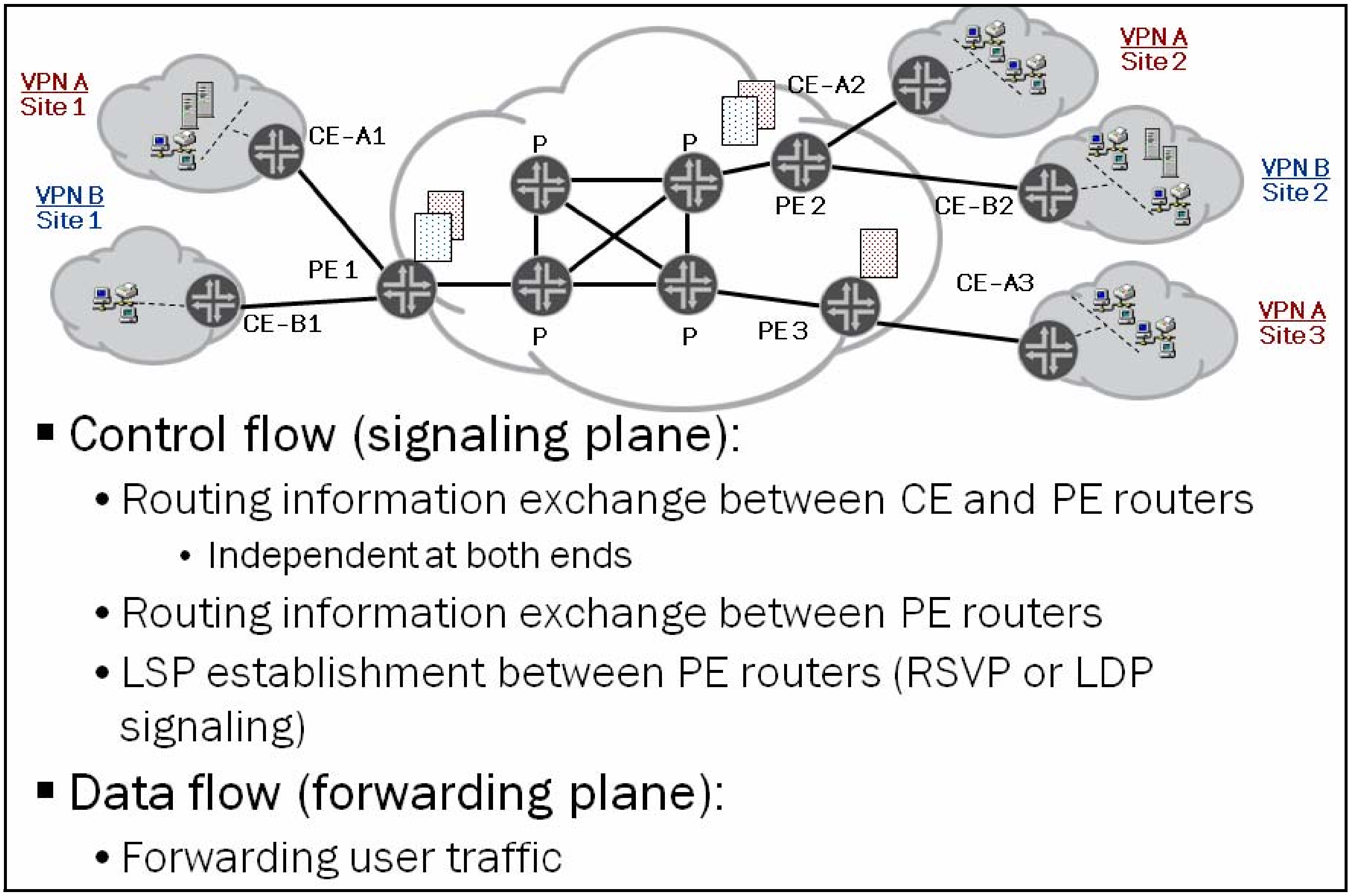


- The overlapping routes from A and B appear to be non-overlapping to PE2 because of the prepended route distinguisher

With the inclusion of the route distinguisher, the overlapping address spaces used by VPN customers A and B do not cause ambiguity at PE 2 because the different route distinguishers make these routes incomparable.

The sole purpose of the route distinguisher is to make what would otherwise be identical addresses incomparable. The PE routers do not interpret or act on the fields in the route distinguisher for any other reason.

Control Flow



- Control flow (signaling plane):
  - Routing information exchange between CE and PE routers
    - Independent at both ends
  - Routing information exchange between PE routers
  - LSP establishment between PE routers (RSVP or LDP signaling)
- Data flow (forwarding plane):
  - Forwarding user traffic

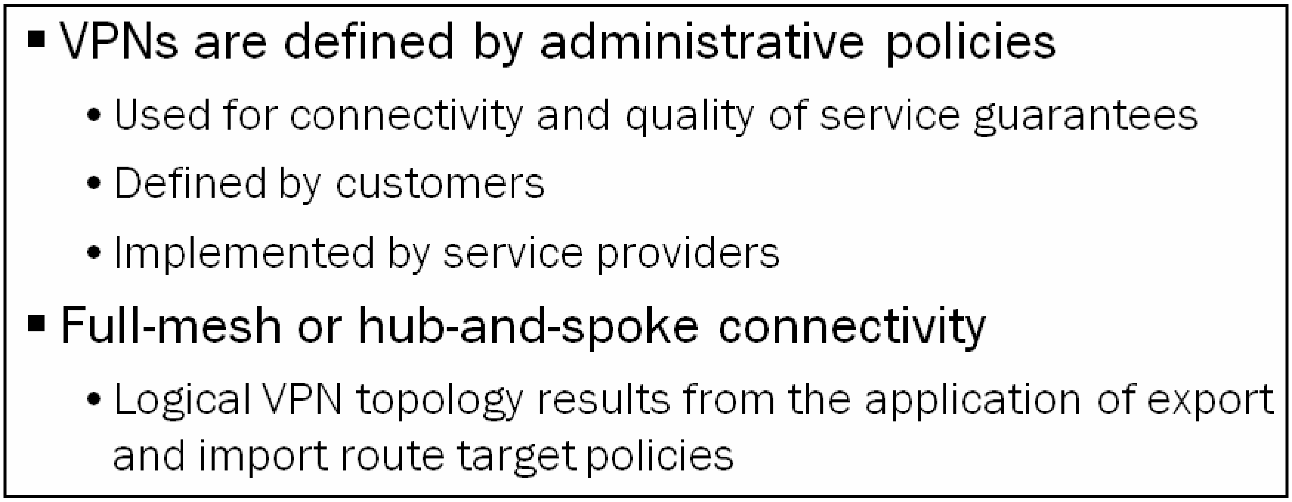
VPN control flows exist at various places in the RFC 4364 environment. First, we have the signaling exchange between CE and PE routers that can take the form of OSPF, RIP, BGP, or even static routing. The control exchanges between PE and CE routers are totally independent, due to the PE routers terminating the local CE-PE signaling flows. The PE routers then use MP-BGP to convey routes from site-specific VRF tables for the purposes of populating the VRF tables on remote PE routers.

Finally, the need for LSPs in the provider's networks results in the presence of MPLS-related signaling in the form of either RSVP or LDP.

Data Flow

Data flow relates to the actual forwarding of VPN traffic from CE router to CE router using MPLS label-based switching through the provider's core.

Administrative Policy



- VPNs are defined by administrative policies
  - Used for connectivity and quality of service guarantees
  - Defined by customers
  - Implemented by service providers
- Full-mesh or hub-and-spoke connectivity
  - Logical VPN topology results from the application of export and import route target policies

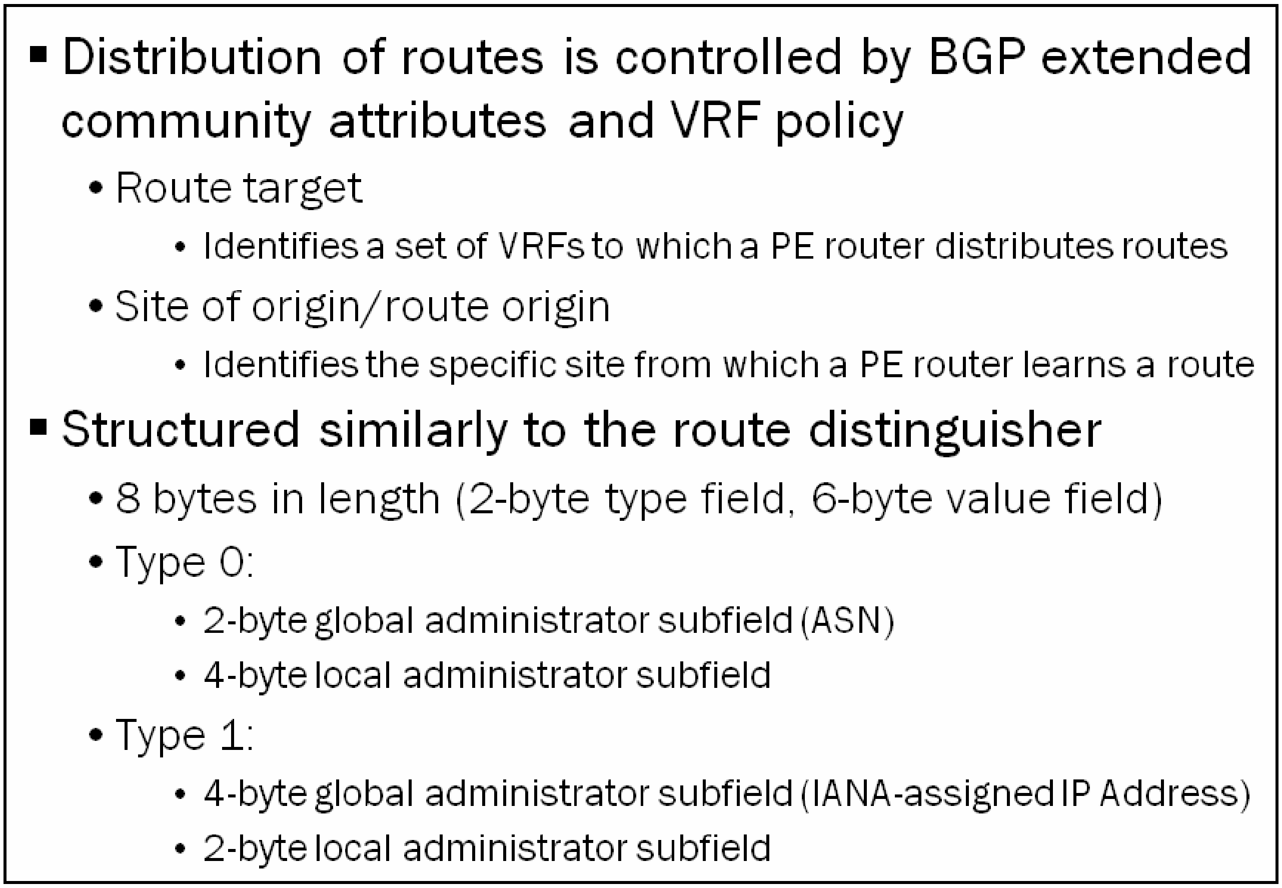
The use of policy in the PE routers determines the connectivity that results between VPN sites. While site connectivity requirements are defined by the VPN customers, the act of implementing this policy is the job of the service provider.

Mistakes made by the provider when defining and implementing VPN policy can lead to security breaches at worst and broken VPN connectivity at best.

VPN Topology Options

VPN policy is extremely flexible and can result in full-mesh, partial-mesh, or hub-and-spoke topologies. The combination of VPN import and export policy determines the resulting site connectivity.

Route Distribution Between PE Routers



- Distribution of routes is controlled by BGP extended community attributes and VRF policy
  - Route target
    - Identifies a set of VRFs to which a PE router distributes routes
  - Site of origin/route origin
    - Identifies the specific site from which a PE router learns a route
- Structured similarly to the route distinguisher
  - 8 bytes in length (2-byte type field, 6-byte value field)
  - Type 0:
    - 2-byte global administrator subfield (ASN)
    - 4-byte local administrator subfield
  - Type 1:
    - 4-byte global administrator subfield (IANA-assigned IP Address)
    - 2-byte local administrator subfield

VPN policy makes use of extended BGP communities that allow PE routers to filter routes for which they have no VPN members. When a PE router has locally attached VPN members, these communities allow the PE router to install the VPN route into the VRF table associated with specific sites.

The most important extended community is the route target, which is used to convey a route's association with a given VPN/VRF table. The site of origin (SoO) community is used in certain corner cases to prevent the unnecessary advertisement of routes back to a site that originated it.

Structure of Extended Communities

BGP extended communities are defined in RFC 4360. Extended communities' attributes have a structure similar to the route distinguisher in that they are 8 bytes in length and support the same type code options and structure.

Route Advertisements

Each VPN-IPv4 route advertised by a PE router contains one or more route target communities. These communities are added using VRF export policy or explicit configuration.

Receiving Routes

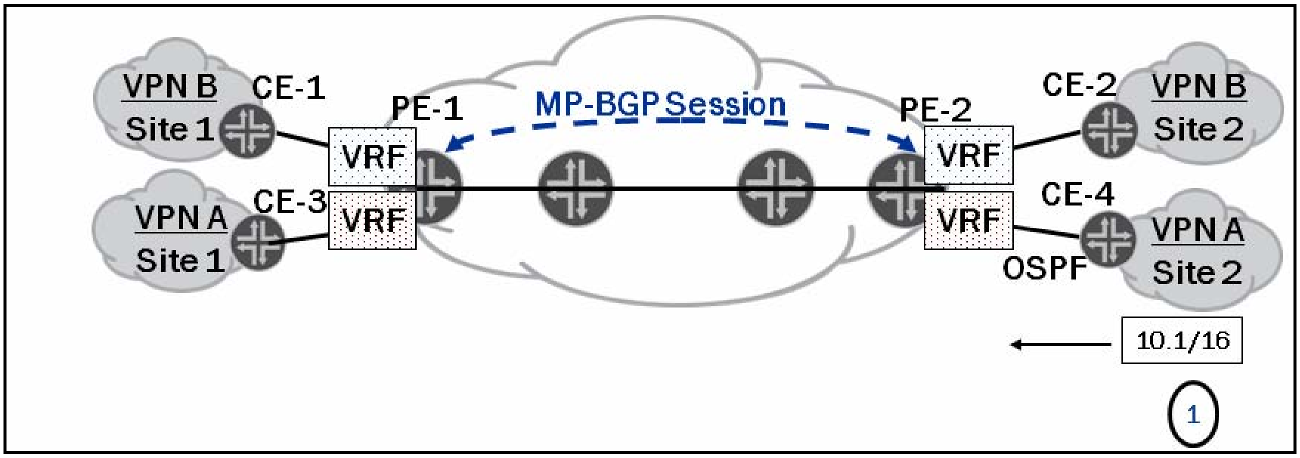
When a PE router receives route advertisements from remote PE routers, it determines whether the associated route target matches one of its local VRF tables. Matching route targets cause the PE router to install the route into all VRF tables whose configuration matches the route target.

Careful Policy Administration

Because the application of policy determines a VPN's connectivity, you must take extra care when writing and applying VPN policy to ensure that the VPN customer's connectivity requirements are faithfully met. Several companies offer automated VPN provisioning tools to minimize the work required when reprovisioning a VPN to meet changing customer requirements. These tools can also limit the errors that tend to occur when changes are manually entered by human operators.

Go to http://www.juniper.net/partners/oss_partners.html to obtain updated information on the alliances that Juniper Networks has formed with the providers of such provisioning tools.

Routing Exchange

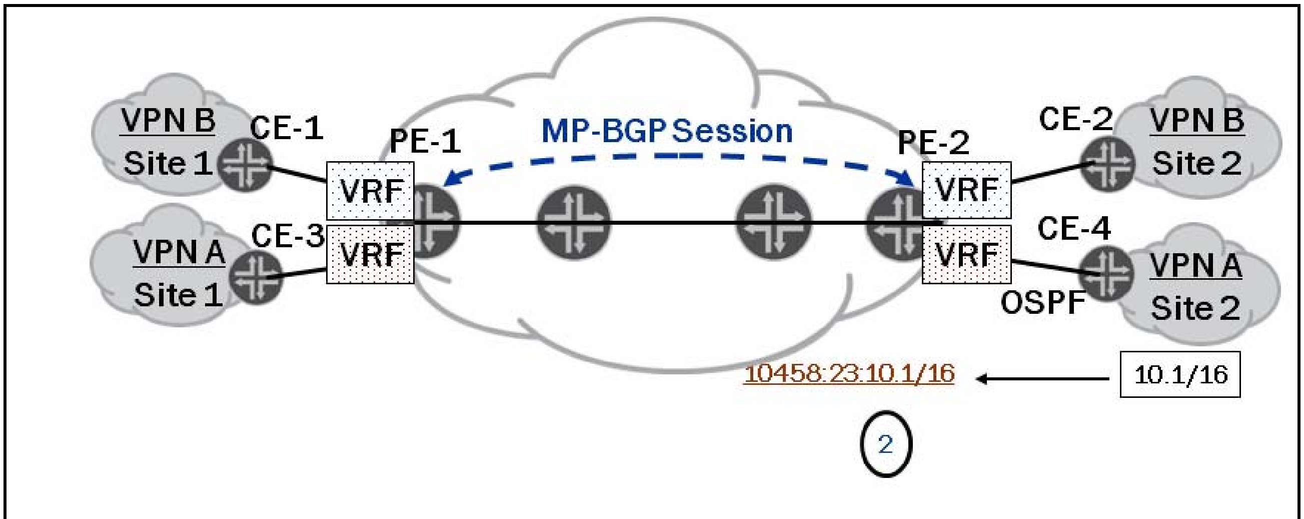


The following sequence of graphics discusses the end-to-end exchange of routing information between CE routers belonging to the same VPN.

CE-4 sends the routes associated with VPN A Site 2 to its attached PE router. The 10.1/16 prefix can be exchanged using OSPF, RIP, or BGP. Static routing can also place a site's routes into the local PE router's VRF table.

Whatever protocol is used between CE-4 and PE-2, the operation of this protocol is terminated by the PE router. This termination provides isolation of the VPN site's routing protocol and the MP-BGP protocols used to convey the routes between PE routers. This isolation improves scalability and stability as malfunctions in the PE-CE routing protocol tend to be limited to that PE-CE pairing.

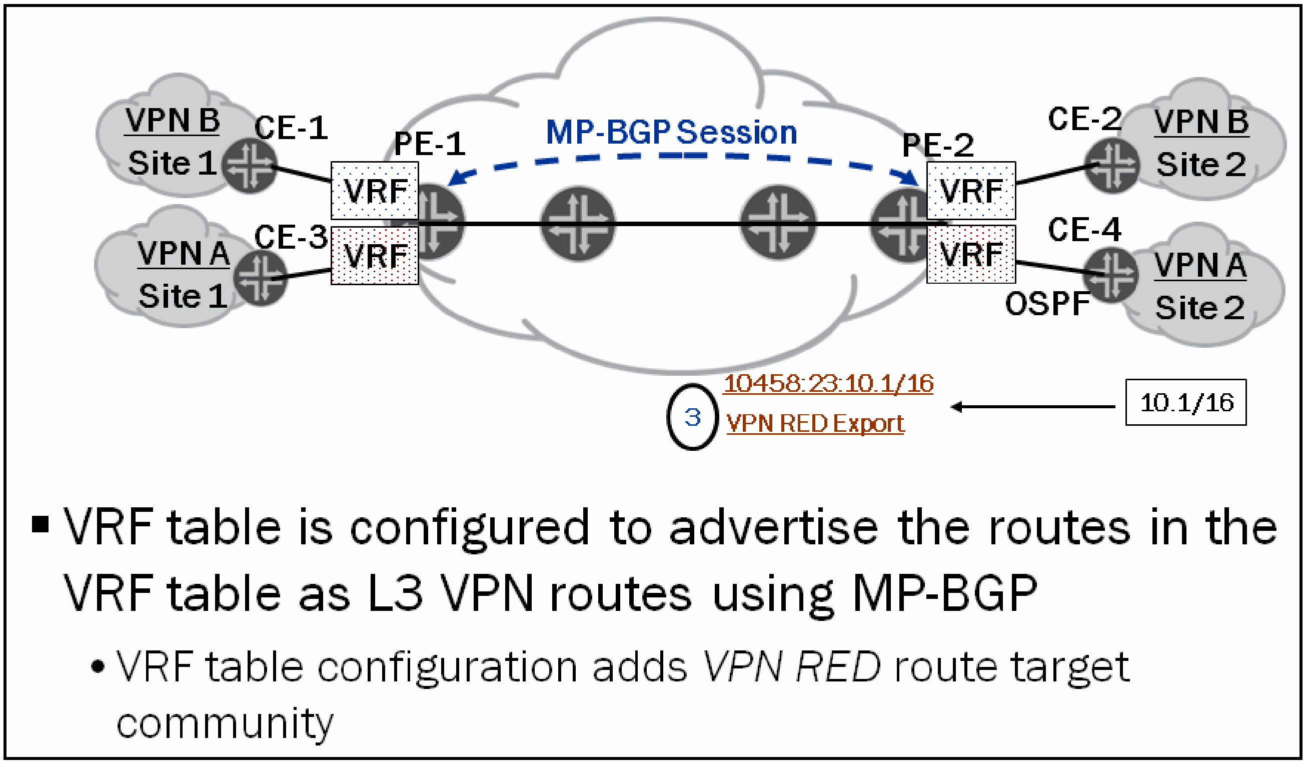
Populating the Local VRF Table



- IPv4 address is added to the appropriate VRF table

Routes received by a local CE device are automatically installed into the VRF table associated with that site.

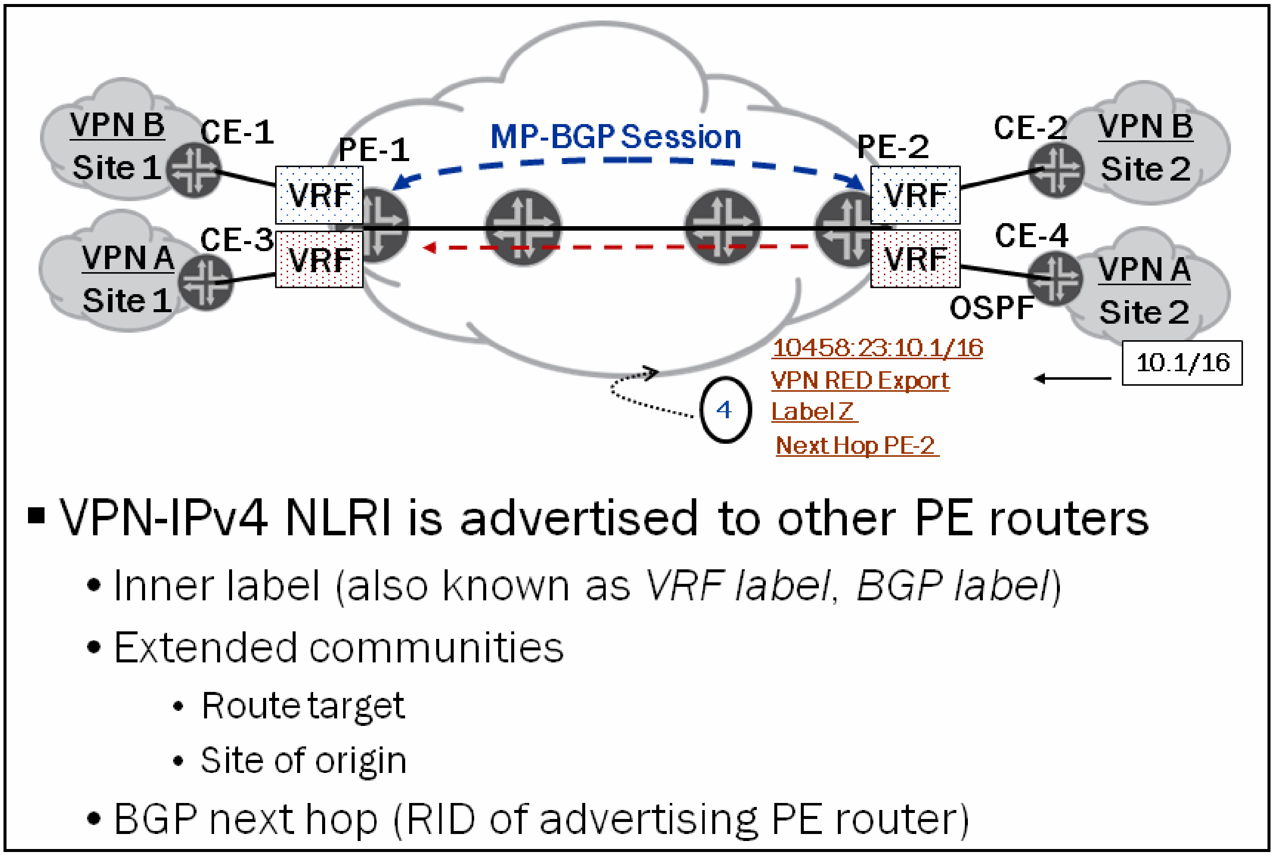
VRF Export Policy



- VRF table is configured to advertise the routes in the VRF table as L3 VPN routes using MP-BGP
  - VRF table configuration adds *VPN RED* route target community

The PE router evaluates the route based on its configuration. If the VRF export policy accepts the route, or if a VRF target is configured, the PE router converts the address into the VPN-IPv4 format by adding the configured route distinguisher. At this time the PE router also chooses a 20-bit MPLS label value used to associate received traffic with this VRF table. Lastly, the PE router associates the route with one or more extended communities. At a minimum, the route will have a route target community added.

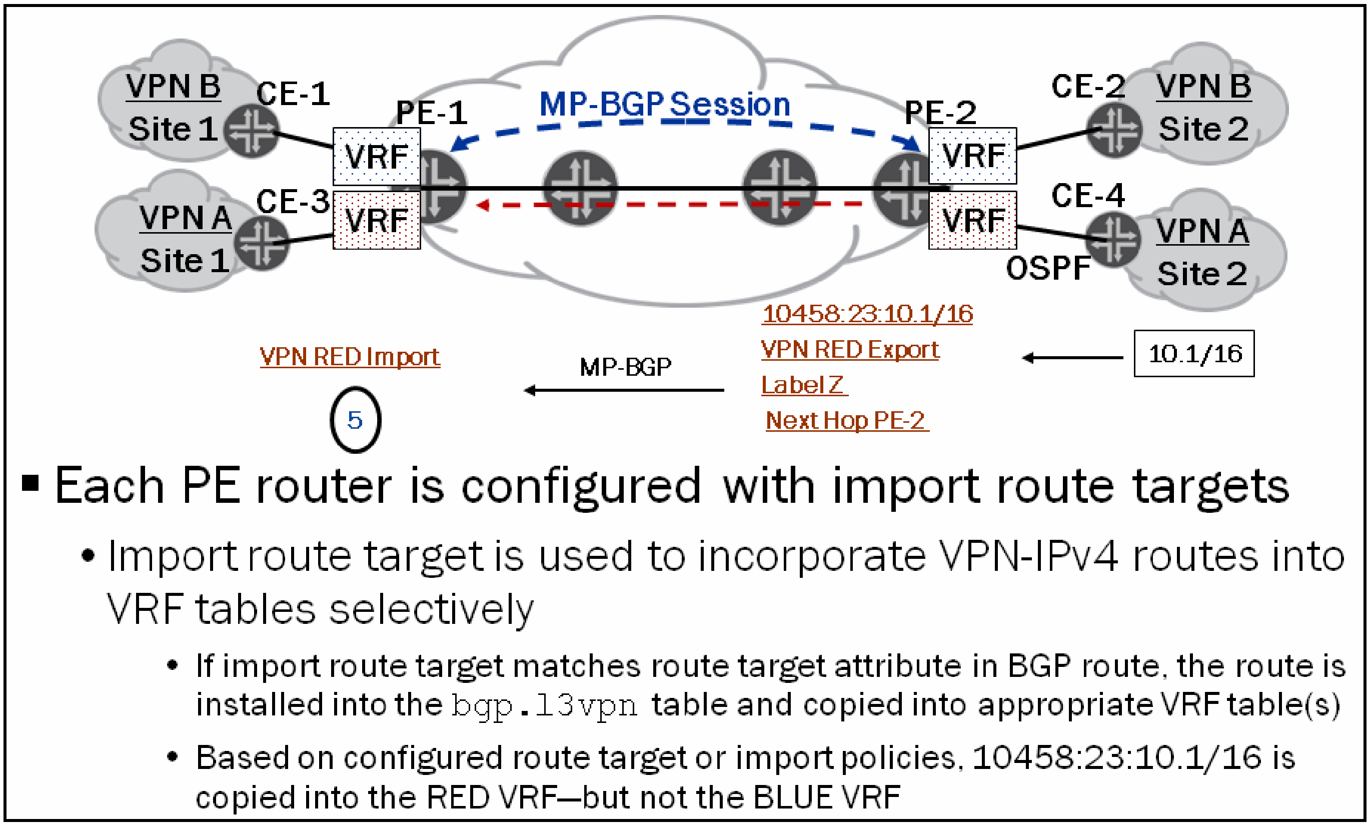
Advertisement to Remote PE Routers



- VPN-IPv4 NLRI is advertised to other PE routers
  - Inner label (also known as *VRF label*, *BGP label*)
  - Extended communities
    - Route target
    - Site of origin
  - BGP next hop (RID of advertising PE router)

In Step 4, PE-2 generates a BGP update message containing the route learned from CE4 at VPN Site A. This route is sent to all MP-BGP peers configured on the PE router that have successfully negotiated the support of the VPN-IPv4 address family. Other routes learned from the CE device that share common community attributes can be packed into a single NLRI advertisement.

Import Targets Determine the Route's Fate



- Each PE router is configured with import route targets
  - Import route target is used to incorporate VPN-IPv4 routes into VRF tables selectively
    - If import route target matches route target attribute in BGP route, the route is installed into the `bgp.l3vpn` table and copied into appropriate VRF table(s)
    - Based on configured route target or import policies, 10458:23:10.1/16 is copied into the RED VRF—but not the BLUE VRF

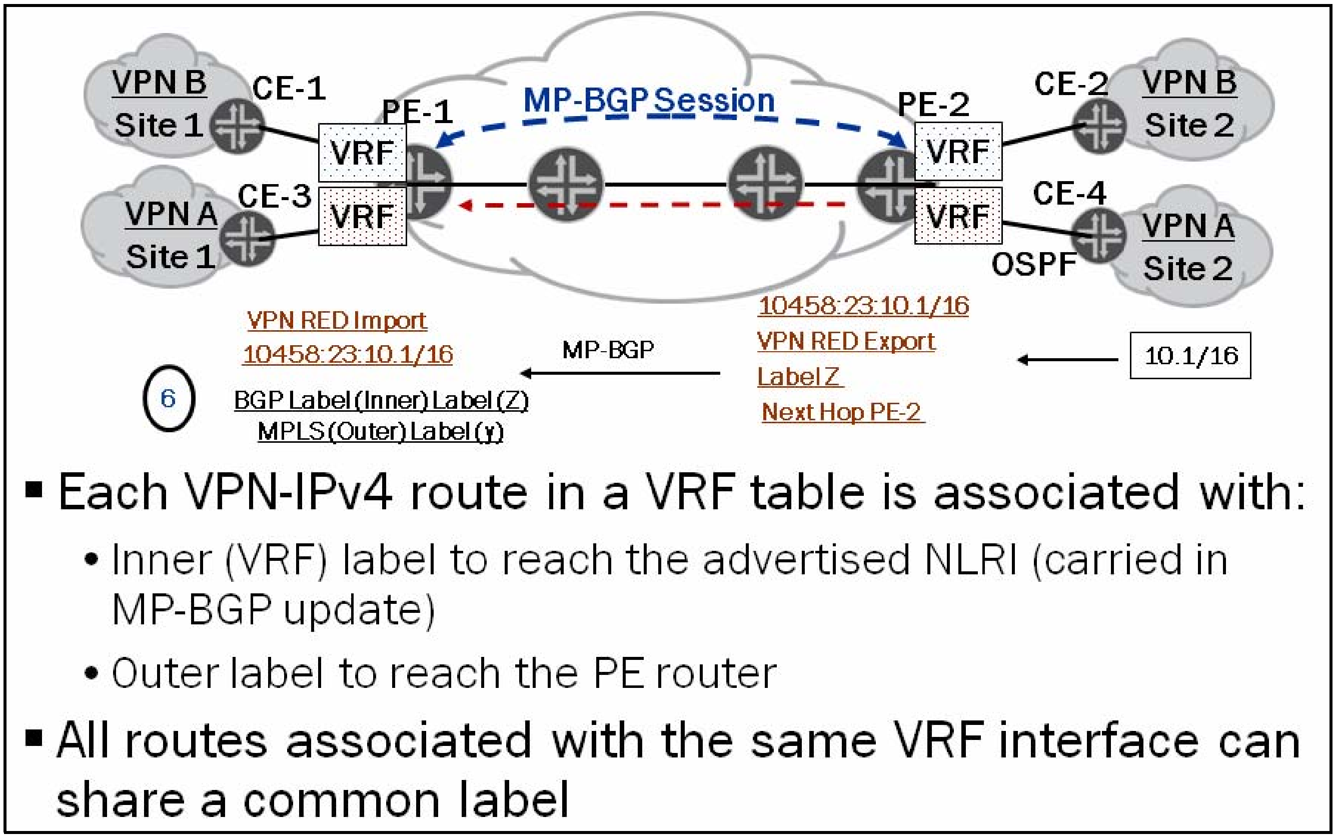
Step 5 shows the remote PE routers receiving the VPN route advertisement. These PE routers use their configured VRF import policy or VRF target to determine if any of their local VRF tables have matching route targets.

If no local configuration matches the route target, the PE router silently discards the route. Thus, a PE router must only carry VPN routes when it has one or more locally attached sites belonging to the same VPN. Should the remote PE router's import policy or VRF target change, BGP route refresh is used to solicit a retransmission of previously advertised routes, because route target matches can now occur due to the policy modifications. Use of BGP route refresh means that BGP sessions do not have to be disrupted when adds, moves, or changes to the VPN topology occur.

When the received route's target does match a VRF table's route target configuration, the PE router copies the VPN route into the `bgp.l3vpn.0` table. This table houses all received VPN routes whose route target matched at least one VPN's configuration. The route is also copied into one or more local VRF tables after having the route distinguisher removed. The result is that prefix 10.1/16 is now present in PE-1's random early detection (RED) VRF table in a native IPv4 format.

PE-1 now associates the RID of PE-2 as the next hop for 10.1/16 when forwarding traffic that matches the prefix and was received on its RED VRF interface.

Label Association



- Each VPN-IPv4 route in a VRF table is associated with:
  - Inner (VRF) label to reach the advertised NLRI (carried in MP-BGP update)
  - Outer label to reach the PE router
- All routes associated with the same VRF interface can share a common label

When VPN routes are advertised, part of the NLRI is the VRF label chosen by the advertising PE router. This label is often called the *inner* label because it is always found at the bottom of the label stack. The purpose of this label is to associate received packets with the correct VRF table.

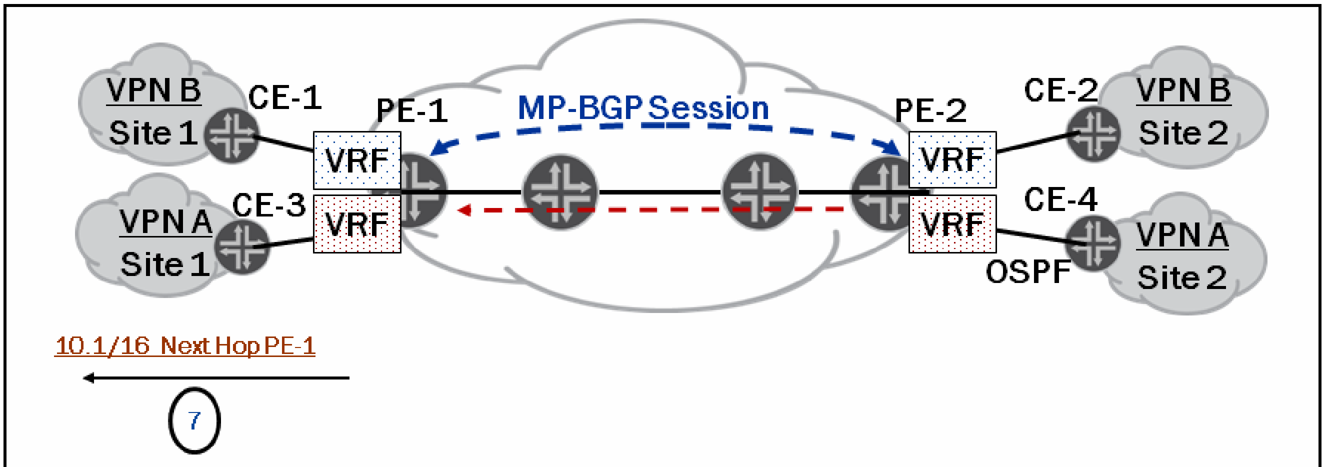
The receiving PE router must be able to resolve the RID of the advertising route to an MPLS LSP stored in the `inet.3` table. If an LSP does not exist to the advertising PE router, the route is hidden due to an unusable next hop. VPN traffic can only be forwarded across the provider's backbone using MPLS switching. If an LSP to the egress PE router does not exist, the VPN route can never be used.

The result of this process is a two-level label stack used to forward packets across the provider's backbone, and then to associate the traffic with a specific VRF table on the receiving PE router.

Common Labels

RFC 4364 allows the PE router to issue a single VRF label for all routes belonging to a common VRF interface or to allocate a unique label for each route being advertised. The Junos OS takes the former approach because it drastically reduces the number of VRF labels that must be managed. Compliant implementations that use per-route VRF label assignment are interoperable with this one-label-per-VRF-interface approach, however.

Advertising Received Routes



- Each IPv4 route installed in a VRF table can be advertised to the CE devices associated with that VRF table
  - For example, RIP, OSPF, and BGP
  - Routing policy can be used on the PE-CE link to control the exchange of routing information further

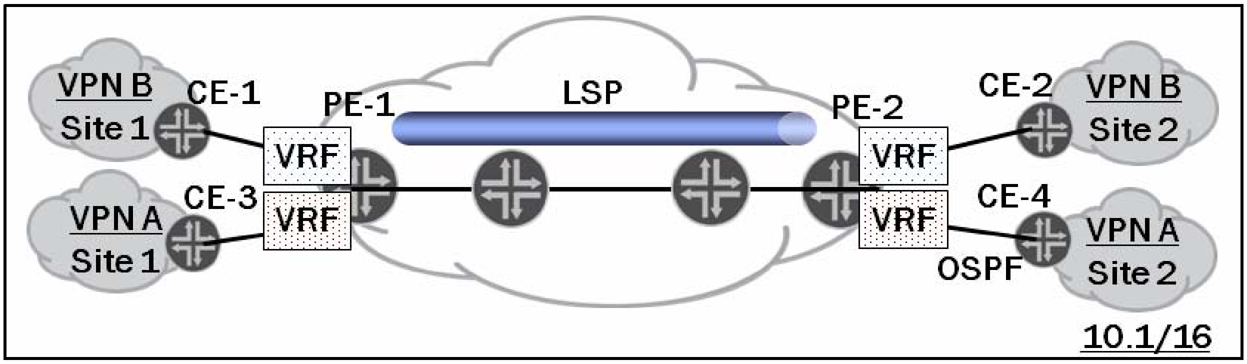
In the last step of the Layer 3 VPN signaling flow, the receiving PE router (PE-1) readvertises the routes learned from remote PE routers to its locally attached CE routers.

These routes can be exchanged using any supported PE-CE routing protocol, or they can be defined statically on the CE device. The CE device associates the PE router’s VRF interface as the next hop for the routes learned from the PE router.

Because the local PE-CE routing protocol is terminated by the PE router’s VRF table, in this example, CE-4 can run EBGP, while CE-3 might be running OSPF or RIP.

Where wanted, you can use routing policy to control or refine the route exchange between PE and CE routers further. This policy would function in addition to the VRF import and export policy discussed in this section.

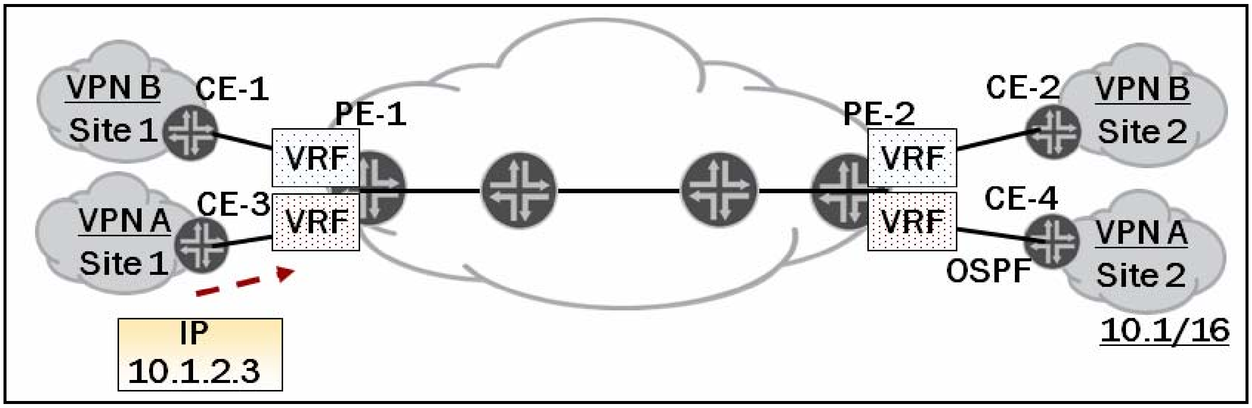
LSP Must Exist Between Ingress and Egress PE Routers



Because VPN traffic is forwarded across the provider’s backbone using MPLS, the presence of an MPLS LSP between ingress and egress PE routers must be in place before VPN packets can be forwarded.

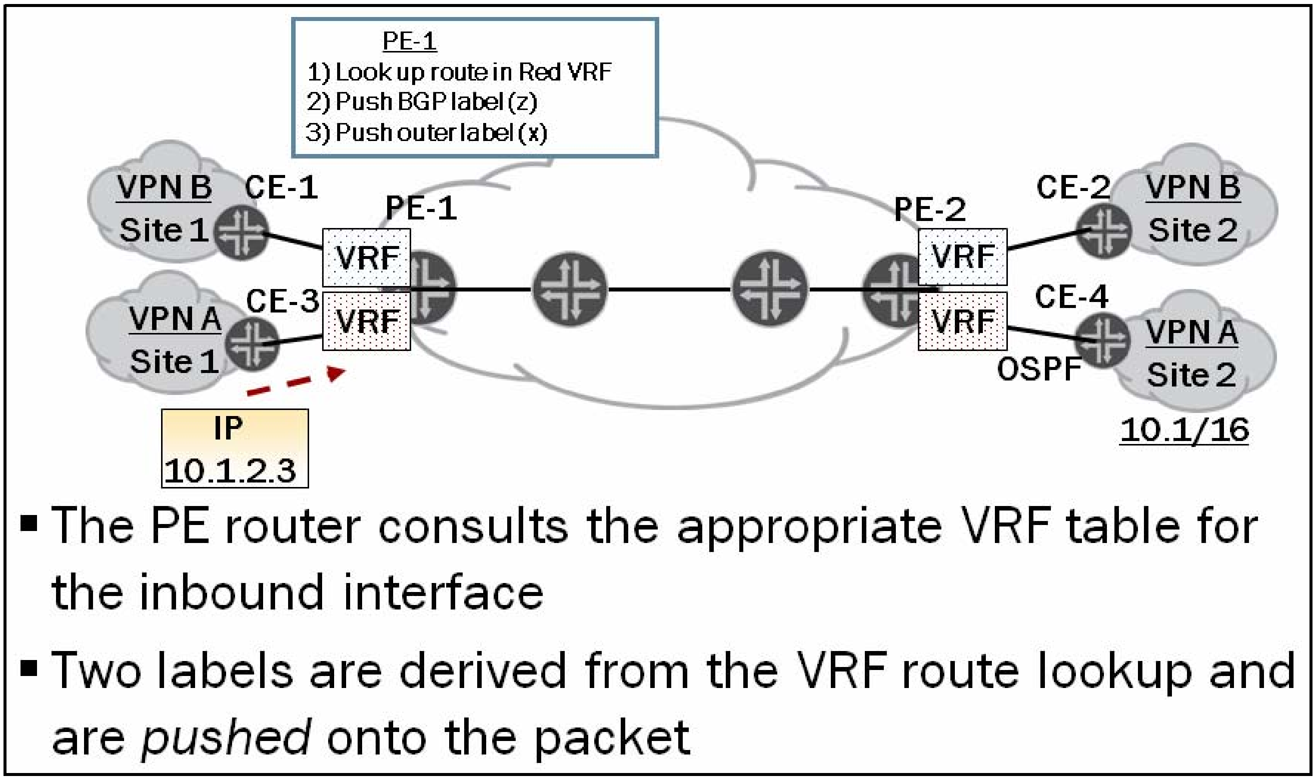
RSVP or LDP can establish the PE-to-PE LSP. The PE-to-PE LSPs can involve PE routers running LDP with the resulting LDP LSPs tunneled over a traffic engineered RSVP LSP.

CE Device Forwards VPN Traffic to PE Router



On this graphic, the CE device performs a longest-match lookup on a packet addressed to 10.1/16. This lookup results in the CE device forwarding the packet to the IP address associated with the PE router's VRF interface.

PE Router Consults VRF Table for Longest Match



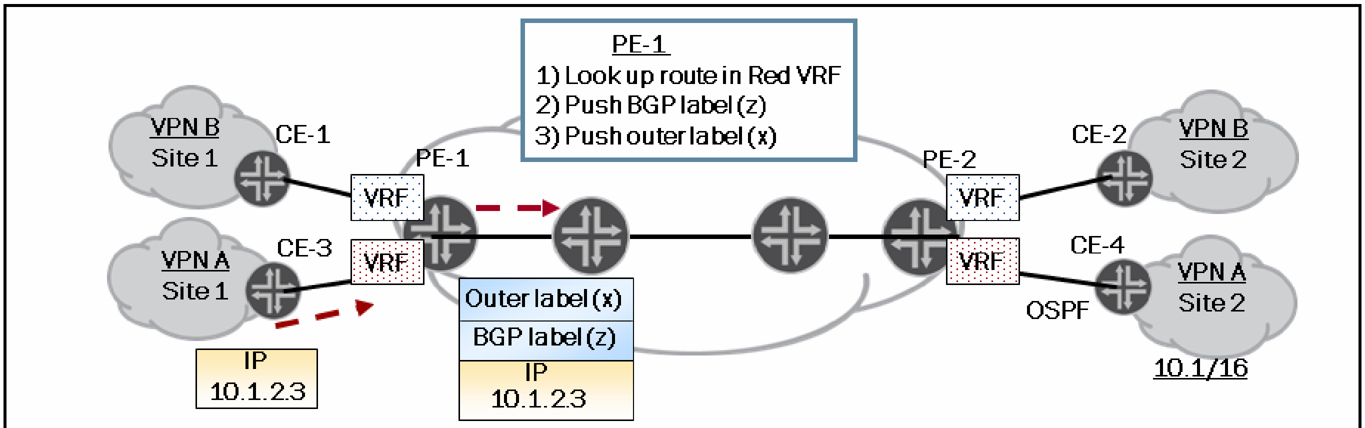
- The PE router consults the appropriate VRF table for the inbound interface
- Two labels are derived from the VRF route lookup and are *pushed* onto the packet

Upon receipt of the packet, the PE router conducts a longest-match route lookup in the VRF table associated with the interface on which the packet arrived.

Two Labels Derived

Assuming that a match is found, the PE router associates the packet with two labels: the VRF label originally advertised with the route, and an outer MPLS label, assigned by either LDP or RSVP, which is used to associate the packet with the LSP between ingress and egress PE routers.

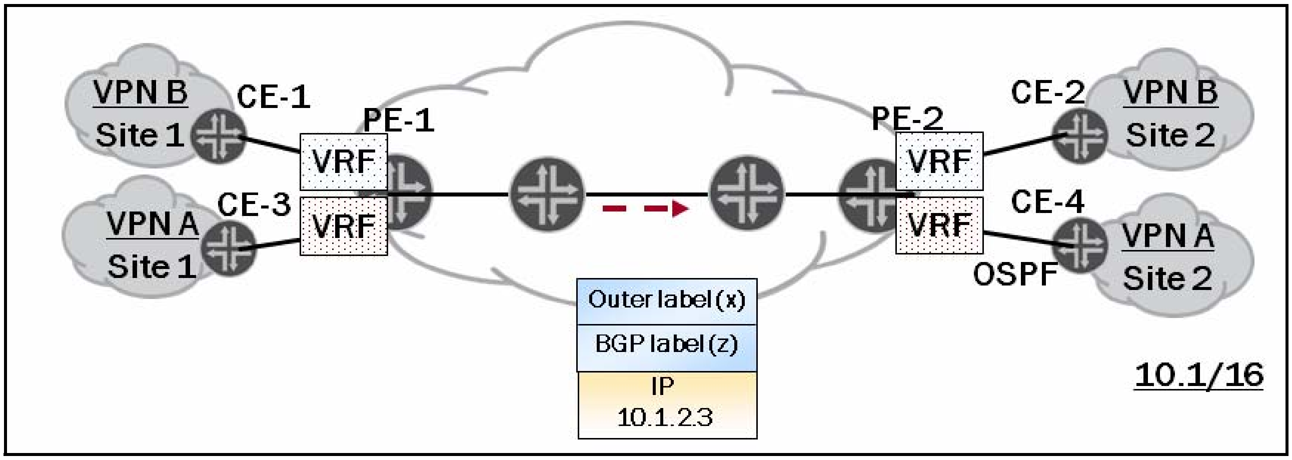
Two-Level Label Stack Required



- Packets are forwarded using two-level label stack
  - Outer (MPLS) label:
    - Identifies the LSP to egress PE router
    - Resolves BGP next hop through `inet.3`
    - Distributed by RSVP or LDP
  - Inner (MP-BGP) label:
    - Identifies outgoing interface from egress PE to CE
    - Communicated in MP-BGP updates (control plane)

The PE router performs a double label push operation involving both the VRF and MPLS labels. The VRF label associates the packet with the correct VRF table on the egress PE router while the MPLS label associates the packet with the LSP that terminates on the egress PE router. The ingress PE router now forwards the labeled packet to the next-hop LSR along the LSP's path.

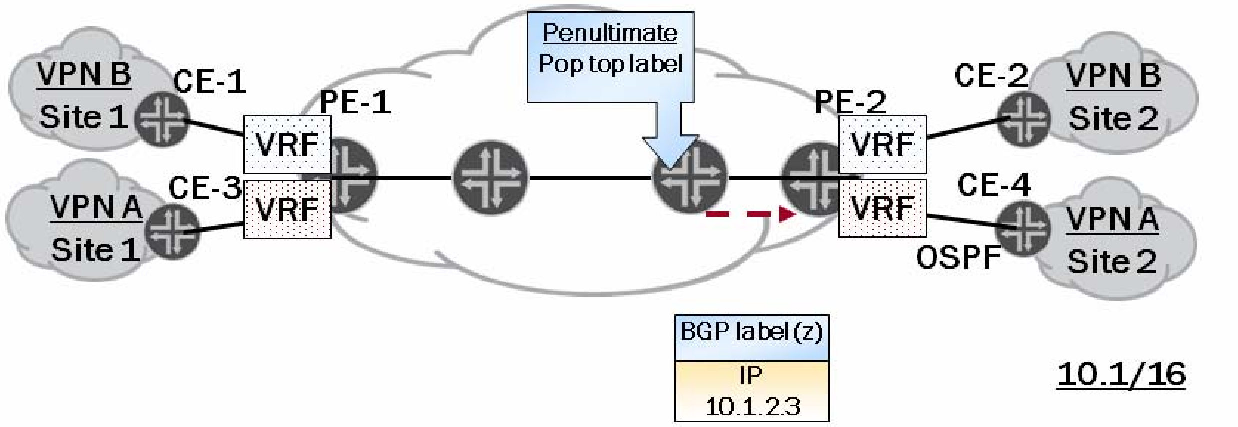
MPLS Forwarding Across Provider Core



As the labeled packet traverses the provider's core, the LSRs that make up the LSP act upon (and swap) the outer MPLS label. In contrast, the inner VRF label remains untouched throughout the labeled packet's journey.

The use of exact match MPLS forwarding allows the P routers to forward the packet towards the egress PE router correctly, without any awareness of the labeled packet's contents. This concept is key to RFC 4364 scalability, because this MPLS capability is what allows P routers to remain unaware of the VPN.

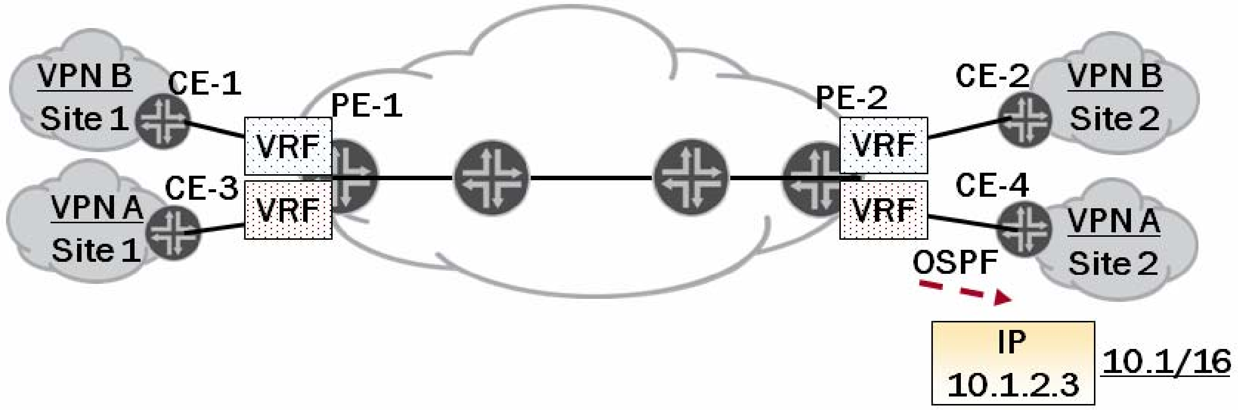
Penultimate Hop Popping



- Penultimate hop popping (before reaching the egress PE router) removes the outer label

The last P router in the LSP's path performs a *pop* operation, which results in a single-level label stack. The packet is now forwarded to the LSP's egress point with only the VRF label.

VRF Label Removed by Egress PE Router



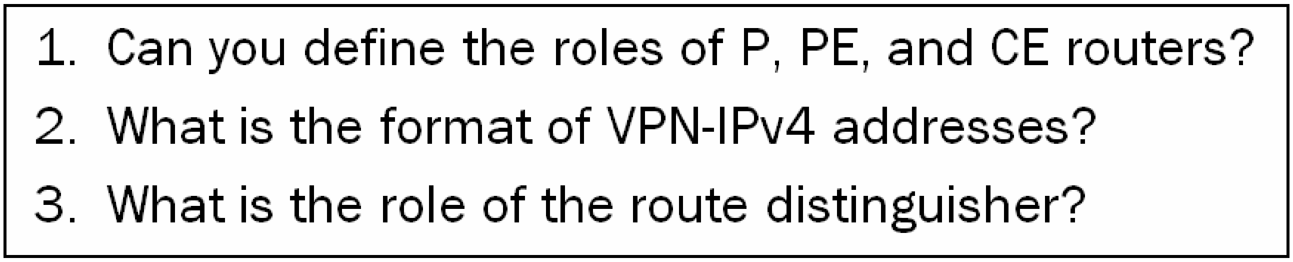
- The inner label is removed at the egress PE router
- The native IPv4 packet is sent to the outbound interface associated with the label

The egress PE router uses the received VRF label to map the packet to a specific VRF interface.

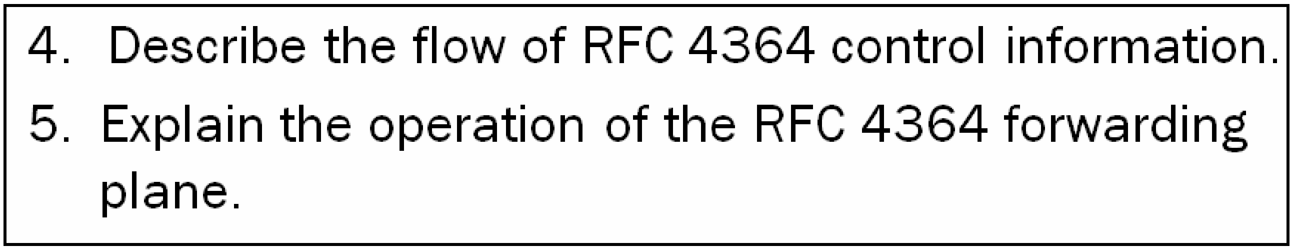
IPv4 Packet Sent to Outbound Interface

After mapping the packet to a specific VRF interface, the VRF label is popped, and the packet is sent to the CE device attached to that VRF interface.

Review Questions



1. Can you define the roles of P, PE, and CE routers?
2. What is the format of VPN-IPv4 addresses?
3. What is the role of the route distinguisher?



4. Describe the flow of RFC 4364 control information.
5. Explain the operation of the RFC 4364 forwarding plane.

Answers to Review Questions

1.

The CE router is located at the customer's VPN site and only participates in the customer's routing. The PE router is located on the edge of the provider network and participates in both the customer's routing and the provider's network. The PE maintains all of the customer specific VRF tables. The P routers participate in the core network and is able to forward VPN traffic using MPLS LSPs without knowledge of the customer's network.

2.

The VPN-IPv4 NLRI consists of an MPLS label, a route distinguisher, an IPv4 address, and a 120 bit mask.

3.

The route distinguisher is used to disambiguate overlapping IPv4 addresses.

4.

Some routing method (OSPF, BGP, static routing) is used to share routes between the customer VPN sites and the PE routers. MP-BGP is used by PE routers to pass customer routes learned from CE routers to other PE routers. PE routers will then pass VPN routes learned from other PE routers to the associated CE routers.

5.

A CE router will forward IPv4 packets to the locally connected PE router. The PE router will perform an route lookup using the VRF table associated with the incoming interface. The PE router will then encapsulate the packets in 2 MPLS headers: the innermost will be the label learned from MP-BGP while the outermost will be the label associated with the LSP that ends at the remote PE. The P routers along the LSP will perform label swapping on the outermost header as the packet traverses the provider's network. The penultimate router along the LSP will pop the outermost label and send a singly labeled packet to the remote PE. The remote PE will analyze the packets label in order to map the packet to a particular routing table, VRF. The remote PE pops the MPLS label and forwards the IPv4 packet to the remote CE router.

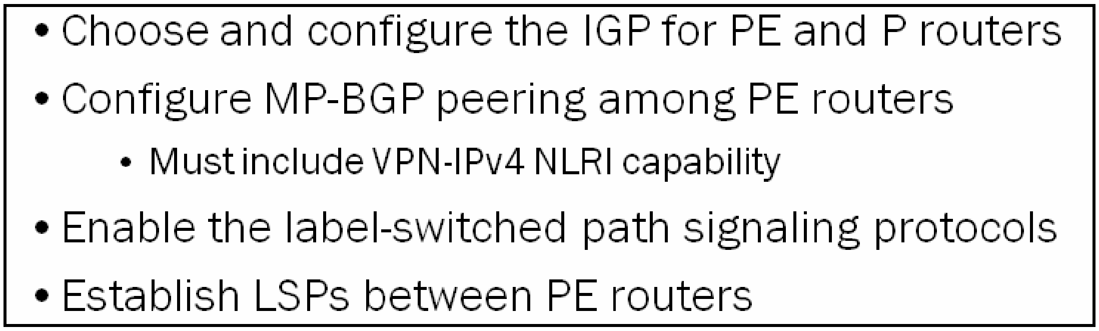
JNCIS-SP Study Guide—Part 3

Chapter 8: Basic Layer 3 VPN Configuration

This Chapter Discusses:

- Creating a routing instance, assigning interfaces, creating routes, and importing/exporting routes within the routing instance using route distinguishers and route targets;
- The purpose of BGP extended communities and how to configure and use these communities;
- The steps necessary for proper operation of a provider edge (PE) to customer edge (CE) dynamic routing protocol; and
- Configuring a simple Layer 3 virtual private network (VPN) using a dynamic CE-PE routing protocol.

Preliminary Steps



- Choose and configure the IGP for PE and P routers
- Configure MP-BGP peering among PE routers
  - Must include VPN-IPv4 NLRI capability
- Enable the label-switched path signaling protocols
- Establish LSPs between PE routers

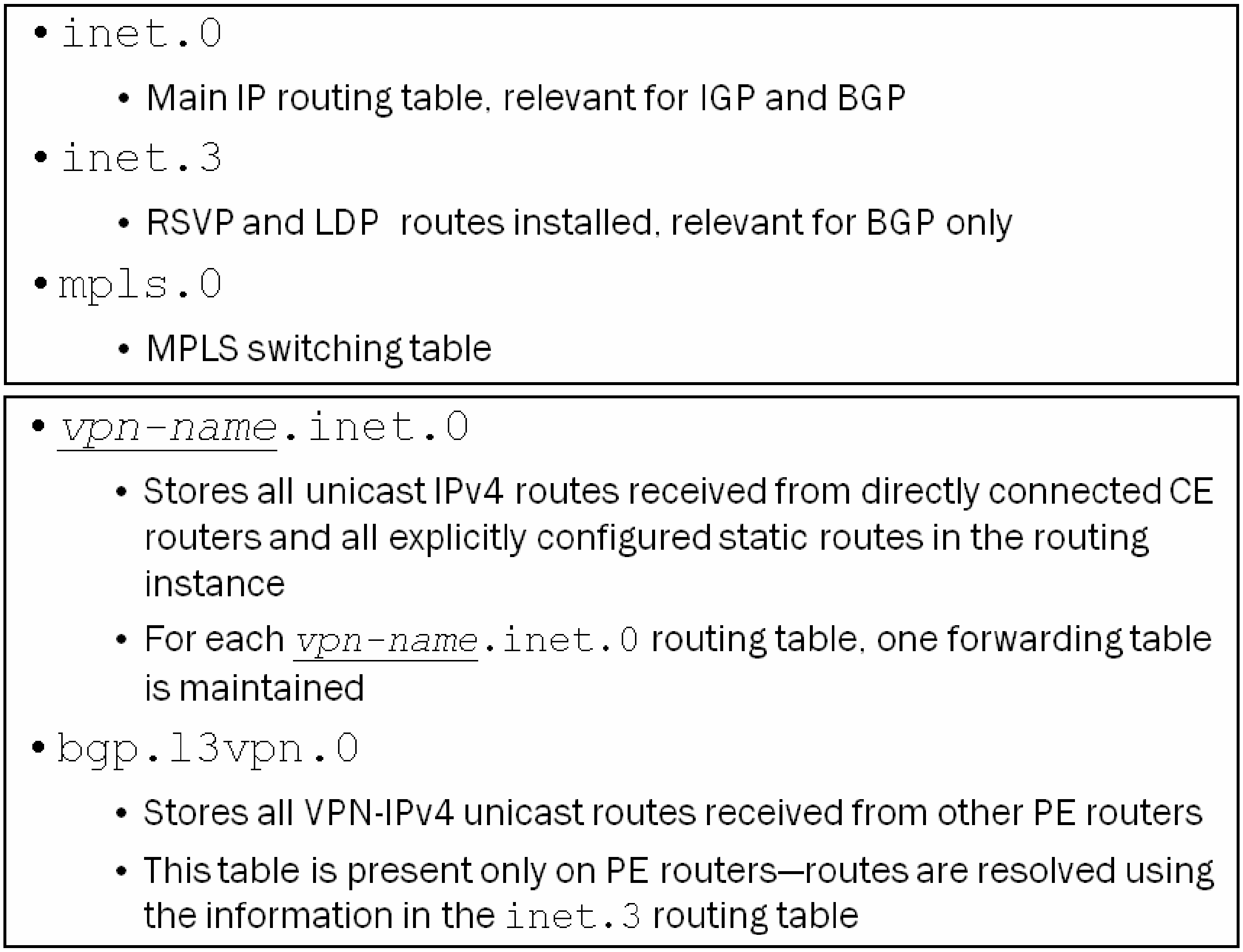
The following steps are needed to establish an IP infrastructure capable of supporting a Layer 3 VPN:

1. The provider core must have a functional interior gateway protocol (IGP) provisioned on the PE and provider (P) routers. Generally speaking, neither internal BGP (IBGP) nor MPLS signaling protocols function without a working IGP. If Constrained Shortest Path First (CSPF) label-switched paths (LSPs) are used, the IGP must support traffic engineering extensions.
2. Next, the Multiprotocol Border Gateway Protocol (MP-BGP) peering sessions should be established between the loopback addresses of the PE routers. PE routers not sharing VPN membership do not have to peer with MP-BGP. However, having the sessions in place can simplify operations later, should the PE routers find themselves attached to sites which are to form a VPN. Because these MP-BGP sessions are used to advertise VPN routes, the VPN-IPv4 address family must be configured and successfully negotiated.
3. You should now decide on an MPLS signaling protocol and provision it on all PE and P routers involved in VPN signaling or traffic forwarding. While it is possible to use a static LSP, the degree of manual labor and lack of operational status makes a static LSP difficult in large-scale networks.
4. Once you have completed the previous steps, configure LSPs between all PE routers that are expected to support a given VPN. The use of RSVP requires that you manually configure each LSP at the ingress node. In contrast, just enabling LDP results in LSP connectivity among all routers.

VPN Configuration in PE Routers Only

When your PE-PE MP-BGP sessions and LSPs are correctly established and operational, you are ready to begin the task of configuring a Layer 3 VPN. The good news is that a subset of your routers, namely the PE routers, perform all VPN-specific configuration.

Routing Tables Used for VPNs



- `inet.0`
  - Main IP routing table, relevant for IGP and BGP
- `inet.3`
  - RSVP and LDP routes installed, relevant for BGP only
- `mpls.0`
  - MPLS switching table

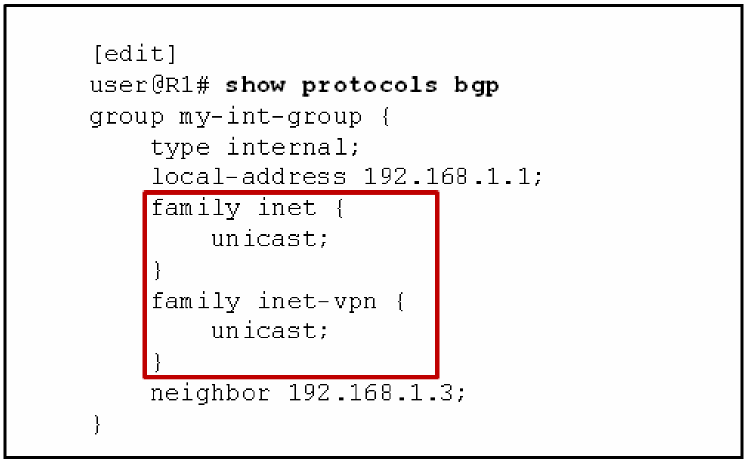
- *`vpn-name`*`.inet.0`
  - Stores all unicast IPv4 routes received from directly connected CE routers and all explicitly configured static routes in the routing instance
  - For each *`vpn-name`*`.inet.0` routing table, one forwarding table is maintained
- `bgp.l3vpn.0`
  - Stores all VPN-IPv4 unicast routes received from other PE routers
  - This table is present only on PE routers—routes are resolved using the information in the `inet.3` routing table

The following list provides information about the routing tables used for VPNs:

- `inet.0`: Stores routes learned by the IGP and IBGP sessions between the PE routers. To provide Internet access to the VPN sites, configure the `vpn.inet.0` routing table to contain a default route to the `inet.0` routing table.
- `inet.3`: Stores all MPLS routes learned from LDP and RSVP signaling done for VPN traffic. VPN IBGP (family `inet-vpn`) relies on next hops in the `inet.3` table.
- `mpls.0`: Stores MPLS switching information. This table contains a list of the next label-switching router (LSR) in each LSP. It is used on transit routers to route packets to the next router along an LSP.
- *`vpn-name`*`.inet.0`: Stores all unicast IPv4 routes received from directly connected CE routers in a routing instance and all explicitly configured static routes in the routing instance. This table is the VPN routing and forwarding (VRF) table and is present only on PE routers. For example, for a routing instance named *`vpn-a`*, the routing table for that instance is named *`vpn-a`*`.inet.0`. The *`vpn-name`*`.inet.0` table also stores routes announced by a remote PE router that match the import criteria for that VPN. This PE router tags the route with the route target that corresponds to the VPN site to which the CE belongs. A label is also distributed with the route. Routes are not redistributed from the *`vpn-name`*`.inet.0` table to the `bgp.l3vpn.0` table; they are directly advertised to other PE routers. For each routing instance, one forwarding table is maintained in addition to the forwarding tables that correspond to the router's `inet.0` and `mpls.0` routing tables.
- `bgp.l3vpn.0`: Stores all VPN-IPv4 unicast routes received from other PE routers. This table is present only on PE routers. When a PE router receives a route from another PE router, it places the route into its `bgp.l3vpn.0`

routing table after evaluating this route against the configured VRF parameters. The route is resolved using the information in the `inet.3` routing table. The resultant route is converted into IP version 4 (IPv4) format and redistributed to the *`vpn-name`*`.inet.0` tables on the PE router if it meets the configured criteria.

The Need for the `inet-vpn` Address Family



```
[edit]
user@R1# show protocols bgp
group my-int-group {
    type internal;
    local-address 192.168.1.1;
    family inet {
        unicast;
    }
    family inet-vpn {
        unicast;
    }
    neighbor 192.168.1.3;
}
```

Because the PE routers are expected to send and receive labeled VPN routes, you must configure the corresponding address family on the MP-BGP sessions between PE routers.

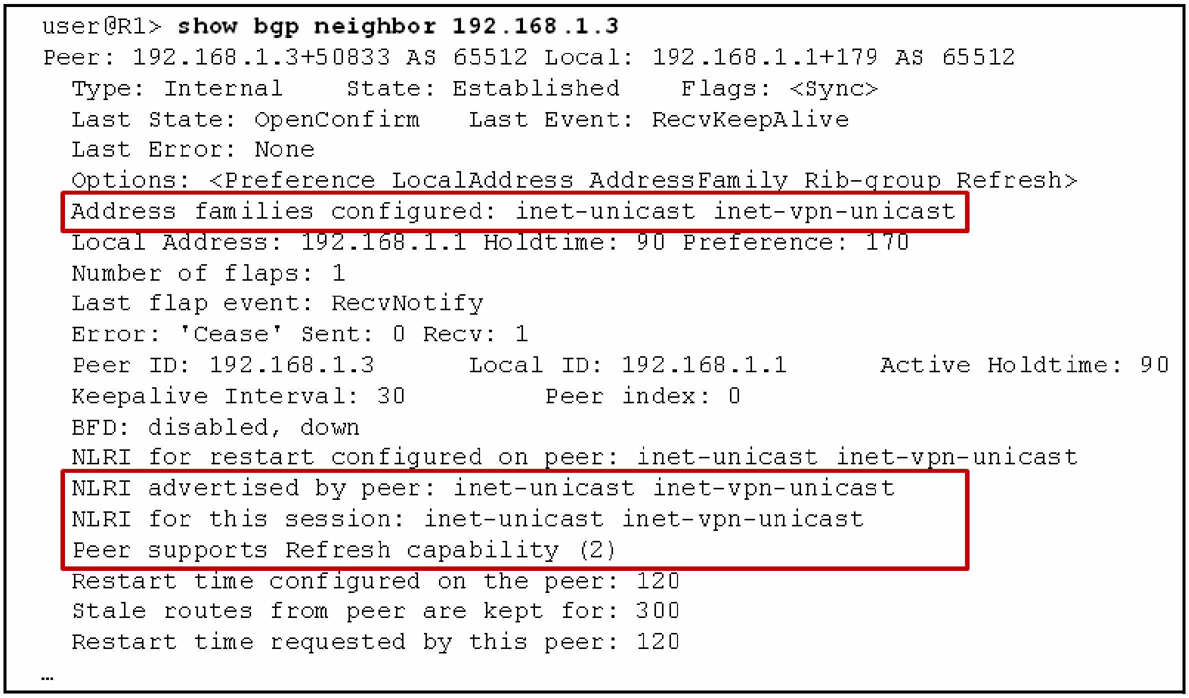
This graphic shows the configuration syntax used to support the VPN-IPv4 network layer reachability information (NLRI) for a BGP session.

Once an address family is explicitly configured on a BGP session, you must also explicitly configure the default address family of `inet unicast` if the PE router is expected to receive both conventional IP as well as VPN routes. Many providers try to keep full Internet routing feeds off their PE routers by using a default route in `inet.0` that points to a P router with a complete BGP table. In such a network, your PE-to-PE MP-BGP peering sessions might not need the `inet unicast` family.

BGP Route Refresh

The BGP route refresh capability is important when supporting VPNs as PE routers summarily discard all route advertisements that do not contain matching route targets. Without the route refresh capability, you would have to clear MP-BGP sessions when changes are made to VPN membership, which would result in disruption to all other VPNs that might share the MP-BGP session. The Junos operating system automatically negotiates the route refresh capability, so this feature does not require any explicit configuration.

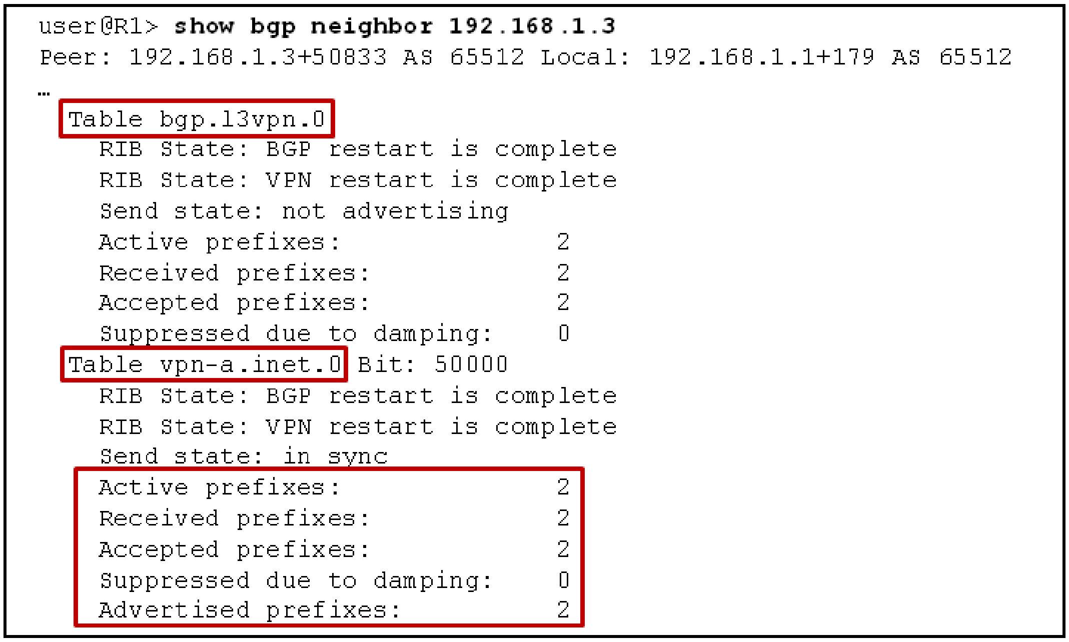
Verifying MP-BGP Peering Session: NLRI Information



```
user@R1> show bgp neighbor 192.168.1.3
Peer: 192.168.1.3+50833 AS 65512 Local: 192.168.1.1+179 AS 65512
  Type: Internal    State: Established    Flags: <Sync>
  Last State: OpenConfirm   Last Event: RecvKeepAlive
  Last Error: None
  Options: <Preference LocalAddress AddressFamily Rib-group Refresh>
  Address families configured: inet-unicast inet-vpn-unicast
  Local Address: 192.168.1.1 Holdtime: 90 Preference: 170
  Number of flaps: 1
  Last flap event: RecvNotify
  Error: 'Cease' Sent: 0 Recv: 1
  Peer ID: 192.168.1.3      Local ID: 192.168.1.1      Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast inet-vpn-unicast
  NLRI advertised by peer: inet-unicast inet-vpn-unicast
  NLRI for this session: inet-unicast inet-vpn-unicast
  Peer supports Refresh capability (2)
  Restart time configured on the peer: 120
  Stale routes from peer are kept for: 300
  Restart time requested by this peer: 120
…
```

This graphic shows the results of the **`show bgp neighbor`** command, where the BGP speakers have successfully negotiated both the VPN-IPv4 address family and the BGP route refresh capability.

Verifying MP-BGP Peering Session: Route Tables



```
user@R1> show bgp neighbor 192.168.1.3
Peer: 192.168.1.3+50833 AS 65512 Local: 192.168.1.1+179 AS 65512
…
  Table bgp.l3vpn.0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: not advertising
    Active prefixes:              2
    Received prefixes:            2
    Accepted prefixes:            2
    Suppressed due to damping:    0
  Table vpn-a.inet.0 Bit: 50000
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: in sync
    Active prefixes:              2
    Received prefixes:            2
    Accepted prefixes:            2
    Suppressed due to damping:    0
    Advertised prefixes:          2
```

A BGP speaker that has negotiated the VPN-IPv4 address family automatically creates the `bgp.l3vpn.0` route table used to store all routes received from other PE routers with at least one matching route target. If no routes have matched the PE router's VRF import policies, the `bgp.l3vpn.0` table is still created, but it remains empty.

Routes with matching route targets are also copied into one or more local VRF tables. In this example, all received routes with a matching route target were copied into the *`vpn-a`*`.inet.0` VRF table. The sum of all VRF table entries should match the total number of routes stored in the `bgp.l3vpn.0` table.

PE Router Configuration

Virtually all VPN-specific configuration and operational monitoring occurs on the PE routers.

PE Routing Instance

VRF tables are created as separate routing instances within the Junos OS. You must associate each instance with one or more logical interfaces. Configuration of the route distinguisher is another mandatory aspect of VRF instances.

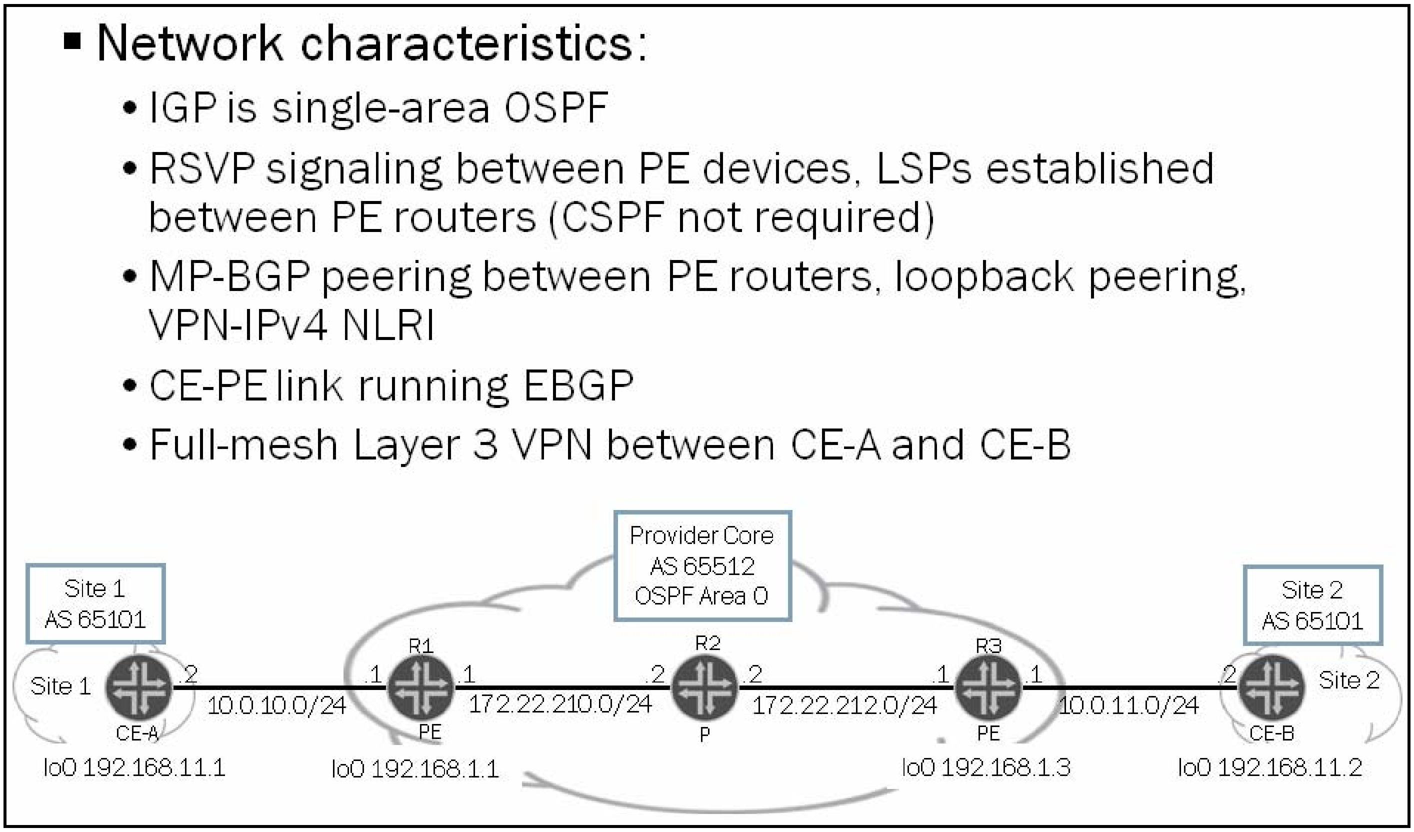
You must also link the VRF instance with either the **`vrf-target`** statement or VRF import and export policy statements. Finally, you must configure the VRF instance with a set of routing protocol properties compatible with the configuration of the attached CE routers.

VPN Policy

In the case where VRF import and export policies are used, the Junos OS does not allow you to commit your VPN configuration until the policy statements to which the VRF table is linked are created.

Minimum policy configuration involves the definition of route target community and the VRF import and export policies that use the route target to associate the route with a particular set of VRF tables.

Layer 3 VPN Example



- Network characteristics:
  - IGP is single-area OSPF
  - RSVP signaling between PE devices, LSPs established between PE routers (CSPF not required)
  - MP-BGP peering between PE routers, loopback peering, VPN-IPv4 NLRI
  - CE-PE link running EBGP
  - Full-mesh Layer 3 VPN between CE-A and CE-B

The diagram serves as the basis for the various configuration and operational mode examples that follow.

The IGP in use is Open Shortest Path First (OSPF), and a single area (Area 0) is configured. This example does not rely on the functionality of CSPF, so traffic engineering extensions are not enabled.

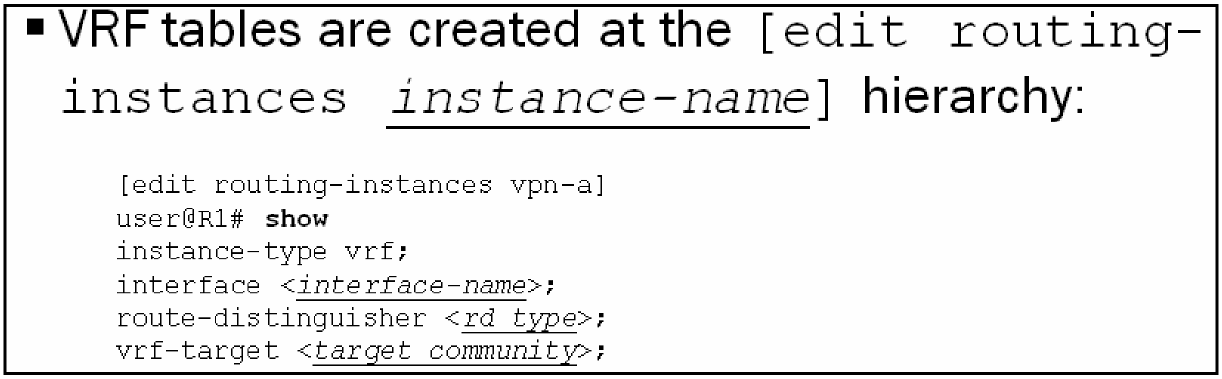
RSVP is deployed as the MPLS signaling protocol, and an LSP is configured between the R1 and R3 PE routers.

An MP-BGP peering session is configured between the loopback addresses of the PE routers. The VPN-IPv4 and `inet unicast` address families are configured.

In this example, the CE routers run EBGP. This results in the need for the PE routers to also run EBGP within their VRF routing instance.

The overall goal of this network is to provide full-mesh (which is point-to-point in this case) connectivity between the two CE routers shown. This application is considered full mesh as the resulting configuration readily accommodates the additional sites with any-to-any connectivity.

VRF Instances



- VRF tables are created at the `[edit routing-instances` *`instance-name`*`]` hierarchy:

```
[edit routing-instances vpn-a]
user@R1# show
instance-type vrf;
interface <interface-name>;
route-distinguisher <rd type>;
vrf-target <target community>;
```

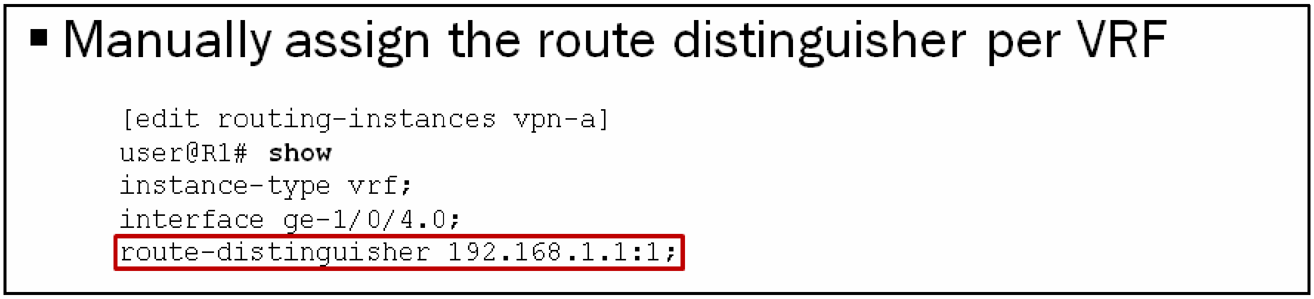
VRF routing instances are configured under the `[edit routing-instances` *`instance-name`*`]` portion of the configuration hierarchy.

This graphic reflects the required parameters for a VRF instance called ***vpn-a*** using the **`vrf-target`** option. Additional details about each required option is outlined in the following section:

- **`instance-type`**: Defines the type of routing instance being created and what parameters you have available to configure;
- **`interface`**: Identifies the logical, private interface between the PE router and the CE router on the PE side;
- **`route-distinguisher`**: An identifier attached to a route, enabling you to distinguish to which VPN the route belongs. Each routing instance must have a unique route distinguisher configured; and
- **`vrf-target`** or **`vrf-import`**/**`vrf-export`** policies:
  - **`vrf-target`**: VRF import and export policies are generated that accept and tag routes with the specified target community.
  - **`vrf-import`**/**`vrf-export`** policies: Defines how routes are imported and exported for the local PE router's VRF table.

You configure the PE-CE routing protocol under the `protocols` subhierarchy; you configure static routing under the `routing-options` sub-hierarchy.

Manually Assigning the Route Distinguisher per VRF Table

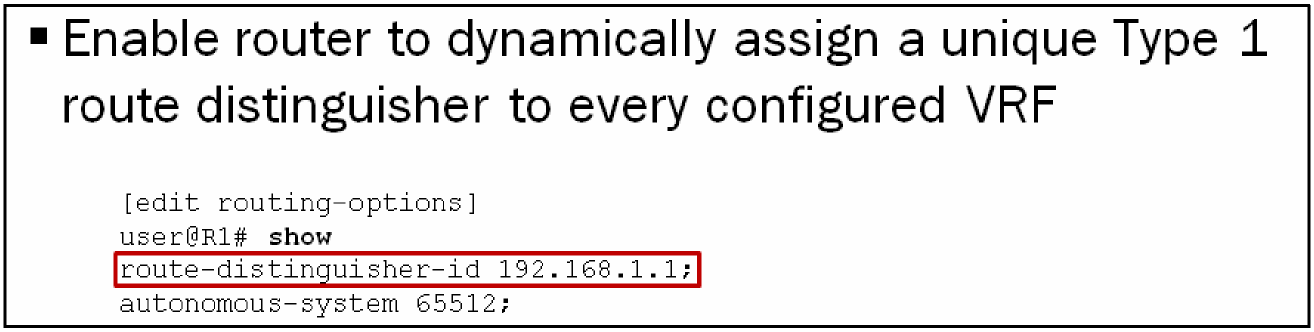


- Manually assign the route distinguisher per VRF

```
[edit routing-instances vpn-a]
user@R1# show
instance-type vrf;
interface ge-1/0/4.0;
route-distinguisher 192.168.1.1:1;
```

We manually assigned the VRF table on the graphic a route distinguisher of 192.168.1.1:1, which is an example of the Type 1 Route Distinguisher format. Each unique instance of a VRF table on this PE router must be given a unique assigned number to ensure that overlapping addresses from multiple VPN customers do not interfere with each other. This task can become daunting when dealing with hundreds of VRF tables on a single PE router.

Dynamic Assignment of the Route Distinguisher

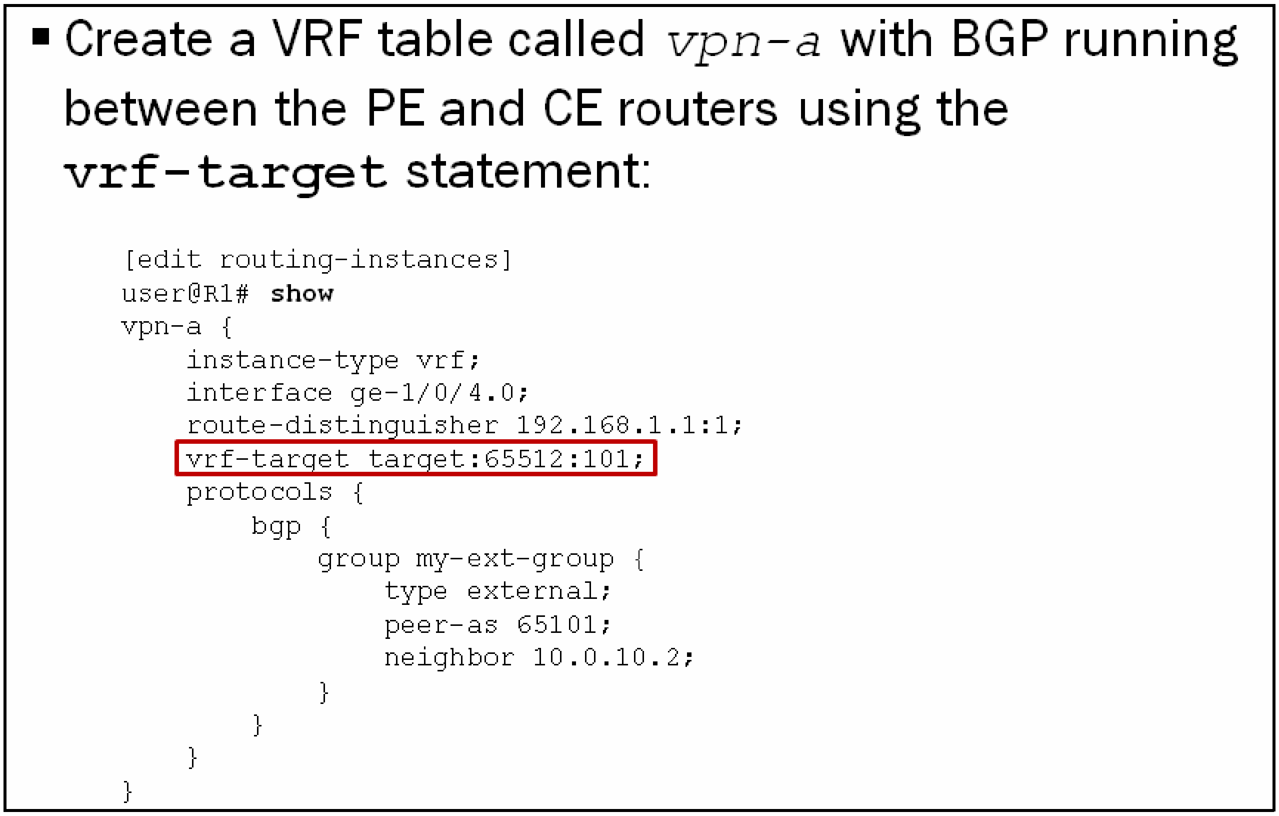


- Enable router to dynamically assign a unique Type 1 route distinguisher to every configured VRF

```
[edit routing-options]
user@R1# show
route-distinguisher-id 192.168.1.1;
autonomous-system 65512;
```

With this configuration, all VRF tables configured on this router will have a dynamically assigned Type 1 route distinguisher based on the `route-distinguisher-id` configured on the graphic (for example, 192.168.1.1:1, 192.168.1.1:2, etc...) The function of assigning a unique route distinguisher per VRF table is no longer your responsibility. You can override the dynamically assigned route distinguisher by manually configuring it under the `[edit routing-instances` ***`vpn-name`***`]` hierarchy.

A VRF Instance Example



- Create a VRF table called *vpn-a* with BGP running between the PE and CE routers using the **vrf-target** statement:

```
[edit routing-instances]
user@R1# show
vpn-a {
    instance-type vrf;
    interface ge-1/0/4.0;
    route-distinguisher 192.168.1.1:1;
    vrf-target target:65512:101;
    protocols {
        bgp {
            group my-ext-group {
                type external;
                peer-as 65101;
                neighbor 10.0.10.2;
            }
        }
    }
}
```

This example has a sample VRF instance configuration that supports EBGP routing on the PE-CE link. VRF tables require a routing instance type of **vrf**.

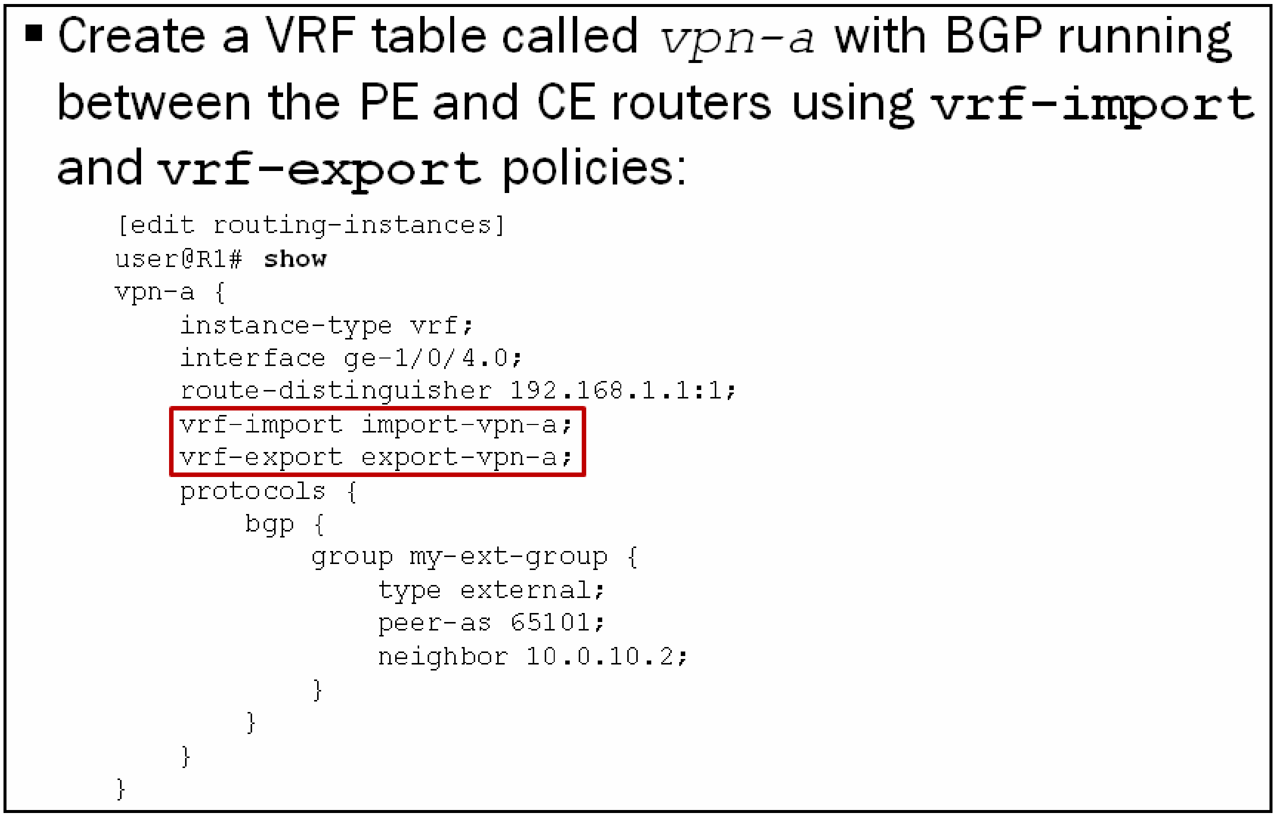
As reflected in the graphic, we configured a single VRF interface (ge-1/0/4.0). You should take care to specify the correct logical unit, especially when non-default unit numbers are in use.

We assigned this VRF table a route distinguisher of 192.168.1.1:1, which is an example of the Type 1 route distinguisher format. In this case, the PE router's loopback address is used as the administration field. Using the PE's router ID (RID) in the route distinguisher can assist with troubleshooting. You can easily track route advertisements back to the PE that generated them, based on the route distinguisher. The assigned number for this VRF table is 1. As mentioned previously, each unique instance of a VRF table on this PE router must be given a unique assigned number to ensure that overlapping addresses from multiple VPN customers remain separate.

We linked this VRF instance to a VRF target community. This method is the easiest way to configure advertisements of Layer 3 VPN routes between PE routers. Another method that can be used is to specify the import community and export community independently (not shown). The **import** statement causes all received Layer 3 VPN MP-BGP routes tagged with the correct target community to be placed into the *vpn-a*.inet.0 table. The **export** statement causes all routes in the *vpn-a*.inet.0 table to be advertised and tagged with the listed target community to all MP-BGP peers.

Finally, the graphic shows a standard EBGP configuration under the protocols hierarchy of the VRF instance's configuration. PE-CE routing protocols are configured for VRF instances the same way as they are configured under the main instance, with the only differences being their association with a VRF instance. If needed, BGP import and export policies can be applied to the BGP instance to refine and control the exchange of BGP routes on the PE-CE routing instance (not shown).

Another VRF Example



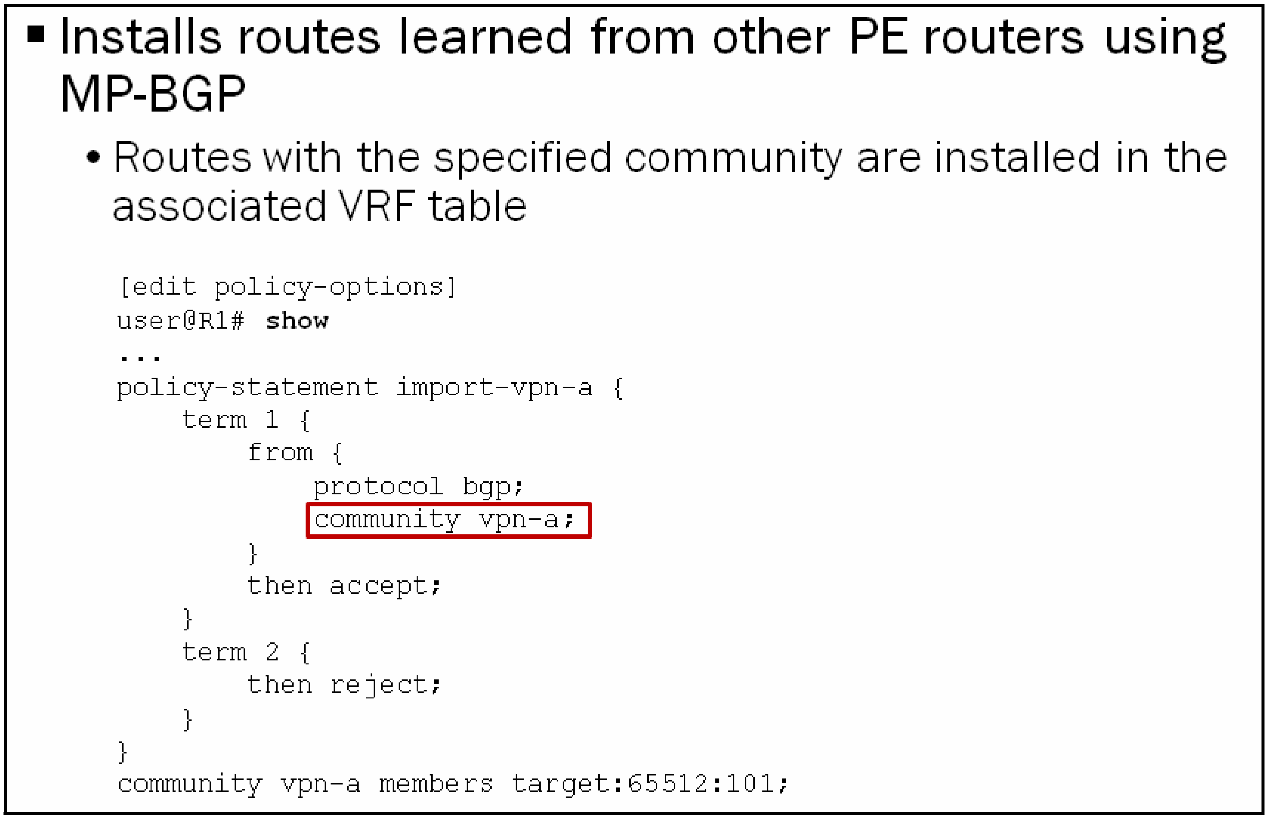
- Create a VRF table called *vpn-a* with BGP running between the PE and CE routers using **vrf-import** and **vrf-export** policies:

```
[edit routing-instances]
user@R1# show
vpn-a {
    instance-type vrf;
    interface ge-1/0/4.0;
    route-distinguisher 192.168.1.1:1;
    vrf-import import-vpn-a;
    vrf-export export-vpn-a;
    protocols {
        bgp {
            group my-ext-group {
                type external;
                peer-as 65101;
                neighbor 10.0.10.2;
            }
        }
    }
}
```

This example shows the use of **vrf-import** and **vrf-export** policies rather than the **vrf-target** statement. This methodology gives an administrator more control over the routes advertised between PE routers, but it requires more configuration.

We linked this VRF instance to VRF policy statements. The router does not allow a commit until the *import-vpn-a* and *export-vpn-a* policy statements are created under the [edit policy-options] hierarchy. We will define these policies next.

VRF Import Filters Routes Learned from Remote PE Routers



- Installs routes learned from other PE routers using MP-BGP
  - Routes with the specified community are installed in the associated VRF table

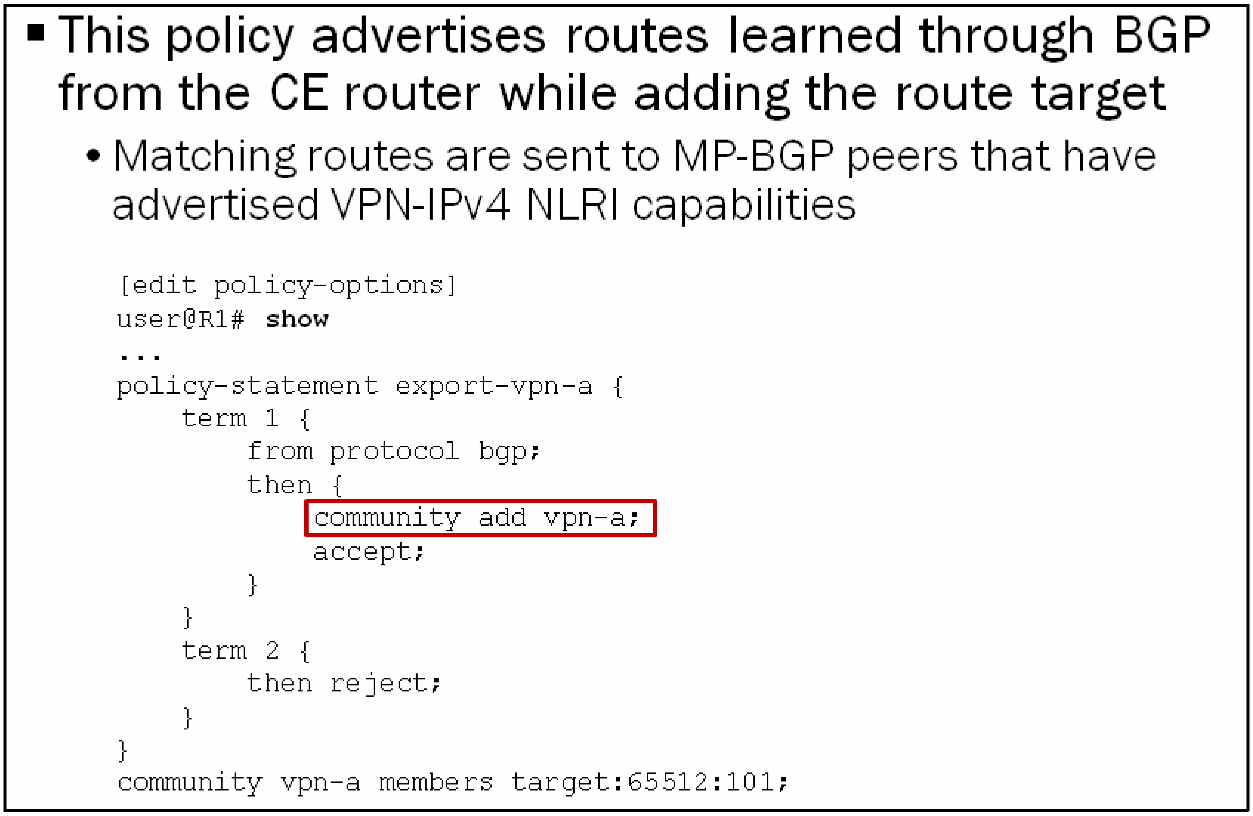
```
[edit policy-options]
user@R1# show
...
policy-statement import-vpn-a {
    term 1 {
        from {
            protocol bgp;
            community vpn-a;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
}
community vpn-a members target:65512:101;
```

This graphic provides an example of a typical VRF import policy. A VRF import policy only filters routes learned from remote PE routers through the MP-BGP peering sessions.

The `term 1` of policy `import-vpn-a` matches BGP routes containing the community string defined under the community name `vpn-a`. You can also see that the community associated with this name is a route target extended community. When a match occurs in `term 1`, the route is accepted and installed into the VRF tables linked to this policy.

The `term 2` of policy `import-vpn-a` serves as an explicit definition of the default policy for VRF import; that is, the PE router rejects all VPN routes by default. Put another way, a PE router only accepts a VPN route when an explicit route target match occurs in conjunction with an `accept` action.When dealing with security, it is usually better to use explicit rather than implicit rules, as explicit rules tend to avoid the misinterpretations, which can lead to unexpected connectivity.

VRF Export Policy Filter Routes Sent to Remote PE Routers



- This policy advertises routes learned through BGP from the CE router while adding the route target
  - Matching routes are sent to MP-BGP peers that have advertised VPN-IPv4 NLRI capabilities

```
[edit policy-options]
user@R1# show
...
policy-statement export-vpn-a {
    term 1 {
        from protocol bgp;
        then {
            community add vpn-a;
            accept;
        }
    }
    term 2 {
        then reject;
    }
}
community vpn-a members target:65512:101;
```

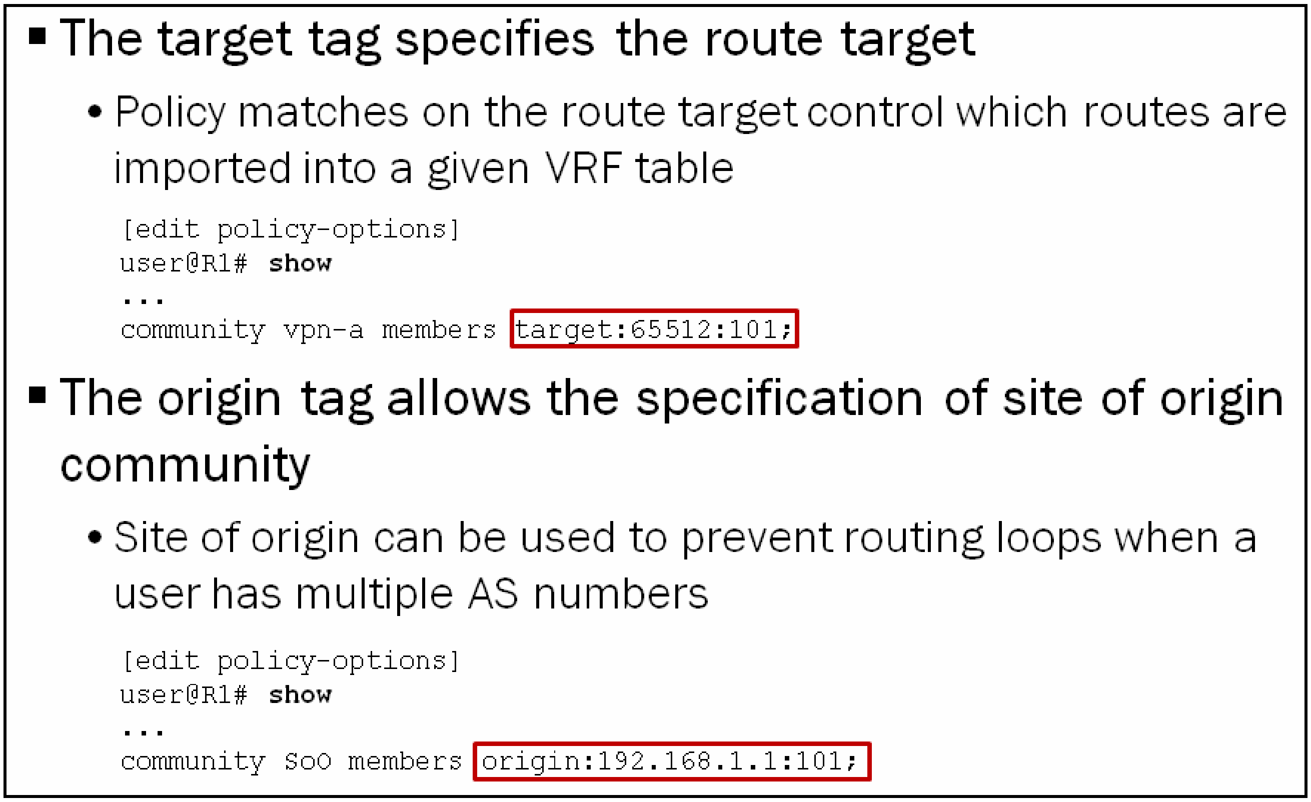
This graphic provides an example of a typical VRF export policy. A VRF export policy is only used to filter routes being advertised to remote PE routers through the MP-BGP peering sessions.

Term 1 of the policy matches routes learned from BGP. Because in this example the PE-CE routing protocol is BGP, all routes learned from the CE router match term 1 of the policy.

Matching routes have the community associated with the named community *vpn-a* added before being accepted for advertisement to the remote PE routers. Again, you can see that the community being added to the route is a route target BGP extended community.

As with the VRF import policy, term 2 provides explicit declaration of the default VRF policy action. Together, the two terms ensure that only routes learned from the CE router using EBGP are accepted for transmission to the remote PE routers with the proper parameters.

The Route Target Community



- The target tag specifies the route target
  - Policy matches on the route target control which routes are imported into a given VRF table

```
[edit policy-options]
user@R1# show
...
community vpn-a members target:65512:101;
```

- The origin tag allows the specification of site of origin community
  - Site of origin can be used to prevent routing loops when a user has multiple AS numbers

```
[edit policy-options]
user@R1# show
...
community SoO members origin:192.168.1.1:101;
```

Named communities under the `[edit policy-options]` portion of the configuration hierarchy define extended BGP communities. The graphic shows the syntax for extended community definition.

The route target community is critical to the operation of Layer 3 VPNs because only routes with matching route targets can be installed into a particular VRF table. This example shows a route target community using the Type 0 format, which uses a 2-byte administration field—set to the provider's autonomous system (AS) number—followed by a 4-byte assigned number field.

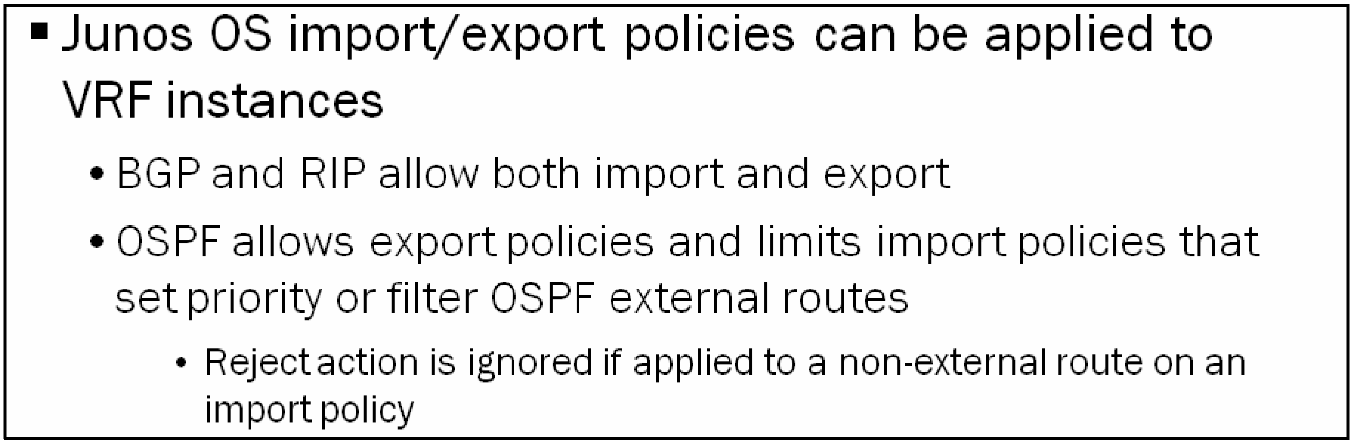
All members of a VPN setting and matching the same route target is common, but not mandatory.

The Site of Origin Community

The site of origin community associates a route with the site that originates the advertisement. A PE filter uses this community to filter the advertisement of a route back to the site from which it originated. Site of origin is optional and is only needed in certain corner cases. A sample application of the site of origin community is shown in subsequent pages.

The site of origin community in the example on the graphic uses the Type 1 format using a 4-byte administration field followed by a 2-byte assigned number field. In this case, the community is coded with the RID of the PE router that attaches to the CE device. The community uses the number 101 to distinguish this site from other sites this PE router might also serve.

PE-CE Routing Policy



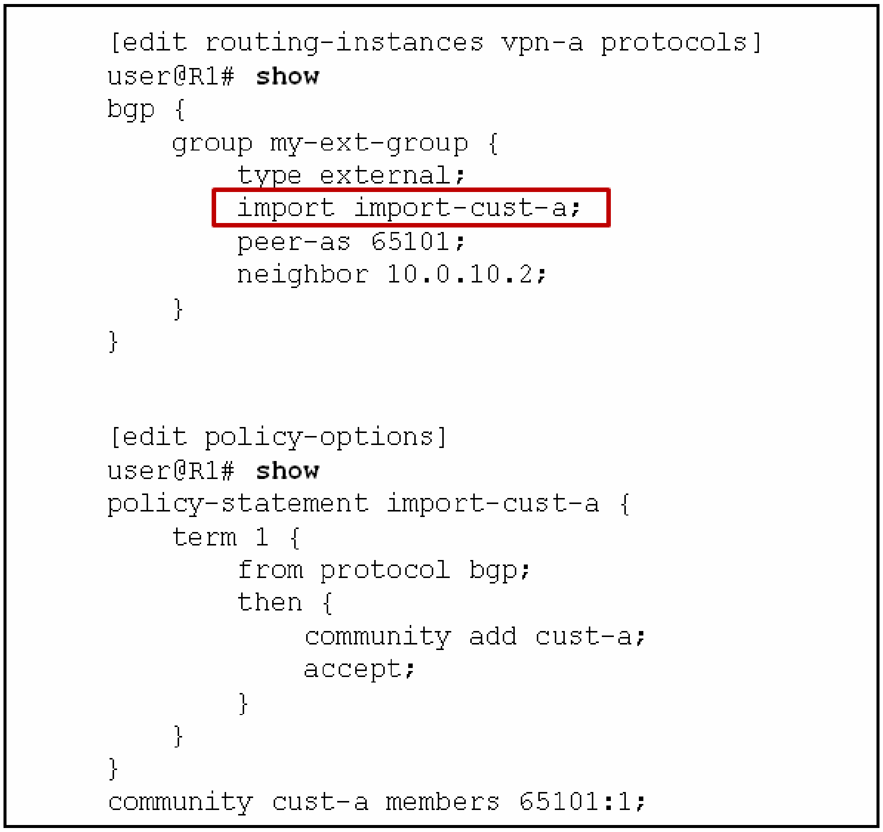
- Junos OS import/export policies can be applied to VRF instances
  - BGP and RIP allow both import and export
  - OSPF allows export policies and limits import policies that set priority or filter OSPF external routes
    - Reject action is ignored if applied to a non-external route on an import policy

In addition to the VRF import and export policies, which control the exchange of routes between PE routers, you can also use routing policy to control the exchange of routes on the PE-CE routing instance. Using BGP or RIP as a PE-CE routing protocol permits both import and export policies. OSPF can also have export policies, but has limited functionality when implementing an import policy. OSPF import policy can only be used to set priority or to filter OSPF external routes. If an OSPF import policy is applied that results in a reject terminating action for a non-external route, then the reject action is ignored and the route is accepted anyway. This behavior prevents traffic black holes, that is, silently discarded traffic, by ensuring consistent routing within the OSPF domain.

Affects PE-CE Route Exchange

Routing policy applied under the `protocols` portion of a VRF table only affects the routes being exchanged on the local PE-CE link. To control the exchange of VPN routes between PE routers, VRF import and export policies must exist.

PE-CE Policy Example

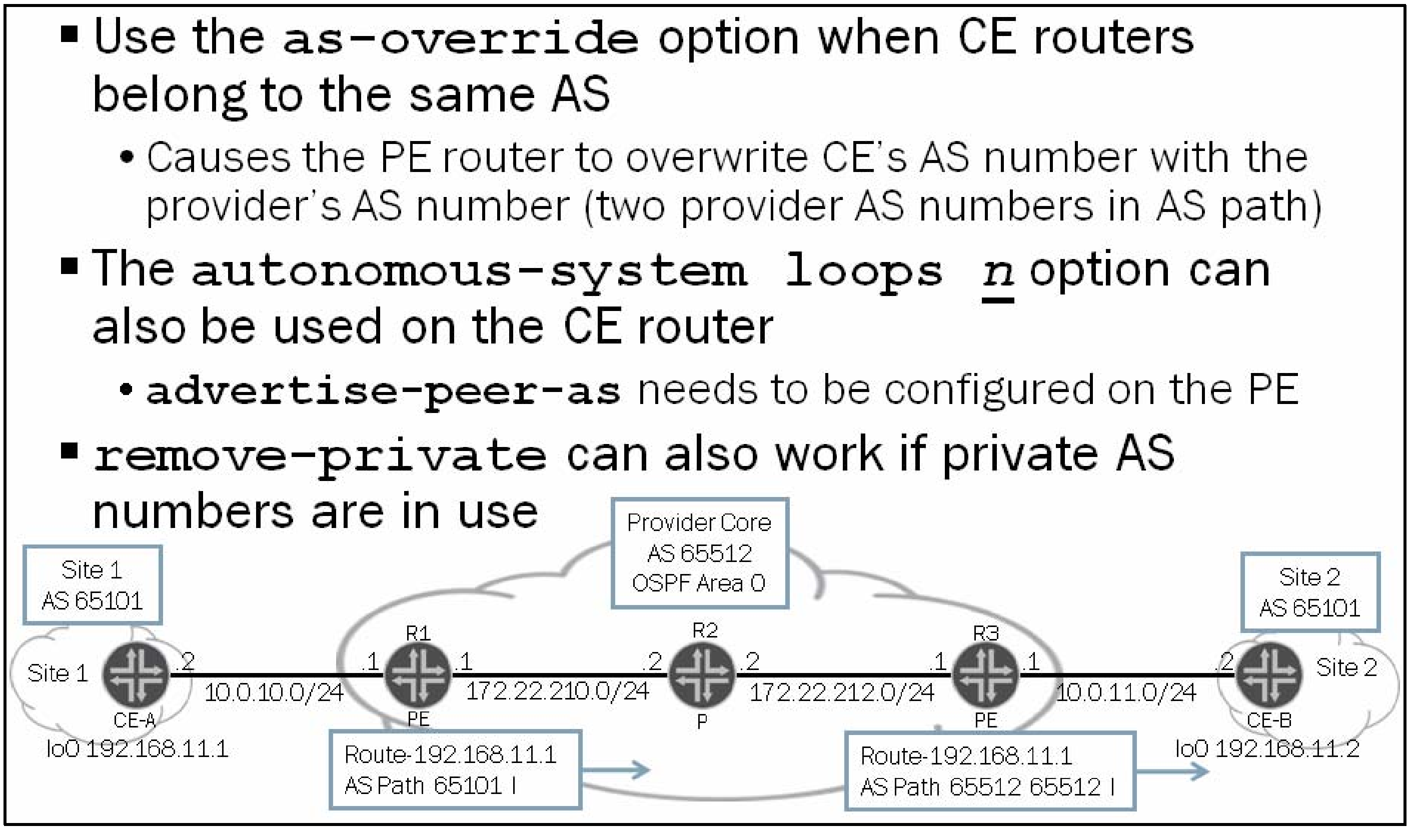


```
[edit routing-instances vpn-a protocols]
user@R1# show
bgp {
    group my-ext-group {
        type external;
        import import-cust-a;
        peer-as 65101;
        neighbor 10.0.10.2;
    }
}


[edit policy-options]
user@R1# show
policy-statement import-cust-a {
    term 1 {
        from protocol bgp;
        then {
            community add cust-a;
            accept;
        }
    }
}
community cust-a members 65101:1;
```

This graphic provides an example of a PE-CE BGP routing instance using an import policy to alter the properties of the routes received from the local CE device. The policy statement *import-cust-a* accepts all BGP routes into the local VRF table after adding the community values associated with the named community *cust-a*.

You can also use route filter statements to accept or reject routes explicitly based on the prefix and mask lengths.

Customer Sites with a Common Autonomous System Number



- Use the `as-override` option when CE routers belong to the same AS
  - Causes the PE router to overwrite CE's AS number with the provider's AS number (two provider AS numbers in AS path)
- The `autonomous-system loops` *n* option can also be used on the CE router
  - `advertise-peer-as` needs to be configured on the PE
- `remove-private` can also work if private AS numbers are in use

Because BGP uses the AS-path attribute to detect loops, problems can arise when VPN sites use the same AS number, as the CE routers ordinarily discard routes indicating an AS path loop. When sites are assigned the same AS number, the **`as-override`** configuration option is one way of supporting the interconnection of customer sites using EBGP as the PE-CE routing protocol (`as-override` is configured on the PE router, under the `protocols bgp` hierarchy within the VRF routing instance).

PE Router Overwrites Site's AS Number

In operation, the egress PE router configured to perform the AS override function replaces the AS number added by the originating VPN site with a second copy of the provider's AS number. This replacement results in two provider AS numbers in the AS-path attribute when the route is delivered to the destination site. The graphic shows the operation, where the 192.168.11.1/32 route is delivered to CE-B, with two instances of AS 65512 at the front of the AS-path attribute.

Allowing AS Loops

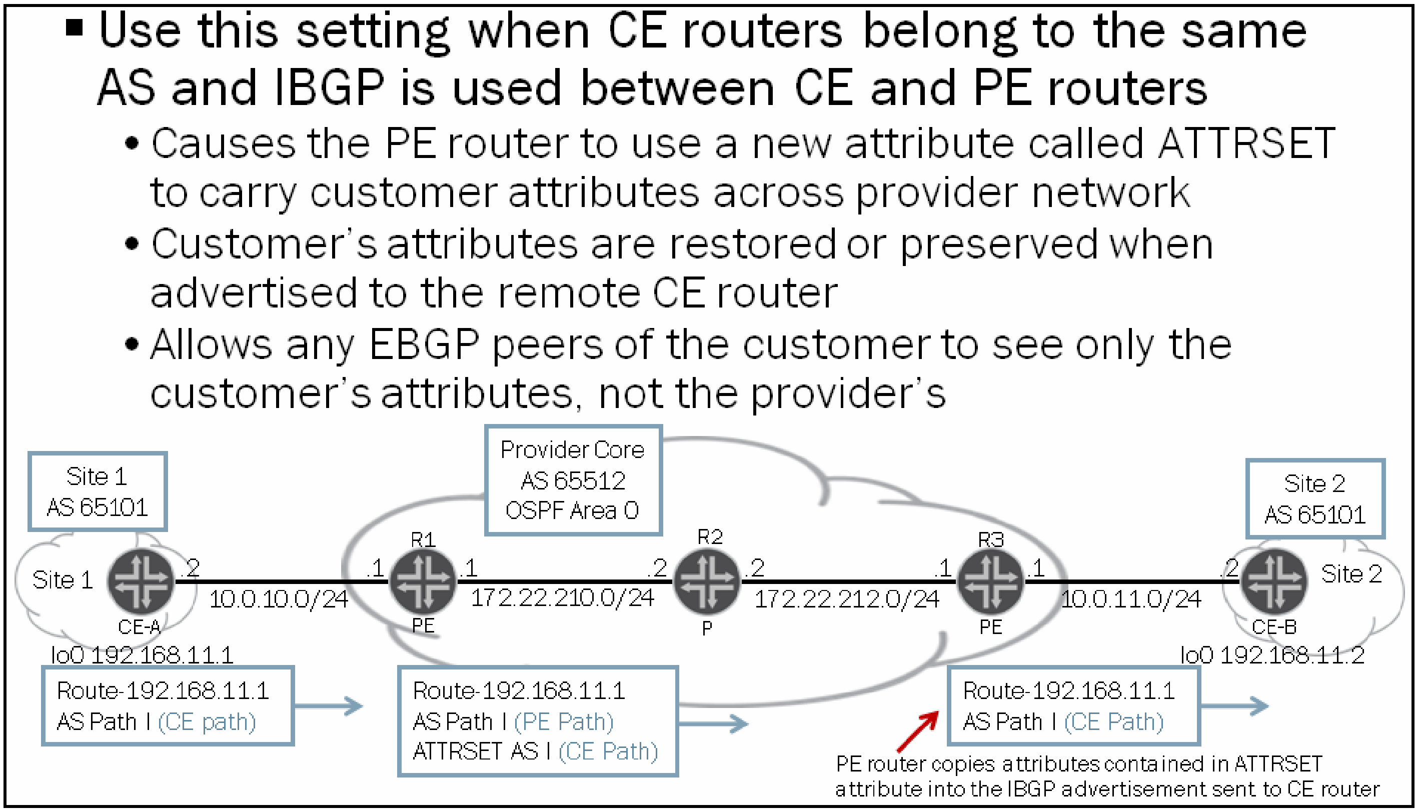
The Junos OS also supports the explicit allowance of AS loops by setting the **`autonomous-system loops`** ***n*** parameter under the BGP routing instance. When configured on the CE router, this parameter allows the CE router to install the VPN-learned routes in its routing table by ignoring up to *n* instances of its own AS in the AS path attribute of received routes.

By default, the Junos OS does not advertise routes whose AS path attribute contains the peer's AS. Because of this default behavior, the **`autonomous-system loops`** ***n*** solution also requires that the PE router be configured with the **`advertise-peer-as`** parameter at the `[edit protocols bgp group` *`group-name`* `neighbor` *`x.x.x.x`*`]` hierarchy. This parameter causes the PE router to include routes whose AS path attribute contains the CE router's AS number in its advertisements the CE router.

Remove Private

The **`remove-private`** option provides yet another way of solving the problem, but only when the customer sites use AS number from the private-use AS numbering space. In this case, you could enable **`remove-private`** under the PE router's main BGP routing instance. Enabling **`remove-private`** causes the PE router to remove any private AS numbers from the front of the AS path when sending MP-BGP updates to remote PE routers.

Independent Domain Setting



- Use this setting when CE routers belong to the same AS and IBGP is used between CE and PE routers
  - Causes the PE router to use a new attribute called ATTRSET to carry customer attributes across provider network
  - Customer's attributes are restored or preserved when advertised to the remote CE router
  - Allows any EBGP peers of the customer to see only the customer's attributes, not the provider's

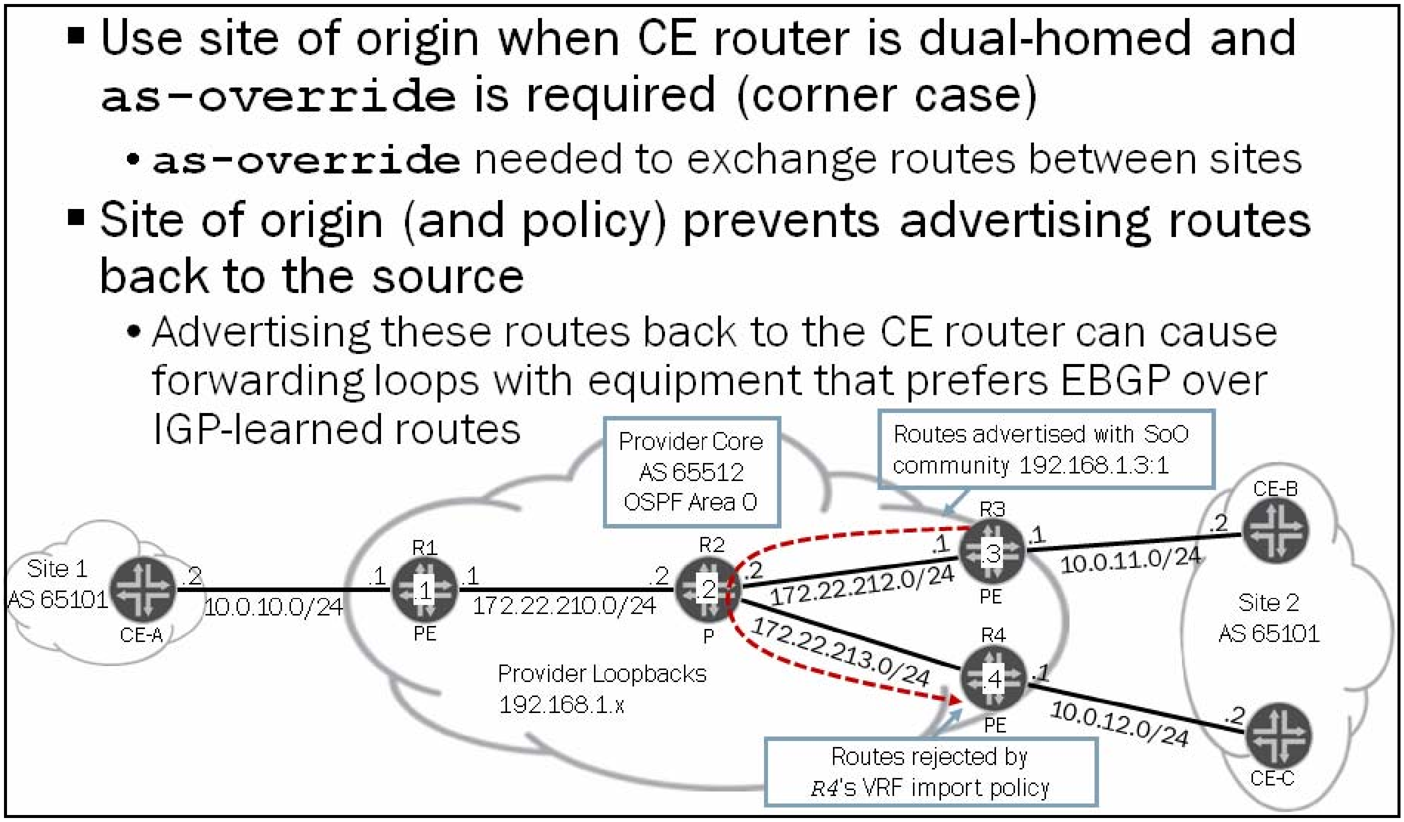
PE router copies attributes contained in ATTRSET attribute into the IBGP advertisement sent to CE router

Here is a sample configuration of **independent-domain** on R1:

```
[edit routing-instances vpn-a]
user@R1# show
instance-type vrf;
…
routing-options {
    autonomous-system 65101 independent-domain;
}
[edit routing-options]
user@R1# show
autonomous-system 65512;
```

As defined in draft-marques-ppvpn-ibgp-*version*.txt, this setting allows the PE routers to preserve the customer's attributes by storing them in the ATTRSET attribute because the routes cross the provider's backbone. Normally, without setting **independent-domain**, the provider's attributes are added to the customer's routes. In the example on the graphic, the **independent-domain** setting allows the remote PE router to advertise routes to the remote CE router using the routes' original attributes. Without the use of **independent-domain** on R1, the routes advertised from R3 to CE-B, would contain an AS path of 65512 65101. This would cause CE-B to detect an AS-path loop and drop the routes. Also, if CE-B allowed for AS-path loops as described on the previous graphic, any downstream EBGP peers of the customer's would evaluate these routes as being three AS hops away (that is, AS path = 65101 65512 65101.)

Dual-Homed Sites Using `as-override`



- Use site of origin when CE router is dual-homed and `as-override` is required (corner case)
  - `as-override` needed to exchange routes between sites
- Site of origin (and policy) prevents advertising routes back to the source
  - Advertising these routes back to the CE router can cause forwarding loops with equipment that prefers EBGP over IGP-learned routes

Provider Core
AS 65512
OSPF Area 0

Routes advertised with SoO community 192.168.1.3:1

Routes rejected by *R4*'s VRF import policy

In this example, we see a two-site VPN, where one of the sites is dual-homed to two different PE routers. If the sites were using different AS numbers, then use of the site of origin community would not be required as routes originated by Site 2 would be rejected based on AS loop detection if they should ever be advertised back to Site 2. In this example, however, because both sites use the same AS number, the **`as-override`** option is needed so that the routes being exchanged between Sites 1 and 2 are not rejected due to AS path-based loop detection.

The resulting scenario is one example of a corner case where you might use the site of origin community.

Preventing Inefficiencies and Potential Loops

In operation, the VRF export policy of the R3 is set to add a site of origin community to the routes it receives from CE-B.This community uses the Type 1 format, with the PE router's RID used as the administrator field. The assigned number, which is 1 in this case, is associated with Site 2.

The VRF import policy statements of R4, match and reject routes having this particular community. The result is that routes learned from Site 2, CE-B, are filtered upon receipt by R4, so they are not sent back to Site 2. To complete the application, similar VRF export and import policies are applied on the R4 and R3 routers so that both PE routers prevent routes from being sent back to Site 2.

In some cases, the use of site of origin as shown in this application just makes things more efficient in that it eliminates the unnecessary transmission of route updates to Site 2. This elimination of unnecessary transmission in turn prevents the BGP speakers in Site 2 from having to carry duplicate BGP routing information. In other cases, the use of site of origin can be necessary to prevent forwarding loops. Some vendors prefer EBGP routes over IGP routes (the Junos OS does not), which could result in a forwarding loop when routes learned from a site are redistributed back to that site using EBGP.

OSPF Routing

The support of OSPF routing on the PE-CE link requires a separate OSPF process for each VRF table. You configure these processes under the `protocols` portion of a VRF table configuration. The actual steps required to configure an OSPF instance in this case are no different from the steps needed to configure the main OSPF routing instance.

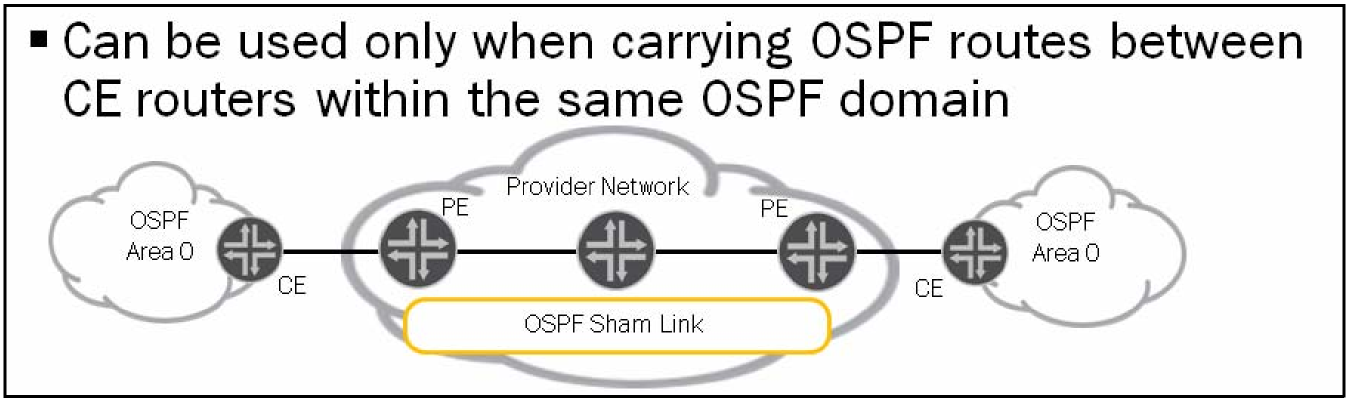
OSPF Routes Transported Between PE Routers Using MP-BGP

Routes learned from the CE device using OSPF are sent to remote PE routers as labeled VPN routes using MP-BGP. The receiving PE router can redistribute these routes to its attached CE device using OSPF or another routing protocol, such as BGP or RIP.

Two Methods

RFC 4577 defines two methods for advertising OSPF routes between CE routers that are running OSPF with their local PE routers. The first method is through the use of the OSPF sham link. The second method is to make use of the BGP extended community, domain ID, to control the link-state advertisement (LSA) translation between PE routers. We will discuss these two options next.

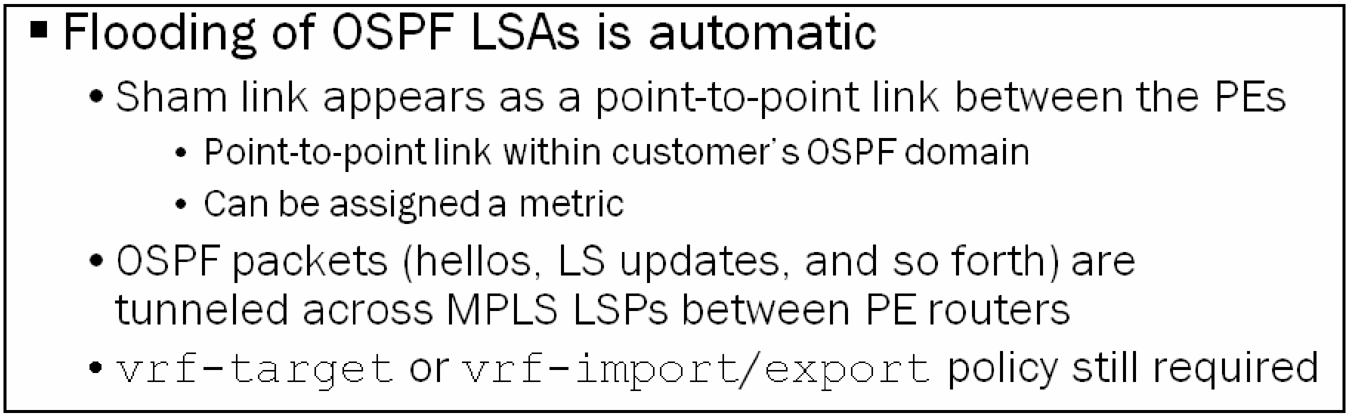
Using Sham Links



- Can be used only when carrying OSPF routes between CE routers within the same OSPF domain

A sham link can be used when the CE routers belong to the same OSPF domain but not necessarily the same OSPF area. The sham link essentially mocks an unnumbered point-to-point link within the VRF routing instance between PE routers.

Automatic Flooding of OSPF LSAs

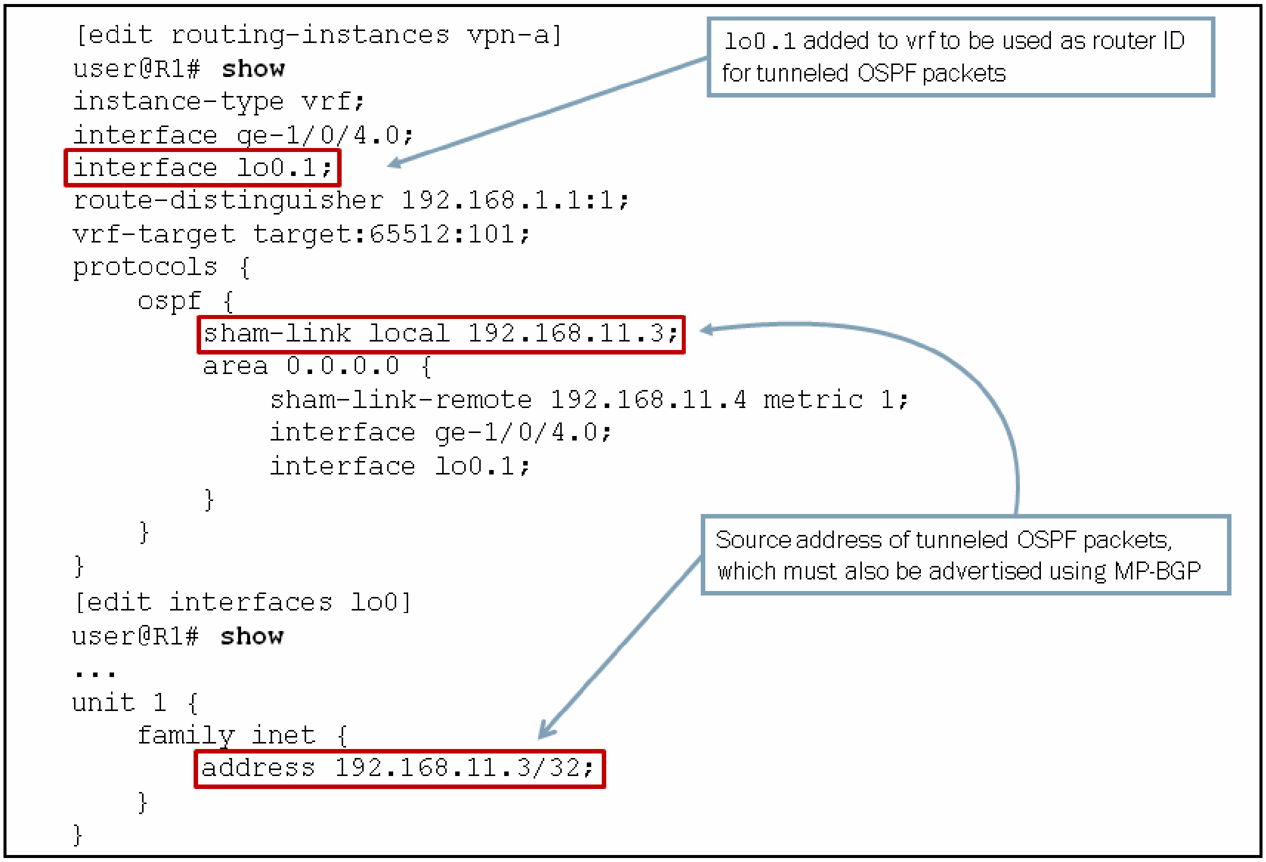


- Flooding of OSPF LSAs is automatic
  - Sham link appears as a point-to-point link between the PEs
    - Point-to-point link within customer's OSPF domain
    - Can be assigned a metric
  - OSPF packets (hellos, LS updates, and so forth) are tunneled across MPLS LSPs between PE routers
  - `vrf-target` or `vrf-import/export` policy still required

Normally, a PE router must redistribute the VPN routes it has learned through its MP-BGP peering sessions into OSPF as external routes (LSA Type 5/7) or summary routes (LSA Type 3). In the case of a sham link, once it is operational, OSPF packets are tunneled between PE routers using the MPLS LSPs that are established between them. A PE router learning the MPLS-tunneled LSAs from a remote PE router floods those LSAs to the local CE router. This behavior allows Type 1 and Type 2 LSAs to be passed across the VPN between CE routers.

Although a PE router learns routes from the remote PE router using OSPF, it cannot use the OSPF routes for forwarding and instead must use MP-BGP learned routes to forward traffic to the remote site. Thus, a PE router must not only learn the routes using OSPF, but it must also learn the same routes using MP-BGP.

OSPF Sham Link Example: Part 1



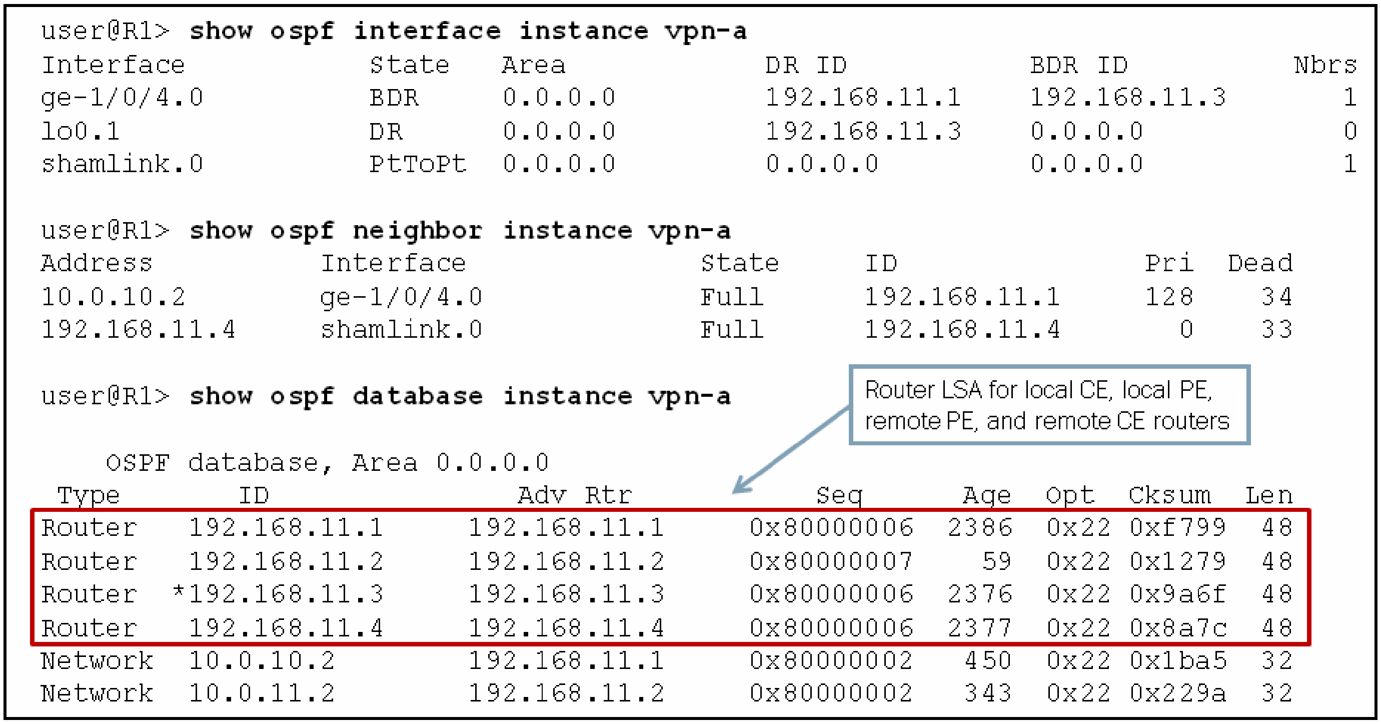
```
[edit routing-instances vpn-a]
user@R1# show
instance-type vrf;
interface ge-1/0/4.0;
interface lo0.1;
route-distinguisher 192.168.1.1:1;
vrf-target target:65512:101;
protocols {
    ospf {
        sham-link local 192.168.11.3;
        area 0.0.0.0 {
            sham-link-remote 192.168.11.4 metric 1;
            interface ge-1/0/4.0;
            interface lo0.1;
        }
    }
}
[edit interfaces lo0]
user@R1# show
...
unit 1 {
    family inet {
        address 192.168.11.3/32;
    }
}
```

`lo0.1` added to vrf to be used as router ID for tunneled OSPF packets

Source address of tunneled OSPF packets, which must also be advertised using MP-BGP

This graphic shows a VRF table configured to support OSPF operation on the PE-CE link as well as a sham link between PE routers. The OSPF sham link's local address must be the loopback VPN interface for the local VPN. To be reachable by the remote PE router, this loopback address must be advertised using MP-BGP (solved with the **`vrf-target`** statement in the graphic.) The OSPF sham link's remote address must be a loopback VPN interface on the remote PE router.

OSPF Sham Link Example: Part 2



```
user@R1> show ospf interface instance vpn-a
Interface            State    Area            DR ID           BDR ID          Nbrs
ge-1/0/4.0           BDR      0.0.0.0         192.168.11.1    192.168.11.3       1
lo0.1                DR       0.0.0.0         192.168.11.3    0.0.0.0            0
shamlink.0           PtToPt   0.0.0.0         0.0.0.0         0.0.0.0            1

user@R1> show ospf neighbor instance vpn-a
Address          Interface              State     ID               Pri  Dead
10.0.10.2        ge-1/0/4.0             Full      192.168.11.1     128    34
192.168.11.4     shamlink.0             Full      192.168.11.4       0    33

user@R1> show ospf database instance vpn-a

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq         Age  Opt  Cksum  Len
Router   192.168.11.1     192.168.11.1     0x80000006  2386  0x22 0xf799  48
Router   192.168.11.2     192.168.11.2     0x80000007    59  0x22 0x1279  48
Router  *192.168.11.3     192.168.11.3     0x80000006  2376  0x22 0x9a6f  48
Router   192.168.11.4     192.168.11.4     0x80000006  2377  0x22 0x8a7c  48
Network  10.0.10.2        192.168.11.1     0x80000002   450  0x22 0x1ba5  32
Network  10.0.11.2        192.168.11.2     0x80000002   343  0x22 0x229a  32
```

Router LSA for local CE, local PE, remote PE, and remote CE routers

This graphic displays some of the operational commands to help troubleshoot and verify OSPF sham links. Notice in the **show ospf neighbor** command that the local PE router has formed an adjacency with the remote PE router over the sham link. The topology used in the example is simply two CE routers, one P router, and two PE routers. Each router is an OSPF Area 0.0.0.0 internal router for the VPN. Notice in the output of the **show ospf database** command that exactly four router LSAs are advertised representing each of the four routers in Area 0.0.0.0.

Carrying Routes Between OSPF Domains or Carrying Interarea Routes

A sham link can be used only when the CE routers belong to the same OSPF domain. OSPF domain IDs can also be used when interconnecting a single OSPF domain and must be used when a Layer 3 VPN to connects multiple OSPF domains. Configuring OSPF domain IDs allows an administrator control LSA translation between the OSPF domains.

Presenting OSPF Routes to the Remote CE Router

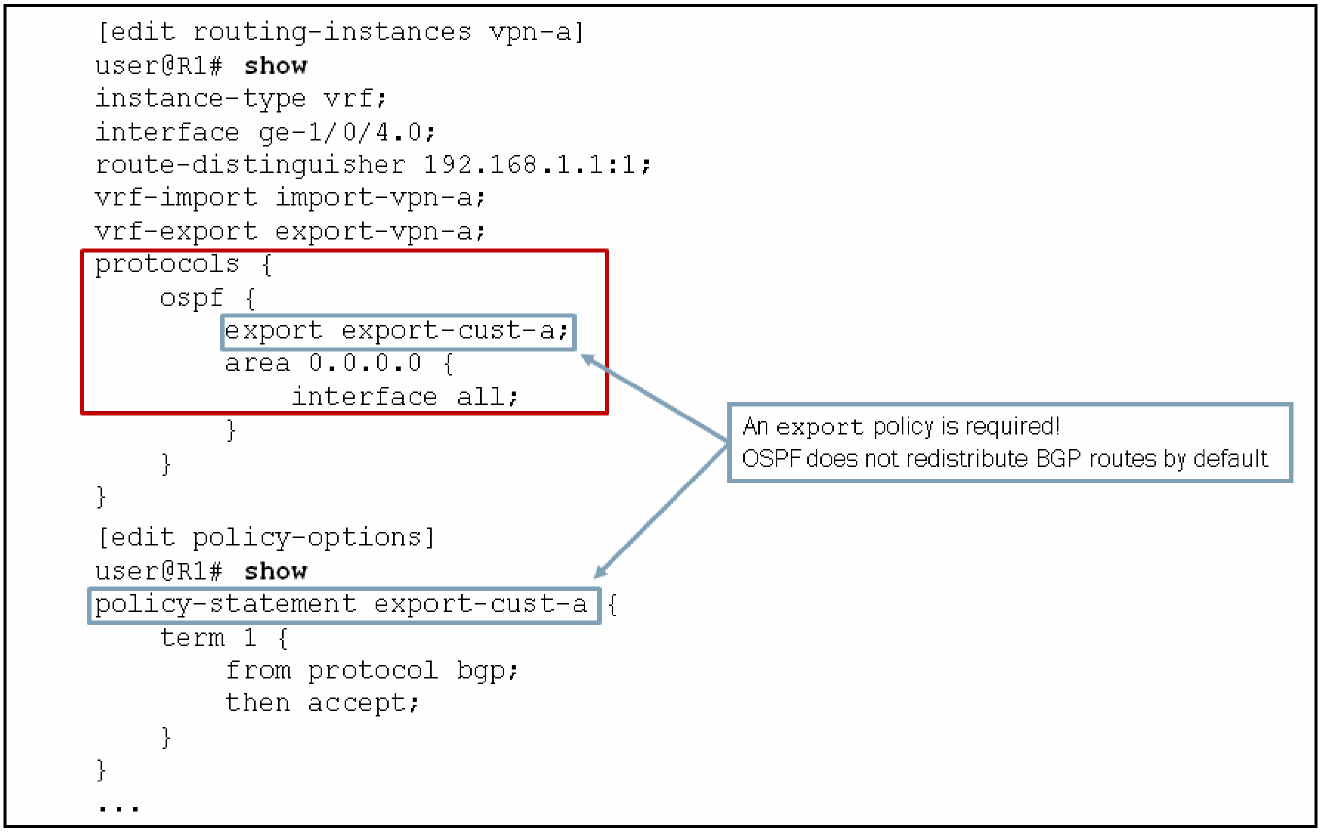
Normally, a PE router redistributes the VPN routes it has learned through its MP-BGP peering sessions as external routes (LSA Type 5). LSA Type 7s are generated when the PE-CE OSPF instances are configured as a not-so-stubby area (NSSA).

The Junos OS supports the OSPF domain ID extended community, as defined in RFC 4577. This community allows the PE router to present OSPF Type 1, 2, or 3 routes within the same OSPF domain to the CE router as network summaries (LSA Type 3) instead of the default external route presentation. Presenting routes as summary LSAs makes it possible to support both a Layer 3 VPN and a legacy backbone with metric-based control over which backbone is actually used. This capability can simplify the rollout of a new Layer 3 VPN-based backbone. OSPF routes with an external type are always presented to the remote CE device as an external LSA, regardless of the domain ID setting. Subsequent pages provide an example of this application.

VRF Import and Export Polices for PE-CE OSPF Support

Users make mistakes in the VRF import and export policies when trying to use OSPF as the PE-CE routing protocol. In this application, the VRF export policy must match and accept OSPF routes, while the VRF import policy must match and accept BGP routes. Additionally, an export policy is required under the OSPF instance to allow the redistribution of BGP routes into OSPF.

OSPF VRF Table Example



```
[edit routing-instances vpn-a]
user@R1# show
instance-type vrf;
interface ge-1/0/4.0;
route-distinguisher 192.168.1.1:1;
vrf-import import-vpn-a;
vrf-export export-vpn-a;
protocols {
    ospf {
        export export-cust-a;
        area 0.0.0.0 {
            interface all;
        }
    }
}
[edit policy-options]
user@R1# show
policy-statement export-cust-a {
    term 1 {
        from protocol bgp;
        then accept;
    }
}
...
```

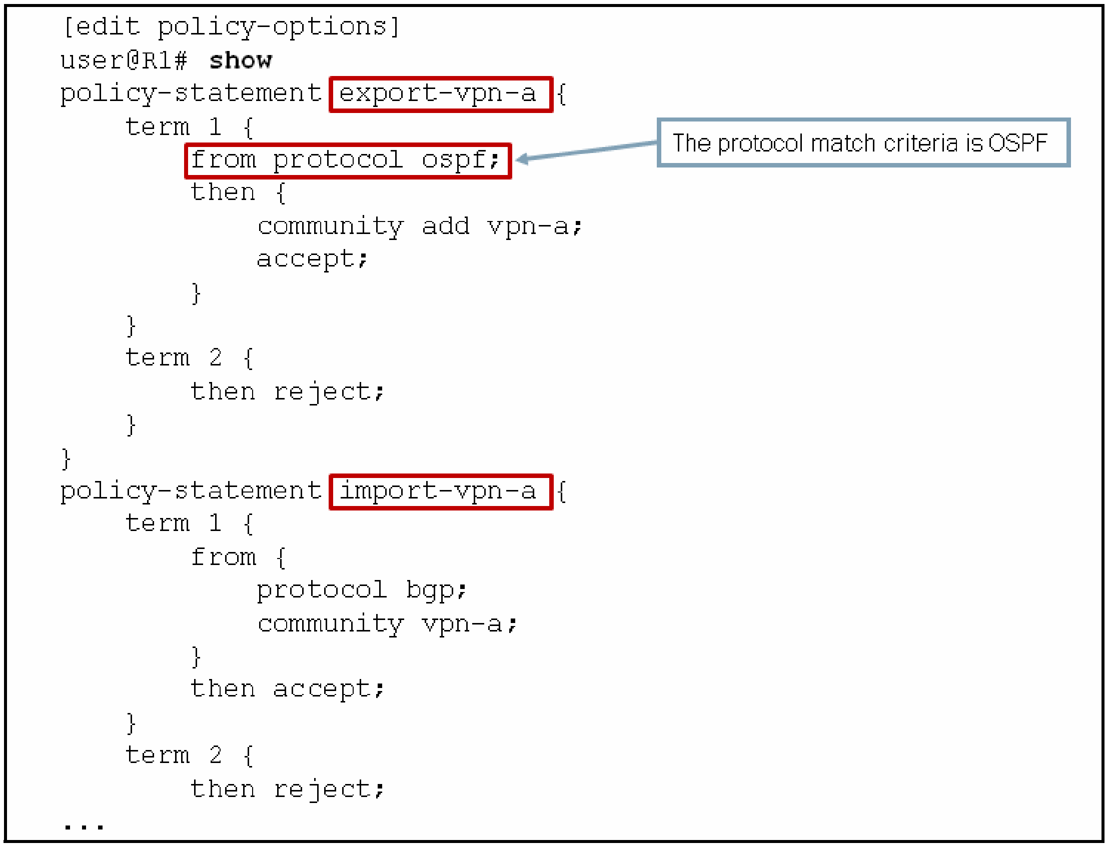
An export policy is required!
OSPF does not redistribute BGP routes by default

This graphic shows a VRF table configured to support basic OSPF operation on the PE-CE link. You can assume that the VRF import policy matches BGP routes, and that the VRF export policy matches OSPF routes (the actual VRF policies are discussed in the next graphic).

As indicated, you must specify an export policy under the OSPF instance to allow the redistribution of BGP into OSPF. This policy is needed because, between PE routers, all routes are learned through the BGP protocol, regardless of what protocol is being used on the PE-CE link.

The actual configuration of the OSPF instance is really no different from the configuration of a main OSPF routing instance. You must specify the OSPF area number and list the VRF interfaces belonging to that area.

Examples of VRF Policies for OSPF Support



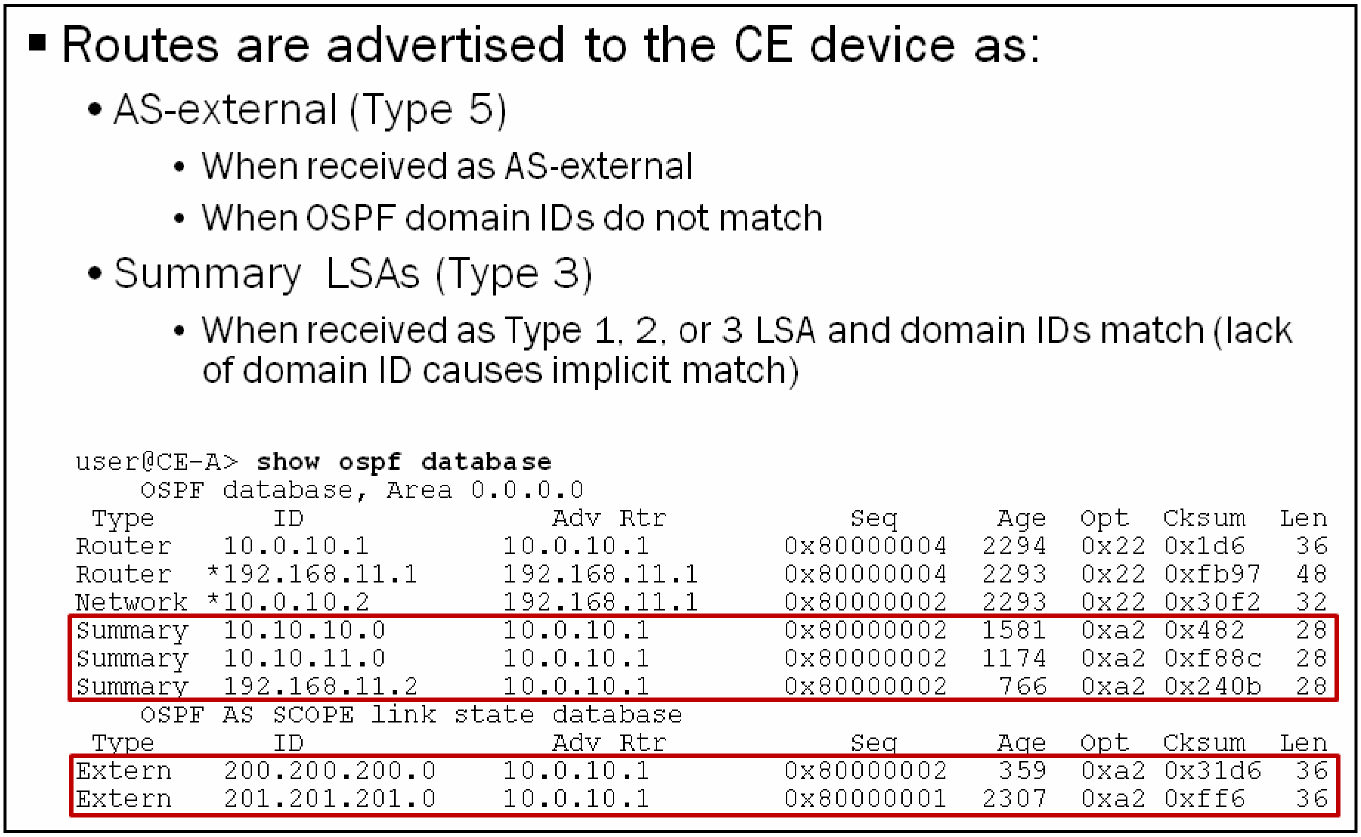
```
[edit policy-options]
user@R1# show
policy-statement export-vpn-a {
    term 1 {
        from protocol ospf;
        then {
            community add vpn-a;
            accept;
        }
    }
    term 2 {
        then reject;
    }
}
policy-statement import-vpn-a {
    term 1 {
        from {
            protocol bgp;
            community vpn-a;
        }
        then accept;
    }
    term 2 {
        then reject;
...
```

The protocol match criteria is OSPF

This graphic shows the VRF import and export policy needed to ensure that routes learned from the CE device using OSPF are sent to remote PE routers, and that routes learned from remote PE routers using MP-BGP are accepted and installed into the local PE router’s VRF table.

These two policies get the routes to and from the PE routers, but remember that a BGP redistribution policy is needed to get the BGP routes learned from remote PE routers sent to the local CE device running OSPF.

OSPF Routes Presented as Summary and External LSAs



- Routes are advertised to the CE device as:
  - AS-external (Type 5)
    - When received as AS-external
    - When OSPF domain IDs do not match
  - Summary LSAs (Type 3)
    - When received as Type 1, 2, or 3 LSA and domain IDs match (lack of domain ID causes implicit match)

```
user@CE-A> show ospf database
    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router   10.0.10.1        10.0.10.1        0x80000004  2294  0x22 0x1d6   36
Router  *192.168.11.1     192.168.11.1     0x80000004  2293  0x22 0xfb97  48
Network *10.0.10.2        192.168.11.1     0x80000002  2293  0x22 0x30f2  32
Summary  10.10.10.0       10.0.10.1        0x80000002  1581  0xa2 0x482   28
Summary  10.10.11.0       10.0.10.1        0x80000002  1174  0xa2 0xf88c  28
Summary  192.168.11.2     10.0.10.1        0x80000002   766  0xa2 0x240b  28
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   200.200.200.0    10.0.10.1        0x80000002   359  0xa2 0x31d6  36
Extern   201.201.201.0    10.0.10.1        0x80000001  2307  0xa2 0xff6   36
```

The result of this basic PE-CE OSPF configuration is shown on the graphic where the contents of CE-A's OSPF link-state database (LSDB) are displayed. Because this basic example does not make use of the OSPF domain ID community, the R1 PE router assumes that the remote CE router belongs to the same OSPF domain and presents them as summary LSAs to the attached CE router when their route type does not indicate external.

As a result, the remote CE router's external routes (the 200.200.200/24 and 201.201.201/24 prefixes), which are being redistributed from static into OSPF, are presented as external LSAs (Type 5s), while the remote CE router's internal OSPF routes (the 192.168.11.2 loopback address, the 10.10.10/24, and the 10.10.10.11/24 OSPF interface routes not shown on the topology) appear in the receiving CE router as OSPF summary routes (LSA Type 3s).

Subsequent sections detail the operation of the OSPF domain ID and show the effect of mismatched domain IDs on these same routes.

Domain ID

Use of the domain ID community allows a PE router to redistribute routes learned from its MP-BGP sessions with remote PE routers as OSPF LSAs when certain conditions are met. The LSAs generated by the PE router make use of a previously undefined bit in the OSPF options field (the high-order bit) to prevent looping. In operation, the PE router sets the down bit when it generates an LSA and ignores any received LSAs having this bit set.

The PE router also includes an OSPF route tag in the LSAs it advertises to the CE router. This VPN OSPF route tag is also used to prevent the looping of LSAs. The VPN route tag is calculated automatically by default but might require manual setting as the VPN route tag must be unique within the OSPF domain. You can set the VPN route tag manually with the **`domain-vpn-tag`** option under the OSPF portion of the VRF table configuration. When computed automatically, the VPN route tag is based on the provider's AS number.

More Extended Communities

The OSPF domain ID specification requires the support of several BGP extended communities:

- The OSPF route type communities indicate the LSA type of the original OSPF route corresponding to this VPN-IPv4 route. Routes with an external type are always delivered to the CE router as an external LSA, whether or not the domain ID of the route matches the local site.

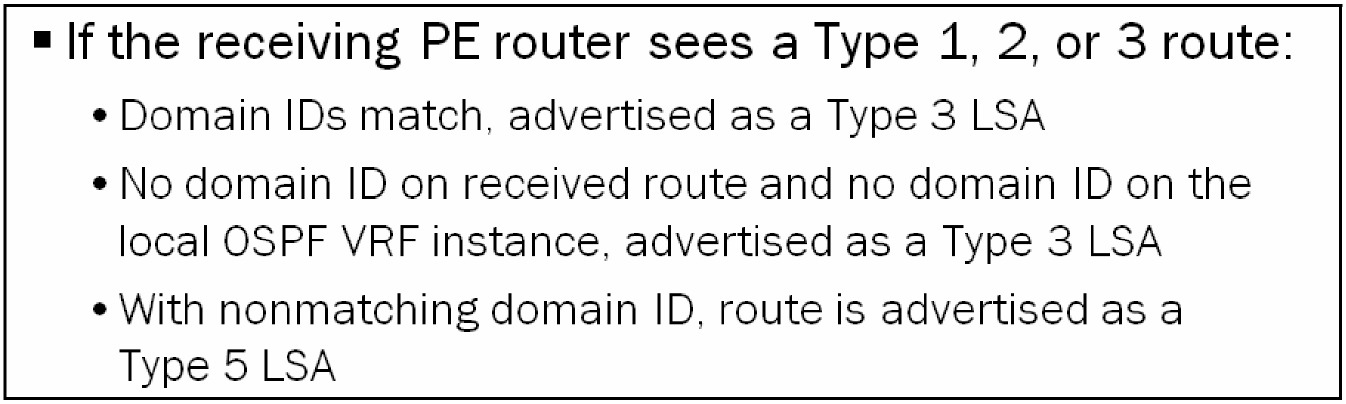
- The OSPF domain ID itself is supported with the VPN of origin extended community. The domain ID is normally coded as a 4-byte IP address with a 0 suffix.
- The OSPF router ID community is only needed when supporting the optional OSPF sham link.

In operation, a PE router generates a summary LSA when the received route type is internal and carries a domain ID community matching the domain ID configured under the local OSPF VRF instance (a missing domain ID on both the received route and the local OSPF VRF instance is also considered to be a match). Mismatched domain IDs, or routes with external types, result in the generation of external LSAs.

Backdoor Links

OSPF domain ID support facilitates a graceful migration from a customer's existing (legacy) WAN backbone onto a Layer 3 VPN, while allowing the use of both the legacy and VPN backbones during the transition period. With routes being presented as summary LSAs, simple adjustments to OSPF metrics can be performed to direct traffic over the backbone of choice. The metric-based selection of one backbone over another is extremely difficult when the routes learned over the VPN appear as external, as a router always chooses internal and intra-area routes over external routing.

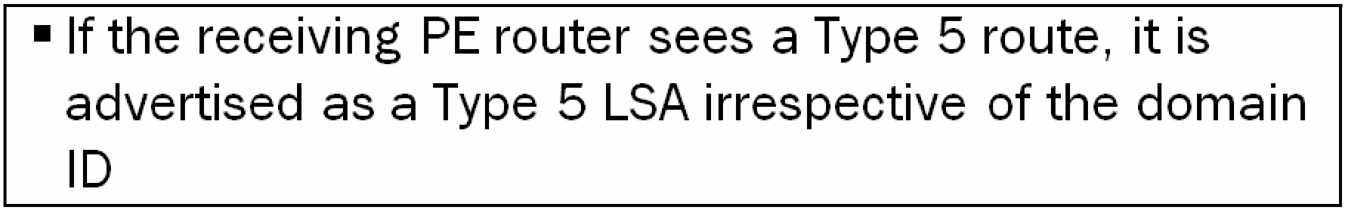
Rules for Receiving Type 1,2, or 3 LSAs



- If the receiving PE router sees a Type 1, 2, or 3 route:
  - Domain IDs match, advertised as a Type 3 LSA
  - No domain ID on received route and no domain ID on the local OSPF VRF instance, advertised as a Type 3 LSA
  - With nonmatching domain ID, route is advertised as a Type 5 LSA

For most OSPF configurations involving Layer 3 VPNs, you do not need to configure an OSPF domain ID. However, for a Layer 3 VPN connecting multiple OSPF domains, configuring OSPF domain IDs can help you to control LSA translation between the OSPF domains and backdoor paths. When a PE router receives a route, it redistributes and advertises the route either as a Type 3 LSA or as a Type 5 LSA, depending on the conditions that are listed on the graphic.

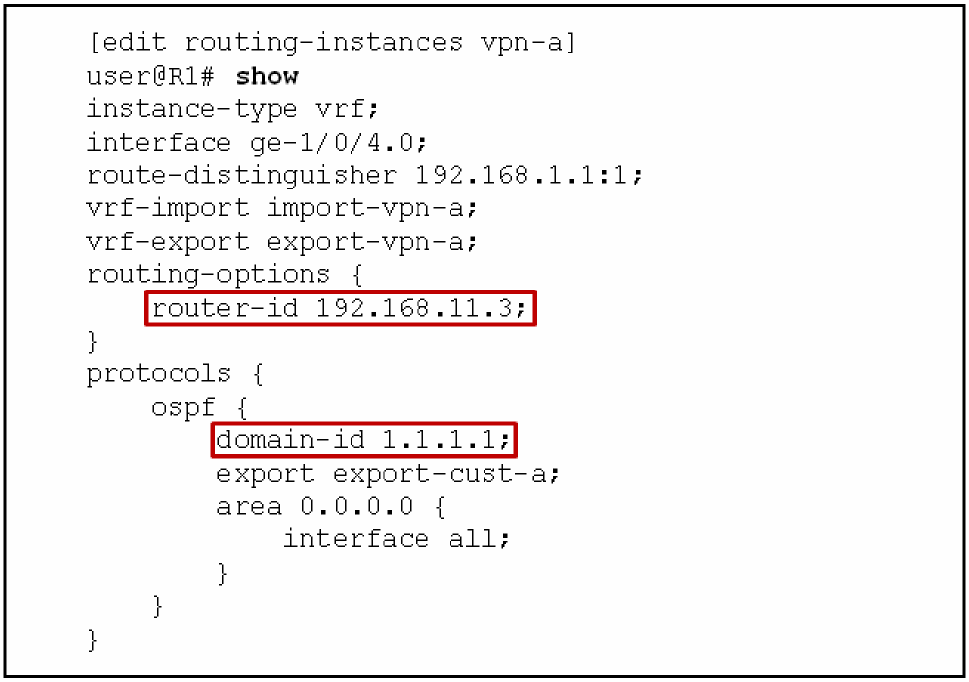
Rules for Receiving Type 5 LSAs



- If the receiving PE router sees a Type 5 route, it is advertised as a Type 5 LSA irrespective of the domain ID

There is only one thing to remember about a PE receiving Type 5 route. A Type 5 route will always be advertised as a Type 5 external LSA regardless of the domain ID configuration.

Domain ID Example



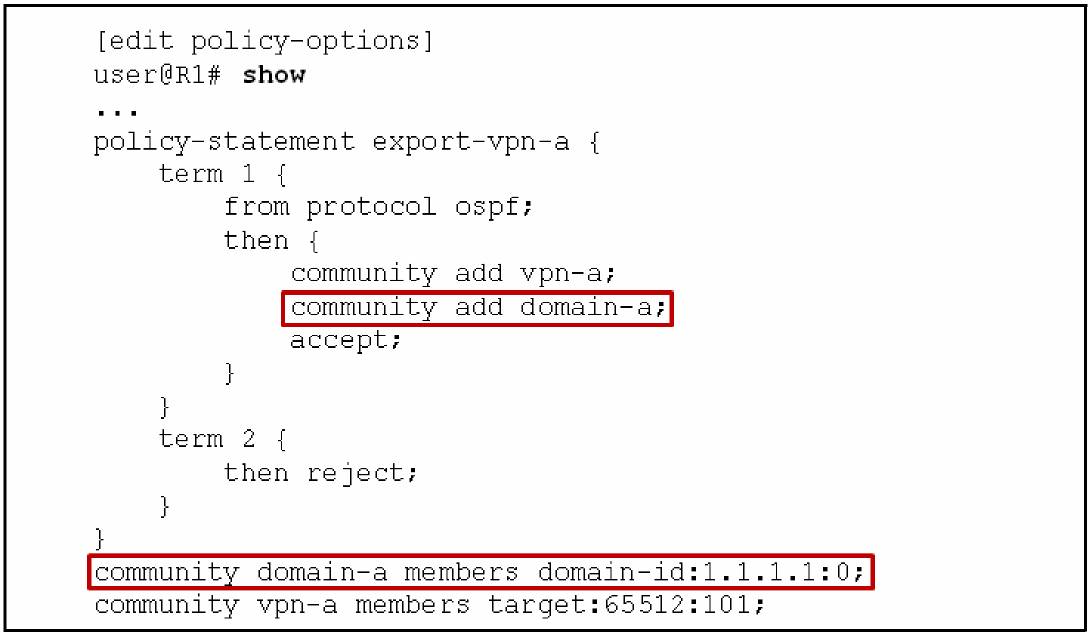
```
[edit routing-instances vpn-a]
user@R1# show
instance-type vrf;
interface ge-1/0/4.0;
route-distinguisher 192.168.1.1:1;
vrf-import import-vpn-a;
vrf-export export-vpn-a;
routing-options {
    router-id 192.168.11.3;
}
protocols {
    ospf {
        domain-id 1.1.1.1;
        export export-cust-a;
        area 0.0.0.0 {
            interface all;
        }
    }
}
```

This graphic shows the OSPF **domain-id** option in use in a VRF table configuration. If this PE router receives a VPN route that has a matching domain ID community and an internal route type, it generates a network summary LSA to the attached CE router. The trailing 0 is the default assigned number; it is not shown explicitly configured in this graphic.

This example also shows how you can configure the RID the PE router uses in the LSAs it generates. Even though the OSPF RID does not have to be *pingable*, many technicians are accustomed to being able to ping the RID of an OSPF router. Because the customer site normally does not carry provider routes, the default PE router action of sourcing its RID from its loopback address can result in the inability to ping the RID.

Where wanted, you can configure a RID from the customer's address space to be used within a particular OSPF instance. Assign the value to the VRF instance carefully, as the RID is unique within the OSPF domain. RIDs do not have to be reachable, but they must be unique within a routing domain. Current versions of the Junos OS, source the RID from the PE router's VRF interface (as opposed to the `lo0` interface), making the explicit setting of the RID purely a matter of choice.

Domain ID Policy

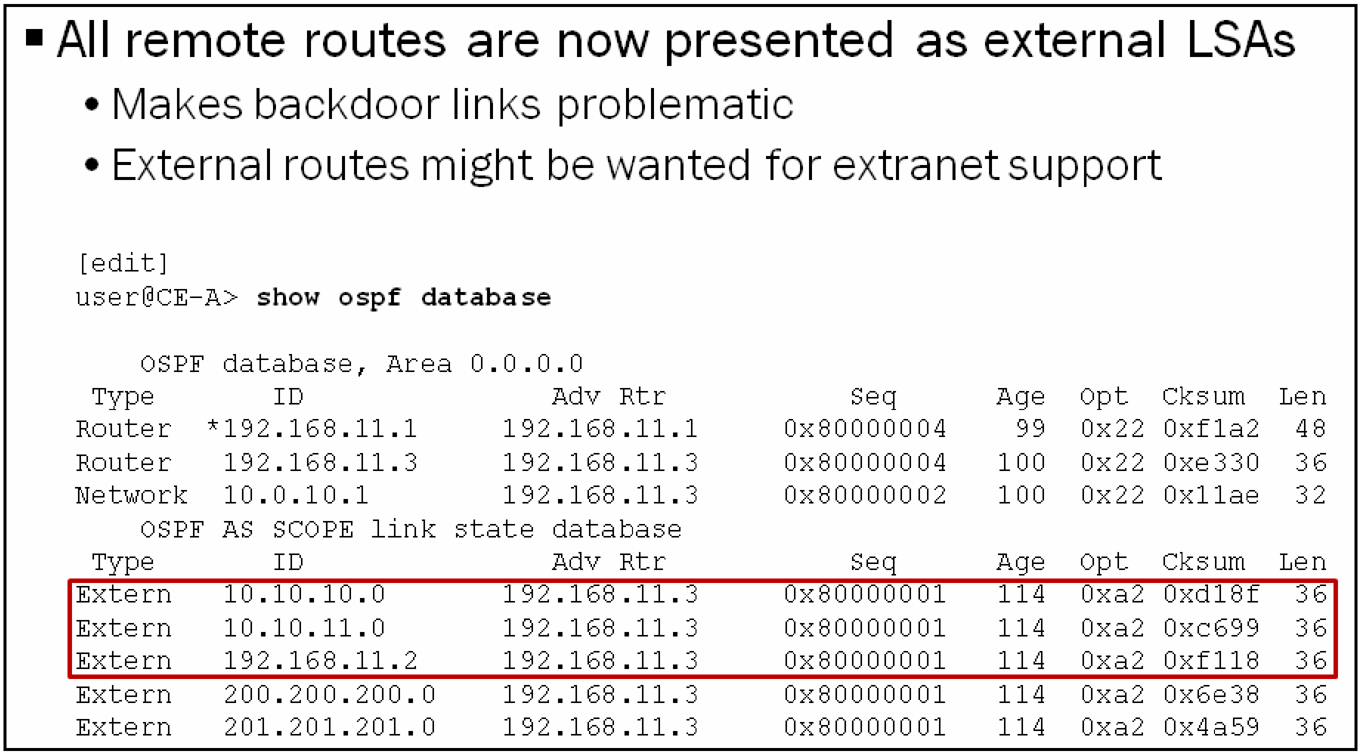


```
[edit policy-options]
user@R1# show
...
policy-statement export-vpn-a {
    term 1 {
        from protocol ospf;
        then {
            community add vpn-a;
            community add domain-a;
            accept;
        }
    }
    term 2 {
        then reject;
    }
}
community domain-a members domain-id:1.1.1.1:0;
community vpn-a members target:65512:101;
```

This graphic shows the policy configuration needed to support the OSPF domain ID. The definition of the `domain-id` community and the VRF export policy causes this community to be attached to the routes sent to the remote PE routers.

When defining the `domain-id` community as a member of a named community, you must include the assigned number portion, even when the default value of 0 is wanted.

Mismatched Domain ID Produced Externals



- All remote routes are now presented as external LSAs
  - Makes backdoor links problematic
  - External routes might be wanted for extranet support

```
[edit]
user@CE-A> show ospf database

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *192.168.11.1     192.168.11.1     0x80000004    99  0x22 0xf1a2  48
Router   192.168.11.3     192.168.11.3     0x80000004   100  0x22 0xe330  36
Network  10.0.10.1        192.168.11.3     0x80000002   100  0x22 0x11ae  32
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.10.10.0       192.168.11.3     0x80000001   114  0xa2 0xd18f  36
Extern   10.10.11.0       192.168.11.3     0x80000001   114  0xa2 0xc699  36
Extern   192.168.11.2     192.168.11.3     0x80000001   114  0xa2 0xf118  36
Extern   200.200.200.0    192.168.11.3     0x80000001   114  0xa2 0x6e38  36
Extern   201.201.201.0    192.168.11.3     0x80000001   114  0xa2 0x4a59  36
```

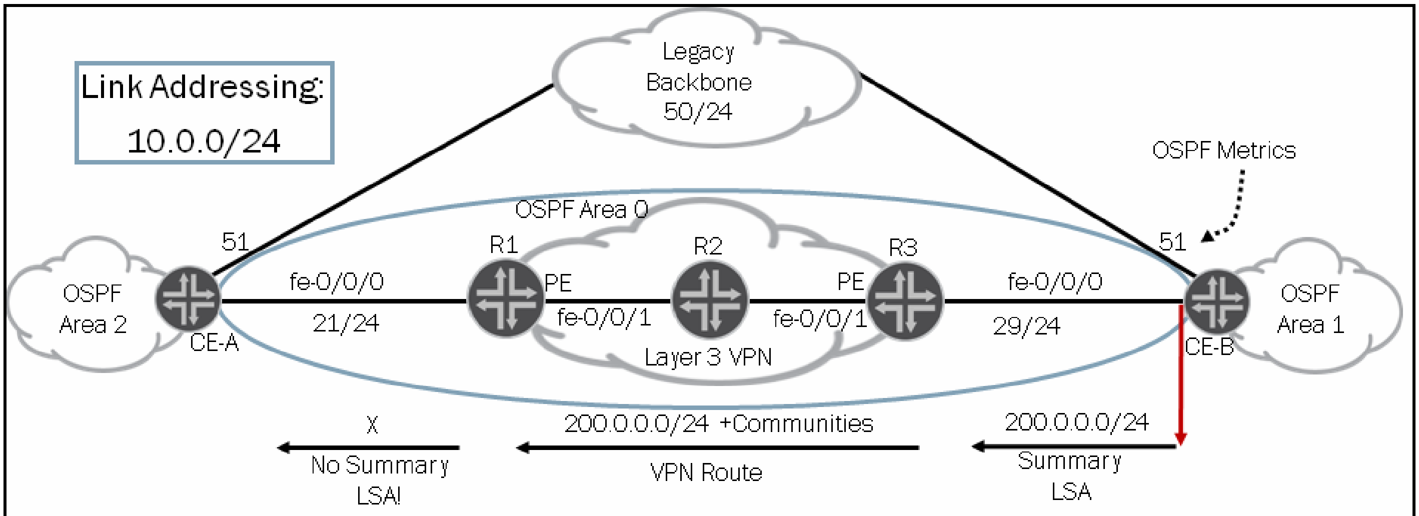
This graphic shows the results of OSPF domain ID use. You should compare this graphic to the basic OSPF configuration results section previously covered for maximum effect.

Looking at CE-A's OSPF database, we now see the CE-B's OSPF internal routes (the 192.168.11.2 loopback address, the 10.10.10/24, and the 10.10.11/24 OSPF interface routes) are now being presented to the CE router as OSPF externals (Type 5s).

You can see the automatically generated VPN route tag in the capture below. Here, the PE router's AS number of 65512 is treated as a hexadecimal value (FF E4), which is displayed in dotted decimal notation (255.232). The high-order 16 bits of the VPN route tag are populated with a 0x D0 00 (as per RFC 4577), which is shown as a 208.0 in dotted decimal notation:

```
user@CE-A> show ospf database external detail
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern   10.10.10.0       192.168.11.3     0x8000001b   632  0xa2 0x9da9  36
  mask 255.255.255.0
  Topology default (ID 0)
    Type: 1, Metric: 2, Fwd addr: 0.0.0.0, Tag: 208.0.255.232
...
```

Backdoor Link: Using Domain ID



- Backdoor case study
  - CE-A forwards traffic to 200.0.0.0/24 over the legacy backbone with a metric of 51
    - Downing the legacy backbone causes CE-A to use the Layer 3 backbone, now with a metric of 3
  - R1 router does not generate a summary LSA for 200.0.0/24 when the legacy backbone is operational

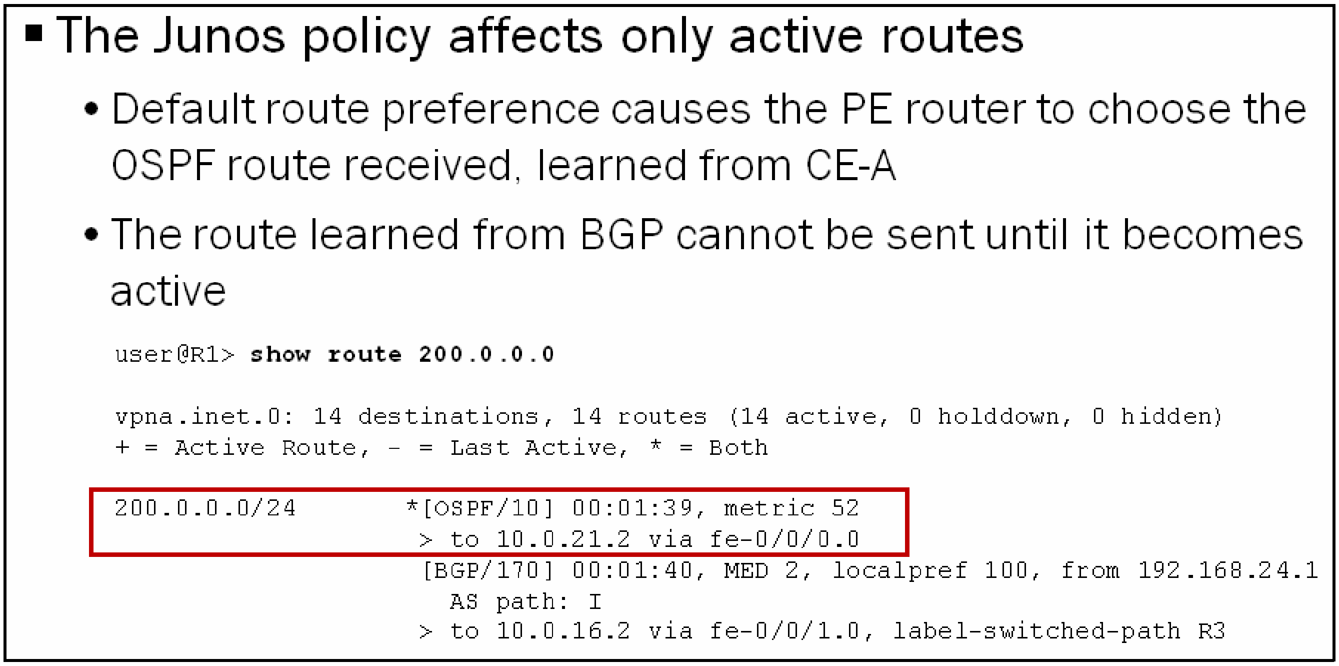
This graphic provides an example of how to use the OSPF domain ID to support backdoor links, which, with the domain ID, allow the control of traffic flow using metric adjustments. This example looks at how CE-A routes information to the 200.0.0/24 prefix, which is associated with an interface on CE-B attached to its OSPF Area 1. The metrics in this example are set in such a way that CE-A should be routing packets addressed to 200.0.0/24 over the Layer 3 VPN backbone.

After committing all changes, we can make the following observations:

- *CE-A is not using the Layer 3 VPN backbone*: Despite the use of OSPF domain ID and the metric setting shown, CE-A is routing to the 200.0.0/24 prefix over the legacy backbone. While it would be easy to assume that the Layer 3 VPN is simply *broken*, CE-A nonetheless forwards data over the Layer 3 VPN backbone when the legacy backbone is taken down. This fact indicates that the problem does not lie in the operational status of the VPN backbone itself.
- *R1 PE router is not generating summaries*: When the legacy backbone is operational, the R1 PE router does not generate a summary LSA for the 200.0.0/24 prefix. When the legacy backbone is taken out of service, the R1 PE router generates the summary LSA. Thus, this does not appear to be a domain ID problem either.

Any ideas?

Policy Only Affects Active Routes!



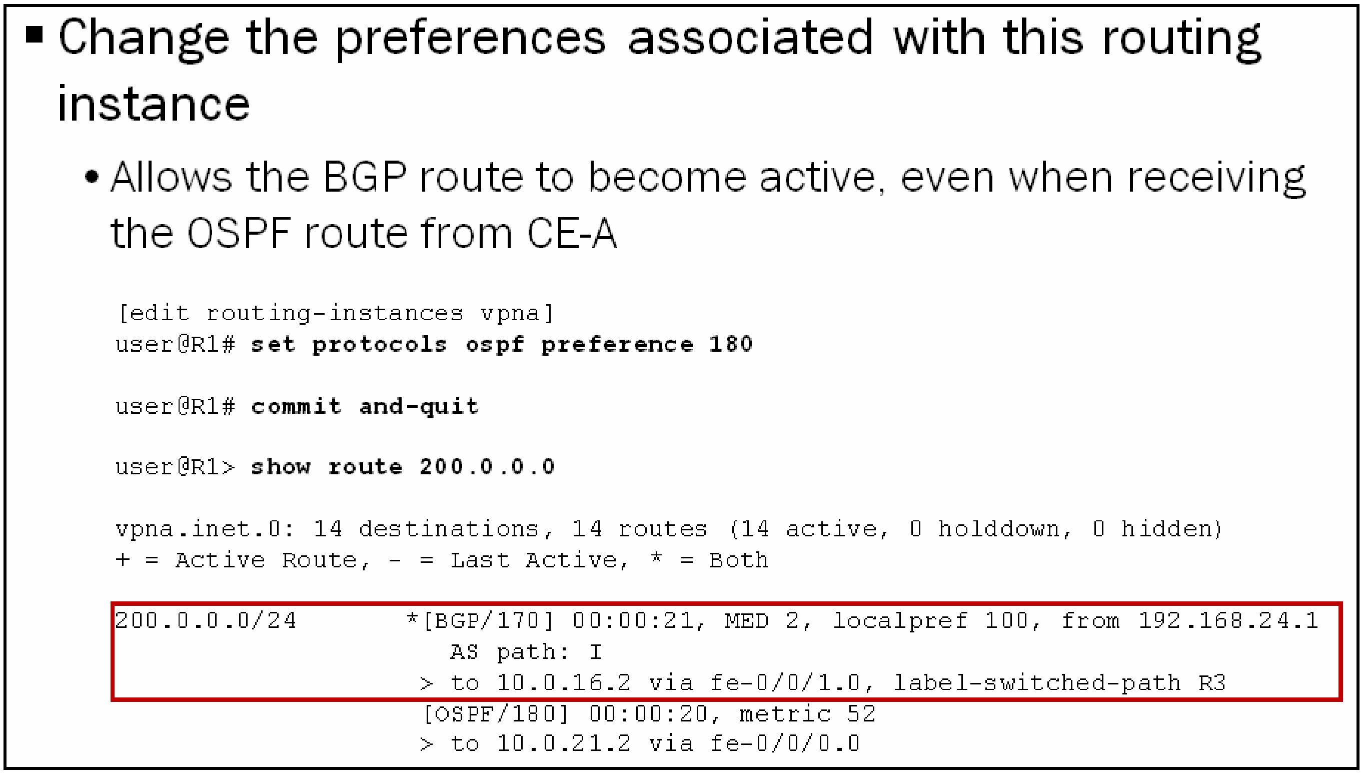
- The Junos policy affects only active routes
  - Default route preference causes the PE router to choose the OSPF route received, learned from CE-A
  - The route learned from BGP cannot be sent until it becomes active

```
user@R1> show route 200.0.0.0

vpna.inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200.0.0.0/24       *[OSPF/10] 00:01:39, metric 52
                    > to 10.0.21.2 via fe-0/0/0.0
                    [BGP/170] 00:01:40, MED 2, localpref 100, from 192.168.24.1
                      AS path: I
                    > to 10.0.16.2 via fe-0/0/1.0, label-switched-path R3
```

Issuing a **`show route 200.0.0.0`** command on the R1 PE router when the legacy backbone is operating provides a vital clue to finding the solution for this problem. The display indicates that the BGP route received from the remote PE router (R3) is not active because the PE router is receiving the same route from the attached CE router through OSPF. Because the Junos OS global route preference prefers OSPF over BGP, the BGP route is inactive whenever the legacy backbone is operating. This inactive BGP route prevents the R1 PE router from generating the summary LSA for the 200.0.0/24 prefix.

Therefore, to fix this problem, you must alter the default Junos OS behavior so that it prefers BGP routes over OSPF routes. The key is how to do this while only affecting this routing instance. A major change such as this, if made to the main routing instances, can result in unanticipated behavior and operational problems.

Change Route Preferences for This Routing Instance



- Change the preferences associated with this routing instance
  - Allows the BGP route to become active, even when receiving the OSPF route from CE-A

```
[edit routing-instances vpna]
user@R1# set protocols ospf preference 180

user@R1# commit and-quit

user@R1> show route 200.0.0.0

vpna.inet.0: 14 destinations, 14 routes (14 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

200.0.0.0/24       *[BGP/170] 00:00:21, MED 2, localpref 100, from 192.168.24.1
                      AS path: I
                    > to 10.0.16.2 via fe-0/0/1.0, label-switched-path R3
                    [OSPF/180] 00:00:20, metric 52
                    > to 10.0.21.2 via fe-0/0/0.0
```

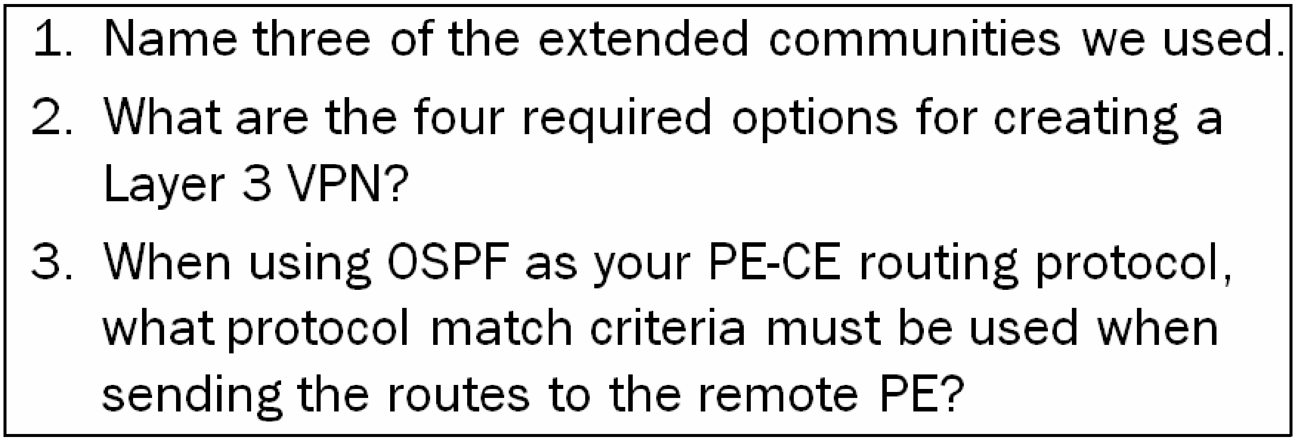
The graphic shows a solution to this problem. We have changed the preference for OSPF such that it is now higher (and therefore less preferred) than the default BGP preference of 170. This change of preference was configured under the routing options portion of a particular VRF table, so the operation of the main instance routing protocols is unaffected.

The **show route** command executed after the change in configuration is committed indicates that, as planned, the BGP route is now active, despite the continued presence of the OSPF route being learned from CE-A.

The end result is a functional Layer 3 VPN based on OSPF routing that yields the ability to force traffic onto one backbone or the other by adjusting the interface metrics on the area border routers (ABRs) (CE routers).

Note: If you break the link between CE-B and the R3 router, the R3 router drops the BGP route to the R1 router. Then, the R1 router sees that the active route now is an OSPF route. The R1 router then advertises a BGP route for a network in OSPF Area 1 (call it Net-A) to the R3 router. If the link from R3 to CE-B is repaired, the R3 router now has a BGP route for Net-A from the R1 router. This causes the OSPF route Net-A to *not* become the active route (because OSPF preference is now 180). To fix this issue, ensure that the PE router's OSPF-to-IBGP export policy matches OSPF routes originated from the local site and rejects any OSPF routes heard through the legacy backbone.

Review Questions



1. Name three of the extended communities we used.
2. What are the four required options for creating a Layer 3 VPN?
3. When using OSPF as your PE-CE routing protocol, what protocol match criteria must be used when sending the routes to the remote PE?

Answers to Review Questions

1.

We used the **`target`**, **`origin`** and **`domain-id`** extended communities.

2.

The four required options for creating a Layer 3 VPN instance are **`instance-type`**, **`interfaces`**, **`route-distinguisher`**, and **`vrf-target`** or **`vrf-import`**/**`vrf-export`** policies.

3.

The protocol match criteria required when exporting OSPF routes across your Layer 3 VPN to a remote PE is OSPF.

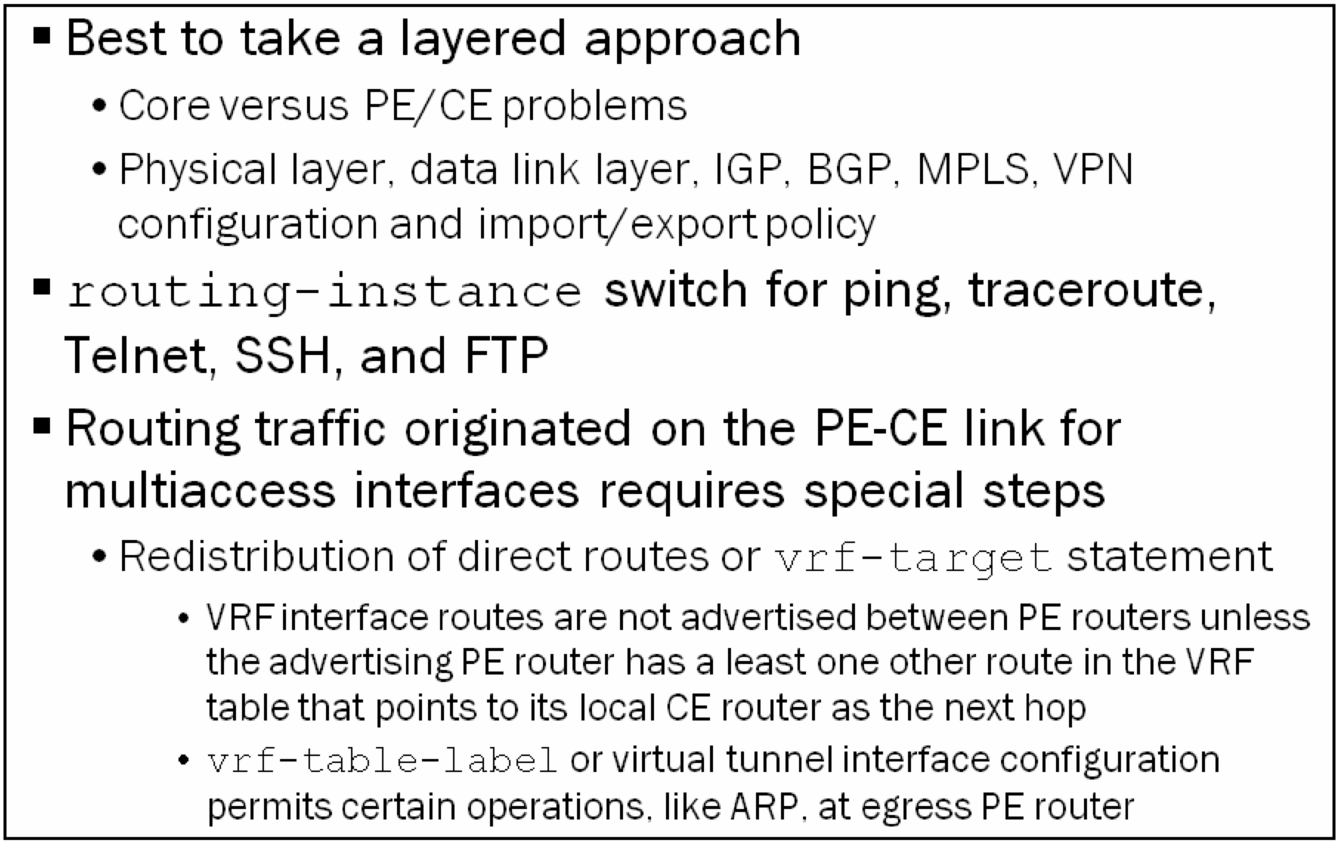
JNCIS-SP Study Guide—Part 3

Chapter 9: Troubleshooting Layer 3 VPNs

This Chapter Discusses:

- The `routing-instance` switch;
- Issues with the support of traffic originating on multiaccess virtual private network (VPN) routing and forwarding table (VRF) interfaces;
- Using operational commands to view Layer 3 VPN control exchanges;
- Using operational commands to display Layer 3 VPN VRF tables; and
- Monitoring and troubleshooting provider edge (PE)-customer edge (CE) routing protocols.

Taking a Layered Approach



- Best to take a layered approach
  - Core versus PE/CE problems
  - Physical layer, data link layer, IGP, BGP, MPLS, VPN configuration and import/export policy
- `routing-instance` switch for ping, traceroute, Telnet, SSH, and FTP
- Routing traffic originated on the PE-CE link for multiaccess interfaces requires special steps
  - Redistribution of direct routes or `vrf-target` statement
    - VRF interface routes are not advertised between PE routers unless the advertising PE router has a least one other route in the VRF table that points to its local CE router as the next hop
    - `vrf-table-label` or virtual tunnel interface configuration permits certain operations, like ARP, at egress PE router

Any number of configuration and operational problems can result in a dysfunctional VPN. With this much complexity, we recommend taking a layered approach to the provisioning and troubleshooting of Layer 3 VPN services.

"Is the problem core or PE-CE related?" and *"Are my pings failing because an interface is down or because a constrained path LSP cannot be established?"* are the types of questions you might ask yourself when faced with a Layer 3 VPN problem. Luckily, Layer 3 VPNs have several natural boundaries that allow for expedient problem isolation. As an example, consider a call reporting that three different VPNs on two different PE routers are down. Here, look for core-related issues (the P routers are common to all VPNs) rather than looking for PE-CE VRF-related problems at the sites reporting problems.

The `routing-instance` Switch

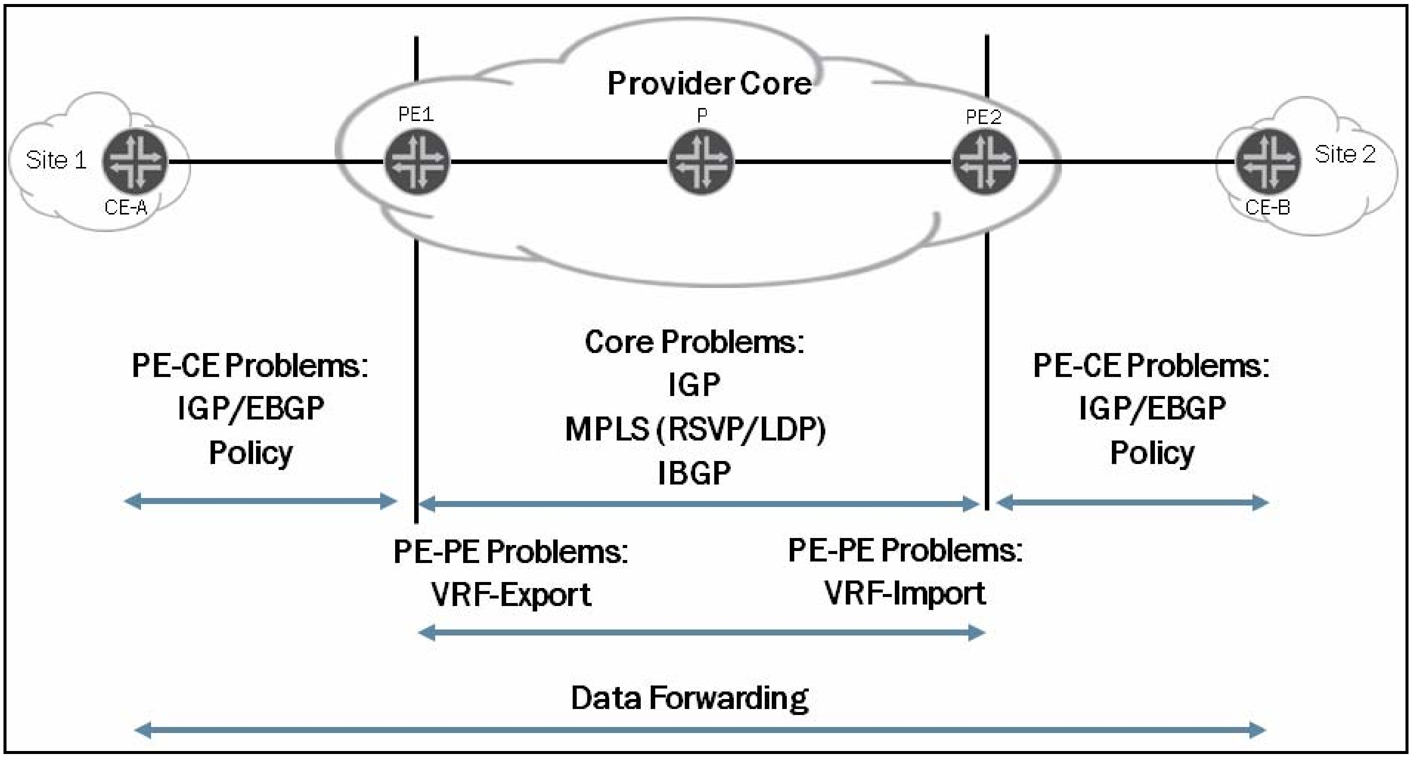
Because the main routing table does not store VRF interface routes, the simple act of pinging a directly attached CE device can prove difficult. The `routing-instance` switch tells the router which VRF table to consult when attempting to route a packet. This information is provided as an argument to commands such as **`ping`**, **`telnet`**, **`ssh`**, and **`ftp`**. Forgetting to use this switch or specifying the wrong VRF table as an argument results in a `no route to host` error message, which can throw technicians off the effective troubleshooting path.

Multiaccess Interfaces

By default, to enhance security on multiaccess interfaces (FE and GE), routers running the Junos operating system do not advertise directly connected routes related to a VRF interface unless at least one route exists in the local VRF table that points to the CE device as a next hop. This is because routers running the Junos OS cannot perform an Address Resolution Protocol (ARP) operation after receiving a labeled packet destined for a multi-access VRF table. However, no special steps are needed to support the traffic originating or terminating behind the CE router at the customer site. This issue only affects traffic being sourced or delivered to the VRF interface itself, such as when a ping is issued from one CE router to the VRF address of a remote CE router.

You can make the Internet Processor II functions available at the egress PE router with use of the `vrf-table-label` option or with a VPN tunnel interface, configured under the VRF instance. Subsequent pages provide more detail on these configuration options.

Keep It Simple



This graphic shows some of the functional boundaries useful in the fault isolation process in a Layer 3 VPN. By verifying the operation of each smaller piece, managing the overall task of troubleshooting the VPN is easier.

A classic example of this layered methodology is the clean separation of problems that can occur in the provider's core versus those associated with the PE-CE VRF interface and protocol exchanges. Because detailed analysis of VRF tables and VRF policy does not fix MPLS LSP issues in the core, you must be able to ascertain quickly if a problem is caused by the provider's infrastructure or if it is related to VPN-specific provisioning in a PE router.

Once you narrow down the nature of the problem to core versus PE-CE, you should continue the modular approach when determining the exact nature of the fault. For example, the lack of a CE route in the attached PE router's VRF table could be caused by physical layer, data link layer, network layer, protocol configuration, or routing policy. Realizing that being able to ping the CE device from the attached PE router generally validates the physical layer, data link layer, and network layer configuration of the local PE-CE VRF interface helps to narrow down further the possible causes for a problem.

Where Is the Route?

The majority of problems in a Layer 3 VPN relate to signaling. When the signaling runs smoothly, data forwarding is almost never a problem. It is a good idea to trace a route all the way from the originating CE device to the receiving CE device. Because VPN routes must resolve to LSPs terminating on the advertising router, hidden routes are often the result of misprovisioned LSPs or network failures.

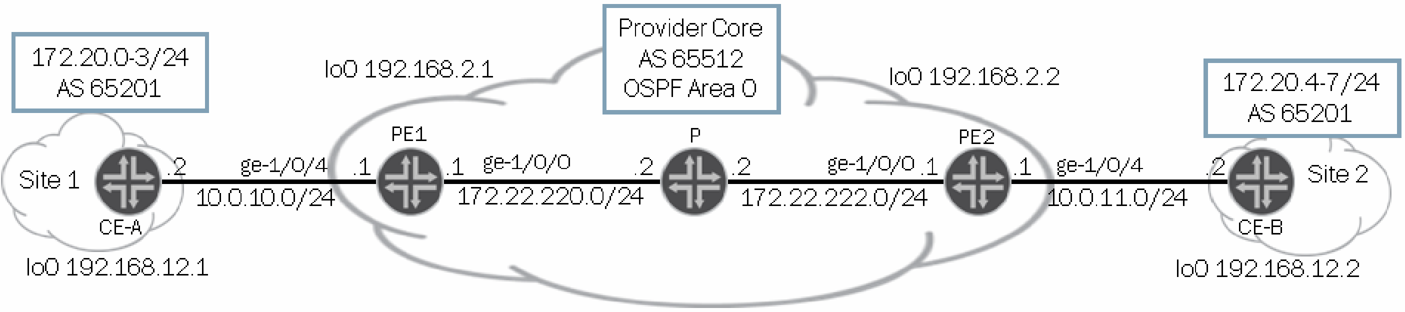
Having routes in the PE router but not in the local CE device (or vice versa) is likely the result of misconfigured routing protocols or policy errors.

Mismatched route targets are the primary cause of this symptom: *one PE router advertises a route while the receiving PE router does not react*.

Getting Started

The remaining pages in this chapter focus on the operational-mode commands critical to Layer 3 VPN troubleshooting and operational monitoring.

Sample Topology



- Network characteristics:
  - 192.168.x.y loopback addresses
  - IGP is single-area OSPF
  - RSVP signaling between PE devices, LSPs established between PE routers (CSPF not required)
  - Full MP-IBGP mesh between PE routers, loopback peering, VPN-IPv4 NLRI
  - CE-PE link running EBGP
  - Full-mesh Layer 3 VPN between CE-A and CE-B

The diagram in this graphic serves as the basis for the various configuration and operational-mode examples that follow.

All PE-CE physical interfaces use addresses from the 10.0/16 address space. The drawings show only the interfaces' subnet and host ID. Loopback addresses are assigned from the 192.168/16 address block.

The IGP in use is OSPF, and a single area (Area 0) is configured. Because the examples in this study guide do not rely on the functionality of CSPF, traffic engineering extensions need not be enabled.

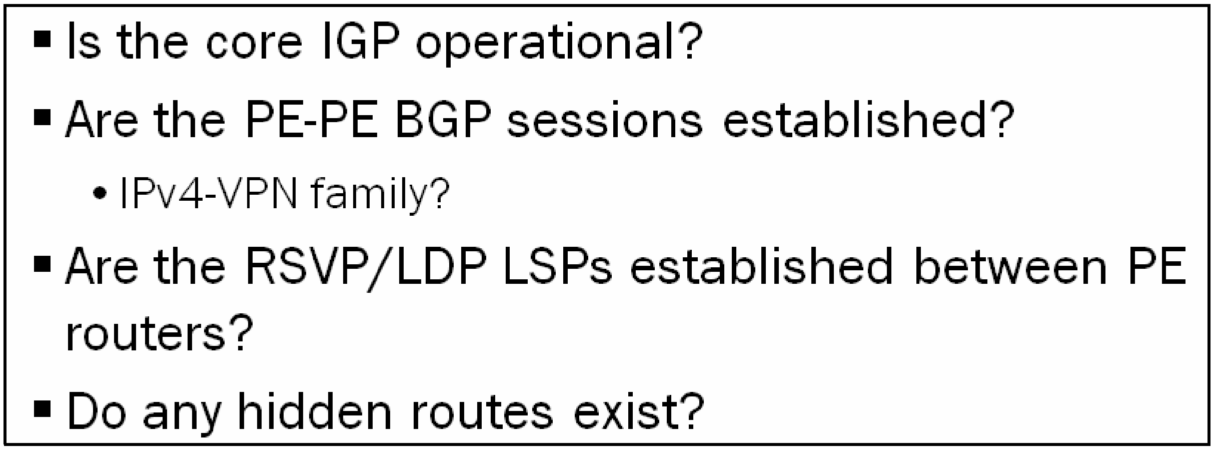
The MPLS signaling protocol is RSVP. LSPs are configured between the PE1 and PE2 routers.

An MP-IBGP peering session is configured between the loopback addresses of the PE routers. The VPN-IPv4 and `inet unicast` addresses families are configured.

In this example, the CE routers run EBGP, which results in the PE routers also needing to run BGP within their VRF routing instance.

The overall goal of this network is to provide full-mesh connectivity (which is point-to-point in this case) between the two CE routers. This network is considered full mesh because the resulting configuration readily accommodates additional sites while providing any-to-any connectivity.

Core IGP



- Is the core IGP operational?
- Are the PE-PE BGP sessions established?
  - IPv4-VPN family?
- Are the RSVP/LDP LSPs established between PE routers?
- Do any hidden routes exist?

LSP signaling protocols and the PE-PE MP-IBGP sessions must have a functional core IGP. You should always check the IGP when LSP or BGP session problems are evident. Generally, to verify IGP operation, look at routing tables and neighbor states (adjacencies), conduct ping and traceroute testing, and so forth.

PE-PE IBGP Sessions

Each PE router must have an MP-IBGP session established to all other PE routers connecting to sites forming a single VPN. If route reflectors are in use, all PE routers must have sessions established to all route reflectors serving the VPNs for which they have attached members. The `inet-vpn` family must be enabled on these sessions.

LSPs

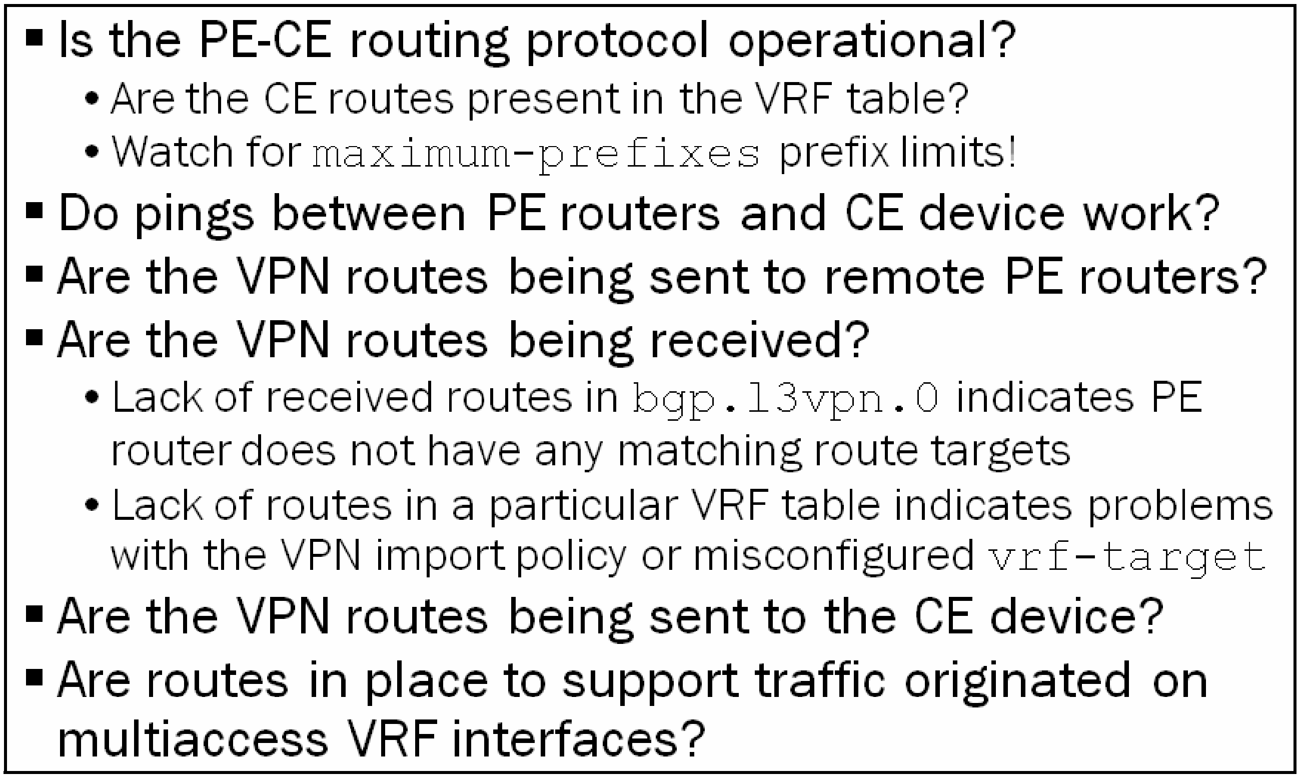
Each pair of PE routers sharing VPN membership must have LSPs established in both directions before traffic can be forwarded over the VPN. Lack of LSPs results in the VPN routes being hidden. When route reflection is in use, LSPs should be established from the route reflector to each PE router that is a client to ensure that hidden routes do not cause failure of the reflection process.

Got Hidden Routes?

Although sometimes hidden routes are the results of normal BGP route filtering, hidden routes in the context of VPNs generally indicate a problem in the prefix-to-LSP resolution process. VPN routes must resolve to an LSP in either the `inet.3` or `inet.0` routing table that egresses at the advertising PE router.

While the Junos OS normally keeps all loop-free BGP routes that are received (though kept, they might be hidden), this is not the case with VPN routes. A PE router that receives VPN updates with no matching route targets acts as if the update never happened. A change in VRF policy triggers a BGP route refresh, and, if you are lucky, the routes appear. When stumped, enable the `keep all` option to force the PE router to retain all BGP routes received. Once you perform fault isolation, turn off this option to prevent excessive resource use on the PE router.

PE-CE Routing Protocol



- Is the PE-CE routing protocol operational?
  - Are the CE routes present in the VRF table?
  - Watch for `maximum-prefixes` prefix limits!
- Do pings between PE routers and CE device work?
- Are the VPN routes being sent to remote PE routers?
- Are the VPN routes being received?
  - Lack of received routes in `bgp.l3vpn.0` indicates PE router does not have any matching route targets
  - Lack of routes in a particular VRF table indicates problems with the VPN import policy or misconfigured `vrf-target`
- Are the VPN routes being sent to the CE device?
- Are routes in place to support traffic originated on multiaccess VRF interfaces?

The operation of the PE-CE routing protocol is an excellent place to start when dealing with suspected PE-CE or VRF interface problems. The presence of link-state adjacencies or established BGP sessions generally indicates that the PE-CE connection is, for the most part, operational. The presence of routes in the local PE router's VRF table is also a good sign that PE-CE routing policy is not indiscriminately tossing out route advertisements.

You can use the `maximum-prefixes` option to tell the PE router to generate log messages, and, if desired, to stop accepting routes from the CE device when the configured parameters have been exceeded. When you use this option, you should check the system log for indications that the route limit has been exceeded to simplify the troubleshooting process. Below is the syntax of this command; the threshold option determines the fill percentage that triggers log messages:

```
[ edit routing-instances name routing-options ]
user@host# set maximum-prefixes route limit [ log-only | { threshold <1-100> } ]
```

PE-CE Ping Testing

If you suspect PE-CE routing problems, you should ping from the PE router to the local CE device. Success requires use of the `routing-instance` switch. When pings fail, check the operation of the physical layer and data link layer, as well as the PE-CE network layer settings.

Are the Routes Being Sent to Remote PE Routers?

If the CE device's routes are in the local PE router's VRF table, you might want to verify that the local PE router is sending the routes through MP-IBGP to the remote PE routers serving sites of this VPN. If the PE router is sending no routes, check the status of the PE-PE MP-IBGP session and the VRF's export policy.

Are Remote PE Routers Receiving the Routes?

Is this PE router receiving any VPN routes from the remote PE routers serving sites in the VPN? The `bgp.l3vpn.0` table stores the received VPN routes with matching route targets, where they are copied into the VRF tables that import based on the route target. Lack of entries could mean that the remote PE router is not advertising any routes or that the routes are being ignored due to lack of route target matches.

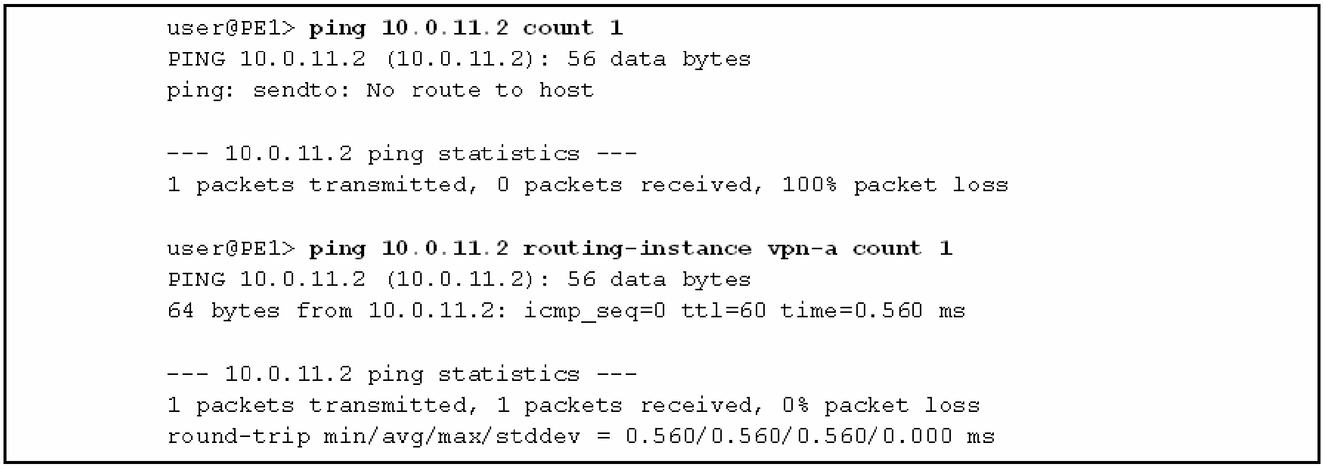
Are the Routes Being Sent to the Remote CE Device?

Once routes are confirmed in the remote PE router's VRF table, verify that these routes are in turn being sent over the VRF interface to the remote CE device. Problems here can relate to PE-CE routing policy or to the operational status of the PE-CE VRF interface and routing protocol.

Routes to Accommodate Multiaccess VRF Interfaces?

When using multiaccess VRF interfaces (Fast Ethernet/Gigabit Ethernet), PE-based redistribution of the connected VRF interface is required to support traffic originating or terminating on a VRF interface. Although you can configure a `vrf-target` or a `vrf-export` policy redistributing direct routes, the VRF table's interface routes sometimes might not get advertised. This is generally caused by not having any routes in the VRF table that were learned from the local CE router. When CE-to-CE pings fail, try sourcing the ping from a non-VRF interface (such as the loopback address) to confirm whether you are dealing with this multiaccess VRF table issue. To fix the multiaccess issue, ensure that the local CE router is advertising at least one route to the local PE router, the local PE router has a static route with the CE device as a next hop, `vrf-table-label` is configured, or a VPN tunnel interface is configured.

VRF Interface Routes Are Not Placed in **`inet.0`**



```
user@PE1> ping 10.0.11.2 count 1
PING 10.0.11.2 (10.0.11.2): 56 data bytes
ping: sendto: No route to host

--- 10.0.11.2 ping statistics ---
1 packets transmitted, 0 packets received, 100% packet loss

user@PE1> ping 10.0.11.2 routing-instance vpn-a count 1
PING 10.0.11.2 (10.0.11.2): 56 data bytes
64 bytes from 10.0.11.2: icmp_seq=0 ttl=60 time=0.560 ms

--- 10.0.11.2 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.560/0.560/0.560/0.000 ms
```

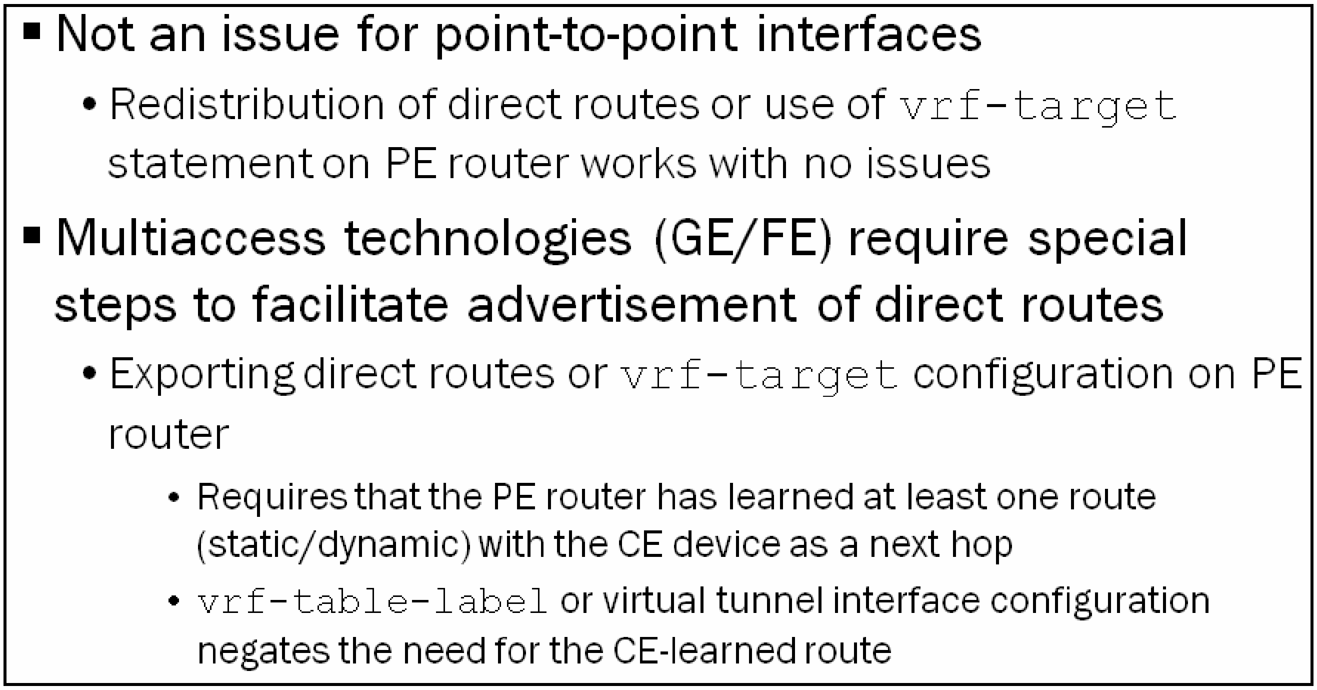
Once you configure an interface as part of a VRF instance, the interface route is no longer placed into the main routing table (`inet.0`). As a result, you must associate commands like **`ping`** and **`telnet`** with the correct VRF table to avoid no route to host error messages.

The **`routing-instance`** Switch

Use of the `routing-instance` switch causes the router to consult the VRF table associated with the interface name specified. This switch thereby enables the use of commands like **`ping`**, **`telnet`**, **`traceroute`**, **`ssh`**, and **`ftp`** in the context of a particular VPN.

On the previous graphic, the first ping attempt from the PE1 router to CE-A fails with a **`No route to host`** error message. When the operator includes the correct VPN instance as an argument to the **`ping`** command, the ping succeeds.

Point-to-Point = No Problem



- Not an issue for point-to-point interfaces
  - Redistribution of direct routes or use of `vrf-target` statement on PE router works with no issues
- Multiaccess technologies (GE/FE) require special steps to facilitate advertisement of direct routes
  - Exporting direct routes or `vrf-target` configuration on PE router
    - Requires that the PE router has learned at least one route (static/dynamic) with the CE device as a next hop
    - `vrf-table-label` or virtual tunnel interface configuration negates the need for the CE-learned route

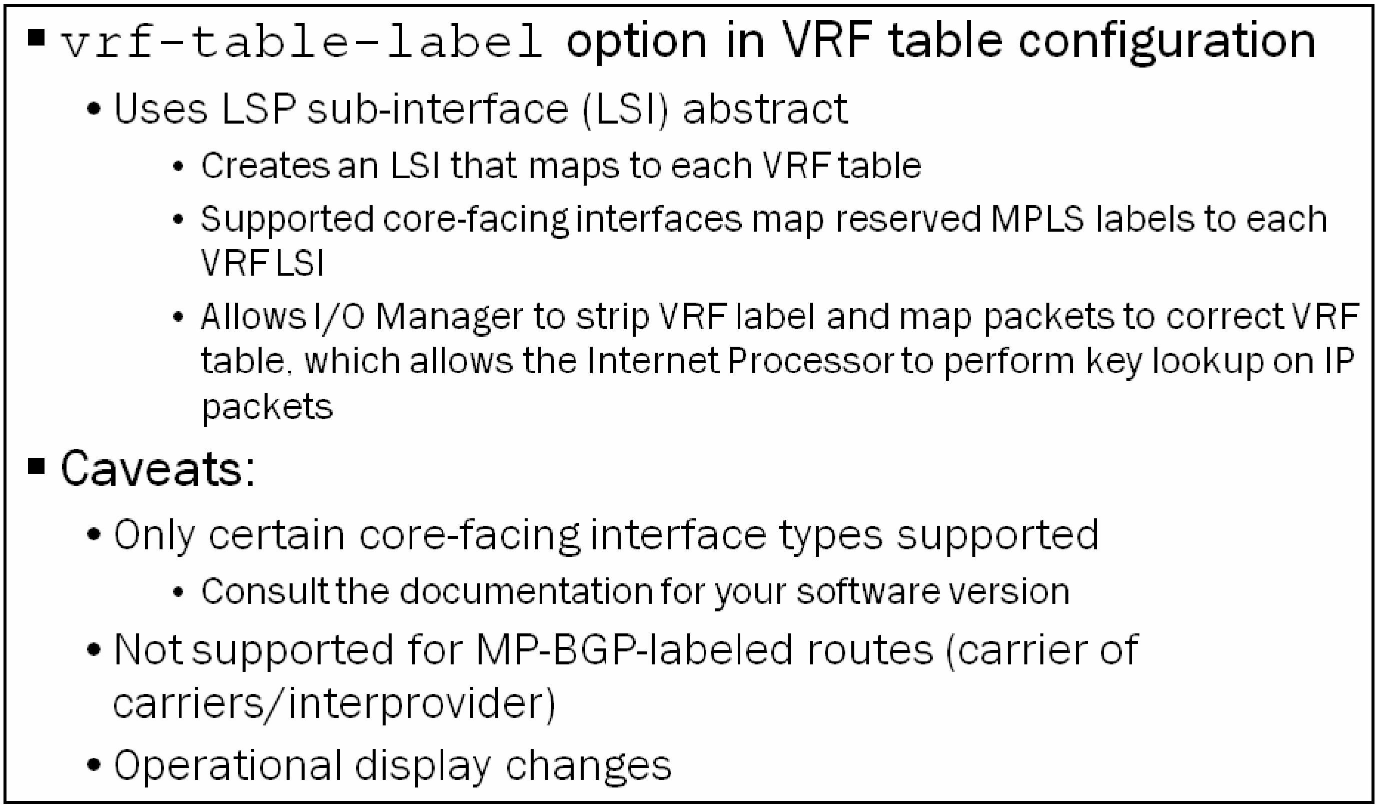
By default, PE routers running the Junos OS can advertise directly-connected VRF interface routes using export policy or the `vrf-target` statement. Because of this, there is no issue with the support of traffic either originating or terminating on point-to-point VRF interfaces.

Special Steps for Multiaccess VRF Interfaces

By default, PE routers running the Junos OS cannot advertise directly connected VRF interface routes using export policy or the `vrf-target` statement on multiaccess interfaces. For this type of configuration to work, the PE router must have learned, through static or dynamic routing, a route from the local CE device with that CE device as the next hop for the route. This is an inherent security feature due to the broadcast nature of FE/GE interfaces.

Without learning a route from the local CE device, another way of fixing this problem is to configure **`vrf-table-label`** on the local PE router. Use of this feature eliminates the need for the routes described above, because the Internet Processor II can now perform ARP operations as needed to determine the MAC address of the correct CE device.

`vrf-table-label` VRF Table Option



- `vrf-table-label` option in VRF table configuration
  - Uses LSP sub-interface (LSI) abstract
    - Creates an LSI that maps to each VRF table
    - Supported core-facing interfaces map reserved MPLS labels to each VRF LSI
    - Allows I/O Manager to strip VRF label and map packets to correct VRF table, which allows the Internet Processor to perform key lookup on IP packets
- Caveats:
  - Only certain core-facing interface types supported
    - Consult the documentation for your software version
  - Not supported for MP-BGP-labeled routes (carrier of carriers/interprovider)
  - Operational display changes

Juniper Networks, starting with Junos OS Release 5.2, added support for Internet Processor II functions for egress PE routers running the Junos OS with a new VRF table configuration option called `vrf-table-label`.

LSP Sub-Interfaces

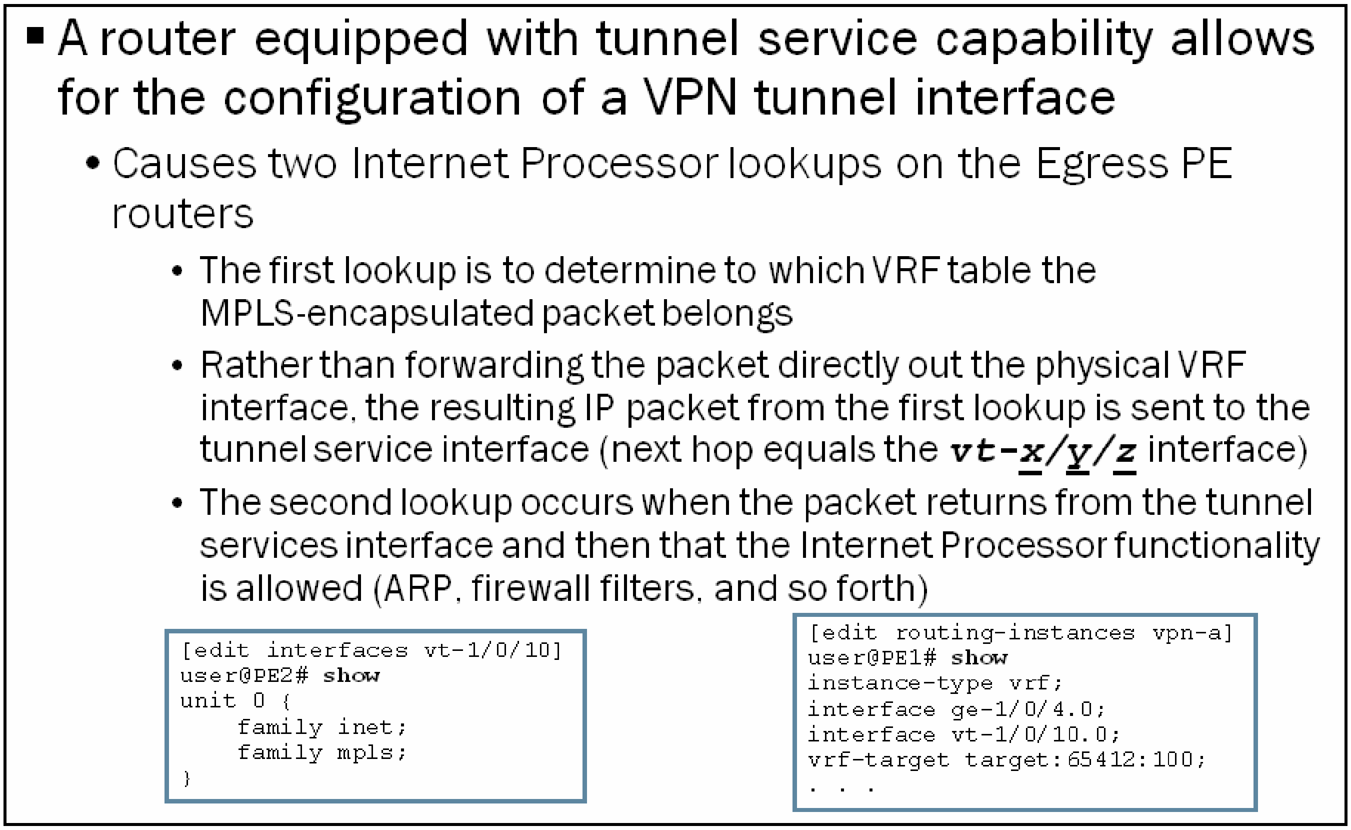
This feature makes use of the LSP sub-interface (LSI) abstract that allows an LSP to be treated as an interface. When you configure the `vrf-table-label` option under a VRF routing instance, an LSI is created for that VRF table. Supported core-facing interfaces assign a label from a special range (currently 1024–2048), which in turn is mapped to the VRF table's LSI.

The result is that the input FPC I/O Manager ASIC now can associate a packet with the correct VRF table, based on the label-to-LSI-to-VRF mapping. This association allows the VRF label to be stripped at the egress router so that the Internet Processor II can perform a key lookup on the IP packet itself. This key lookup supports Internet Processor II functions such as ARP generation, rate limiting, and firewall filtering.

Feature Restrictions

The `vrf-table-label` feature is only supported on certain core-facing interfaces types. Consult the documentation for your software version for a list of supported interface types. LSI-based labels are not used for MP-BGP label routes to avoid operational problems with carrier-of-carriers and interprovider applications. You can view LSIs with the **`show interfaces`** command. Also, a *dummy* route is added to the main `mpls.0` table, which shows the LSI-to-VRF mapping. This route is never used, however, because the inner label is now stripped at the input FPC I/O Manager ASIC.

VPN Tunnel Interface



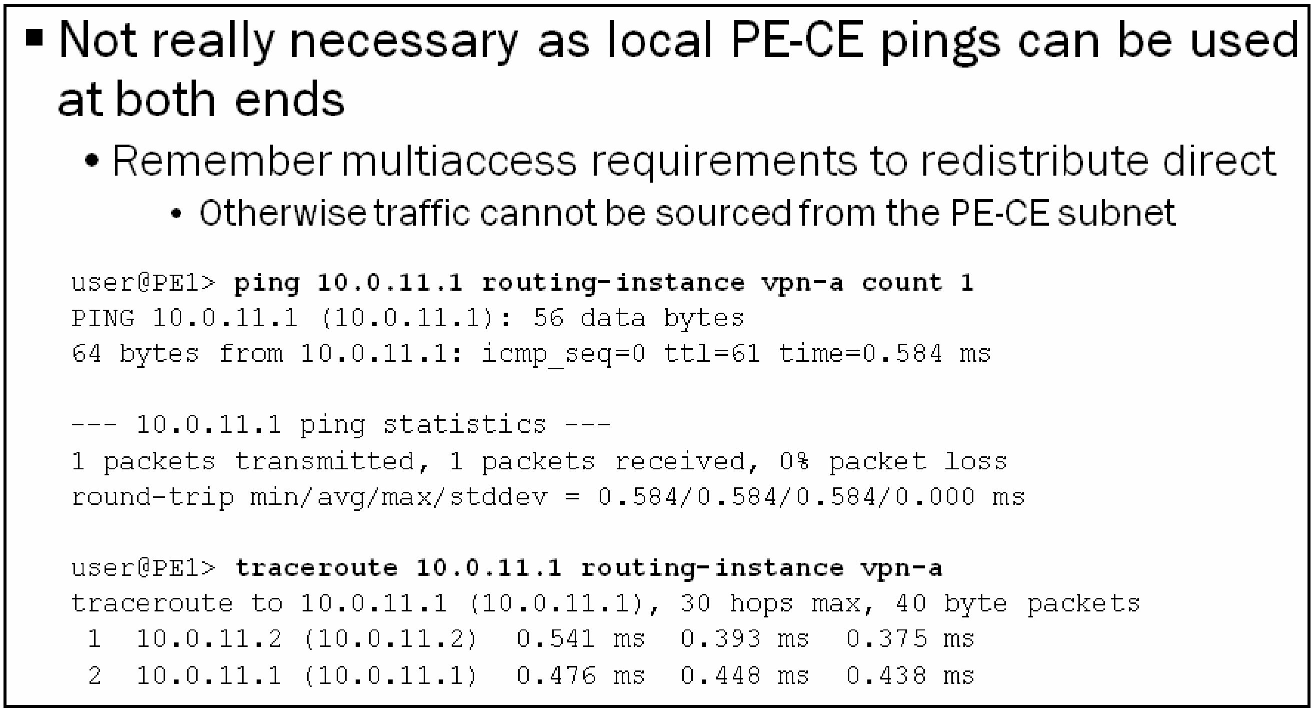
- A router equipped with tunnel service capability allows for the configuration of a VPN tunnel interface
  - Causes two Internet Processor lookups on the Egress PE routers
    - The first lookup is to determine to which VRF table the MPLS-encapsulated packet belongs
    - Rather than forwarding the packet directly out the physical VRF interface, the resulting IP packet from the first lookup is sent to the tunnel service interface (next hop equals the **vt-x/y/z** interface)
    - The second lookup occurs when the packet returns from the tunnel services interface and then that the Internet Processor functionality is allowed (ARP, firewall filters, and so forth)

```
[edit interfaces vt-1/0/10]
user@PE2# show
unit 0 {
    family inet;
    family mpls;
}
```

```
[edit routing-instances vpn-a]
user@PE1# show
instance-type vrf;
interface ge-1/0/4.0;
interface vt-1/0/10.0;
vrf-target target:65412:100;
. . .
```

To allow a PE router running the Junos OS to perform Internet Processor II functionality, you can configure a VPN tunnel interface for the egress VRF table. To configure a VPN tunnel interface, a router must have a tunnel services enabled. Tunnel services can be enabled in simple configuration on an MX Series Ethernet Services Router, while other routers require either a Tunnel Services or Adaptive Services PIC installed. The graphic shows the configuration steps for a VPN tunnel interface. With a VPN tunnel interface configured, the PE router pops the label stack as normal. Before passing the remaining packet to the CE device as it usually would, the packet is passed through a logical VPN tunnel interface. The packet is then sent back to the Internet Processor II from the VPN tunnel interface where the Internet Processor II performs the functions it could not on the first pass.

PE-to-PE VRF Interface Pings Are Optional



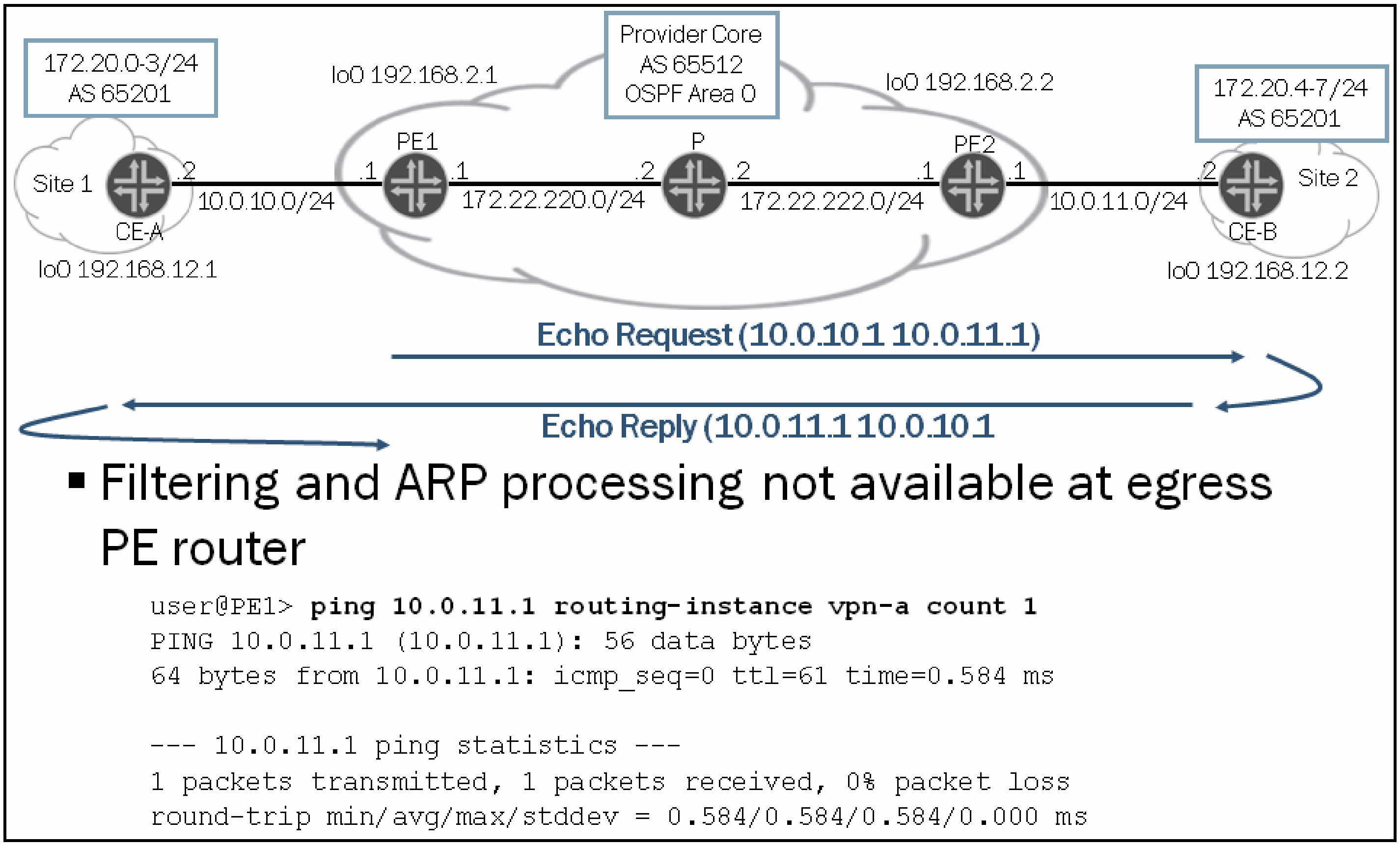
- Not really necessary as local PE-CE pings can be used at both ends
  - Remember multiaccess requirements to redistribute direct
    - Otherwise traffic cannot be sourced from the PE-CE subnet

```
user@PE1> ping 10.0.11.1 routing-instance vpn-a count 1
PING 10.0.11.1 (10.0.11.1): 56 data bytes
64 bytes from 10.0.11.1: icmp_seq=0 ttl=61 time=0.584 ms

--- 10.0.11.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.584/0.584/0.584/0.000 ms

user@PE1> traceroute 10.0.11.1 routing-instance vpn-a
traceroute to 10.0.11.1 (10.0.11.1), 30 hops max, 40 byte packets
 1  10.0.11.2 (10.0.11.2)  0.541 ms  0.393 ms  0.375 ms
 2  10.0.11.1 (10.0.11.1)  0.476 ms  0.448 ms  0.438 ms
```

Conducting PE-to-PE VRF interface pings is optional, because you can test the PE-CE links individually by performing local PE-CE pings from each PE router. You must redistribute the PE router's connected VRF interface routes while conducting PE-to-PE VRF interface pings when multi-access VRF interfaces are deployed.

Regardless of interface type, you must use the **`routing-instance`** switch to associate the traffic with the correct VRF instance on the PE router.

Cannot Process Twice



- Filtering and ARP processing not available at egress PE router

```
user@PE1> ping 10.0.11.1 routing-instance vpn-a count 1
PING 10.0.11.1 (10.0.11.1): 56 data bytes
64 bytes from 10.0.11.1: icmp_seq=0 ttl=61 time=0.584 ms

--- 10.0.11.1 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.584/0.584/0.584/0.000 ms
```

The design of routers running the Junos OS is optimized for the single-pass processing of transit traffic. As a result, the Internet Processor II ASIC cannot process the same packet twice when transiting the router. Because the Internet Processor II ASIC is used to process the labeled packet at ingress, the features of the Internet Processor II ASIC are not available for packet processing at the egress of the PE router unless you configure `vrf-table-label` or a VPN tunnel (`vt`) interface.

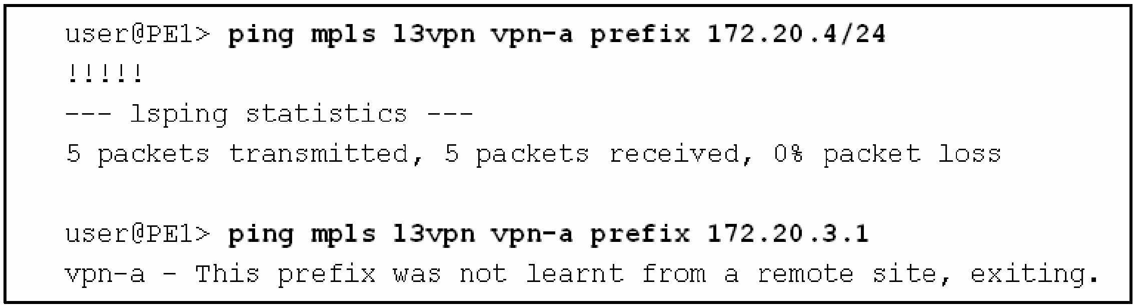
After popping the VRF label, the router must forward the packet out the associated VRF interface. The Juniper Networks implementation assigns VRF labels on a per-VRF interface basis, which accommodates this forwarding behavior.

An interesting side effect of this implementation is the rather convoluted path of a PE-to-PE VRF interface ping. Therefore, we recommend testing each PE-CE VRF connection using traffic sourced from the local PE router.

On this graphic, a ping generated by PE1 is addressed to the PE2 router’s VRF interface. Because PE2 cannot perform a route lookup after popping the VRF label, it must forward the packet to the attached CE router. The CE router recognizes that this packet is addressed to PE2, and it therefore sends the packet right back to PE2. On ingress, the Internet Processor II can process the incoming packet, so PE2 recognizes that it is the target and generates the ICMP echo-reply.

Similarly, at the PE1 end, when the echo-request is received, a PE-to-PE VRF interface ping only succeeds when all aspects of both the PE-CE VRF interfaces function correctly.

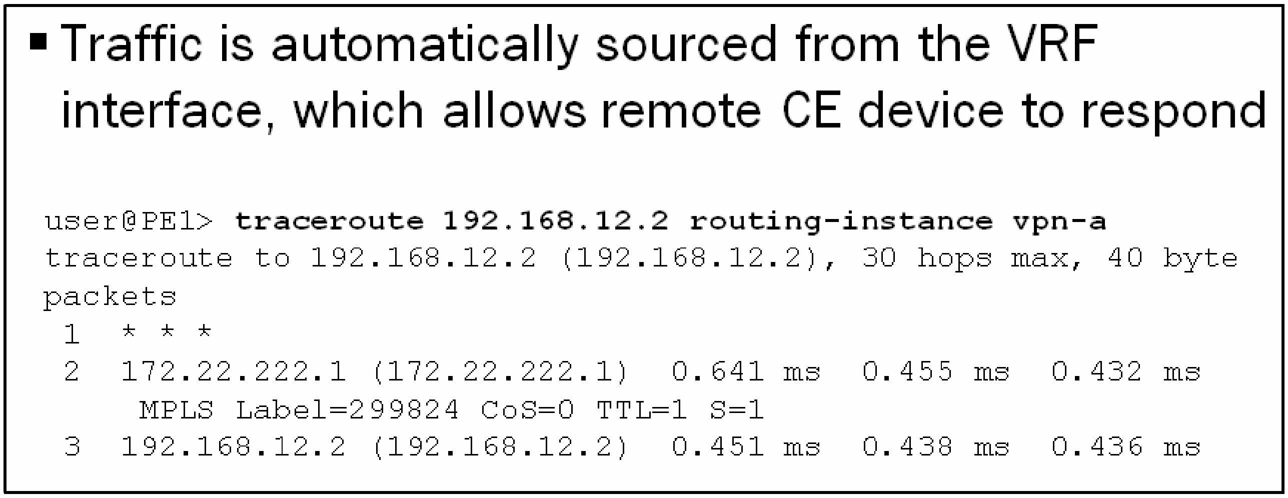
Testing PE-to-PE L3 VPN Connectivity



```
user@PE1> ping mpls l3vpn vpn-a prefix 172.20.4/24
!!!!!
--- lsping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss

user@PE1> ping mpls l3vpn vpn-a prefix 172.20.3.1
vpn-a - This prefix was not learnt from a remote site, exiting.
```

The **`ping mpls l3vpn`** ***`instance`*** **`prefix`** ***`prefix/length`*** command allows you to test two things. If the ping succeeds, as in the first example on the graphic, this proves that the MPLS LSP is up and also that the route to the destination prefix exists in the VRF table.

Tracing the Remote PE-CE VRF Interface

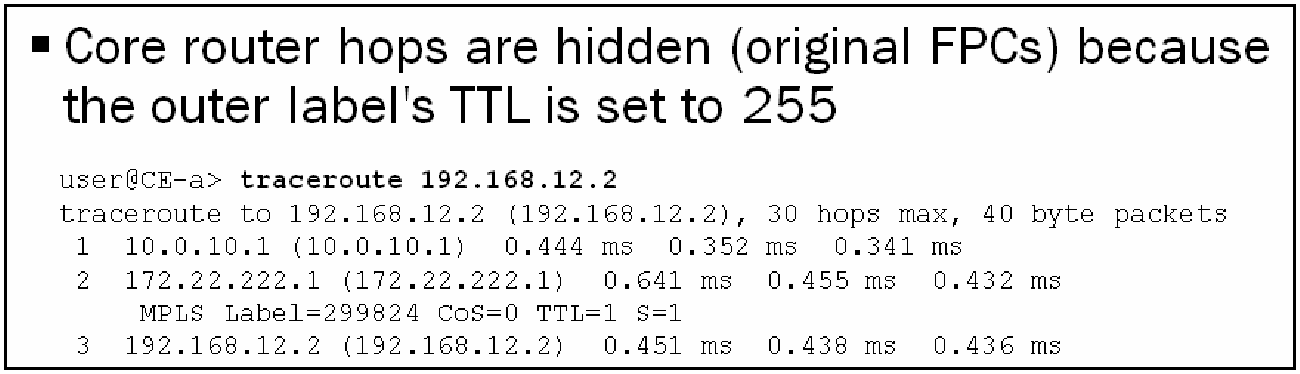


- Traffic is automatically sourced from the VRF interface, which allows remote CE device to respond

```
user@PE1> traceroute 192.168.12.2 routing-instance vpn-a
traceroute to 192.168.12.2 (192.168.12.2), 30 hops max, 40 byte
packets
 1  * * *
 2  172.22.222.1 (172.22.222.1)  0.641 ms  0.455 ms  0.432 ms
     MPLS Label=299824 CoS=0 TTL=1 S=1
 3  192.168.12.2 (192.168.12.2)  0.451 ms  0.438 ms  0.436 ms
```

Routers running the Junos OS automatically source pings and traceroutes from the VRF interface when using the **`routing-instance`** switch. Therefore, you should be able trace the route at the remote PE-CE VRF interface.

Core Hops with Original FPC

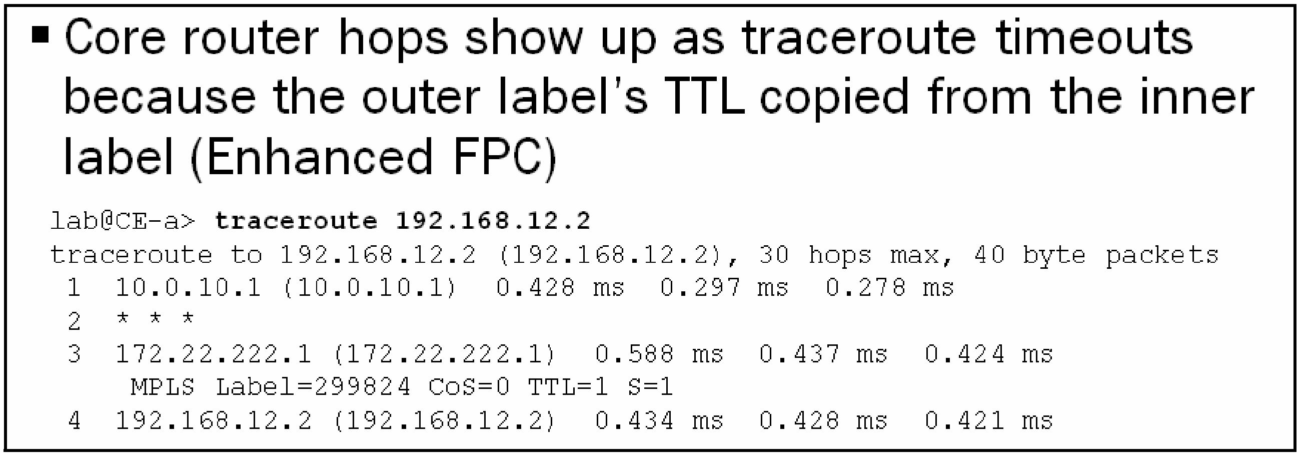


- Core router hops are hidden (original FPCs) because the outer label's TTL is set to 255

```
user@CE-a> traceroute 192.168.12.2
traceroute to 192.168.12.2 (192.168.12.2), 30 hops max, 40 byte packets
 1  10.0.10.1 (10.0.10.1)  0.444 ms  0.352 ms  0.341 ms
 2  172.22.222.1 (172.22.222.1)  0.641 ms  0.455 ms  0.432 ms
     MPLS Label=299824 CoS=0 TTL=1 S=1
 3  192.168.12.2 (192.168.12.2)  0.451 ms  0.438 ms  0.436 ms
```

The graphic shows the results of a CE-to-CE traceroute with the local PE router equipped with the original FPC. Core router hops are hidden due to the outer MPLS label having its TTL set to 255 for traffic received over the local VRF interface. A normal CE-CE traceroute, therefore, shows the ingress VRF interface, the remote PE router's core-facing interface (it can respond because the label associates the packet with the correct VRF table), and the attached CE router.

Core Hops with Enhanced FPC

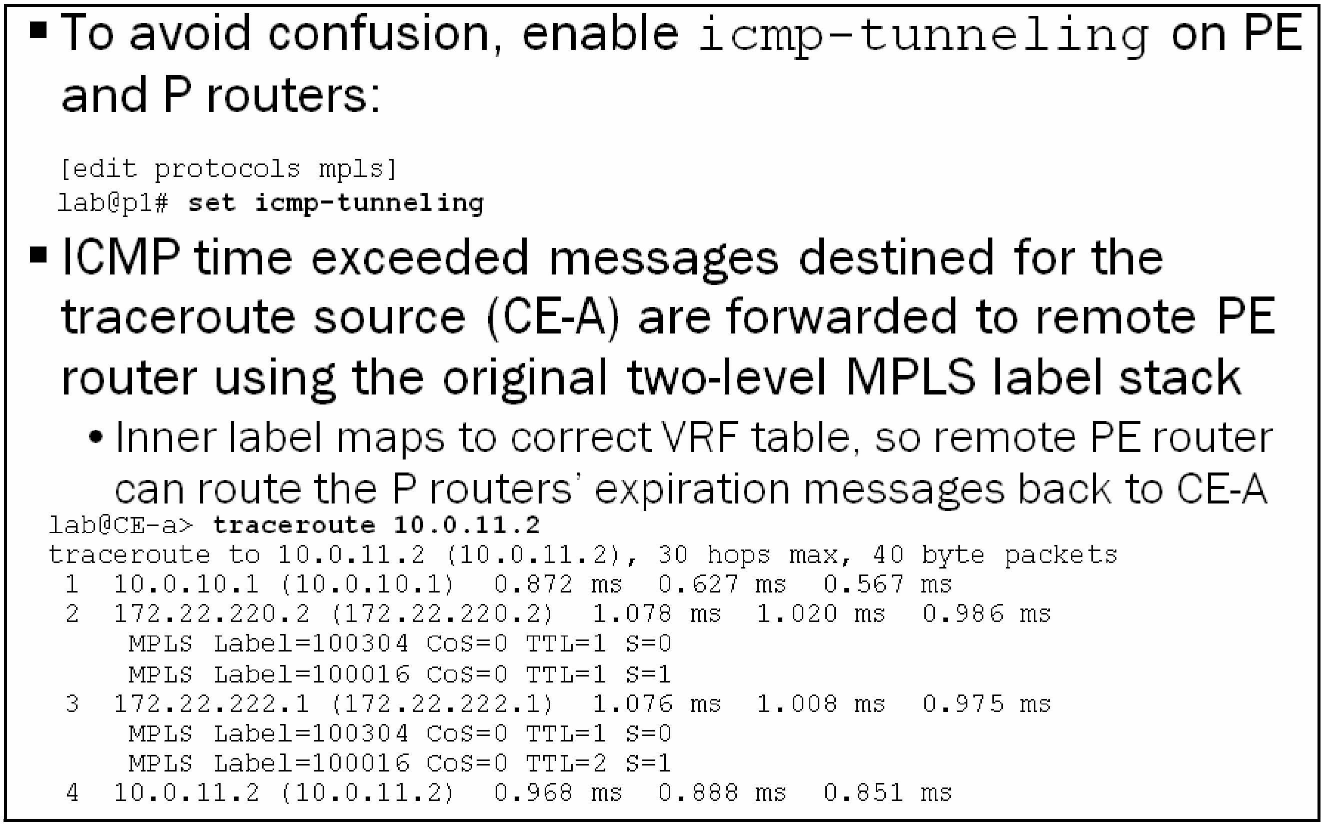


- Core router hops show up as traceroute timeouts because the outer label's TTL copied from the inner label (Enhanced FPC)

```
lab@CE-a> traceroute 192.168.12.2
traceroute to 192.168.12.2 (192.168.12.2), 30 hops max, 40 byte packets
 1  10.0.10.1 (10.0.10.1)  0.428 ms  0.297 ms  0.278 ms
 2  * * *
 3  172.22.222.1 (172.22.222.1)  0.588 ms  0.437 ms  0.424 ms
     MPLS Label=299824 CoS=0 TTL=1 S=1
 4  192.168.12.2 (192.168.12.2)  0.434 ms  0.428 ms  0.421 ms
```

If you have an Enhanced FPC the outer label TTL tracks the inner label's TTL for Layer 3 VPN tracerouting. This can result in confusion when performing traceroutes, because you can get timeouts where P routers cannot route ICMP time exceeded messages back to sources in the VPN.

Enabling **`icmp-tunneling`**



- To avoid confusion, enable `icmp-tunneling` on PE and P routers:

```
[edit protocols mpls]
lab@p1# set icmp-tunneling
```

- ICMP time exceeded messages destined for the traceroute source (CE-A) are forwarded to remote PE router using the original two-level MPLS label stack
  - Inner label maps to correct VRF table, so remote PE router can route the P routers' expiration messages back to CE-A

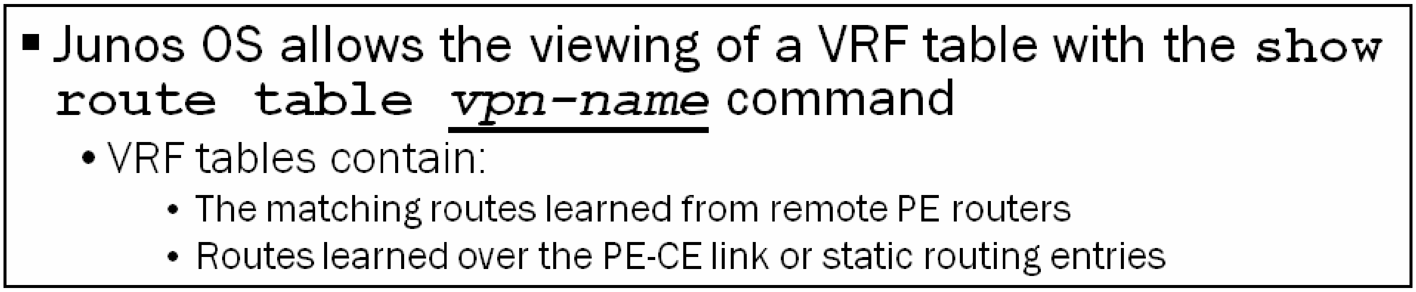
```
lab@CE-a> traceroute 10.0.11.2
traceroute to 10.0.11.2 (10.0.11.2), 30 hops max, 40 byte packets
 1  10.0.10.1 (10.0.10.1)  0.872 ms  0.627 ms  0.567 ms
 2  172.22.220.2 (172.22.220.2)  1.078 ms  1.020 ms  0.986 ms
     MPLS Label=100304 CoS=0 TTL=1 S=0
     MPLS Label=100016 CoS=0 TTL=1 S=1
 3  172.22.222.1 (172.22.222.1)  1.076 ms  1.008 ms  0.975 ms
     MPLS Label=100304 CoS=0 TTL=1 S=0
     MPLS Label=100016 CoS=0 TTL=2 S=1
 4  10.0.11.2 (10.0.11.2)  0.968 ms  0.888 ms  0.851 ms
```

To aid in troubleshooting, consider enabling ICMP tunneling on the P and PE routers.

ICMP Tunneling

ICMP tunneling allows the time exceeded messages that occur during a traceroute to reach their destination. The second example on the previous graphic shows that P router time exceeded messages normally do not reach their destination, which is the local CE router. This is because the P routers have no knowledge of VPN routes. To get around this problem, ICMP tunneling causes the label stack to be copied from the original packet to the ICMP message. The ICMP message is then label-switched across the network. This label switching causes the message to be forwarded along the path towards the original packet's destination rather than toward its source. Once the MPLS-encapsulated ICMP packets arrive on the remote PE router, the remote PE router pops the label stack (inner label maps to the correct VRF table) and sends the ICMP messages to their final destination, which is the local CE router in the reverse direction across the MPLS core.

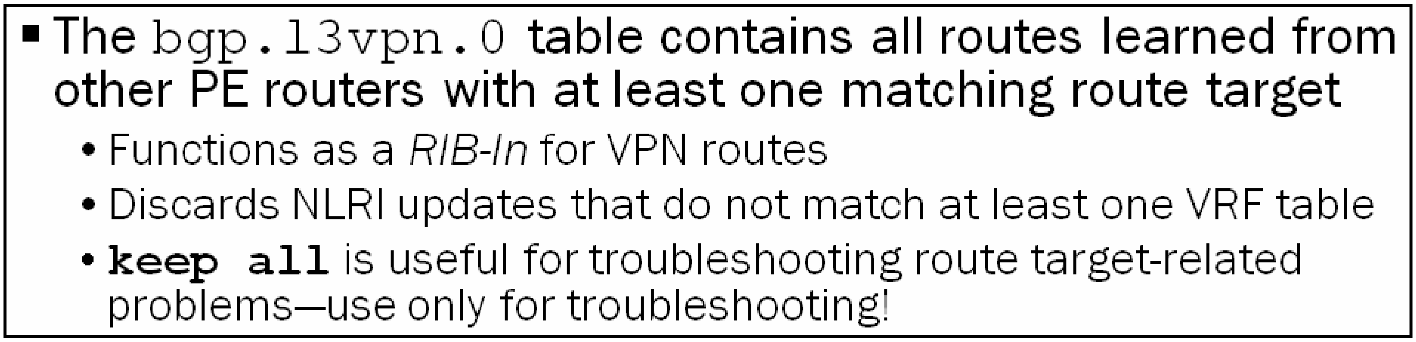
Viewing VRF Tables



- Junos OS allows the viewing of a VRF table with the `show route table` ***vpn-name*** command
  - VRF tables contain:
    - The matching routes learned from remote PE routers
    - Routes learned over the PE-CE link or static routing entries

You can view the contents of a specific VRF table using the **`show route table`** ***vpn-name*** operational command. This table shows entries learned from the local CE router (or static route definitions) as well as routes learned from the remote PE routers having matching route targets.

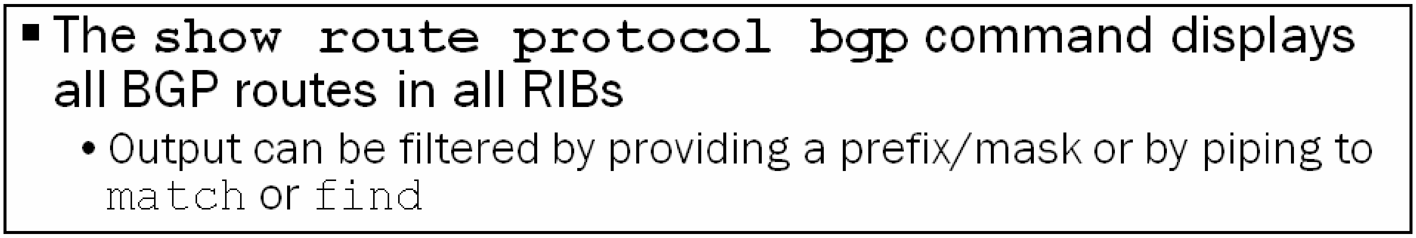
The **`bgp.l3vpn.0`** Table



- The `bgp.l3vpn.0` table contains all routes learned from other PE routers with at least one matching route target
  - Functions as a *RIB-In* for VPN routes
  - Discards NLRI updates that do not match at least one VRF table
  - **`keep all`** is useful for troubleshooting route target-related problems—use only for troubleshooting!

The `bgp.l3vpn.0` table houses routes learned from all remote PE routers having at least one matching route target. This table functions as a RIB-In for VPN routes that match at least one local route target. When troubleshooting route target-related problems, enable the `keep all` option under the BGP configuration stanza. This option places all received VPN routes into the `bgp.l3vpn.0` table, regardless of whether matching route targets are present. Do not leave this option enabled in a production PE router, however, due to the increased memory and processing requirements that can result. In normal operation, a PE router should only house VPN routes that relate to its directly connected sites.

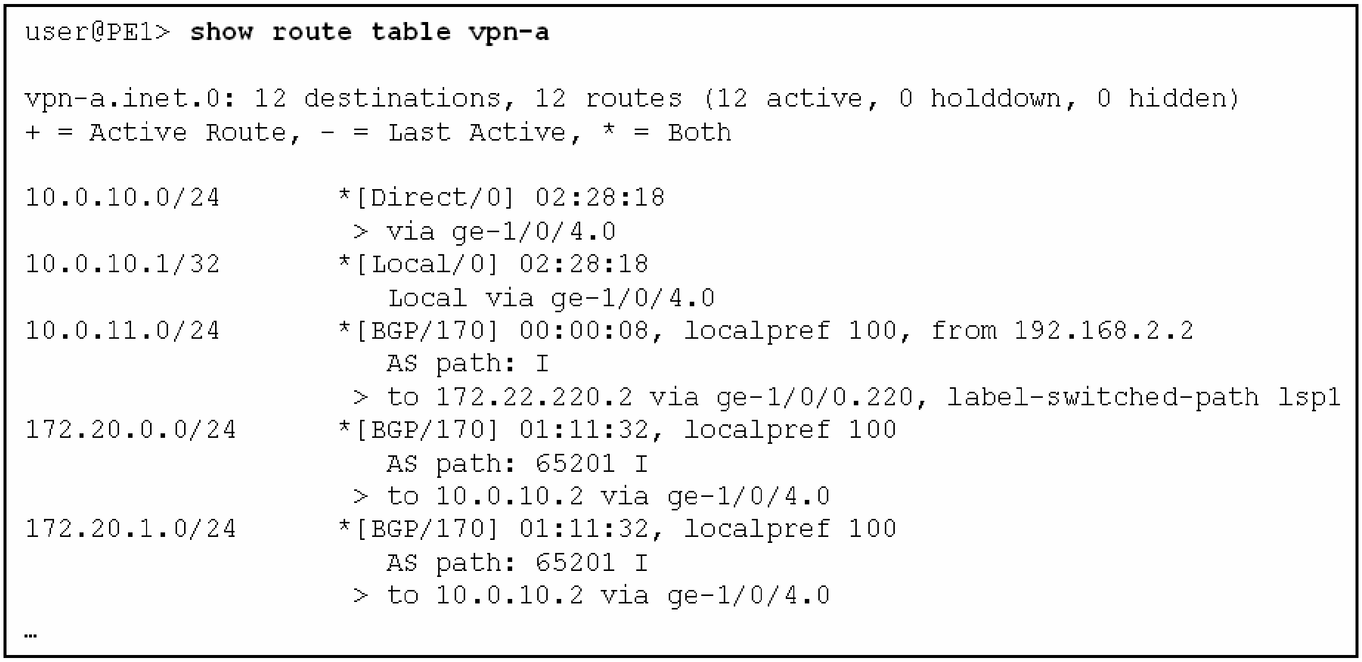
A Shortcut



- The **`show route protocol bgp`** command displays all BGP routes in all RIBs
  - Output can be filtered by providing a prefix/mask or by piping to `match` or `find`

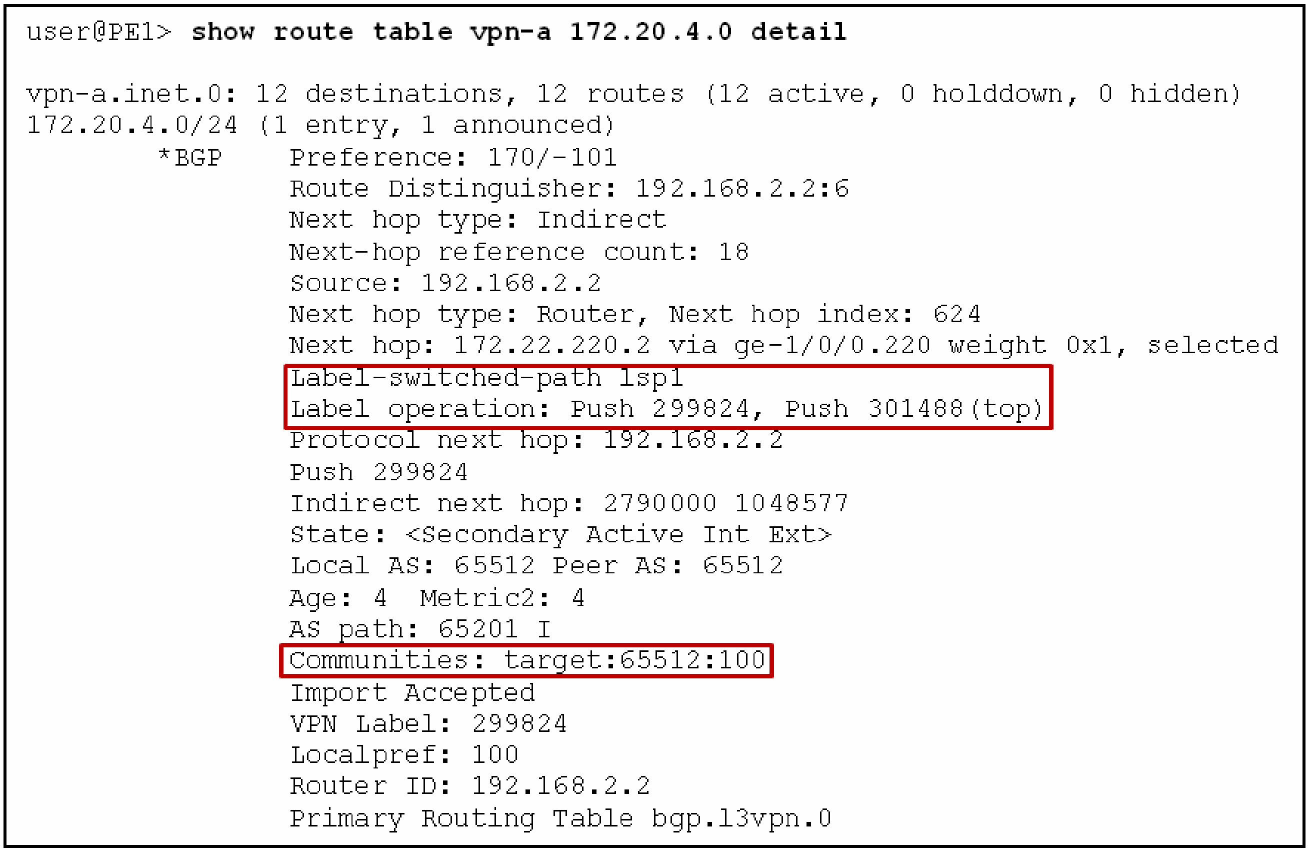
By issuing a **`show route protocol bgp`** command, you can view all BGP routes, regardless of the table in which they are placed. This approach proves helpful when you cannot recall the exact name of a particular VPN's routing instance. You can include a prefix and mask pair to filter some of the output; you also can use the pipe command in conjunction with the `match` or `find` arguments.

Viewing VRF Tables



```
user@PE1> show route table vpn-a

vpn-a.inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.10.0/24       *[Direct/0] 02:28:18
                    > via ge-1/0/4.0
10.0.10.1/32       *[Local/0] 02:28:18
                      Local via ge-1/0/4.0
10.0.11.0/24       *[BGP/170] 00:00:08, localpref 100, from 192.168.2.2
                      AS path: I
                    > to 172.22.220.2 via ge-1/0/0.220, label-switched-path lsp1
172.20.0.0/24      *[BGP/170] 01:11:32, localpref 100
                      AS path: 65201 I
                    > to 10.0.10.2 via ge-1/0/4.0
172.20.1.0/24      *[BGP/170] 01:11:32, localpref 100
                      AS path: 65201 I
                    > to 10.0.10.2 via ge-1/0/4.0
…
```

This graphic provides an example of how you can view a local VRF table. In this case, ***vpn-a*** is the name of the VRF instance. The resulting display shows the local VRF routes and routes learned from the local CE router using EBGP, and routes learned from the remote PE router using MP-IBGP. It is easy to tell the difference between the two sources of BGP routes because routes learned from remote PE routers always point to an LSP as the next hop.

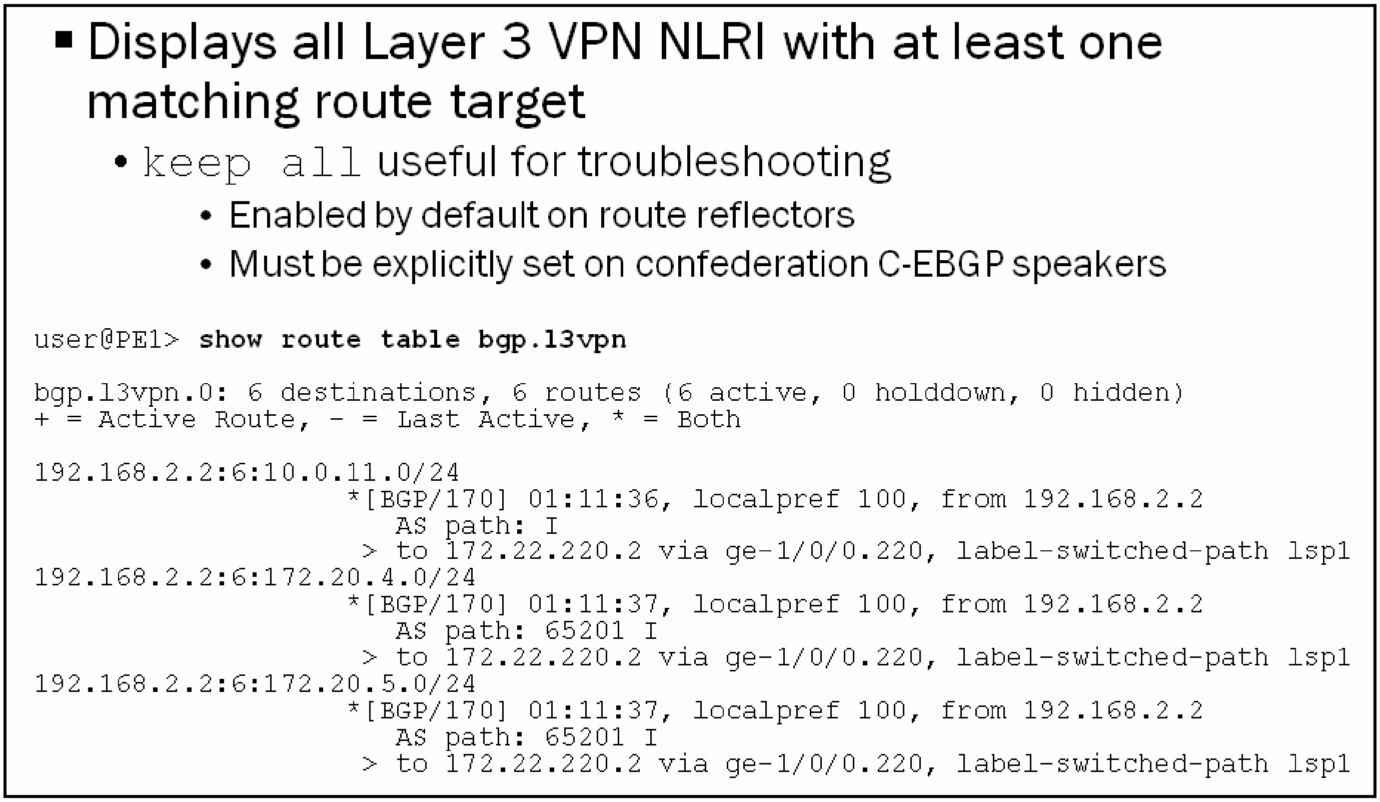


```
user@PE1> show route table vpn-a 172.20.4.0 detail

vpn-a.inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
172.20.4.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Route Distinguisher: 192.168.2.2:6
                Next hop type: Indirect
                Next-hop reference count: 18
                Source: 192.168.2.2
                Next hop type: Router, Next hop index: 624
                Next hop: 172.22.220.2 via ge-1/0/0.220 weight 0x1, selected
                Label-switched-path lsp1
                Label operation: Push 299824, Push 301488(top)
                Protocol next hop: 192.168.2.2
                Push 299824
                Indirect next hop: 2790000 1048577
                State: <Secondary Active Int Ext>
                Local AS: 65512 Peer AS: 65512
                Age: 4  Metric2: 4
                AS path: 65201 I
                Communities: target:65512:100
                Import Accepted
                VPN Label: 299824
                Localpref: 100
                Router ID: 192.168.2.2
                Primary Routing Table bgp.l3vpn.0
```

This graphic shows the optional prefix/mask pair and the `detail` switch added to the same command as the preceding graphic.

Here, the 172.20.4.0/24 route is associated with two labels. The BGP next hop is the PE2 router (192.168.2.2). This next hop is associated with an LSP named ***lsp1***.

Viewing the **bgp.l3vpn.0** Table



- Displays all Layer 3 VPN NLRI with at least one matching route target
  - `keep all` useful for troubleshooting
    - Enabled by default on route reflectors
    - Must be explicitly set on confederation C-EBGP speakers

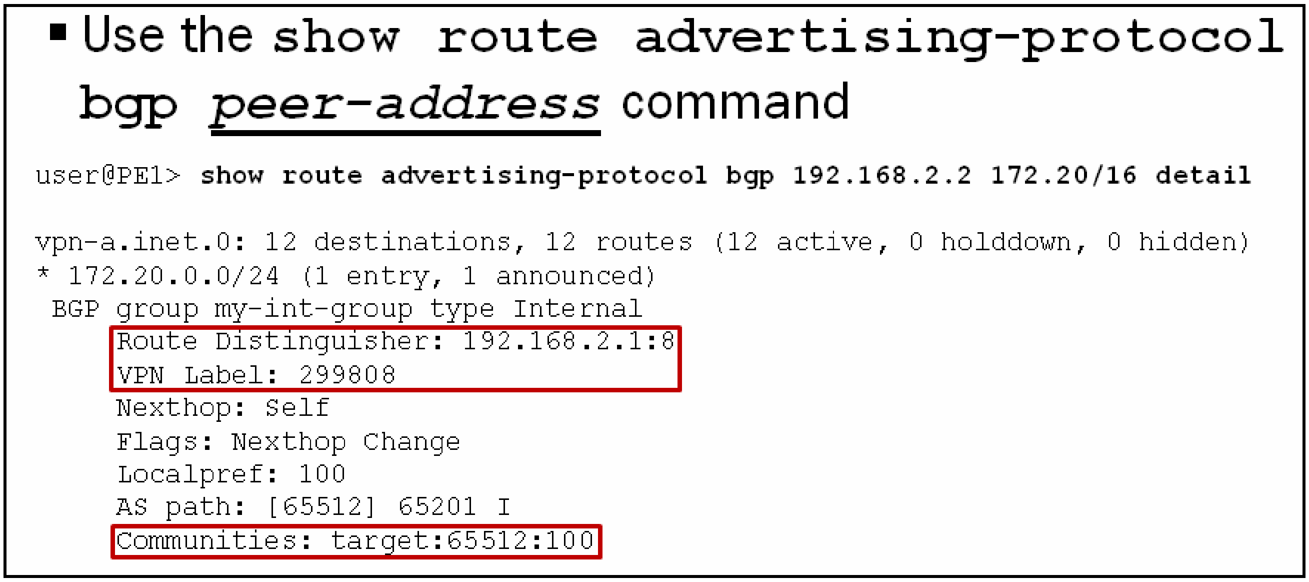
```
user@PE1> show route table bgp.l3vpn

bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

192.168.2.2:6:10.0.11.0/24
                   *[BGP/170] 01:11:36, localpref 100, from 192.168.2.2
                      AS path: I
                    > to 172.22.220.2 via ge-1/0/0.220, label-switched-path lsp1
192.168.2.2:6:172.20.4.0/24
                   *[BGP/170] 01:11:37, localpref 100, from 192.168.2.2
                      AS path: 65201 I
                    > to 172.22.220.2 via ge-1/0/0.220, label-switched-path lsp1
192.168.2.2:6:172.20.5.0/24
                   *[BGP/170] 01:11:37, localpref 100, from 192.168.2.2
                      AS path: 65201 I
                    > to 172.22.220.2 via ge-1/0/0.220, label-switched-path lsp1
```

This graphic shows the output associated with the viewing of the `bgp.l3vpn.0` routing table. This table holds all received MP-IBGP routes containing at least one matching route target. Note that the routes in the `bgp.l3vpn.0` table have the 8-byte route distinguisher associated with the VPN prefixes. Also, the route distinguisher is only used in the control plane.

The `keep all` option forces the PE router to retain all VPN route advertisements, which can assist with route target-related troubleshooting. Route reflectors have the `keep all` option enabled by default because they do not maintain VRF tables and can therefore never be expected to find route target matches. When using confederations, the C-EBGP speakers must have this option explicitly set so that all VPN routes are exchanged across sub-confederation boundaries.

The example here is from the PE1 router. The route distinguisher indicates that the PE2 router originated the routes because the route distinguisher is coded based on the originating PE router’s loopback address in these examples.

Viewing PE-PE Route Advertisements



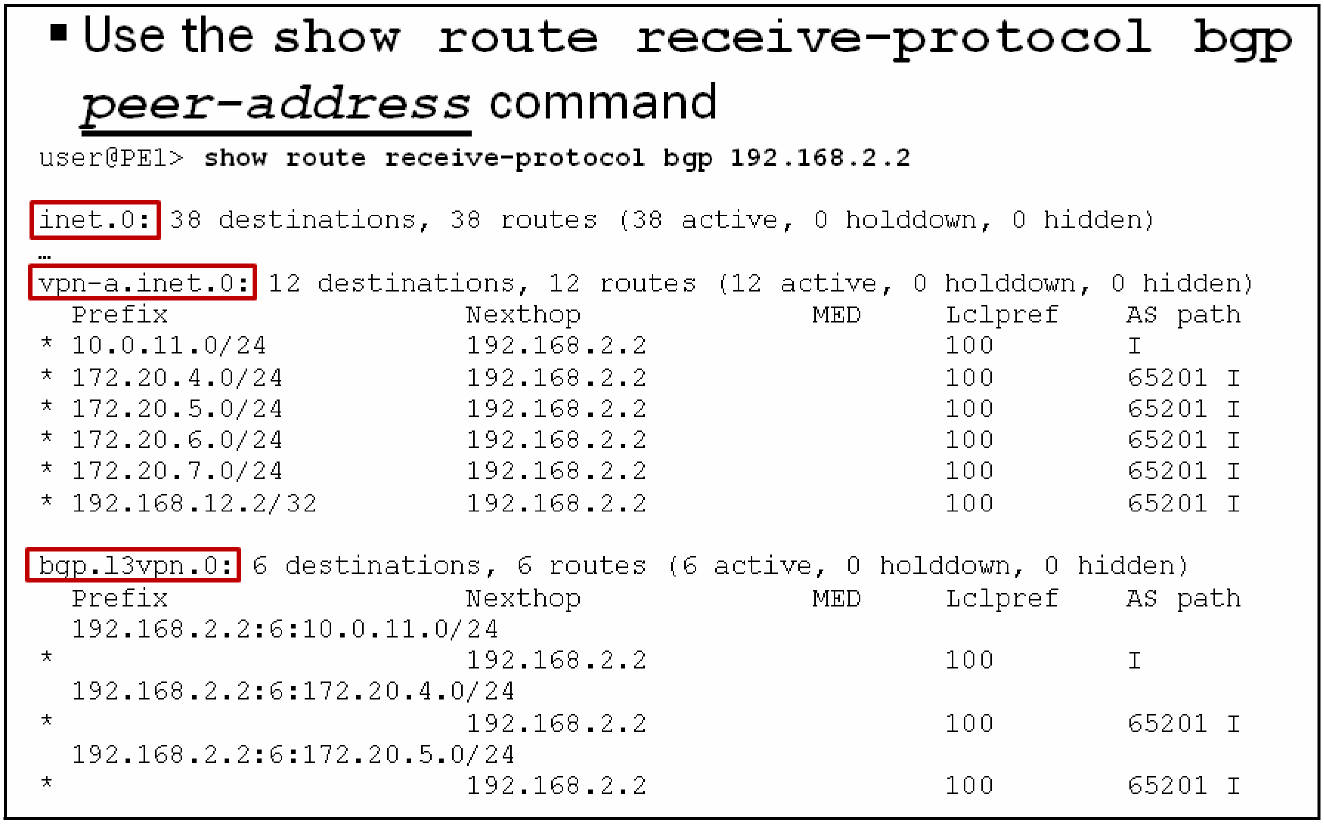
- Use the **show route advertising-protocol bgp *peer-address*** command

```
user@PE1> show route advertising-protocol bgp 192.168.2.2 172.20/16 detail

vpn-a.inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
* 172.20.0.0/24 (1 entry, 1 announced)
 BGP group my-int-group type Internal
     Route Distinguisher: 192.168.2.1:8
     VPN Label: 299808
     Nexthop: Self
     Flags: Nexthop Change
     Localpref: 100
     AS path: [65512] 65201 I
     Communities: target:65512:100
```

You can use the **show route advertising-protocol bgp *peer-address*** command to view the route advertisements being sent to a remote PE router. In this example, the provided prefix/mask pair and the optional `detail` switch control the output.

The resulting output displays the route distinguisher, the assigned VRF label, and the communities attached to the route.

Viewing Routes Learned from Remote PE Routers



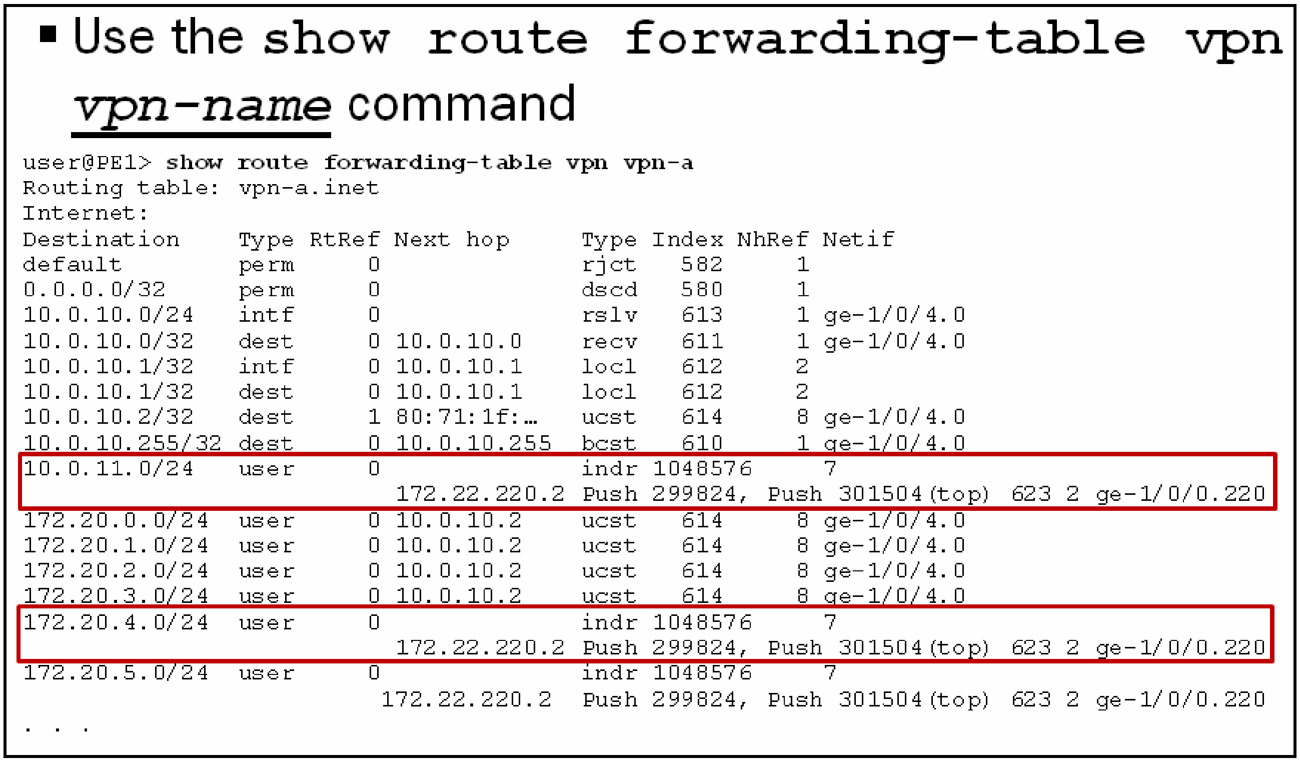
- Use the **show route receive-protocol bgp *peer-address*** command

```
user@PE1> show route receive-protocol bgp 192.168.2.2

inet.0: 38 destinations, 38 routes (38 active, 0 holddown, 0 hidden)
…
vpn-a.inet.0: 12 destinations, 12 routes (12 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.11.0/24            192.168.2.2                  100        I
* 172.20.4.0/24           192.168.2.2                  100        65201 I
* 172.20.5.0/24           192.168.2.2                  100        65201 I
* 172.20.6.0/24           192.168.2.2                  100        65201 I
* 172.20.7.0/24           192.168.2.2                  100        65201 I
* 192.168.12.2/32         192.168.2.2                  100        65201 I

bgp.l3vpn.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  192.168.2.2:6:10.0.11.0/24
*                         192.168.2.2                  100        I
  192.168.2.2:6:172.20.4.0/24
*                         192.168.2.2                  100        65201 I
  192.168.2.2:6:172.20.5.0/24
*                         192.168.2.2                  100        65201 I
```

You can use the **show route receive-protocol bgp *peer-address*** command to view the route advertisements being received from the remote PE router specified on the command line.

In this example, the local PE router has received the 172.20.4.0/24 prefix from 192.168.2.2. Because this route has a matching route target, it is copied into both the **bgp.l3vpn.0** table and the ***vpn-a*** VRF table, which matches that route target.

Viewing VRF Tables



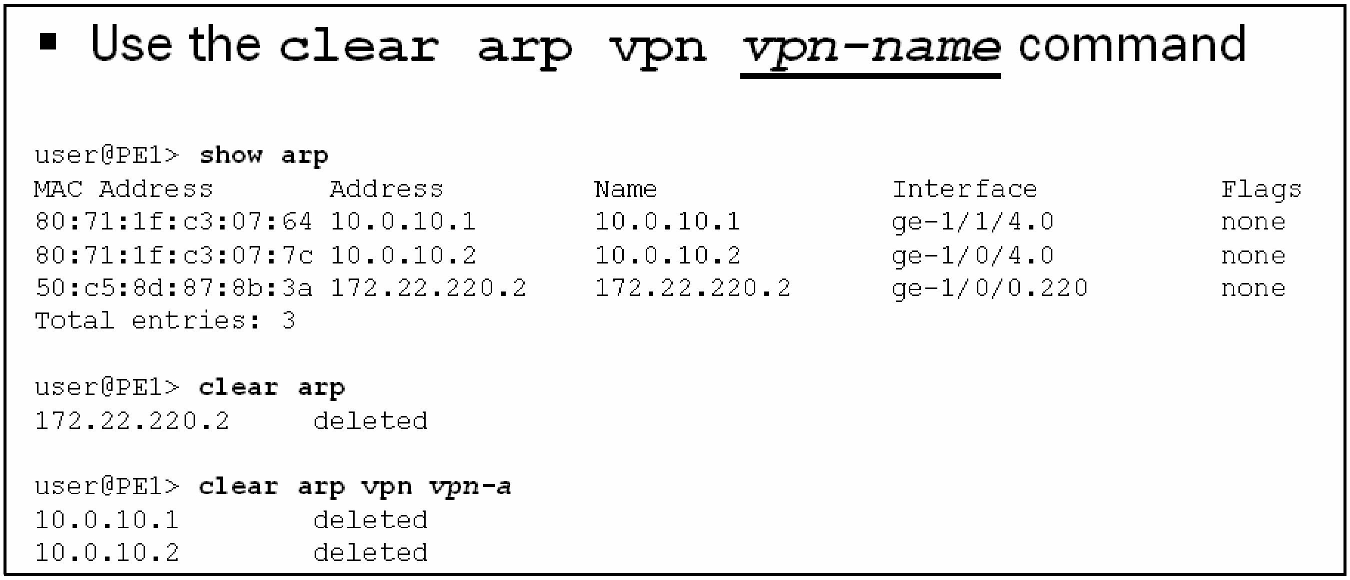
- Use the `show route forwarding-table vpn` ***`vpn-name`*** command

```
user@PE1> show route forwarding-table vpn vpn-a
Routing table: vpn-a.inet
Internet:
Destination    Type RtRef Next hop     Type Index NhRef Netif
default        perm     0              rjct   582     1
0.0.0.0/32     perm     0              dscd   580     1
10.0.10.0/24   intf     0              rslv   613     1 ge-1/0/4.0
10.0.10.0/32   dest     0 10.0.10.0    recv   611     1 ge-1/0/4.0
10.0.10.1/32   intf     0 10.0.10.1    locl   612     2
10.0.10.1/32   dest     0 10.0.10.1    locl   612     2
10.0.10.2/32   dest     1 80:71:1f:…   ucst   614     8 ge-1/0/4.0
10.0.10.255/32 dest     0 10.0.10.255  bcst   610     1 ge-1/0/4.0
10.0.11.0/24   user     0              indr 1048576     7
                          172.22.220.2 Push 299824, Push 301504(top)  623 2 ge-1/0/0.220
172.20.0.0/24  user     0 10.0.10.2    ucst   614     8 ge-1/0/4.0
172.20.1.0/24  user     0 10.0.10.2    ucst   614     8 ge-1/0/4.0
172.20.2.0/24  user     0 10.0.10.2    ucst   614     8 ge-1/0/4.0
172.20.3.0/24  user     0 10.0.10.2    ucst   614     8 ge-1/0/4.0
172.20.4.0/24  user     0              indr 1048576     7
                          172.22.220.2 Push 299824, Push 301504(top)  623 2 ge-1/0/0.220
172.20.5.0/24  user     0              indr 1048576     7
                         172.22.220.2  Push 299824, Push 301504(top)  623 2 ge-1/0/0.220
. . .
```

You can use the **`show route forwarding-table vpn`** ***`vpn-name`*** command to view the forwarding table associated with the specified VRF instance. This command is useful because of its compact (and dense) display, which is easier to parse through when compared to viewing received routes with the **`show route`** commands.

The resulting output shows local forwarding table entries as well as entries for routes learned from remote PE routers.

Clearing VRF ARP Entries



- Use the `clear arp vpn` ***`vpn-name`*** command

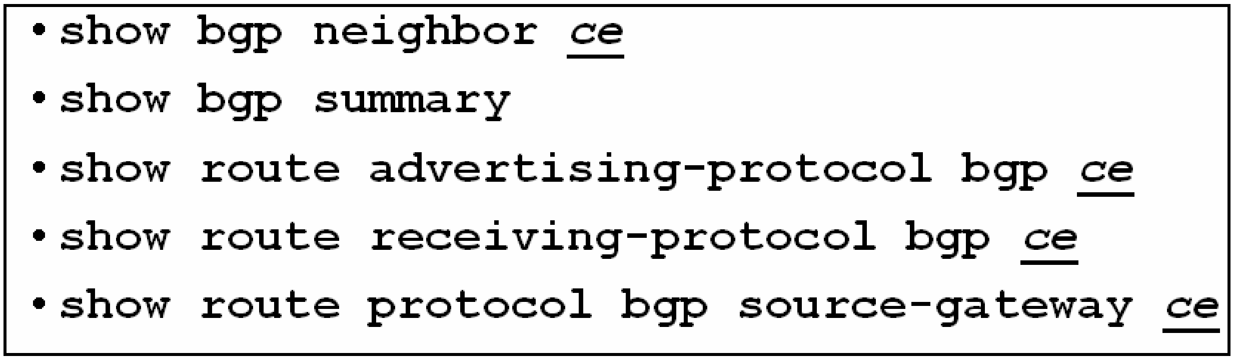
```
user@PE1> show arp
MAC Address       Address         Name            Interface         Flags
80:71:1f:c3:07:64 10.0.10.1       10.0.10.1       ge-1/1/4.0        none
80:71:1f:c3:07:7c 10.0.10.2       10.0.10.2       ge-1/0/4.0        none
50:c5:8d:87:8b:3a 172.22.220.2    172.22.220.2    ge-1/0/0.220      none
Total entries: 3

user@PE1> clear arp
172.22.220.2     deleted

user@PE1> clear arp vpn vpn-a
10.0.10.1        deleted
10.0.10.2        deleted
```

When needed, you can flush ARP entries associated with a VRF instance by including the VPN name as an argument to the **`clear arp`** command. The graphic shows VRF ARP entries both before and after the VRF ARP cache is cleared. It also shows that the **`clear arp`** command by itself only affects ARP entries associated with the main routing instance.

`show arp` Operational Command

To display ARP entries in both the main routing instance and for VRF instances, use the **`show arp`** operational command.

PE-CE BGP Monitoring Uses Standard Commands



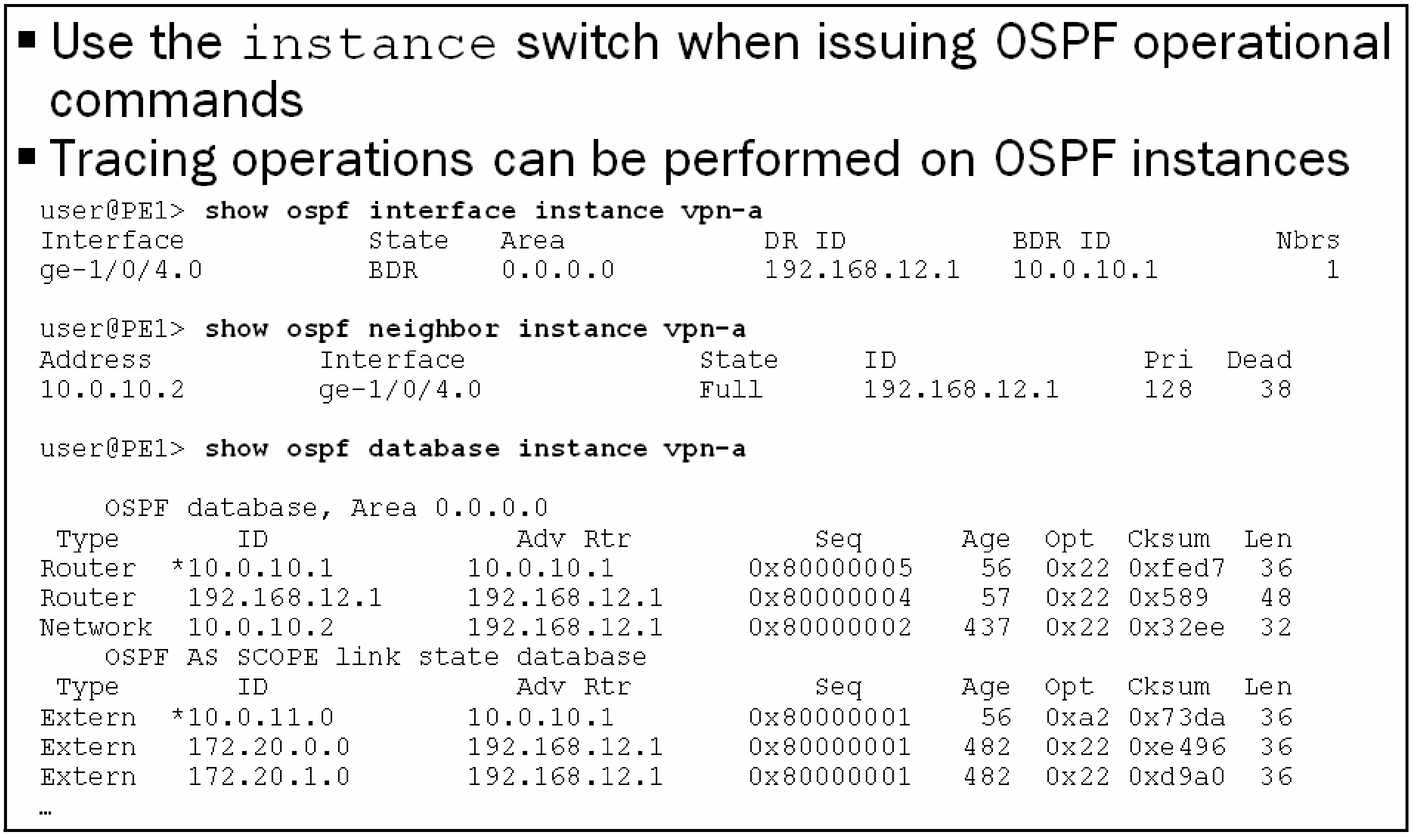
- `show bgp neighbor` *ce*
- `show bgp summary`
- `show route advertising-protocol bgp` *ce*
- `show route receiving-protocol bgp` *ce*
- `show route protocol bgp source-gateway` *ce*

You can use the standard set of BGP-related CLI operational-mode commands to monitor and troubleshoot BGP instances operating over a VRF interface.

BGP Tracing

When needed, you can configure standard protocol tracing under the VRF table's BGP instance to provide additional debugging information.

Use the `instance` Switch When Monitoring OSPF



- Use the `instance` switch when issuing OSPF operational commands
- Tracing operations can be performed on OSPF instances

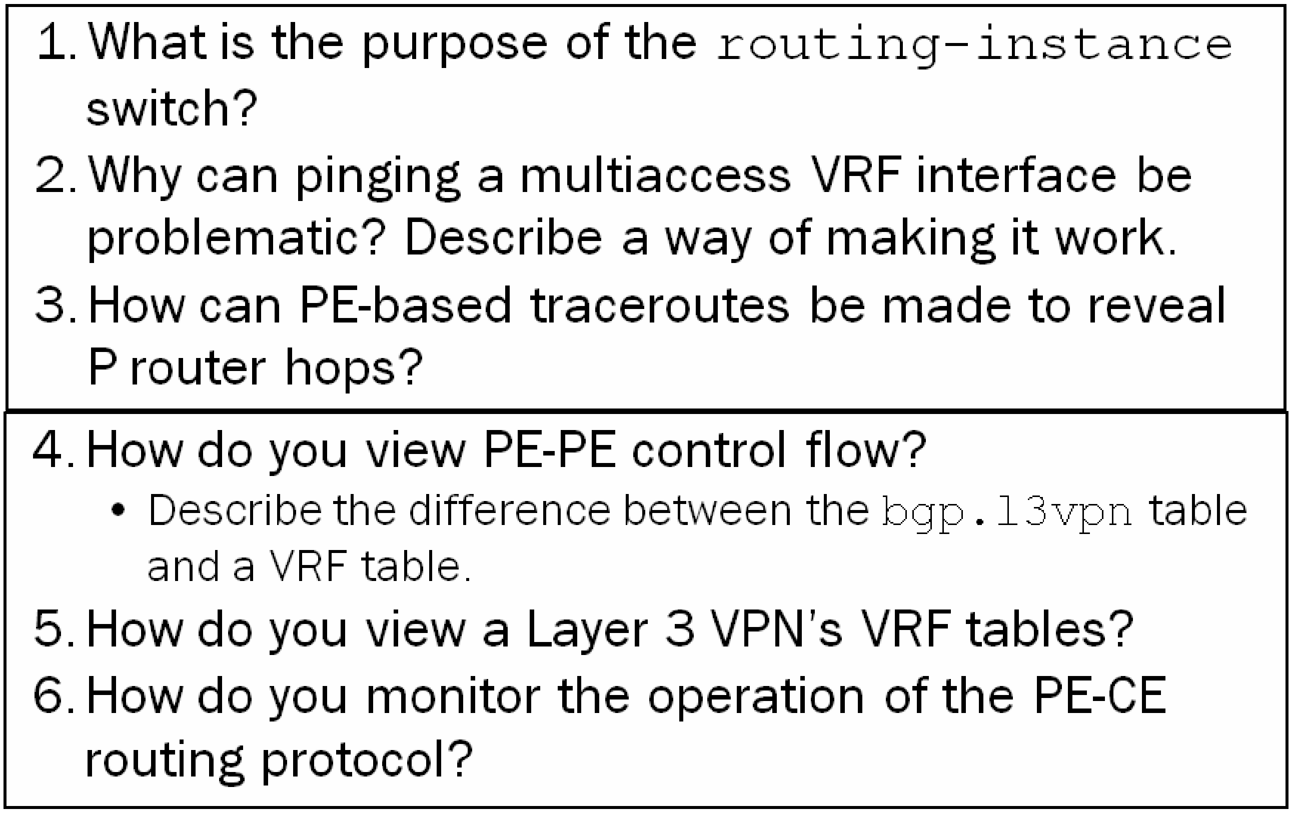
```
user@PE1> show ospf interface instance vpn-a
Interface           State   Area            DR ID           BDR ID          Nbrs
ge-1/0/4.0          BDR     0.0.0.0         192.168.12.1    10.0.10.1          1

user@PE1> show ospf neighbor instance vpn-a
Address          Interface              State     ID               Pri  Dead
10.0.10.2        ge-1/0/4.0             Full      192.168.12.1     128    38

user@PE1> show ospf database instance vpn-a

    OSPF database, Area 0.0.0.0
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Router  *10.0.10.1       10.0.10.1        0x80000005    56  0x22 0xfed7  36
Router   192.168.12.1    192.168.12.1     0x80000004    57  0x22 0x589   48
Network  10.0.10.2       192.168.12.1     0x80000002   437  0x22 0x32ee  32
    OSPF AS SCOPE link state database
 Type       ID               Adv Rtr           Seq      Age  Opt  Cksum  Len
Extern  *10.0.11.0       10.0.10.1        0x80000001    56  0xa2 0x73da  36
Extern   172.20.0.0      192.168.12.1     0x80000001   482  0x22 0xe496  36
Extern   172.20.1.0      192.168.12.1     0x80000001   482  0x22 0xd9a0  36
…
```

When monitoring the operation of a PE-CE OSPF instance, you must include the `instance` switch with a VRF instance as an argument. With the exception of needing instance specification, the standard set of OSPF-related CLI operational-mode commands are available for OSPF monitoring and troubleshooting.

OSPF Tracing

When needed, you can configure standard protocol tracing under the VRF table's OSPF instance to provide additional debugging information.

Review Questions



1. What is the purpose of the `routing-instance` switch?
2. Why can pinging a multiaccess VRF interface be problematic? Describe a way of making it work.
3. How can PE-based traceroutes be made to reveal P router hops?
4. How do you view PE-PE control flow?
   - Describe the difference between the `bgp.l3vpn` table and a VRF table.
5. How do you view a Layer 3 VPN's VRF tables?
6. How do you monitor the operation of the PE-CE routing protocol?

Answers to Review Questions

1.

When using network commands like ping, traceroute, and ssh, the routing-instance switch is used to specify the routing table that should be used to forward packets for the session. By default, the router will use the inet.0 table not the VRF table.

2.

By default, an egress PE that has an Ethernet VRF interface cannot perform both a pop of the MPLS label and an ARP for packets that come from the core. Therefore, an ARP must be performed by the egress router prior to receiving packet from the core. This can be achieved simply by receiving at least one route from the connected CE (which causes an ARP to occur to determine next hop). Also, a static route can be configured within the VRF instance that points to the connected CE. This is generally sufficient. However, if it is necessary to ping the VRF interface without adding routes to the VRF table, **`vrf-table-label`** or a VT interface can be used to allow for both a pop and ARP operation by the egress router.

3.

ICMP tunneling can be configured which allows for P router hops to be revealed.

4.

To view PE to PE control flow use the **`show route receive-protocol bgp`** and **`show route advertise-protocol bgp`** commands which show received and sent BGP routes. Routes that are located in the **`bgp.l3vpn.0`** table have been accepted by at least one vrf-import policy with a matching route target.

5.

To view a VRF table, use the **`show route table`** ***`vpn-name`*** command.

6.

To view the status of the PE-CE routing protocols, use the standard protocol troubleshooting commands modified with the **`instance`** switch.

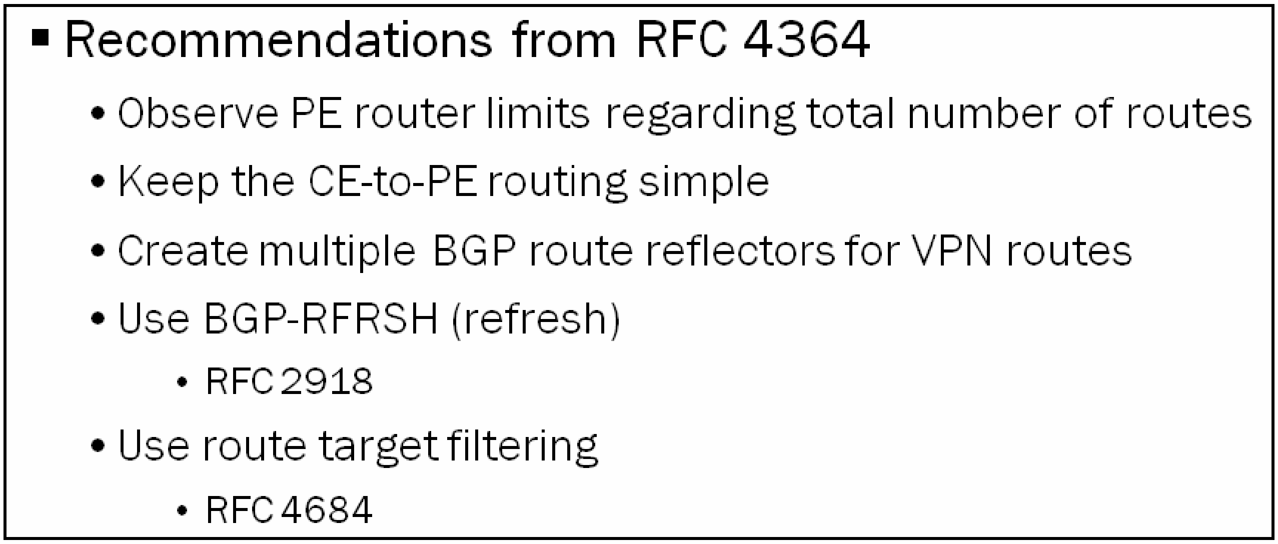
JNCIS-SP Study Guide—Part 3

Chapter 10: Layer 3 VPN Scaling and Internet Access

This Chapter Discusses:

- Four ways to improve Layer 3 virtual private network (VPN) scaling; and
- Three methods for providing Layer 3 VPN customers with Internet access.

Observe Vendor-Specific PE Router Limits



- Recommendations from RFC 4364
  - Observe PE router limits regarding total number of routes
  - Keep the CE-to-PE routing simple
  - Create multiple BGP route reflectors for VPN routes
  - Use BGP-RFRSH (refresh)
    - RFC 2918
  - Use route target filtering
    - RFC 4684

Determining how many VPNs can be supported by a given provider edge (PE) router is a somewhat difficult question. There are many variables that come into play when factoring the VPN load on a PE router. For example, having 1000 VPN routing and forwarding (VRF) tables that use static routing might be no problem at all, but the same number of VRF tables using Open Shortest Path First (OSPF) routing could result in a PE router becoming overloaded. Where possible, PE routers should not carry full Internet routing tables in addition to their VRF table-related burden. A simple static default route to a provider (P) router with a full BGP table normally makes this possible.

Additional PE router scaling factors include memory, processing power, limits on total numbers of labels, and limits on logical interface counts.

Keep the PE-CE Routing Simple

A large portion of the processing burden placed on a PE router is the need to maintain multiple routing protocol instances. Receiving large numbers of routes from customer edge (CE) routers can also present a resource drain. Where possible, you should implement static routing and address aggregation to help ensure that PE routers are not over-taxed.

Route Reflection

A key aspect of the RFC 4364 model is that no single PE router has to carry all VPN state for the provider's network. This concept can be extended to route reflection by deploying multiple route reflectors that are responsible for different pieces of the total VPN customer base. Route reflection has the added advantage of minimizing the number of Multiprotocol Border Gateway Protocol (MP-BGP) peering sessions in the provider's network.

BGP Route Refresh

The use of the BGP refresh message (defined in RFC 2918) allows for non-disruptive adds, moves, and changes, which in turn reduces routing disruption by not forcing the termination of PE-PE MP-BGP sessions when changes are made to the VPN topology or membership.

Route Target Filtering

The use of route target filtering (defined in RFC 4684) can improve efficiencies, because it allows a route reflector to reflect only those routes which a particular client PE router cares about.

Two-Level Label Stacks Spare P Routers

The use of a two-level label stack allows P routers to remain ignorant of all things having to do with the VPN. P routers should only carry the provider's internal routes and, in most cases, full BGP tables. Because VPN traffic is label-switched across the core, the P routers never have to *route* to VPN destinations.

PE Routers Only Keep What Concerns Them

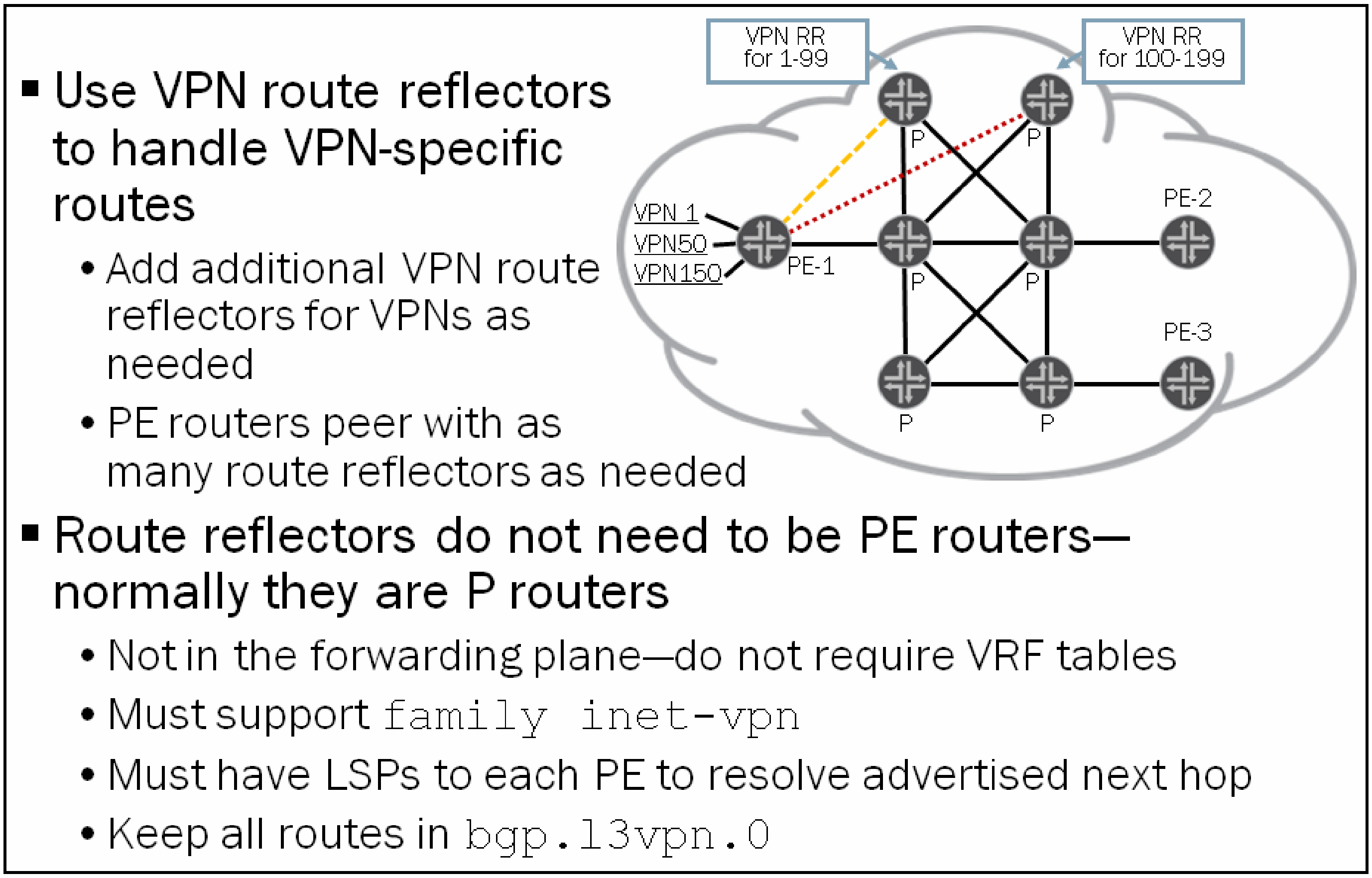
PE routers use route targets to filter out VPN routes that do not concern the PE router's directly connected VPN sites. Therefore, no single PE router must ever carry a state for all VPN customers using a Layer 3 service.

Route Reflection

The use of route reflection could result in a single router (the route reflector) having to store all VPN states. To eliminate this serious scaling issue, it is possible to deploy multiple route reflectors with each reflector servicing a subset of the total VPN population. You must ensure, however, that PE routers are configured correctly to peer with all route reflectors serving VPN customers for which this PE router has locally attached sites.

Based on these methods, Layer 3 VPNs can be scaled virtually without bounds, as no single device must ever carry the total VPN states for the provider's VPN service.

Use VPN-Specific Route Reflectors



- Use VPN route reflectors to handle VPN-specific routes
  - Add additional VPN route reflectors for VPNs as needed
  - PE routers peer with as many route reflectors as needed
- Route reflectors do not need to be PE routers—normally they are P routers
  - Not in the forwarding plane—do not require VRF tables
  - Must support `family inet-vpn`
  - Must have LSPs to each PE to resolve advertised next hop
  - Keep all routes in `bgp.l3vpn.0`

The use of route reflection is an important part of Layer 3 VPN scaling because their presence dramatically reduces the numbers of MP-BGP peering sessions on the PE routers.

We recommend using one or more P routers to provide VPN-related reflection services. If possible, you should not use the VPN route reflectors for conventional (non-VPN) reflection duties.

This graphic illustrates how you can deploy multiple route reflectors so that no single reflector is required to carry all VPN routes. A PE router with local VPN sites ranging from 1 through 99 MP-BGP peers with the reflector on the left, while a PE router with local sites in the 100–199 space peers with the reflector on the right. A PE router must peer with both reflectors if it has local sites belonging to both VPN spaces.

Route Reflectors Should Not Be PE Routers

While a PE router could serve double duty as a reflector, we recommend that you use a P router for VPN reflection. The VPN route reflector automatically keeps all received VPN routes in the `bgp.l3vpn.0` table and is not required to maintain VRF tables, as the reflector is not in the data forwarding path. The automatic use of the `keep all` option in VPN route reflectors means that route target matching is not performed. Therefore, you do not need VRF import and export policy.

The route reflector configuration must include the VPN IP version 4 (IPv4) family because it receives and reflects VPN routes. A BGP route can only be active when it has a resolvable next hop. Also, because VPN routes must resolve to a label-switched path (LSP), the route reflector requires LSPs terminating at each PE router to avoid hidden routes and the resulting problems these hidden routes cause with regards to reflection. When running LDP in the provider core, all routers are connected to all other routers with LSPs, so this requirement is not an issue. The use of RSVP generally requires that an LSP be defined from the route reflector to each of its client PE routers. Work-arounds to this requirement do exist, for example, placing a static default route into `inet.3`.

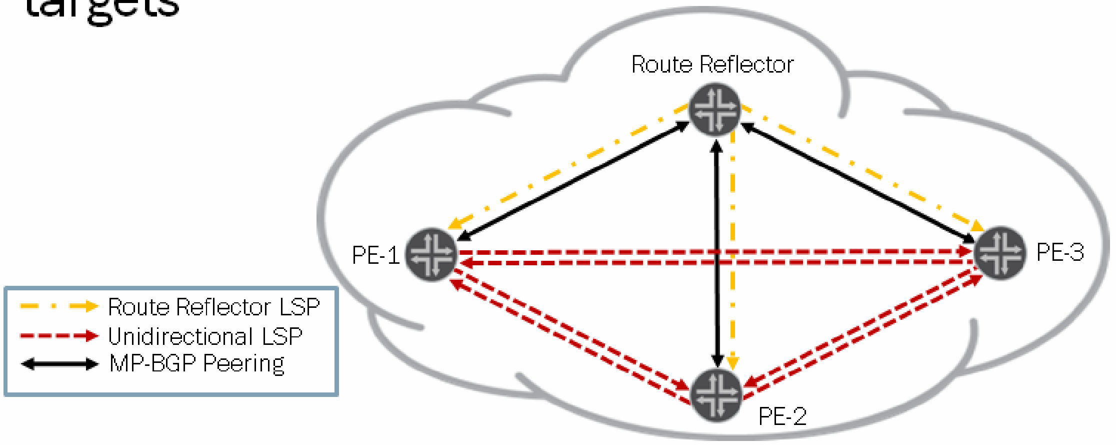
Route Reflector Has No VRF Tables

As previously mentioned, a route reflector does not require VRF tables or VRF-related routing policy. The reflector must support the `inet-vpn` family and must support BGP refresh to accommodate non-disruptive moves and changes. The Junos operating system will automatically negotiate BGP refresh, and the `keep all` option is automatically enabled when a cluster ID is configured (thereby making the device a route reflector).

For a P router to become a VPN route reflector, the only things needed are the configuration of a cluster ID, the declaration of PE routers with which it peers, and configuration of the `inet-vpn` address family. These steps accommodate the BGP peering and VPN route reflector functionality, but remember that LSPs are also needed to ensure that routes are considered usable by the route reflector.

LSPs Needed for Next-Hop Resolution

- PE routers filter received routes based on route targets

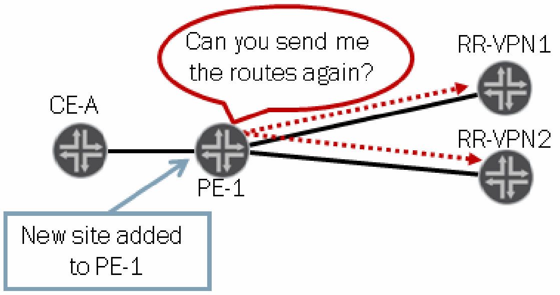


As the graphic shows, LSPs are generally needed from the route reflector to each of its PE clients so that the BGP next hops of the VPN routes can be resolved to an LSP. Routes that cannot be resolved are hidden on the route reflector. Therefore, these routes cannot be re-advertised to other clients. Because the reflector is not in the forwarding path, there is no need for LSPs in the PE client-to-route reflector direction.

When a PE router peers with a VPN route reflector, it is sent all routes contained in the reflector's `bgp.l3vpn.0` table. The PE router uses its VRF import policies to match and keep the routes relating to its locally attached sites.

BGP Is Stateful

- BGP is a stateful protocol
  - Once peers are synchronized, they do not exchange routes again until a change occurs or the session is disrupted



Unlike most routing protocols, BGP uses the reliable delivery services of Transmission Control Protocol (TCP). As a result, once a BGP speaker receives a TCP acknowledgment for network layer reachability information (NLRI) updates sent to a peer, it does not advertise the same routes again unless it must modify the NLRI or the path attributes, or the BGP session itself is disrupted.

PE Routers Filter Routes Based on Route Targets

As discussed, a PE router immediately discards all VPN routes not containing at least one matching route target.

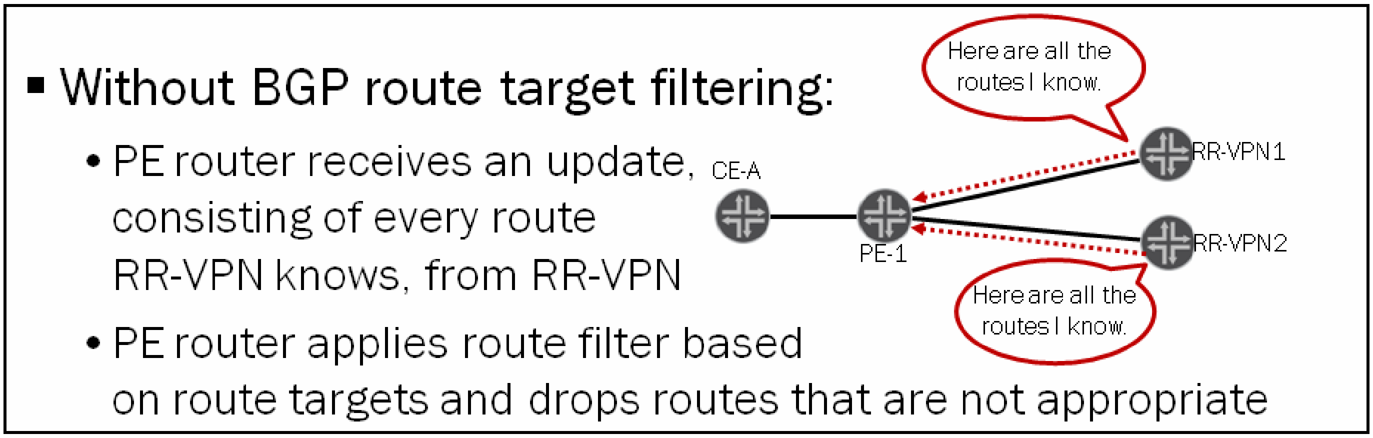
Adds and Changes

When the PE router's VPN-related configuration is modified, it must reevaluate all routes as changes in VRF policy, which might result in route target matches for routes previously ignored. The dilemma that faces our PE router is how to get the BGP peer to resend routes that have already been received and acknowledged!

BGP Refresh

The solution is using the BGP refresh capability as defined in RFC 2918, Route Refresh Capability for BGP-4. BGP refresh allows a BGP speaker to request that its BGP peer readvertise all NLRI associated with the session. Support of BGP refresh is critical to the Layer 3 VPN model, as it allows for non-disruptive changes to VPN membership. The Junos OS supports BGP refresh by default.

Without Route Target Filtering

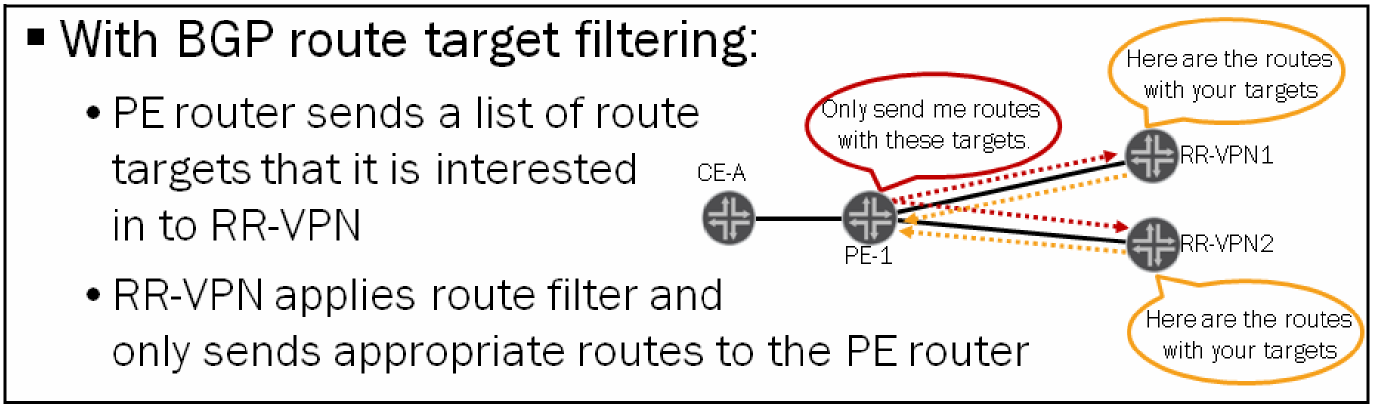


- Without BGP route target filtering:
  - PE router receives an update, consisting of every route RR-VPN knows, from RR-VPN
  - PE router applies route filter based on route targets and drops routes that are not appropriate

Here are all the routes I know.
CE-A
PE-1
RR-VPN1
RR-VPN2
Here are all the routes I know.

Another BGP enhancement, called *route target filtering*, also promises to improve Layer 3 VPN scalability. Without route target filtering, a PE router must receive all VPN routes from all BGP peers. Upon receipt, the PE router's VRF policy can result in the vast majority of these routes being ignored. This problem is most pronounced when route reflection is in use, as a single route reflector might be servicing a large portion of the provider's VPN routes.

With Route Target Filtering



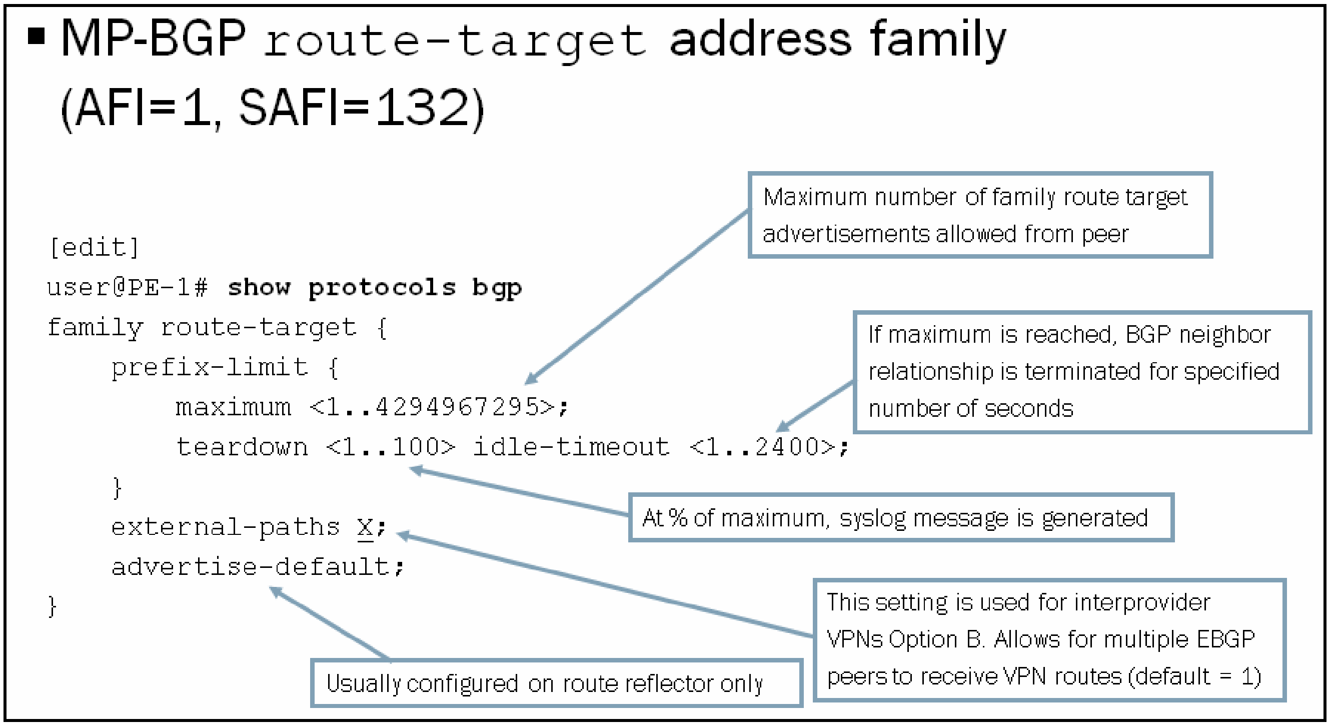
- With BGP route target filtering:
  - PE router sends a list of route targets that it is interested in to RR-VPN
  - RR-VPN applies route filter and only sends appropriate routes to the PE router

Here are the routes with your targets
Only send me routes with these targets.
CE-A
PE-1
RR-VPN1
RR-VPN2
Here are the routes with your targets

With route target filtering, the PE router sends a list of route targets it is interested in, based on its local VRF policy, to the BGP peer. The BGP peer applies this route target list as an outbound filter so that the routes sent to the PE router match at least one of its configured route targets. Route target filtering improves efficiency because fewer BGP updates and protocol traffic are required.

The BGP route target filtering functionality is defined by RFC 4684.

The **route-target** Address Family



- MP-BGP `route-target` address family (AFI=1, SAFI=132)

```
[edit]
user@PE-1# show protocols bgp
family route-target {
    prefix-limit {
        maximum <1..4294967295>;
        teardown <1..100> idle-timeout <1..2400>;
    }
    external-paths X;
    advertise-default;
}
```

Maximum number of family route target advertisements allowed from peer

If maximum is reached, BGP neighbor relationship is terminated for specified number of seconds

At % of maximum, syslog message is generated

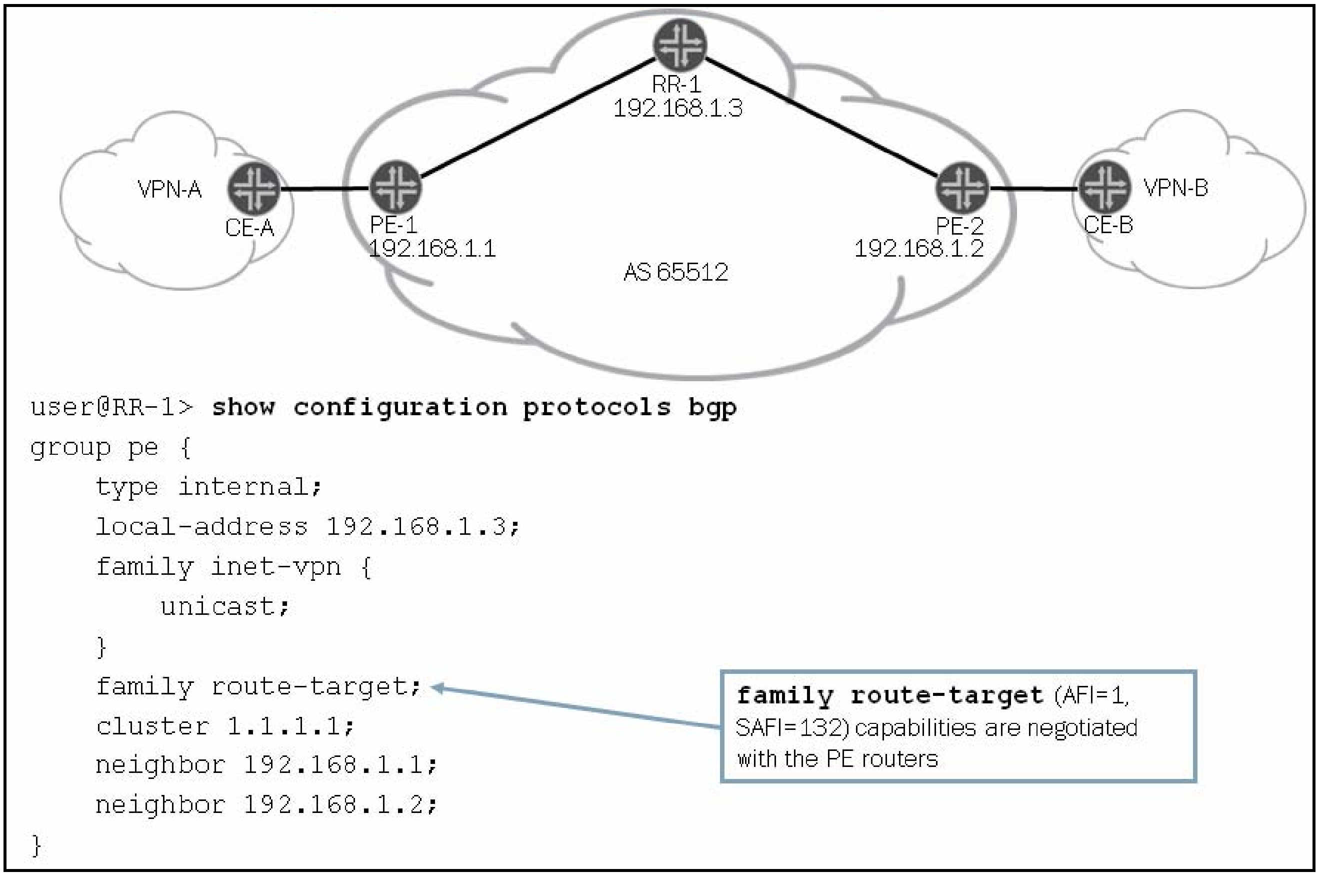
This setting is used for interprovider VPNs Option B. Allows for multiple EBGP peers to receive VPN routes (default = 1)

Usually configured on route reflector only

The graphic shows the optional settings for the `route-target` address family.

- **`prefix-limit`**: Limits the number of route-target advertisements that can be received from a peer router. By default, when the limit is reached, the router stops accepting route-target advertisements from the peer. Using the optional `teardown` statement causes the neighbor relationship with a peer to be torn down when the `maximum` limit is reached. The diagram also shows the usage of the optional percentage and `idle-timeout` configuration.
- **`external-paths`** (default value=1): Affects EBGP Layer 3 VPN route advertisements between autonomous system (AS) boundary routers when performing interprovider VPNs Option B (described in a future chapter). When a router learns the same `route-target` advertisement from multiple EBGP peers, this option allows for Layer 3 VPN advertisements to be sent to more than only one of those EBGP peers.
- **`advertise-default`**: Causes the router to advertise the default route target route (0:0:0/0) and suppress all routes that are more specific. A router reflector can use this on BGP groups consisting of neighbors that act as PE routers only. PE routers often must advertise all routes to the route reflector. Suppressing all route target advertisements other than the default route reduces the amount of information exchanged between the route reflector and the PE routers.

Route Target Filtering: Part 1

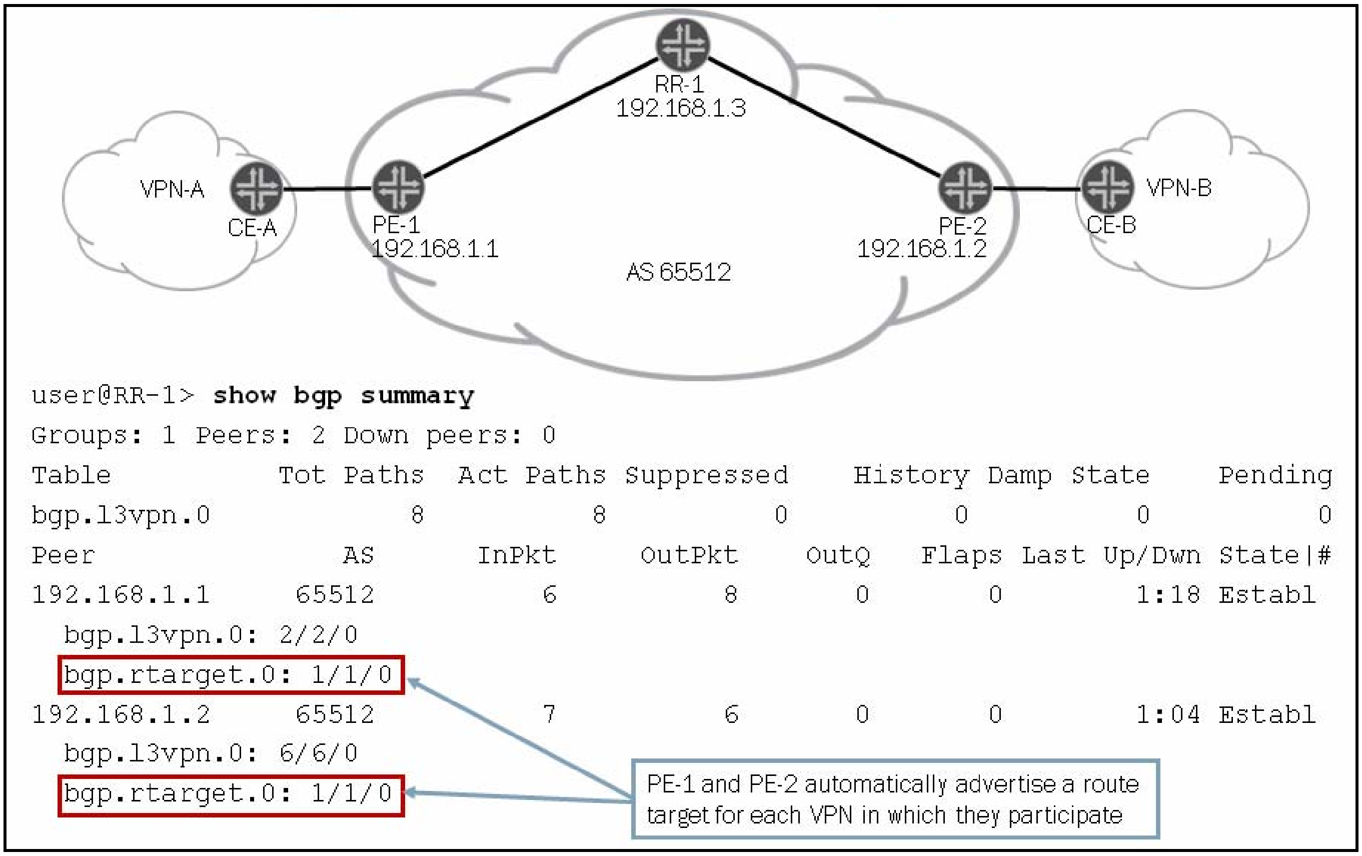


```
user@RR-1> show configuration protocols bgp
group pe {
    type internal;
    local-address 192.168.1.3;
    family inet-vpn {
        unicast;
    }
    family route-target;
    cluster 1.1.1.1;
    neighbor 192.168.1.1;
    neighbor 192.168.1.2;
}
```

family route-target (AFI=1, SAFI=132) capabilities are negotiated with the PE routers

The graphic shows the minimal configuration needed on a route reflector to negotiate the **route-target** address family with its peers. A similar configuration is needed on the two PE routers.

Route Target Filtering: Part 2

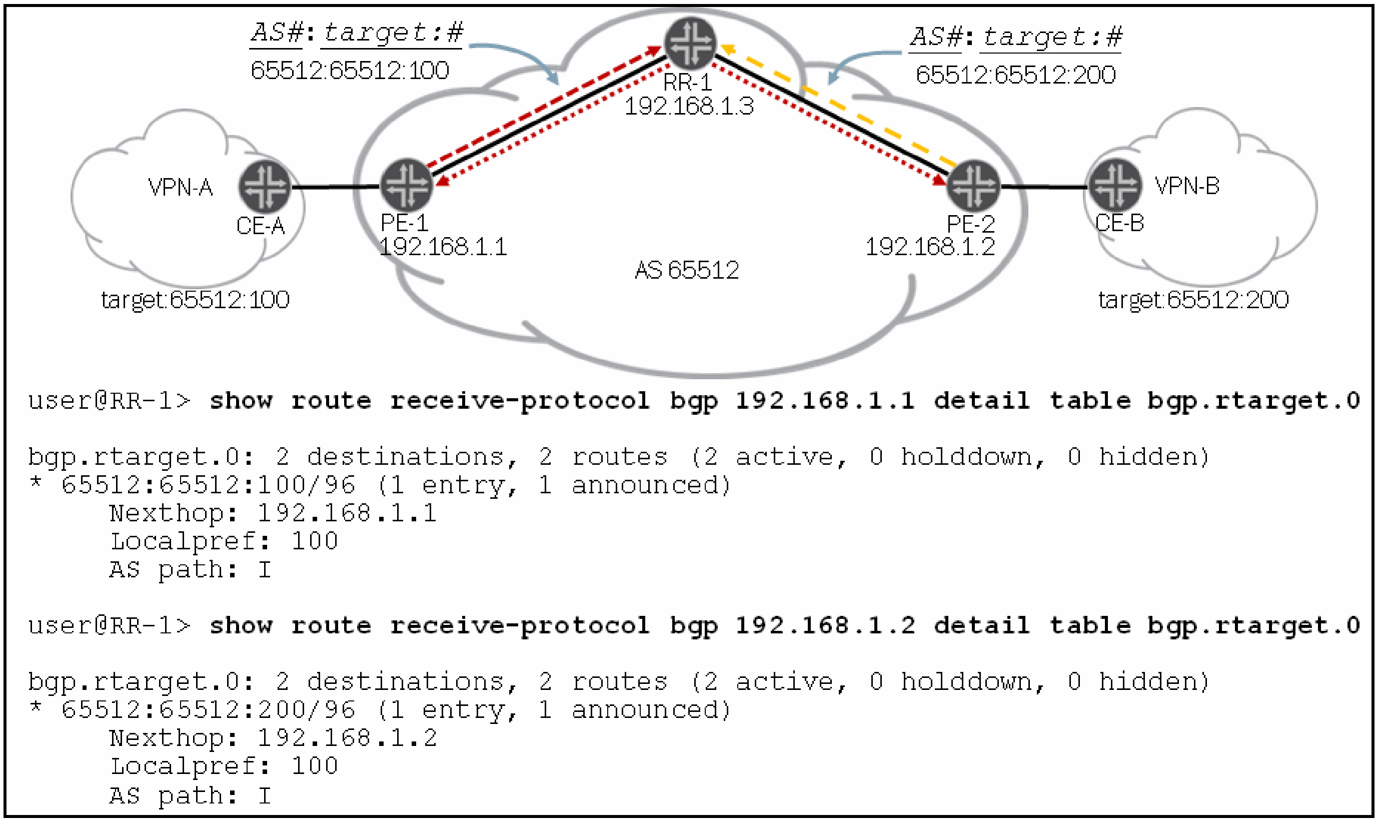


```
user@RR-1> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
bgp.l3vpn.0            8          8          0          0          0          0
Peer               AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn State|#
192.168.1.1     65512          6          8       0       0        1:18 Establ
  bgp.l3vpn.0: 2/2/0
  bgp.rtarget.0: 1/1/0
192.168.1.2     65512          7          6       0       0        1:04 Establ
  bgp.l3vpn.0: 6/6/0
  bgp.rtarget.0: 1/1/0
```

PE-1 and PE-2 automatically advertise a route target for each VPN in which they participate

The graphic shows that when the `route-target` address family is correctly negotiated between routers, all route target advertisements are placed into a new table called `bgp.rtarget.0`. Notice that PE-1 and PE-2 automatically advertise a route target to the route reflector. The specifics relating to these advertisements are shown on the next page.

Route Target Filtering: Part 3

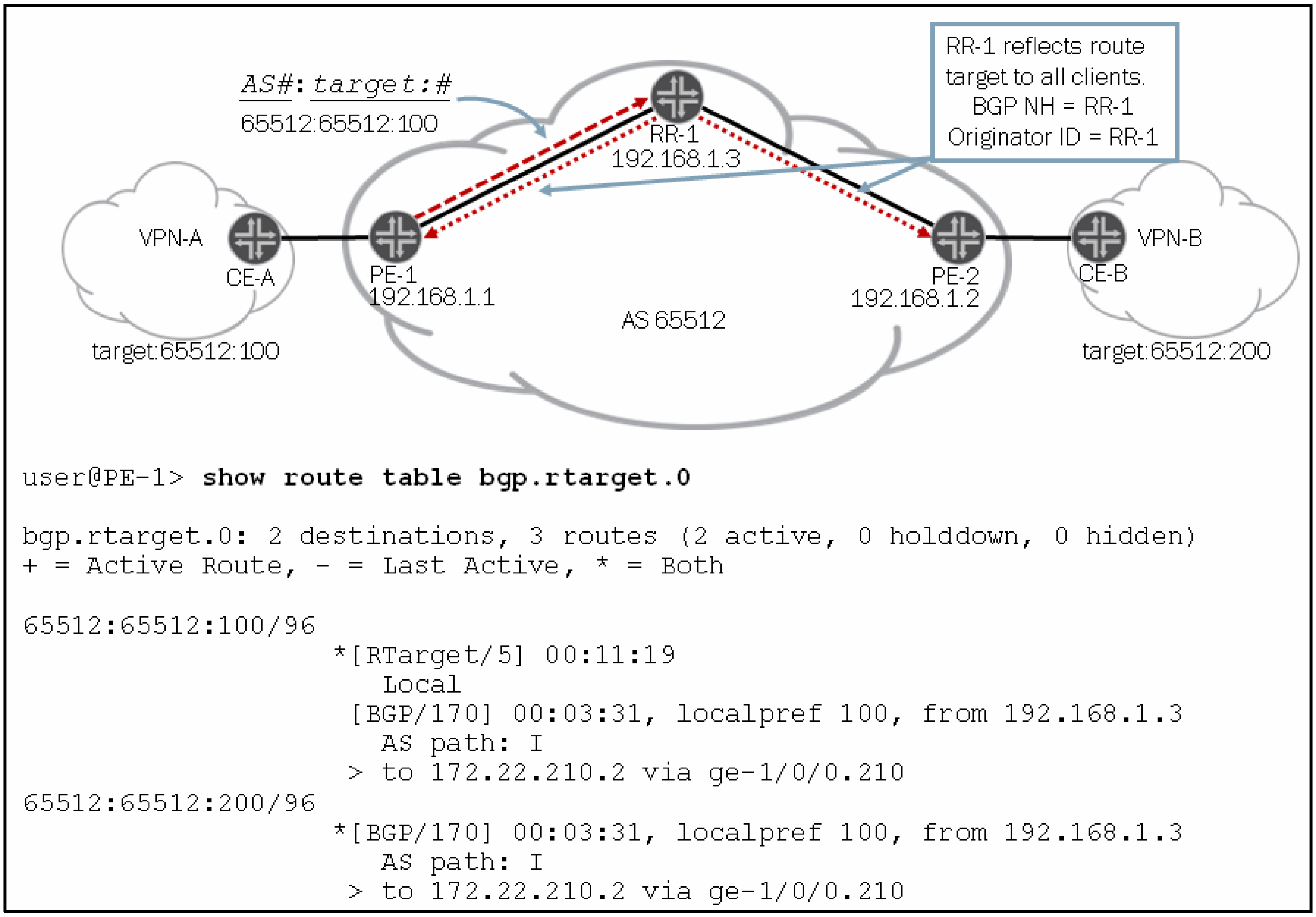


```
user@RR-1> show route receive-protocol bgp 192.168.1.1 detail table bgp.rtarget.0

bgp.rtarget.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
* 65512:65512:100/96 (1 entry, 1 announced)
     Nexthop: 192.168.1.1
     Localpref: 100
     AS path: I

user@RR-1> show route receive-protocol bgp 192.168.1.2 detail table bgp.rtarget.0

bgp.rtarget.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
* 65512:65512:200/96 (1 entry, 1 announced)
     Nexthop: 192.168.1.2
     Localpref: 100
     AS path: I
```

Once the BGP peering relationships are established using the `route-target` address family, the PE routers send a route target advertisement for each of their configured import route targets. For example, because PE-1 has a VRF table configured with a `vrf-target import target:65512:100` statement, PE-1 is requesting that the route reflector send all routes tagged with that community. This is demonstrated in the output of a **`show route-receive protocol bgp 192.168.21.1`** command in the graphic. A route target advertisement takes the form of *`originating AS#`*:*`target community`*.

Notice that the route reflector reflects the route target advertisement to all clients, including the client that originally sent the advertisement. The diagram only shows the route reflector reflecting PE-1's advertisement, but it also reflects PE-2's advertisement in a similar fashion.

Route Target Filtering: Part 4

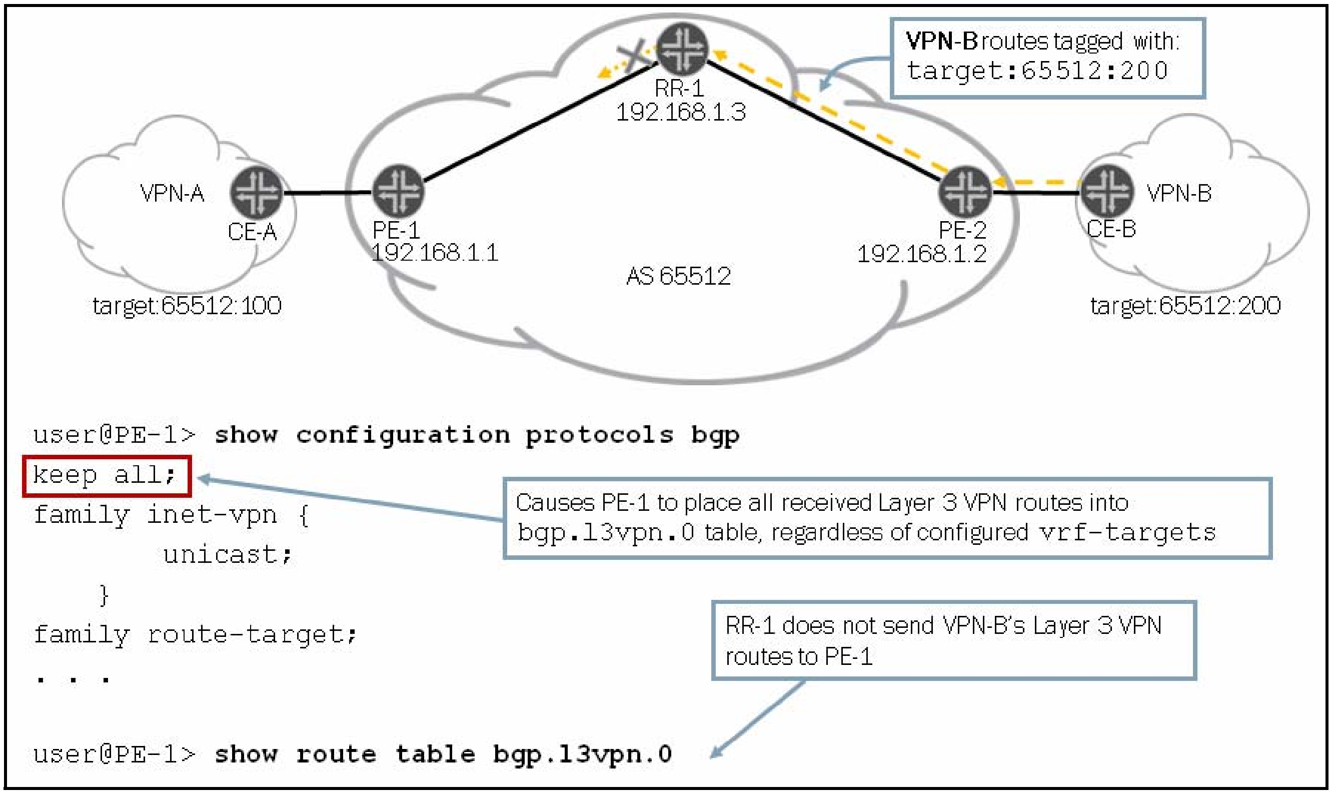


RR-1 reflects route target to all clients.
BGP NH = RR-1
Originator ID = RR-1

```
user@PE-1> show route table bgp.rtarget.0

bgp.rtarget.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

65512:65512:100/96
                   *[RTarget/5] 00:11:19
                      Local
                    [BGP/170] 00:03:31, localpref 100, from 192.168.1.3
                      AS path: I
                    > to 172.22.210.2 via ge-1/0/0.210
65512:65512:200/96
                   *[BGP/170] 00:03:31, localpref 100, from 192.168.1.3
                      AS path: I
                    > to 172.22.210.2 via ge-1/0/0.210
```

When the route reflector reflects the route target advertisements to its internal BGP (IBGP) peers, it changes the BGP next-hop and originator ID to reflect itself. Without these modifications, PE-1 would discard the route target advertisement that it originated. The output of a **`show route table bgp.rtarget.0`** command shows that PE-1 considers the reflected route to be a valid route because of the alterations made by the route reflector. Because the route reflector changes the next hop and originator ID to itself, there must be an LSP on PE-1 that egresses on the route reflector. This LSP is required to resolve the route target NLRIs received on PE-1. PE-1 now knows that it must send any routes contained in locally configured VRF tables that are using export targets of target:65512:100 and target:65512:200 to the route reflector.

Route Target Filtering: Part 5



VPN-B routes tagged with: `target:65512:200`

```
user@PE-1> show configuration protocols bgp
keep all;
family inet-vpn {
        unicast;
    }
family route-target;
. . .

user@PE-1> show route table bgp.l3vpn.0
```

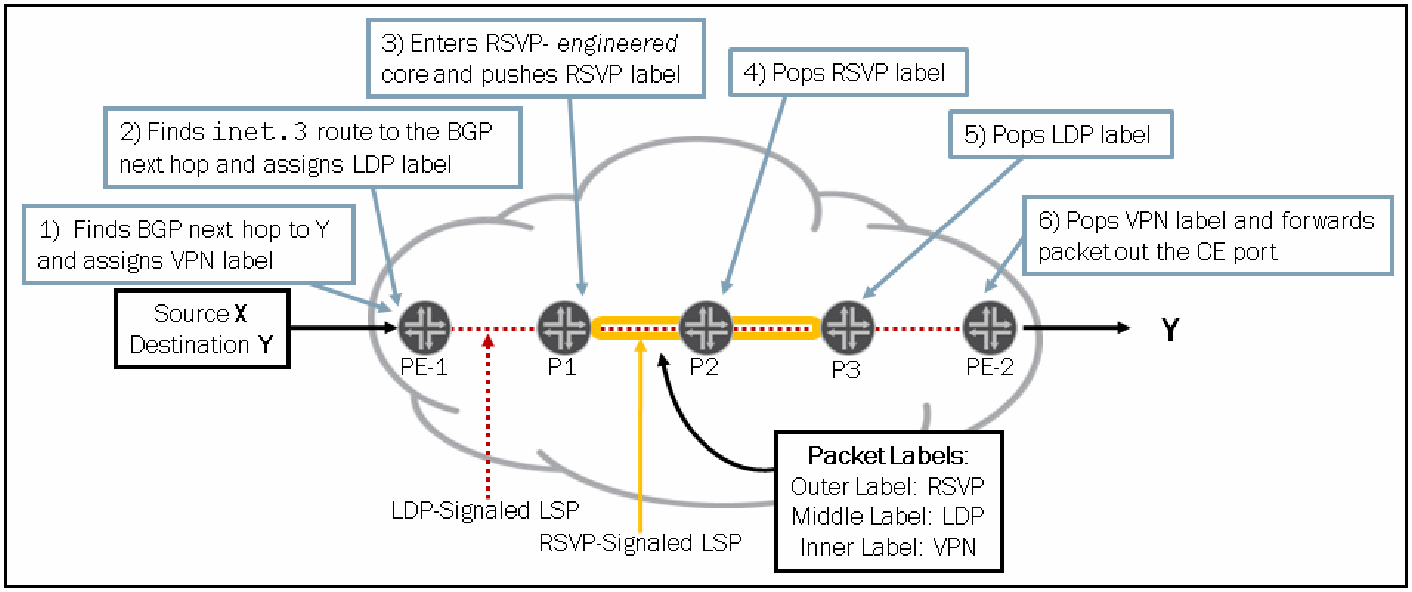
Causes PE-1 to place all received Layer 3 VPN routes into `bgp.l3vpn.0` table, regardless of configured `vrf-targets`

RR-1 does not send VPN-B's Layer 3 VPN routes to PE-1

Without route target filtering, PE-1 would unnecessarily receive routes from PE-2. Because of route target filtering, the route reflector knows that it should only send routes tagged with the `target:65512:100` community to PE-1.

In the example in the graphic, the **`keep all`** statement is used to force PE-1 to store all Layer 3 VPN routes received from the route reflector in its `bgp.l3vpn.0` table. Notice that the `bgp.l3vpn.0` table is empty on PE-1. The empty table shows that route target filtering is working because the route reflector is not sending the Layer 3 VPN routes learned from PE-2 to PE-1.

Adding Traffic Engineering



1) Finds BGP next hop to Y and assigns VPN label

2) Finds `inet.3` route to the BGP next hop and assigns LDP label

3) Enters RSVP- *engineered* core and pushes RSVP label

4) Pops RSVP label

5) Pops LDP label

6) Pops VPN label and forwards packet out the CE port

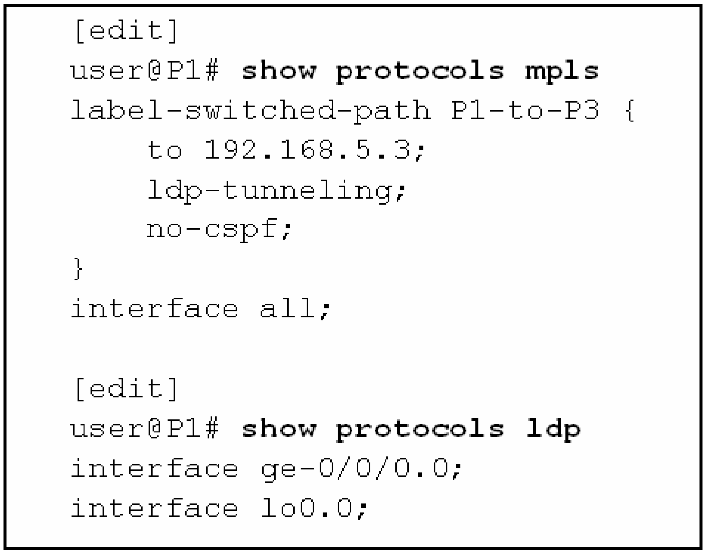
Packet Labels:
Outer Label: RSVP
Middle Label: LDP
Inner Label: VPN

The use of MPLS traffic engineering can also improve VPN scaling and performance. With the Junos OS, you can extend RSVP-based engineered LSPs all the way to the PE routers. The ability to map VPN traffic onto LSPs routed over specific facilities in the provider's core is useful when the VPN service is associated with a service level agreement of some kind.

RFC 4364 requires that the PE router support the LDP signaling protocol; RSVP is optional. As a result, some non-Juniper Networks equipment might only support LDP at the PE router. Because LDP does not support traffic engineering, it might seem that all hope for traffic engineered VPNs is lost. With the Junos OS, you can tunnel LDP-based LSPs over an RSVP traffic engineered LSP. Therefore, traffic engineering across the core is still possible, even though the PE routers might only support LDP signaling.

This graphic shows how the tunneling of LDP over RSVP results in a three-level label stack for VPN traffic. The ingress PE router pushes both a VRF label (inner) and LDP label (middle) before forwarding the labeled packet to the P1 router. The P1 router now pushes a third label (outer) to be swapped as the packet traverses the RSVP core. Penultimate-hop popping (PHP) results in a two-level label stack when the packet arrives at the P3 router. The P3 router also performs PHP so that the PE-2 router receives a packet with a single-level label stack. PE-2 uses the remaining VRF label to associate the packet with the correct VRF interface.

LDP Tunneling



```
[edit]
user@P1# show protocols mpls
label-switched-path P1-to-P3 {
    to 192.168.5.3;
    ldp-tunneling;
    no-cspf;
}
interface all;

[edit]
user@P1# show protocols ldp
interface ge-0/0/0.0;
interface lo0.0;
```

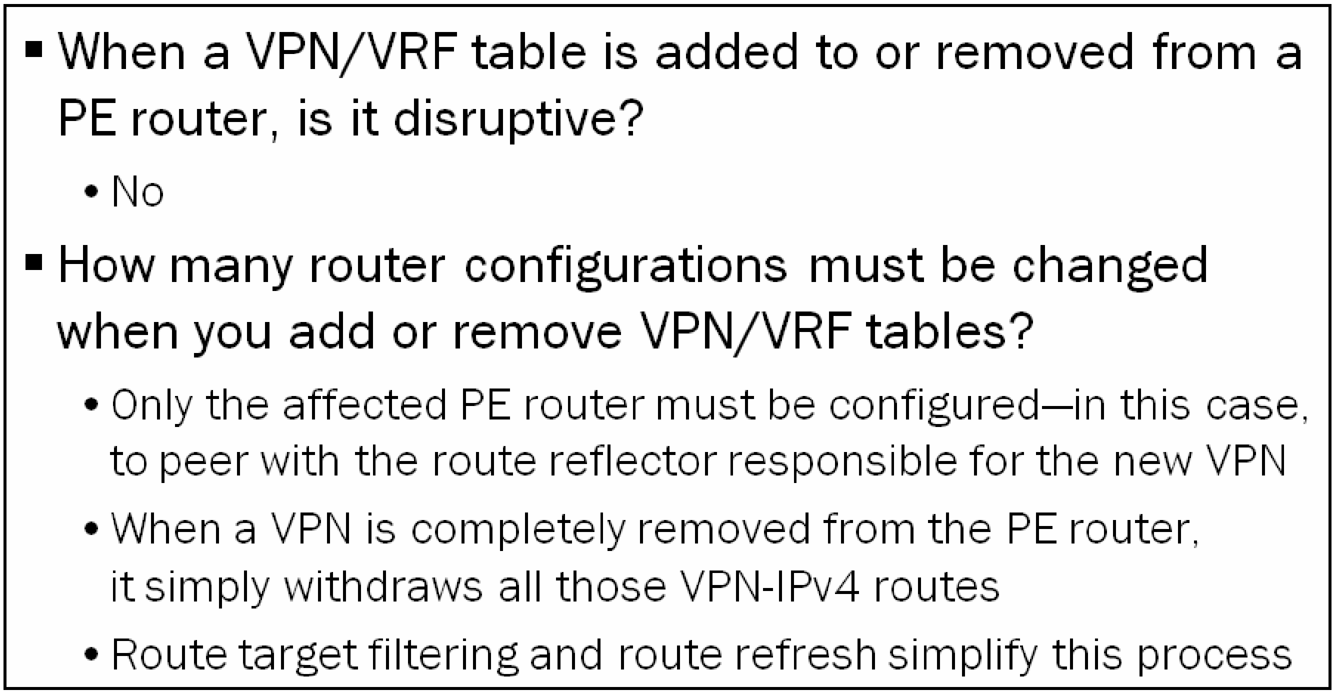
This graphic shows the key configuration steps required for LDP tunneling over RSVP. In this case, the P1 router has an RSVP session to the P3 router that includes the **ldp-tunneling** statement. This router is also configured to run LDP on the PE-facing interface (ge-0/0/0) as well as on its lo0 interface, because you must run LDP on the router's lo0 interface when performing tunneling.

The following capture is from a core P router, where LDP tunneling over RSVP is configured. It clearly shows the three-level label stack that results from LDP tunneling:

```
Frame 25 (110 on wire, 110 captured)
    Ethernet II
    Destination: 00:d0:b7:3f:b5:0c (00:d0:b7:3f:b5:0c)
    Source: 00:d0:b7:3f:b4:ce (00:d0:b7:3f:b4:ce)
    Type: MPLS label switched packet (0x8847)
MultiProtocol Label Switching Header
    MPLS Label: Unknown (100003)
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 0
    MPLS TTL: 254
MultiProtocol Label Switching Header
    MPLS Label: Unknown (100003)
    MPLS Experimental Bits: 0
    MPLS Bottom Of Label Stack: 0
    MPLS TTL: 254
MultiProtocol Label Switching Header
    MPLS Label: Unknown (100002)
    MPLS Experimental Bits: 4
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 254
```

```
Internet Protocol
    Version: 4
    Header length: 20 bytes
    . . .
```

VRF Table Modifications Are Non-Disruptive



- When a VPN/VRF table is added to or removed from a PE router, is it disruptive?
  - No
- How many router configurations must be changed when you add or remove VPN/VRF tables?
  - Only the affected PE router must be configured—in this case, to peer with the route reflector responsible for the new VPN
  - When a VPN is completely removed from the PE router, it simply withdraws all those VPN-IPv4 routes
  - Route target filtering and route refresh simplify this process

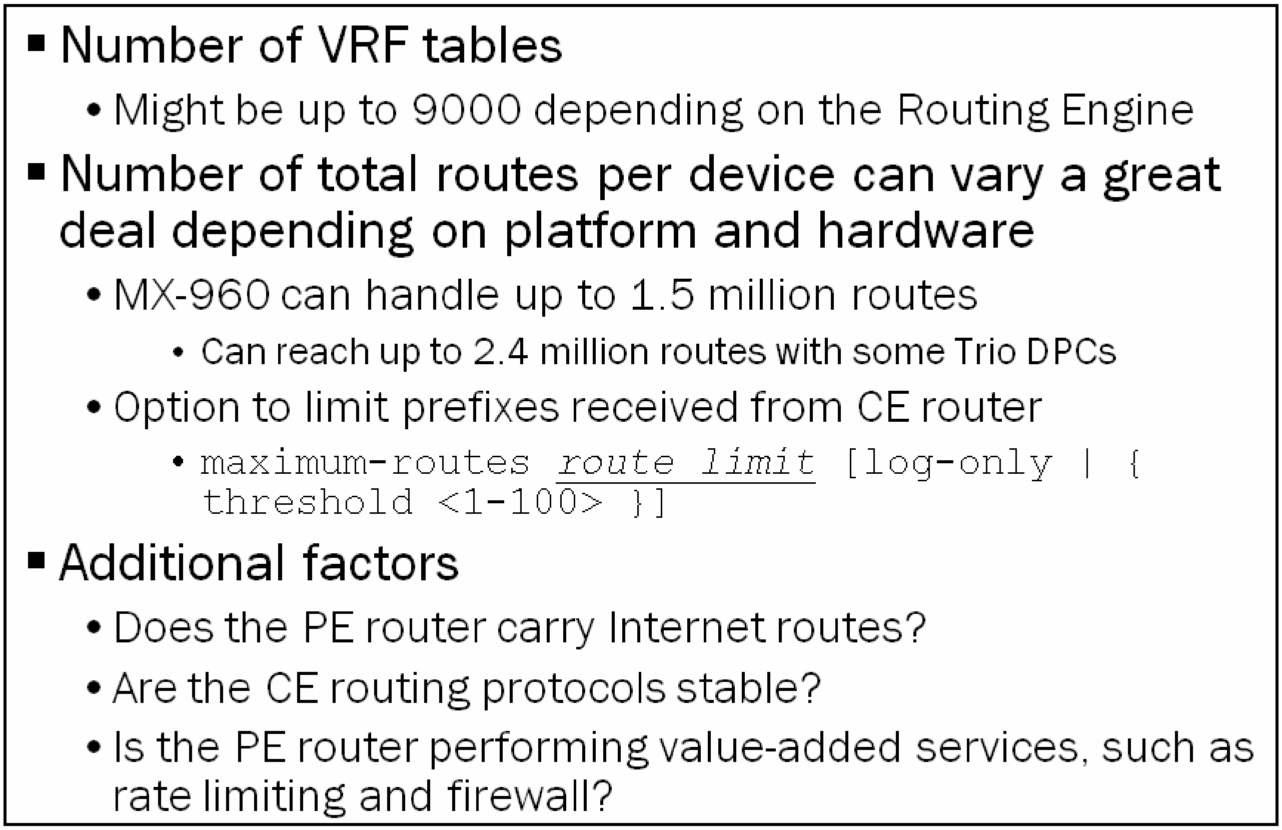
The support of BGP refresh means that modification to a PE router's VRF-related configuration is non-disruptive to the operation of the other VPNs configured on that PE router. Without refresh, changes to VPN membership would require clearing the shared MP-BGP sessions between PE routers, which would be disruptive to all VPNs supported by that PE router.

Adding a New VPN

In general, adding a new VPN site to a PE router does not require the modification of the configuration in the remote PE router, the provider core, or in any route reflectors that are deployed, especially when all PE routers have full MP-BGP and MPLS connectivity (with or without route reflectors). With these prerequisites, the new VPN site requires configuration changes only to the PE router attaching to the new site.

If full MP-BGP and MPLS connectivity is not preprovisioned among the PE routers, the remote PE routers require changes; the remote PE routers must then establish LSP and MP-BGP sessions to the PE router serving the new VPN site.

Number of VRF Tables



- Number of VRF tables
  - Might be up to 9000 depending on the Routing Engine
- Number of total routes per device can vary a great deal depending on platform and hardware
  - MX-960 can handle up to 1.5 million routes
    - Can reach up to 2.4 million routes with some Trio DPCs
  - Option to limit prefixes received from CE router
    - `maximum-routes` *route limit* `[log-only | { threshold <1-100> }]`
- Additional factors
  - Does the PE router carry Internet routes?
  - Are the CE routing protocols stable?
  - Is the PE router performing value-added services, such as rate limiting and firewall?

This graphic outlines some of the recommended guidelines for scaling. You should understand, that many of the limitation are related to available Routing Engine memory. The total number of VRF instances per device can be as high as 9000. This total number might be more or possibly less depending on the resources needed for the PE-CE routing. It is recommended that the PE-CE routing protocol be kept as simple as possible. For instance, static or RIP routing presents less processing load on the PE router when compared to the OSPF or BGP routing protocols.

Number of Routes Per Device

The total number of routes that are supported widely varies depending on the platform and the hardware combinations being used. The MX960 for instance, can handle up to 1.5 million routes. With certain Trio Modular Port Concentrators (MPCs) the number of routes can reach as high as 2.4 million.

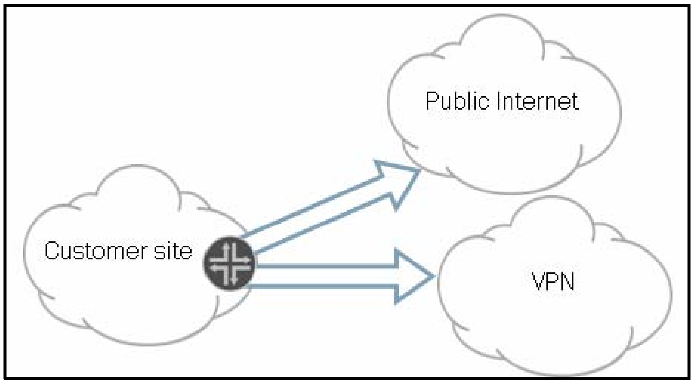
The Junos OS allows you to limit the number of prefixes received from the CE router using a dynamic routing protocol. When the prefix limit is reached, you can choose to have warning messages logged, or to stop accepting additional routes. The **`maximum-routes`** option is configured under the `routing-options` portion of a routing instance's configuration. Take care when opting to ignore CE routes in excess of the limit, as the results can lead to some interesting troubleshooting!

Additional Scaling Factors

With the preceding guidelines in mind, you must also consider other factors that can affect the processing and resource loads on a PE router. For example, does the PE router serve double duty by providing non-VPN Internet access services? Does it carry a full BGP table? Is the CE routing protocol stable? Instability (route flap) within a VPN site can result in substantial processing loads on the PE router. Have valued-added services such as firewall filtering been deployed on the PE router?

Because so many variables exist, it is difficult to provide a concrete rule defining how much is too much. Adhering to these scaling recommendations should result in successful VPN deployment.

Private Addressing Requires NAT



If a VPN site uses private addressing, Internet access requires some form of Network Address Translation (NAT). Either the CE or PE routers can perform the NAT function.

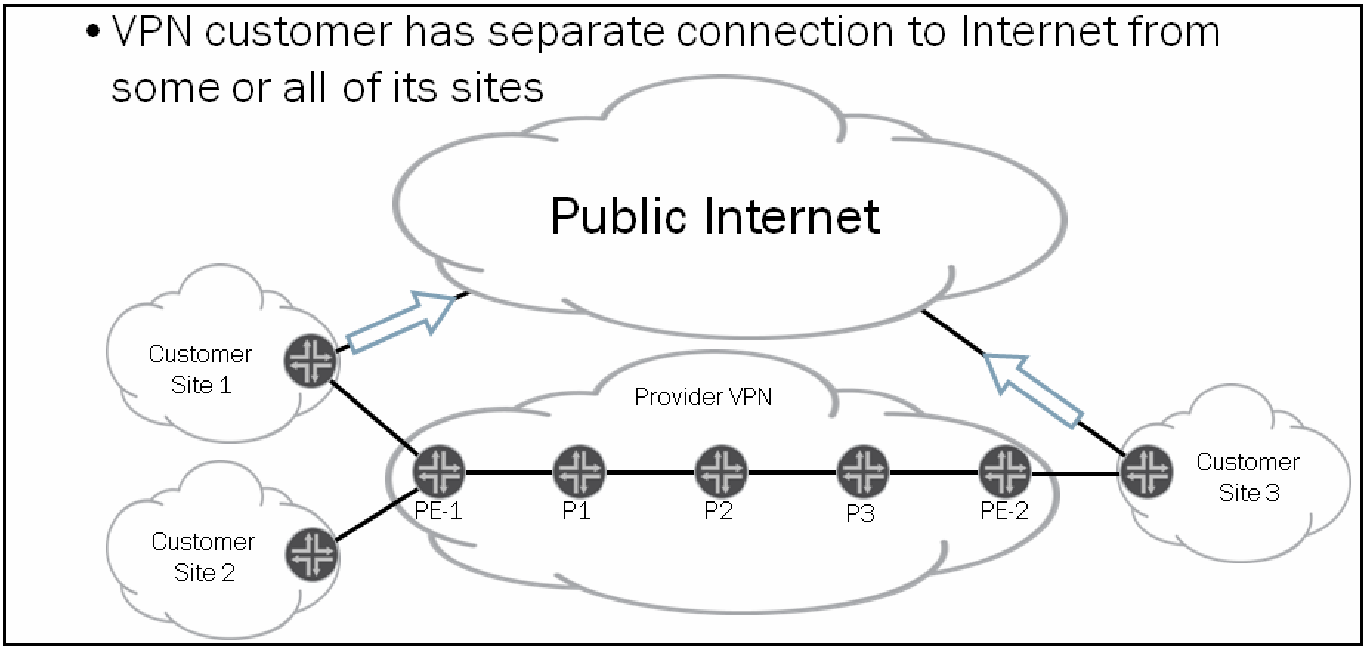
RFC 4364 Internet Access Options

The following list provides details about the various RFC 4364 Internet access options:

- *Option 1*: RFC 4364 defines several Internet access options. In Option 1, the PE router does not exchange routes between its main routing instance and the instances associated with its VRF tables. Option 1 solutions are often referred to as *non-VRF Internet access* because Internet traffic ultimately crosses a non-VRF interface before leaving or entering a VPN site.
- *Option 2*: Option 2 defines Internet access options in which the PE router maintains partial or full Internet routes in its main routing table and has the ability to redistribute routes between the VRF tables and the main routing instance. Option 2 solutions can provide VRF interface-based Internet access, depending on implementation specifics. Option 2 might require the PE router to place some or all of the VPN's routes into the main forwarding table to accommodate return traffic. The routes copied into the main forwarding table must represent globally unique addresses.
- *Option 3*: Option 3 defines Internet access options in which a central CE location is used to provide Internet access to other sites using both a VRF and non-VRF interface. Option 3 solutions are referred to as *VRF-based Internet access*, because remote CE locations use a VRF interface to connect to the central site providing Internet access.

In operation, the central site advertises a default route placed into the VRF tables of the remote locations. When the central CE device receives nonlocal traffic, it turns the traffic around and sends it to the PE router using a non-VRF interface.

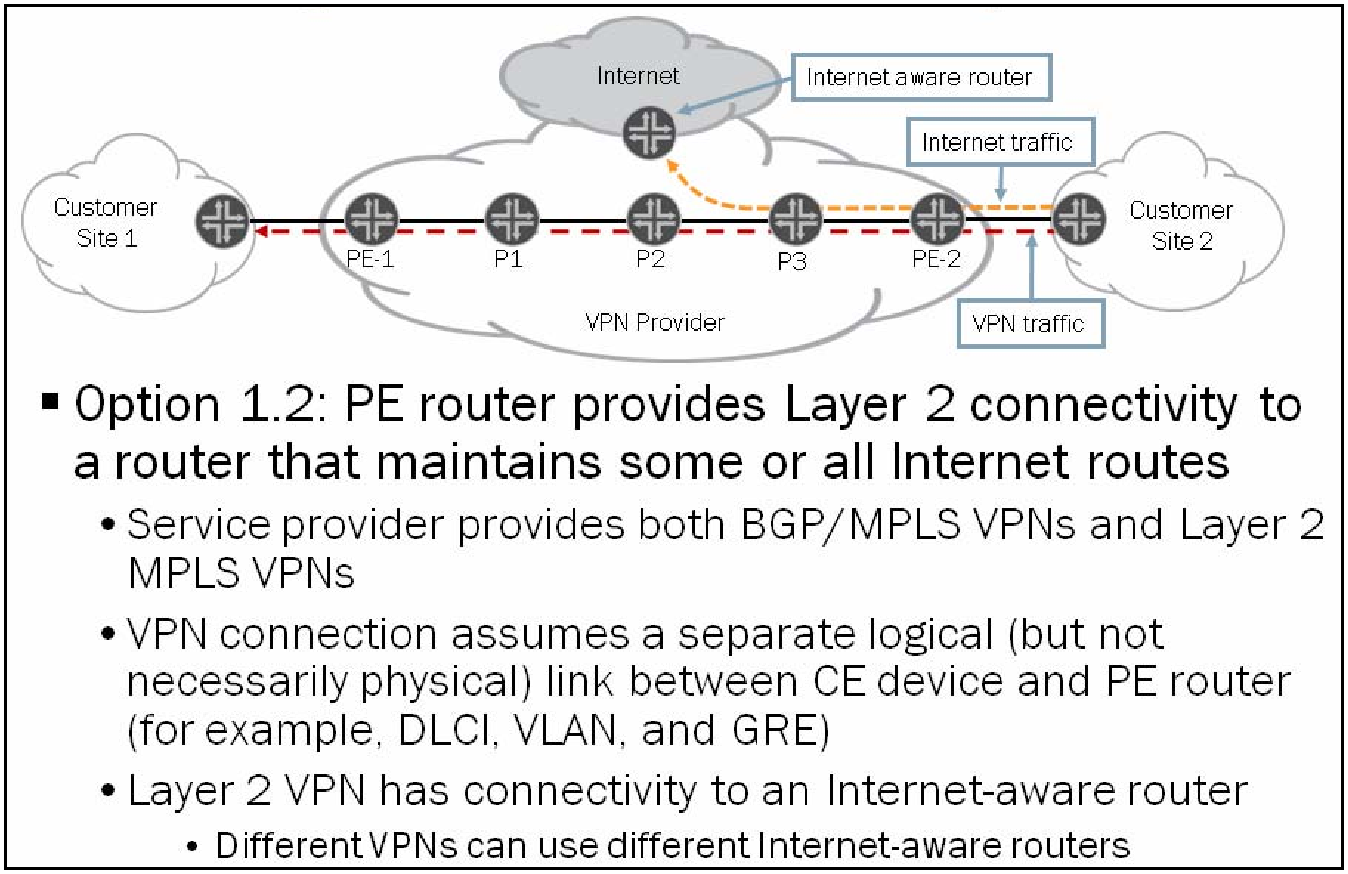
Option 1.1



In Option 1.1, the VPN service provider and PE router provide no Internet access functionality. The customer sites have separate connections for Internet access, so the PE router never receives nor transmits Internet traffic.

By default, the Junos OS supports Option 1.1.

Option 1.2

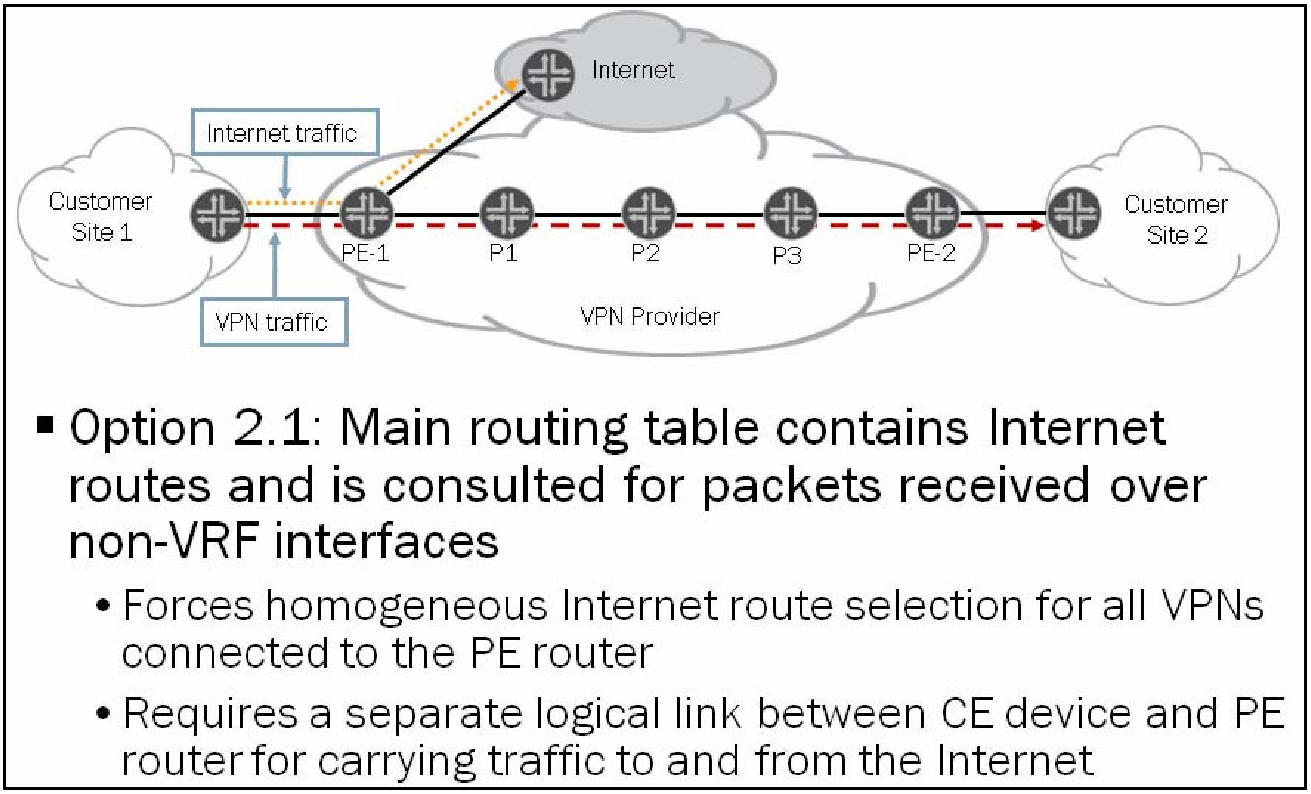


- Option 1.2: PE router provides Layer 2 connectivity to a router that maintains some or all Internet routes
  - Service provider provides both BGP/MPLS VPNs and Layer 2 MPLS VPNs
  - VPN connection assumes a separate logical (but not necessarily physical) link between CE device and PE router (for example, DLCI, VLAN, and GRE)
  - Layer 2 VPN has connectivity to an Internet-aware router
    - Different VPNs can use different Internet-aware routers

In Option 1.2, the PE router provides a Layer 2 connection to an Internet-aware router. This Layer 2 VPN connection does not require a separate physical interface, as it can function over a second logical unit on the interface used for VPN access. Each CE router can be connected to the same Internet-aware device or can be attached to separate Internet-aware provider routers.

The Junos OS supports Option 1.2 using either circuit cross-connect (CCC) or Layer 2 VPN technology.

Option 2.1: Separate Interfaces for VPN and Internet



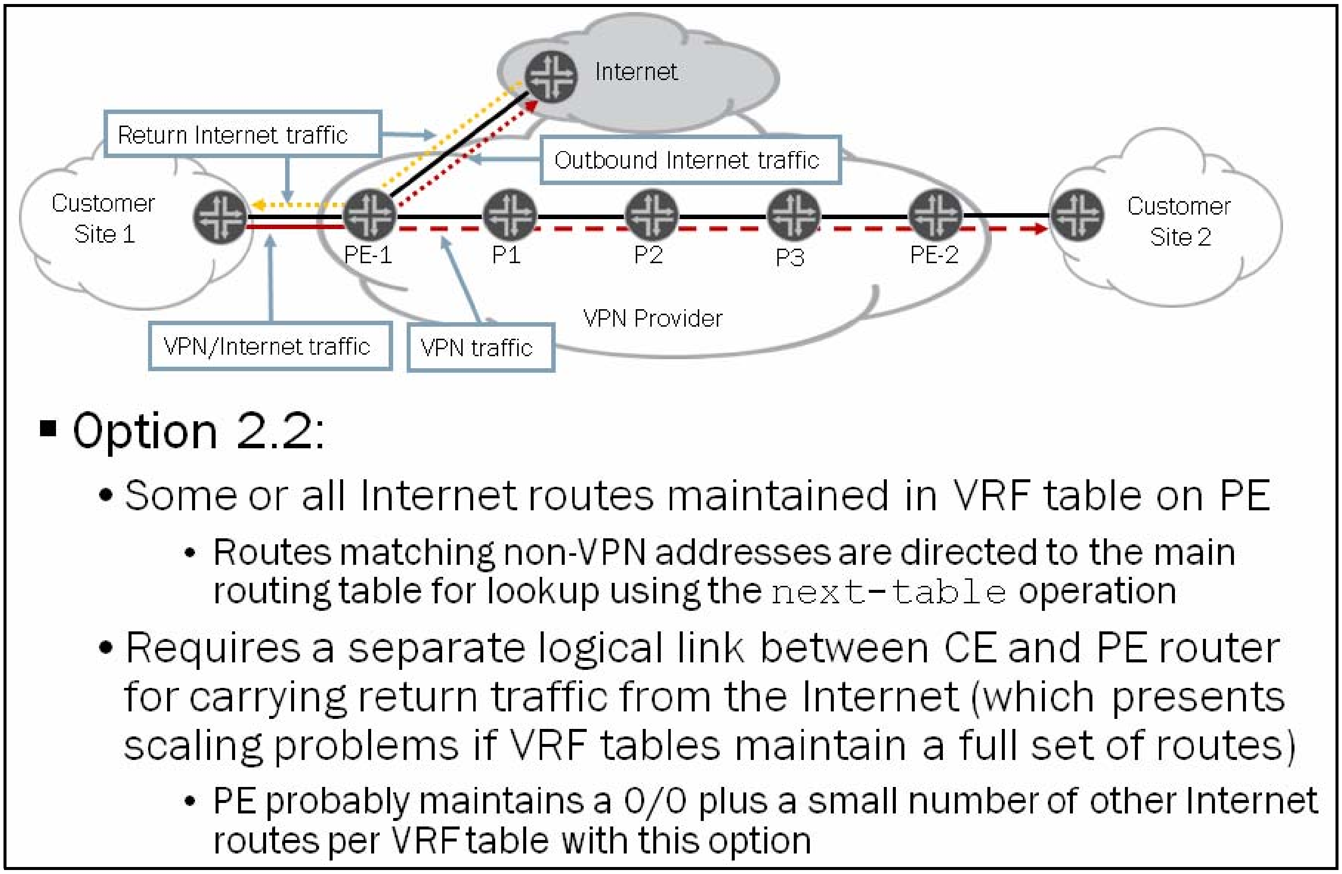
- Option 2.1: Main routing table contains Internet routes and is consulted for packets received over non-VRF interfaces
  - Forces homogeneous Internet route selection for all VPNs connected to the PE router
  - Requires a separate logical link between CE device and PE router for carrying traffic to and from the Internet

In Option 2.1, the PE router maintains partial or full Internet routes in its main routing instance. All VPN customers attached to the PE router share these routes, which forces homogenous Internet access.

With this option, the CE router attaches to the PE router using both a VRF and a non-VRF interface. Traffic received over the non-VRF interface is matched against the main routing table, while traffic received over the VRF interface is matched against the VRF table.

The VPN site’s global addresses are associated with the non-VRF interface. These routes are placed into the main routing table to attract reverse traffic. Because all VPN customers have the non-VRF interface traffic matched against a common routing table, the result is homogenous routing for Internet access.

Option 2.2: Separate Interface for Returning Internet Traffic

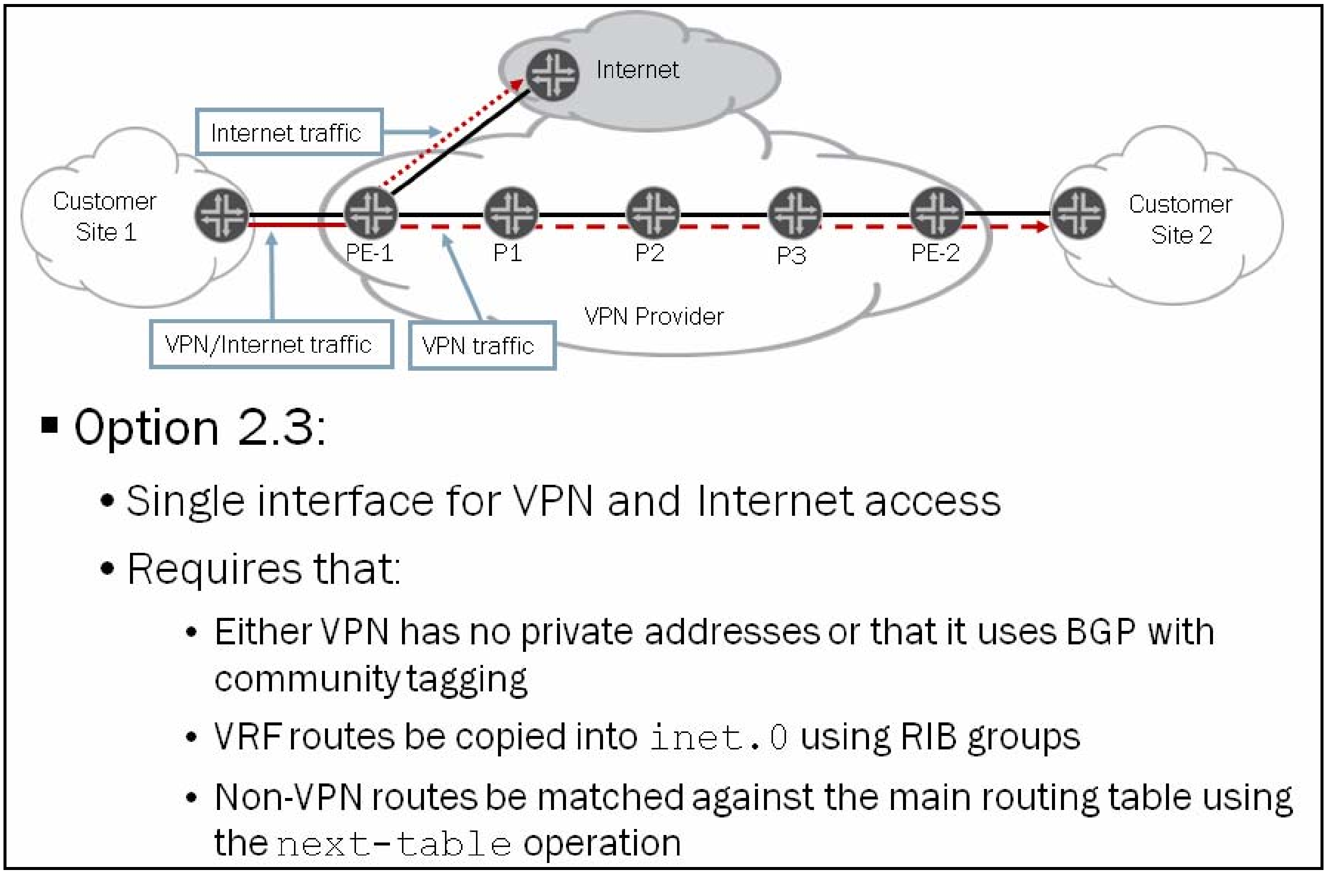


- Option 2.2:
  - Some or all Internet routes maintained in VRF table on PE
    - Routes matching non-VPN addresses are directed to the main routing table for lookup using the `next-table` operation
  - Requires a separate logical link between CE and PE router for carrying return traffic from the Internet (which presents scaling problems if VRF tables maintain a full set of routes)
    - PE probably maintains a 0/0 plus a small number of other Internet routes per VRF table with this option

In Option 2.2, the PE router and CE device once again attach with both a VRF and non-VRF interface. In this case, the CE router sends both VPN and Internet traffic to the PE router using the VRF interface. The PE router must be able to match the packet against the VRF table as well as the main routing table. The Junos OS can accomplish this matching by placing a static route (usually a default route) in the VRF table that uses the **`next-table`** option to force a second lookup in the default routing table.

Traffic returning from the Internet is matched against the main routing instance and delivered to the customer using the non-VRF interface. All the global addresses associated with the VPN site must be added to the main routing table to allow reverse traffic.

Option 2.3: Single VRF Interface for VPN and Internet Access

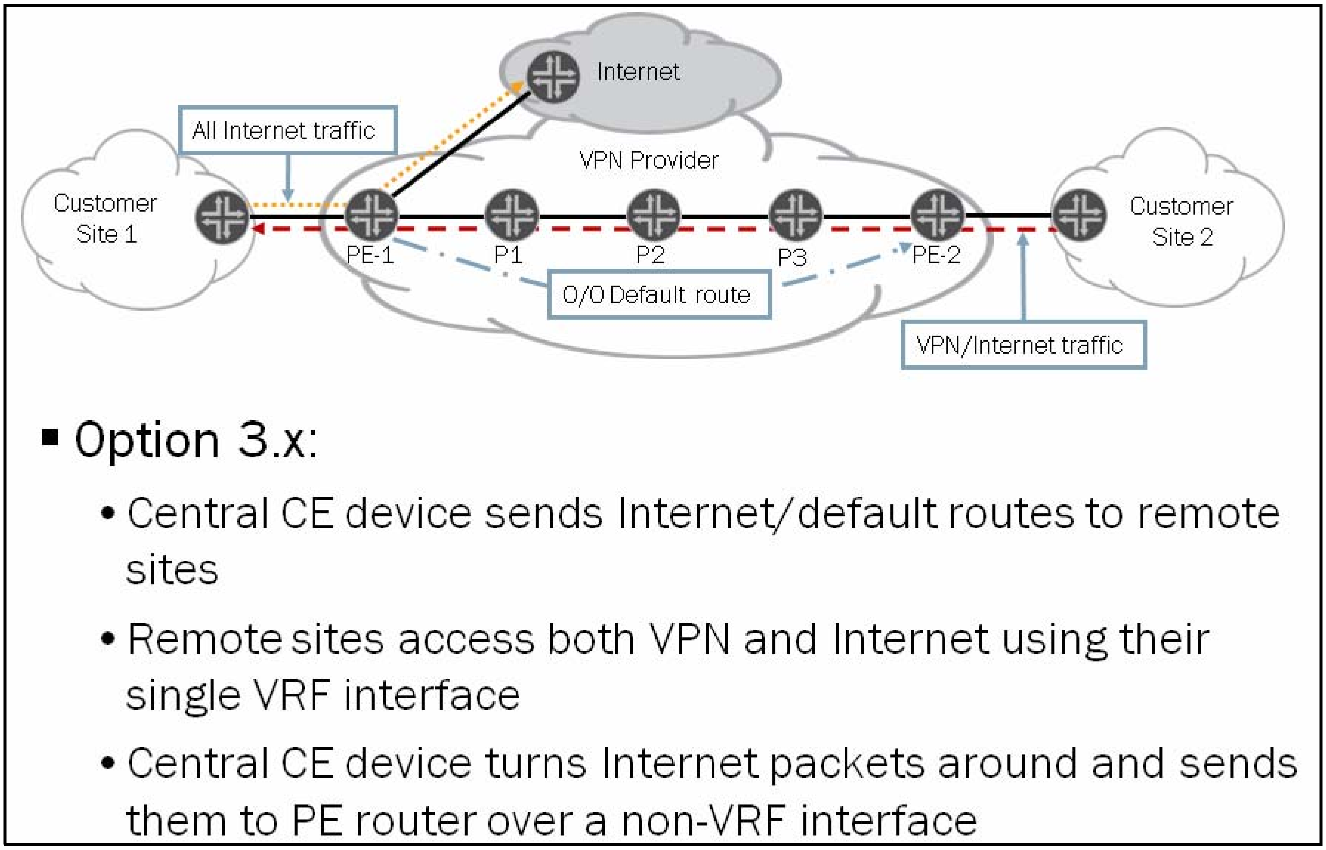


- Option 2.3:
  - Single interface for VPN and Internet access
  - Requires that:
    - Either VPN has no private addresses or that it uses BGP with community tagging
    - VRF routes be copied into `inet.0` using RIB groups
    - Non-VPN routes be matched against the main routing table using the `next-table` operation

If the VPN does not use private addresses space, both VPN and Internet access can be achieved with a single VRF interface by copying the routes from the VRF table into the main routing table using `RIB` groups. This option also requires that the VRF table carry a default route using the **`next-table`** option to direct nonmatching packets to the main routing table for longest-match lookup.

When the PE-CE protocol is BGP, Option 2.3 can be achieved when the VPN site is using a mix of private and global addressing. In this case, BGP community tags differentiate between global and private addressing. By sorting out the global from private addresses from the tags values, the global routes that are to be copied into the main routing instance can be determined.

Option 3.x: VRF Internet Access



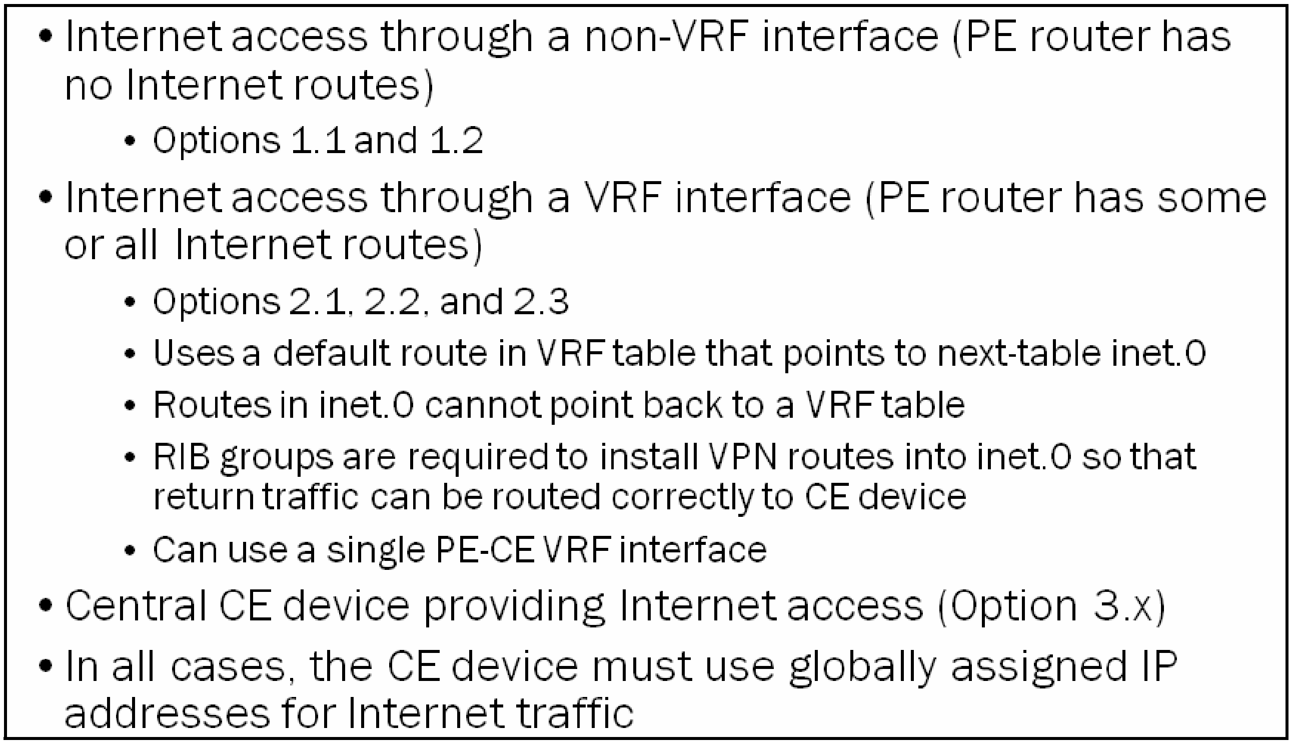
- Option 3.x:
  - Central CE device sends Internet/default routes to remote sites
  - Remote sites access both VPN and Internet using their single VRF interface
  - Central CE device turns Internet packets around and sends them to PE router over a non-VRF interface

The Option 3 solutions use a central CE site to provide Internet access for remote CE sites belonging to the same VPN. These solutions often are called VRF Internet access because from the perspective of the remote CE locations, the same VRF interface and VRF table is used to access VPN and Internet destinations.

In operation, the central CE route normally connects to the PE router using both a VRF and a non-VRF interface. The VPN's global addresses are associated with the non-VRF interface. These routes are placed in the main routing table. The central CE router then redistributes Internet routes (this can be a default route) to the remote CE locations.

Therefore, the central CE device receives both VPN and Internet traffic over its VRF interface. In the case of Internet traffic, the central CE device turns the packets around and sends them back to the PE router using its non-VRF interface.

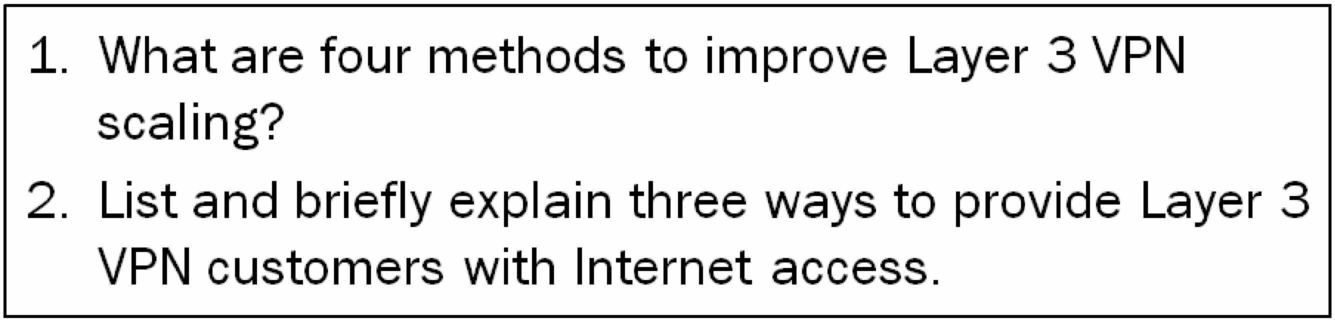
The Junos OS Internet Access Support



- Internet access through a non-VRF interface (PE router has no Internet routes)
  - Options 1.1 and 1.2
- Internet access through a VRF interface (PE router has some or all Internet routes)
  - Options 2.1, 2.2, and 2.3
  - Uses a default route in VRF table that points to next-table inet.0
  - Routes in inet.0 cannot point back to a VRF table
  - RIB groups are required to install VPN routes into inet.0 so that return traffic can be routed correctly to CE device
  - Can use a single PE-CE VRF interface
- Central CE device providing Internet access (Option 3.x)
- In all cases, the CE device must use globally assigned IP addresses for Internet traffic

This graphic summarizes the Internet access options supported by the Junos OS.

Review Questions



1. What are four methods to improve Layer 3 VPN scaling?
2. List and briefly explain three ways to provide Layer 3 VPN customers with Internet access.

Answers to Review Questions

1.

Some of the recommended methods are: observing PE router limits regarding total number of routes, keeping the CE-to-PE routing simple, using BGP route reflectors for VPN routes, using the BGP refresh option, and using route target filtering.

2.

First there is Option 1 which provides Internet access through a non-VRF interface (PE router has no Internet routes). Second there is Option 2 which provides Internet access through a VRF interface (the PE router has some or all Internet routes). And finally there is Option 3 which provides Internet access through a central CE device.

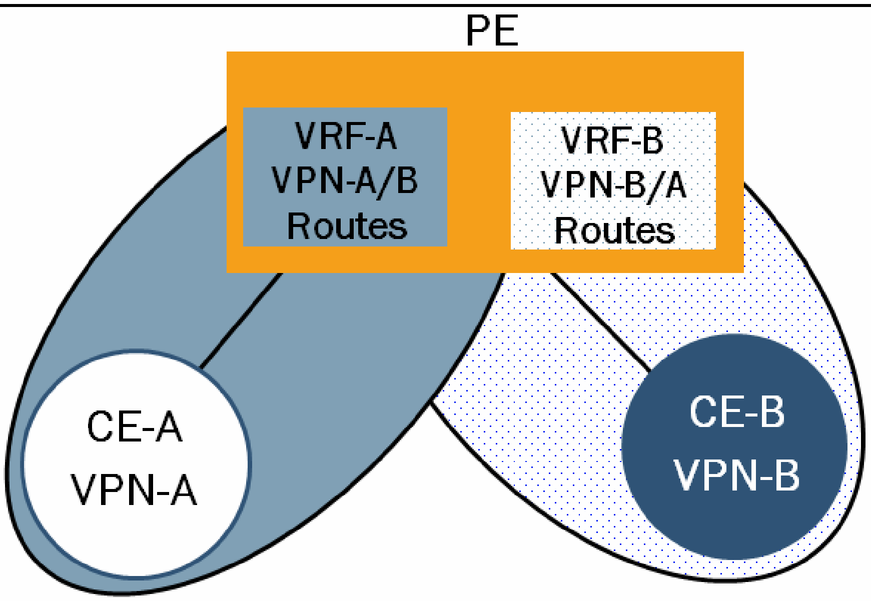
JNCIS-SP Study Guide—Part 3

Chapter 11: Layer 3 VPNs—Advanced Topics

This Chapter Discusses:

- How the auto-export command and routing table groups can be used to support communications between sites attached to a common provider edge (PE) router;
- The flow of control and data traffic in a hub-and-spoke topology;
- The various Layer 3 virtual private network (VPN) class of service (CoS) mechanisms supported by the Junos operating system; and
- Junos OS support for generic routing encapsulation (GRE) and IP Security (IPsec) tunnels in Layer 3 VPNs.

Allowing Communication



- Goal: Allow communications between CE-A and CE-B without placing them into the same VPN
- Solution: Use the `auto-export` command or RIB groups

At this point, you should be well versed in the procedures used to populate VPN routing and forwarding tables (VRFs) with routes learned from local customer edge (CE) devices and with the routes learned from remote PE routers. However, what if you want to allow communications between two different VPN sites that attach to the same PE router?

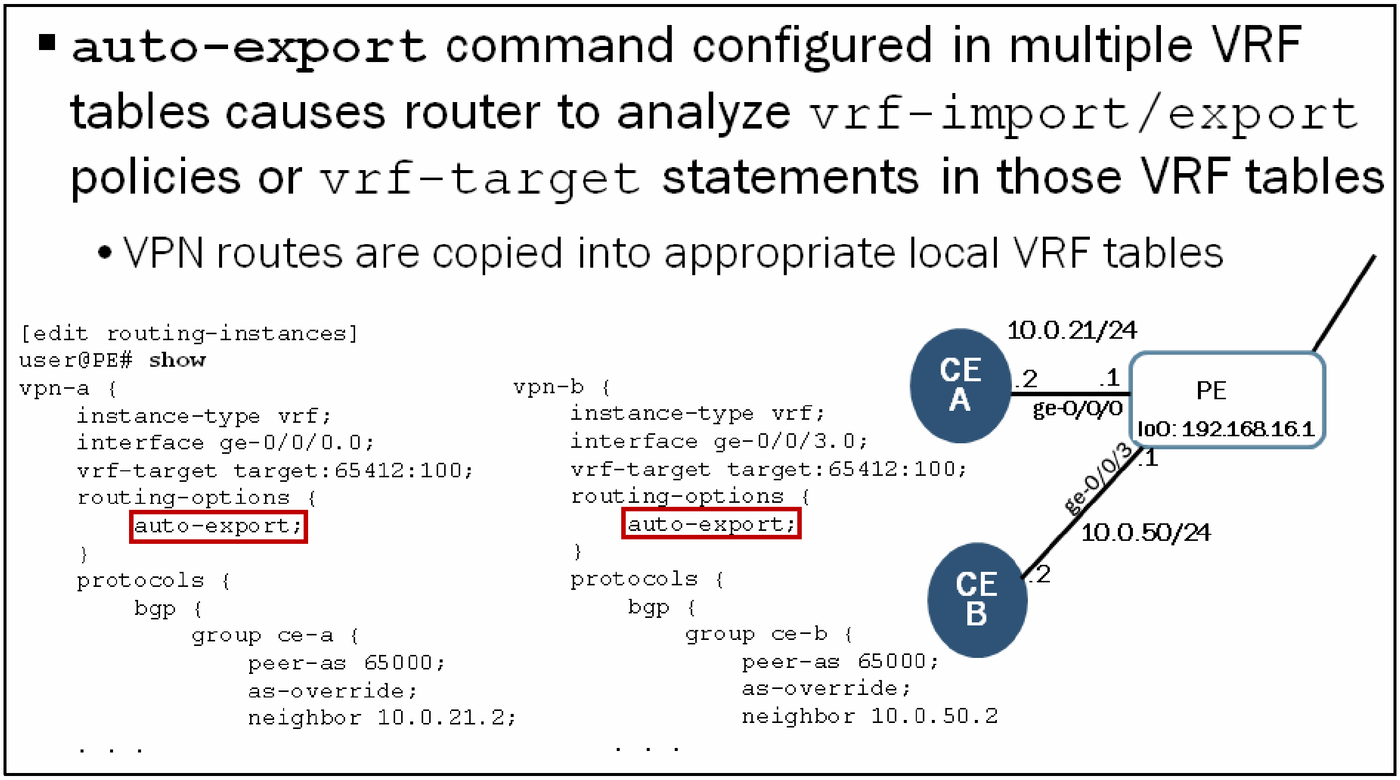
A common example why this might be necessary is a provider's network management system requiring communications with CE routers at customer sites that are attached to the same PE router. In some cases, it might be possible to resolve this dilemma by simply combining the two VRF tables into one VRF table by placing both CE routers into the same VPN. Unfortunately, this straightforward solution does not work well for the example on the graphic because administrative boundaries (which are the whole purpose of VPNs) are difficult to maintain when different VPNs suddenly merge into one VPN.

VRF policy does not solve this problem either. VRF policy normally only affects routes exchanged between PE routers. Because the sites shown on the graphic are attached to the same PE router, no Multiprotocol Border Gateway Protocol (MP-BGP) session exists to which you could even apply VRF policy. However, if you configure the **`auto-export`** command in each VRF table, the import and export VRF policies are evaluated without the need for MP-BGP sessions to exist, as described in the following pages.

Using Routing Table Groups

Another solution to this problem involves routing table groups. Routing table groups allow the linking of different routing tables within the router so that routes can be exchanged between them. The use of routing table groups to solve this problem is demonstrated as well in following pages.

`auto-export` Example



- `auto-export` command configured in multiple VRF tables causes router to analyze `vrf-import/export` policies or `vrf-target` statements in those VRF tables
  - VPN routes are copied into appropriate local VRF tables

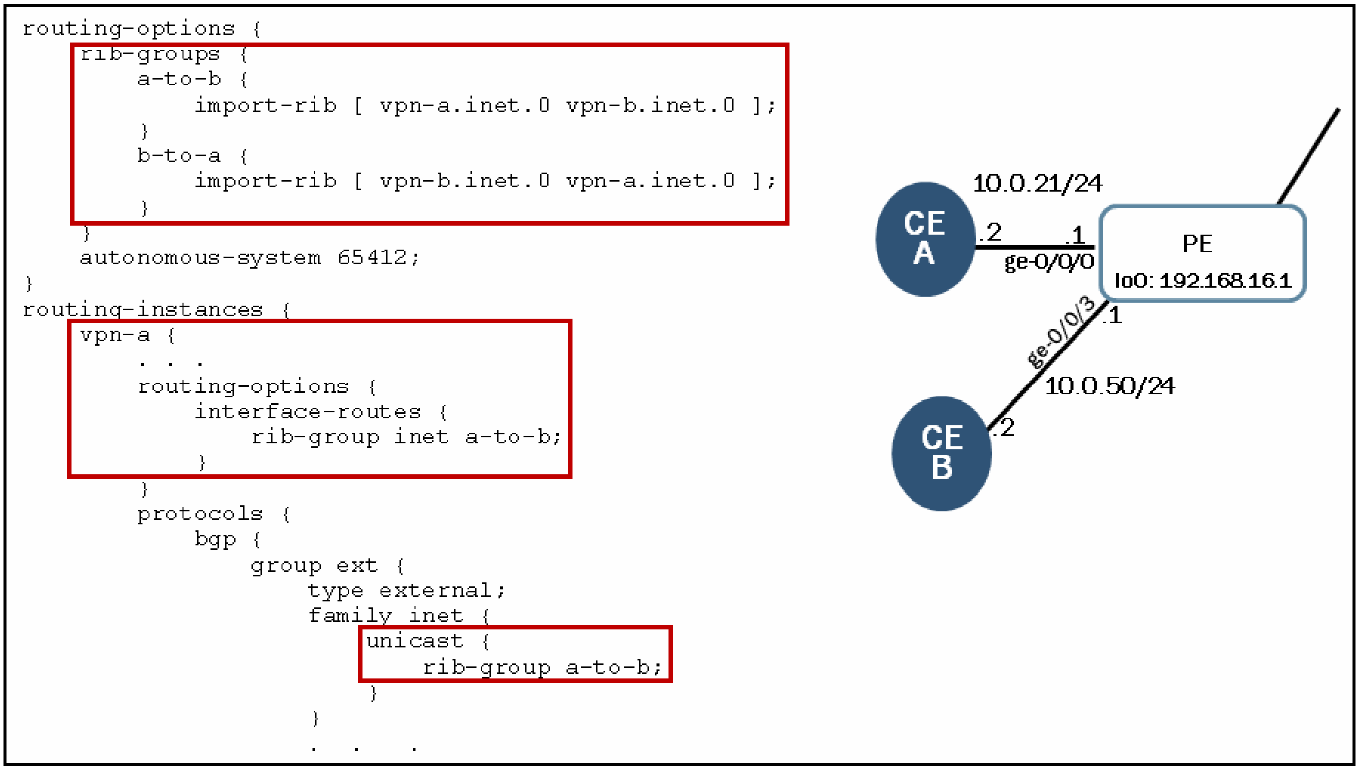
```
[edit routing-instances]
user@PE# show
vpn-a {
    instance-type vrf;
    interface ge-0/0/0.0;
    vrf-target target:65412:100;
    routing-options {
        auto-export;
    }
    protocols {
        bgp {
            group ce-a {
                peer-as 65000;
                as-override;
                neighbor 10.0.21.2;
    . . .
```

```
vpn-b {
    instance-type vrf;
    interface ge-0/0/3.0;
    vrf-target target:65412:100;
    routing-options {
        auto-export;
    }
    protocols {
        bgp {
            group ce-b {
                peer-as 65000;
                as-override;
                neighbor 10.0.50.2
    . . .
```

This graphic provides an example of how to use the **`auto-export`** command to *leak* routes between VRF tables in the same PE router. The drawing on the graphic shows the PE router that now has CE-B attached to its ge-0/0/3 interface. In each VRF table on the PE, the **`auto-export`** command is enabled. This command causes the router to analyze some combination of the `vrf-import` policy, `vrf-export` policy, and the `vrf-target` statements of each VRF table that has the **`auto-export`** command configured. Any routes with the correct target communities are then copied between these VRF tables.

In the preceding example, because both VRF tables use the same import and export VRF target, all routes in the ***vpn-a*** table are copied into the ***vpn-b*** table, and vice versa.

VRF Routing Table Group Example



```
routing-options {
    rib-groups {
        a-to-b {
            import-rib [ vpn-a.inet.0 vpn-b.inet.0 ];
        }
        b-to-a {
            import-rib [ vpn-b.inet.0 vpn-a.inet.0 ];
        }
    }
    autonomous-system 65412;
}
routing-instances {
    vpn-a {
        . . .
        routing-options {
            interface-routes {
                rib-group inet a-to-b;
            }
        }
        protocols {
            bgp {
                group ext {
                    type external;
                    family inet {
                        unicast {
                            rib-group a-to-b;
                        }
                    }
                    .  .   .
```

This graphic provides an example of how to use routing table groups to *leak* routes between VRF tables in the same PE router. The code snippet begins with the creation of two routing table groups under the `[edit routing-options]` hierarchy. In this example, the *a-to-b* routing table group is told to place its routes into its own instance (*vpn-a*`.inet.0`) and into the routing table associated with the *vpn-b*`.inet.0` instance. The same effect is configured for the opposite direction with the *b-to-a* routing table group.

When listing the `import-rib` variables, the first routing table listed is considered the owner of the routing table group. Therefore, the *vpn-a*`.inet.0` is listed before the *vpn-b*`.inet.0` in the *a-to-b* routing table group. This order prevents the *a-to-b* routing table group from functioning if it is applied later to the *vpn-b* routing instance.

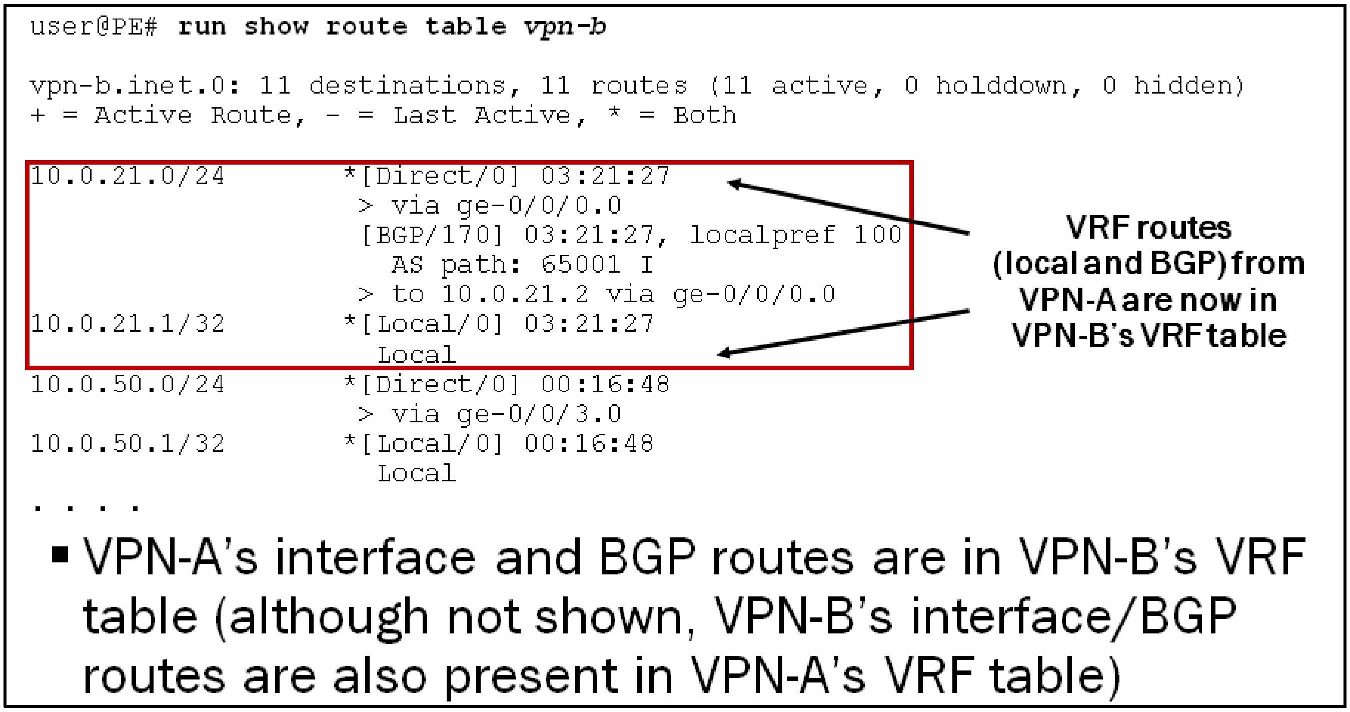
The next code snippet shows the relevant portions of the ***vpn-a*** VRF table. While the VRF table configuration for ***vpn-b*** is not shown, that instance requires similar configuration steps. In this case, the VRF instance has its **`routing-options`** configured to place the VRF interface routes into the ***a-to-b*** routing table group. This configuration is required so that the interface routes associated with each VRF table are copied into the VRF tables of the other sites with which it is to communicate. If the VRF interface routes are not copied into the other VRF tables, the routes that are copied will be unresolvable (and therefore unusable) by virtue of their pointing to an unknown interface as part of the packet's next hop.

The last relevant portion of *vpn-a*'s VRF configuration is the need to link the CE-PE routing protocol to the *a-to-b* routing table group. This step causes the BGP routes learned from CE-A to be copied into both the *vpn-a* and *vpn-b* VRF tables. EBGP, OSPF, and RIP support routing table groups. You can define static routes in each site's VRF table, or they can be specified in a routing table group that imports into the VRF tables.

The Junos OS also allows the use of policy to control the exchange of routes between routing table groups. To use this feature, include the `import-policy` option when defining the routing table groups:

```
user@PE# show routing-options
rib-groups {
    a-to-b {
        import-rib [ vpn-a.inet.0 vpn-b.inet.0 ];
        import-policy rib-policy;
    . . .
```

Verifying the Results



```
user@PE# run show route table vpn-b

vpn-b.inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

10.0.21.0/24       *[Direct/0] 03:21:27
                    > via ge-0/0/0.0
                    [BGP/170] 03:21:27, localpref 100
                      AS path: 65001 I
                    > to 10.0.21.2 via ge-0/0/0.0
10.0.21.1/32       *[Local/0] 03:21:27
                     Local
10.0.50.0/24       *[Direct/0] 00:16:48
                    > via ge-0/0/3.0
10.0.50.1/32       *[Local/0] 00:16:48
                     Local
. . . .
```

VRF routes (local and BGP) from VPN-A are now in VPN-B's VRF table

- VPN-A's interface and BGP routes are in VPN-B's VRF table (although not shown, VPN-B's interface/BGP routes are also present in VPN-A's VRF table)

To verify the results, we issued a command to display the VRF table associated with the *vpn-b* routing instance. The display confirms that the interface and BGP routes contained in the *vpn-a* VRF table are now present in the *vpn-b* VRF table. The screen capture also confirms that the interface routes associated with the *vpn-b* instance are also present in the *vpn-b* VRF table.

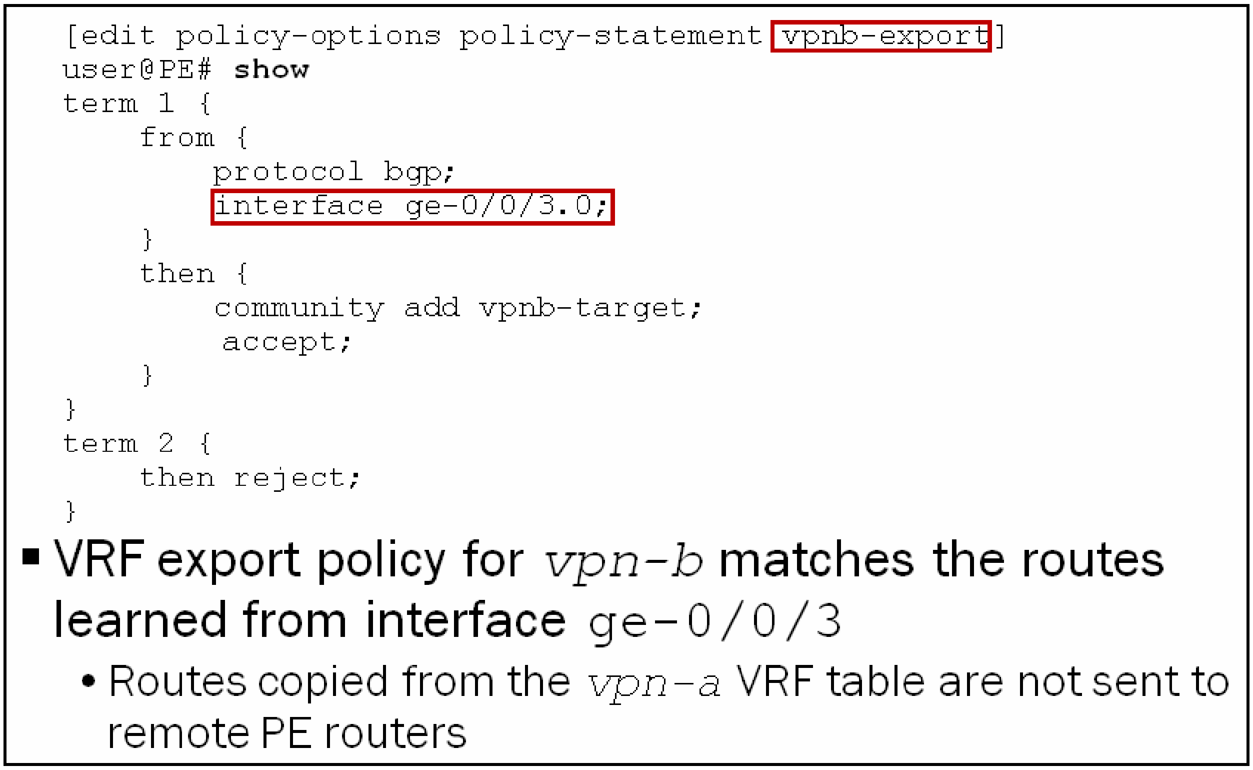
The 10.0.21/24 interface route is listed twice because it is both a direct route and a route learned through BGP (the CE-A router has a BGP policy to redistribute direct routes). Because policy is not used in this routing table group example, both routes are copied from the *vpn-a* VRF table to the *vpn-b* VRF table even though only the direct route is currently active in the *vpn-a* VRF table.

Although not shown in the graphic, the configuration steps performed under the *vpn-b* routing instance cause the interface and BGP routes in the *vpn-b* VRF table to be copied into the *vpn-a* VRF table.

Site A and Site B Can Communicate

Because both VPN sites now have routes for each other's site, the two locations can now communicate freely through the PE router.

vpn-b’s Modified VRF Export Policy: The Final Step



```
[edit policy-options policy-statement vpnb-export]
user@PE# show
term 1 {
    from {
        protocol bgp;
        interface ge-0/0/3.0;
    }
    then {
        community add vpnb-target;
         accept;
    }
}
term 2 {
    then reject;
}
```

- VRF export policy for `vpn-b` matches the routes learned from interface `ge-0/0/3`
  - Routes copied from the `vpn-a` VRF table are not sent to remote PE routers

Now that we have the two VRF tables sharing routes, the question might arise as to how we can keep these routes from being sent to remote PE routers. Assuming this is a problem, the answer is making an easy modification to the VRF export polices of the affected VRF tables.

The example shows ***vpn-b***’s VRF export policy, which now includes an interface condition in **`term 1`**’s **`from`** clause. The result is that only routes learned from the ge-0/0/3 interface are accepted for export to remote PE routers. This result prevents the ***vpn-b*** instance from advertising routes leaked from the ***vpn-a*** VRF table.

Reduces the Number of BGP Sessions and LSPs Required

Layer 3 VPNs can be deployed in a hub-and-spoke topology in which remote sites communicate through the hub site CE router. This topology is well suited for centralized data processing environments where spoke-to-spoke communications are the exception rather than the norm. A hub-and-spoke VPN has the added advantage of reducing BGP peering and LSP requirements in that spoke locations only require a single BGP session and LSP to the hub site. The hub site must support $n-1$ LSPs and BGP sessions, however, because it must connect back to each spoke site.

Two VRF Instances Required at Hub

For proper operation, the hub PE router requires two VRF instances. The spoke instance receives routes from the spoke locations and conveys them to the hub CE router. The hub instance receives routes from the hub CE router and redistributes them out to the spoke sites.

Two VRF Interfaces Required at Hub

A separate VRF interface is required to back up each VRF instance in the hub PE router. In practice, this interface is normally one physical interface with two logical units.

Two Route Targets Needed

The hub-and-spoke topology uses two route targets. Spoke sites advertise routes to the spoke instance using one route target and receive routes from the hub instance with another route target. You can implement a hub-and-spoke topology with a single route distinguisher used for both the hub and spoke instances, but the presence of route reflection forces a unique route distinguisher value for each instance. This requirement is needed to ensure that the route reflector does not attempt to compare the routes advertised (and choose a best route) by the two instances.

AS Path Loops and Domain ID Issues

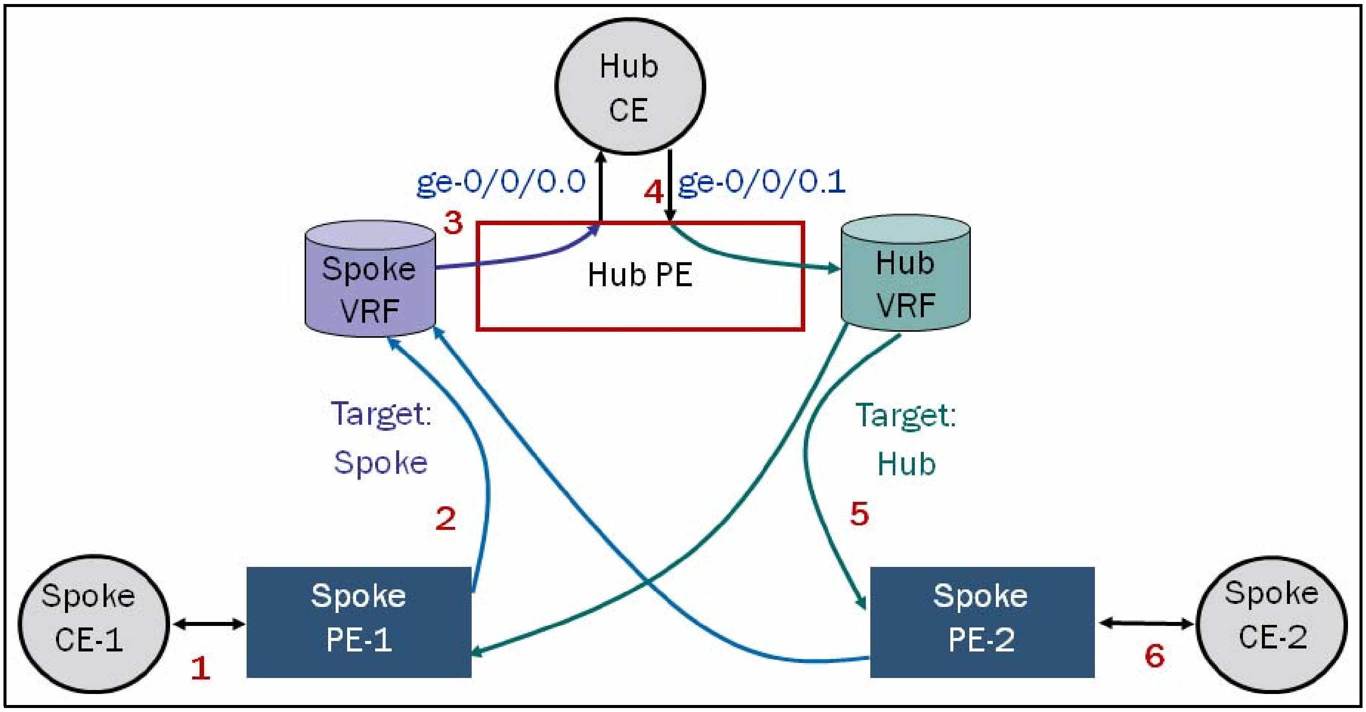
The use of BGP in a hub-and-spoke topology can result in problems with AS loop detection. Enabling autonomous system (AS) `loops` on the hub PE router might be required, even when using `as-override` and `remove-private`.

The use of OSPF as the hub PE-CE routing protocol can present problems due to the up/down bit that prevents link-state advertisement (LSA) looping. A PE router that receives an LSA with this bit set will not install the corresponding route. By default, this bit is set on all LSAs that the PE router advertises to the CE router. You can disable this functionality by explicitly configuring **`domain-vpn tag 0`**. Hub sites must manually configure this VPN route tag in their *spoke* instance so that the *hub* instance will install the routes to spoke CE routers.

Locally Attached Spokes

The presence of multiple spokes attached to the same PE router, or a spoke site attached to the hub PE router, requires additional configuration steps to ensure the hub CE device is transited for spoke-to-spoke communications.

Signaling Flow Between Spoke Locations



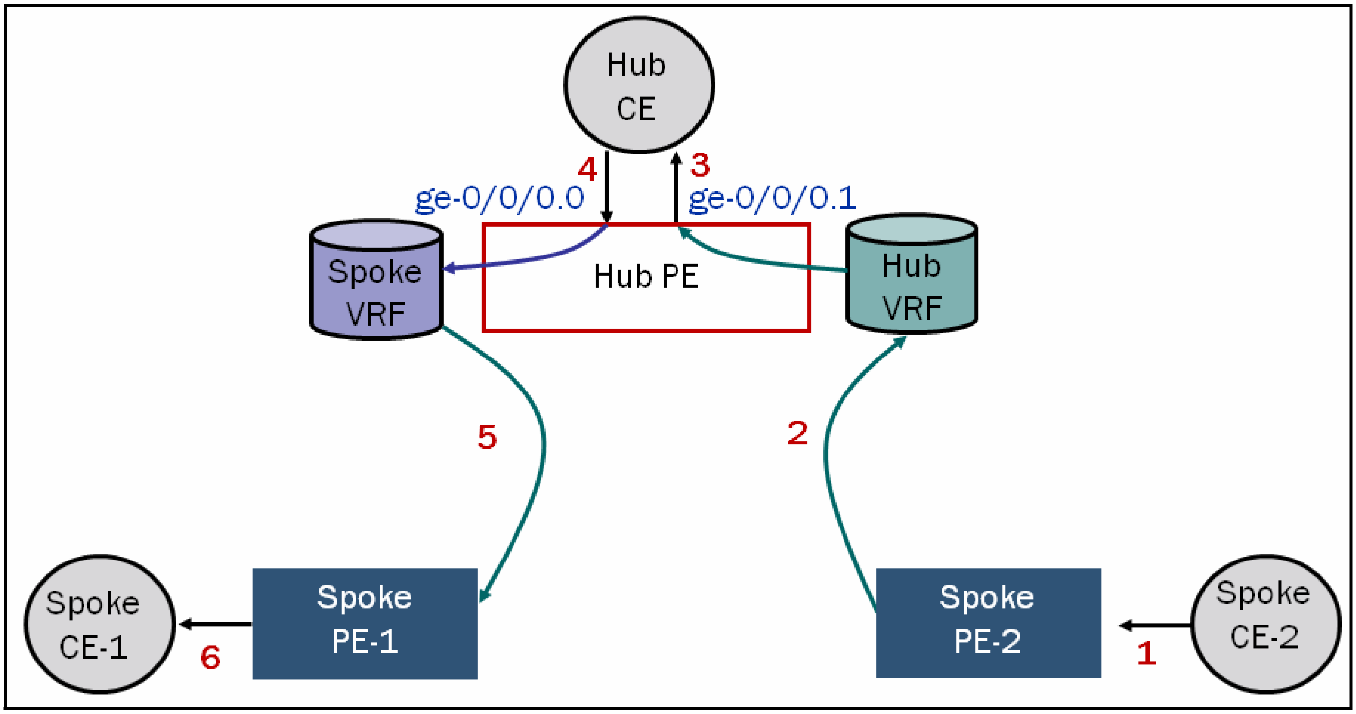
This graphic highlights the flow of signaling (routing protocol exchanges) between two spoke locations. The result is that spokes learn each other's routes through the hub PE router, thereby causing the hub CE router to act as a transit point for all traffic between spoke locations.

The following list provides details of the signaling flow shown in the graphic:

1. Spoke CE-1 advertises a route.
2. Spoke PE-1 advertises this route to the spoke instance on the hub PE router using the spoke route target.
3. The spoke instance in the hub PE router sends the route to the hub CE router using the ge-0/0/0.0 VRF interface.
4. The hub CE router either readvertises the route or generates an aggregate for all spoke sites, which is sent to the hub PE router's hub instance using the ge-0/0/0.1 VRF interface.
5. The hub instance in the hub PE router advertises this route to the spoke sites using the hub route target.

The spoke sites match the routes with the hub route target and install the route in their VRF table. For spoke PE-2, the route is sent to the attached spoke CE router (CE-2).

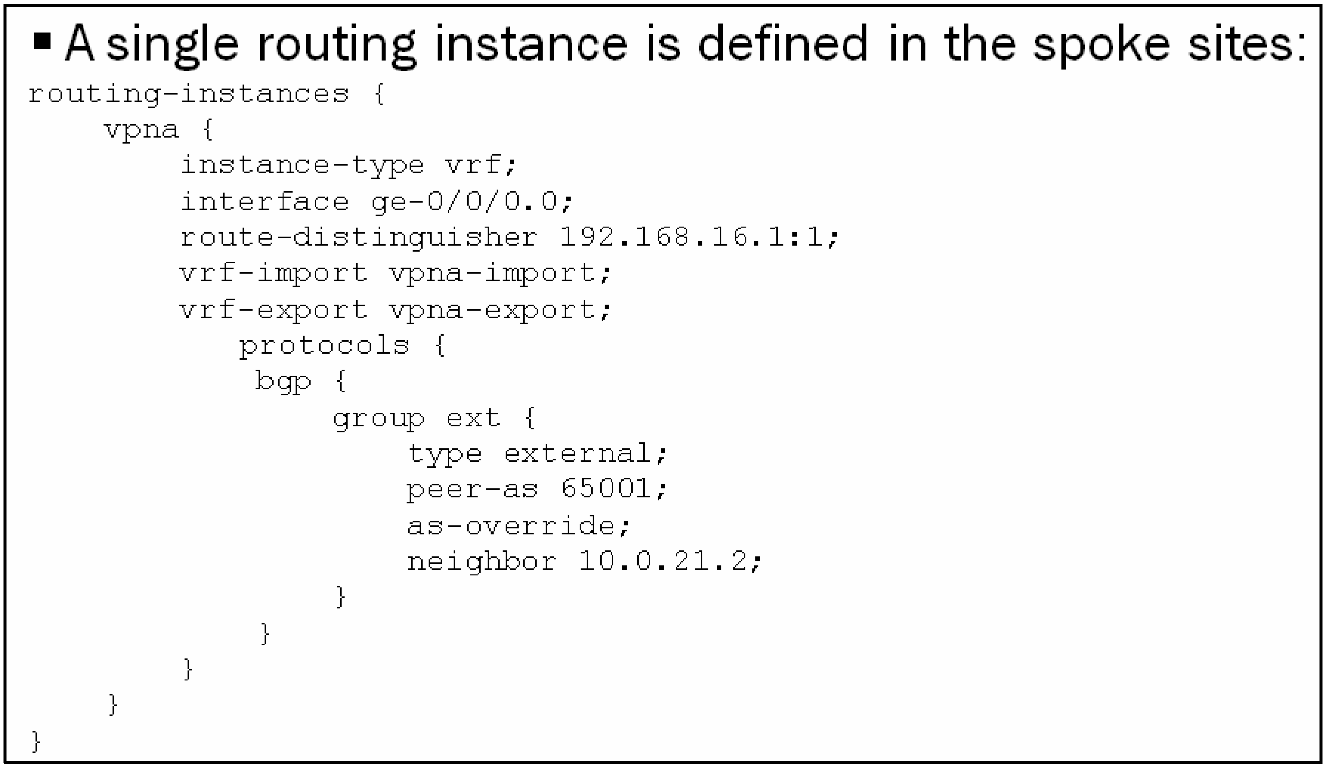
Data Flow Between Spoke Locations



This graphic highlights the flow of data (forwarding plane) between two spoke locations. The following list provides the details of this flow:

1. CE-2 sends a packet addressed to the CE-1 site.
2. PE-2 has learned the routes for Site 1 through the hub instance, so it forwards the packet to the hub PE router.
3. The packet is received by the hub PE router's hub instance. It is forwarded out the ge-0/0/0.1 VRF interface, because the hub instance has learned these routes from the hub CE router.
4. The hub CE router has learned about Site 1's routes from the hub PE router's spoke instance. Therefore, the packet is turned around by the hub CE router and is sent back to the hub PE router on the ge-0/0/0.0 VRF interface.
5. The spoke instance in the hub PE router forwards the packet to spoke PE-1.
6. Spoke PE-1 forwards the packet to CE-1.

Sample Spoke VRF Table

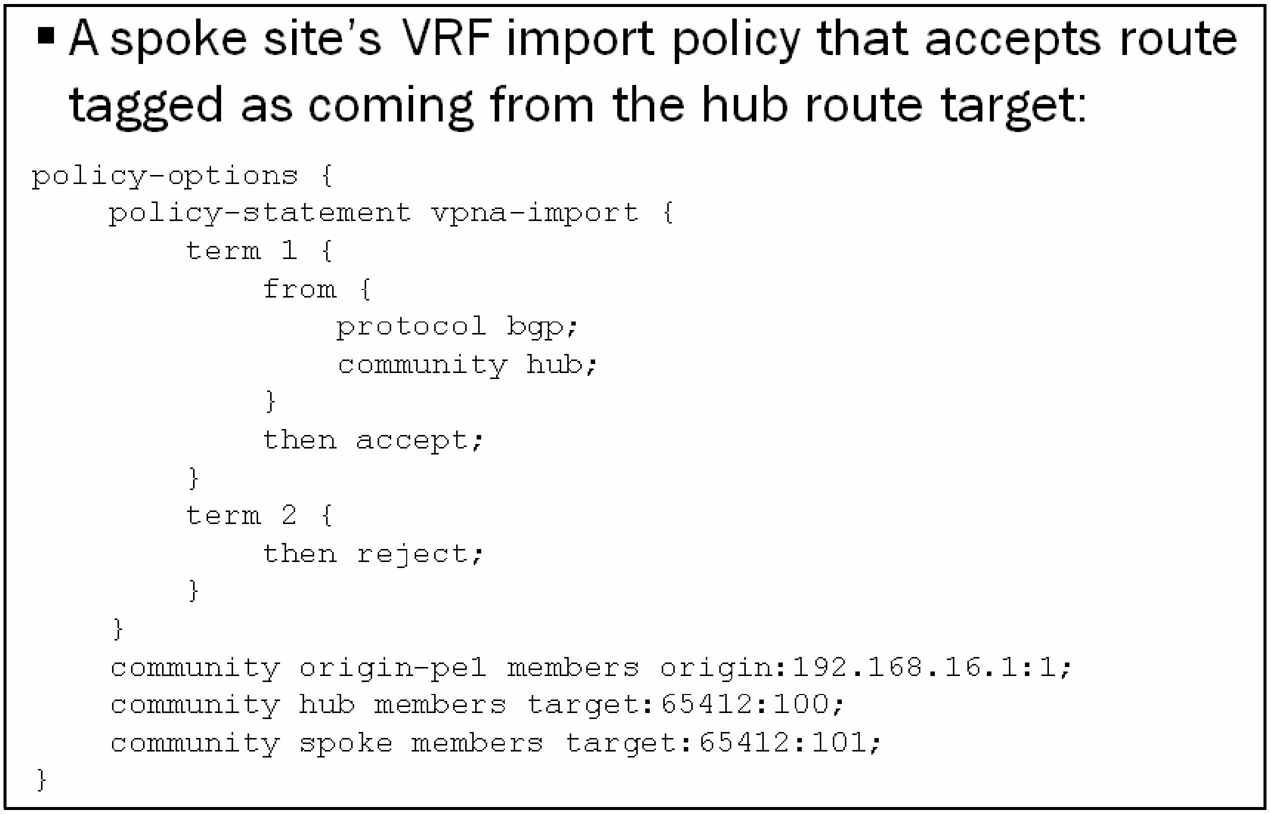


- A single routing instance is defined in the spoke sites:

```
routing-instances {
    vpna {
        instance-type vrf;
        interface ge-0/0/0.0;
        route-distinguisher 192.168.16.1:1;
        vrf-import vpna-import;
        vrf-export vpna-export;
           protocols {
            bgp {
                group ext {
                    type external;
                    peer-as 65001;
                    as-override;
                    neighbor 10.0.21.2;
                }
            }
        }
    }
}
```

This graphic provides an example of a spoke PE router's VRF table configuration. Only one instance is required for spoke sites. In this example, the PE-CE routing protocol is EBGP.

Sample Spoke VRF Import Policy

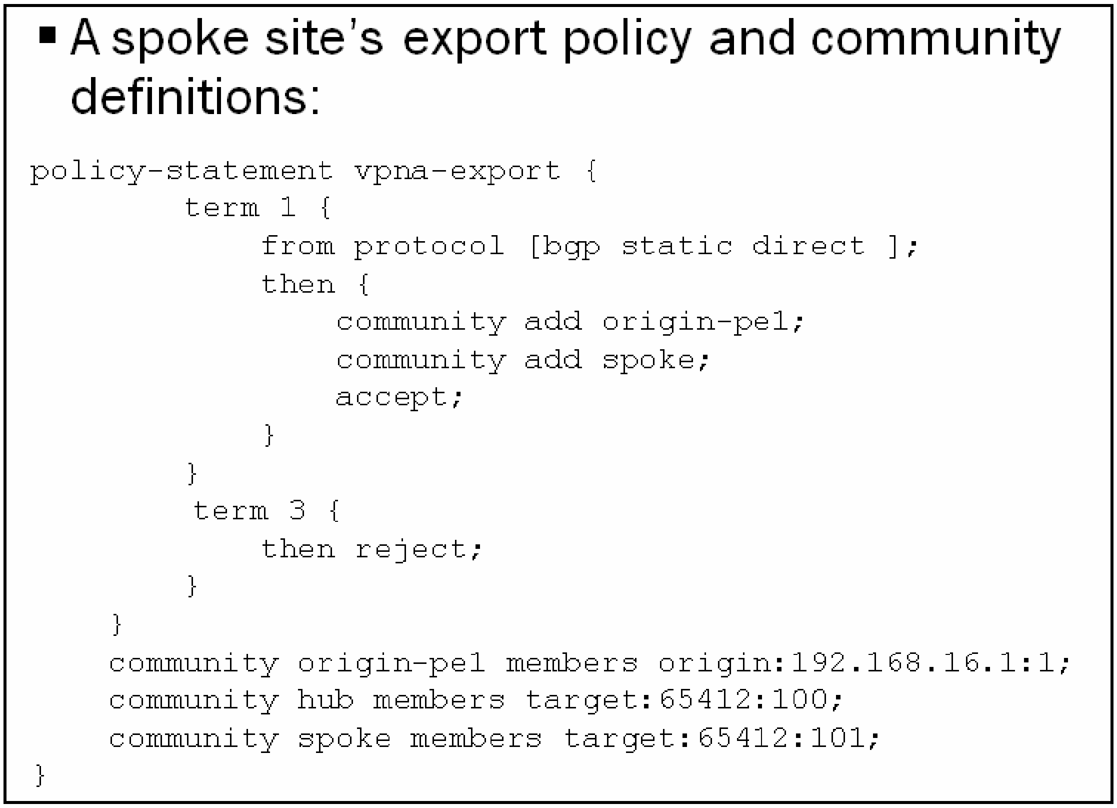


- A spoke site's VRF import policy that accepts route tagged as coming from the hub route target:

```
policy-options {
    policy-statement vpna-import {
        term 1 {
            from {
                protocol bgp;
                community hub;
            }
            then accept;
        }
        term 2 {
            then reject;
        }
    }
    community origin-pe1 members origin:192.168.16.1:1;
    community hub members target:65412:100;
    community spoke members target:65412:101;
}
```

This graphic shows a spoke site's VRF import policy set to match the routes with the hub route target.

Sample Spoke VRF Export Policy



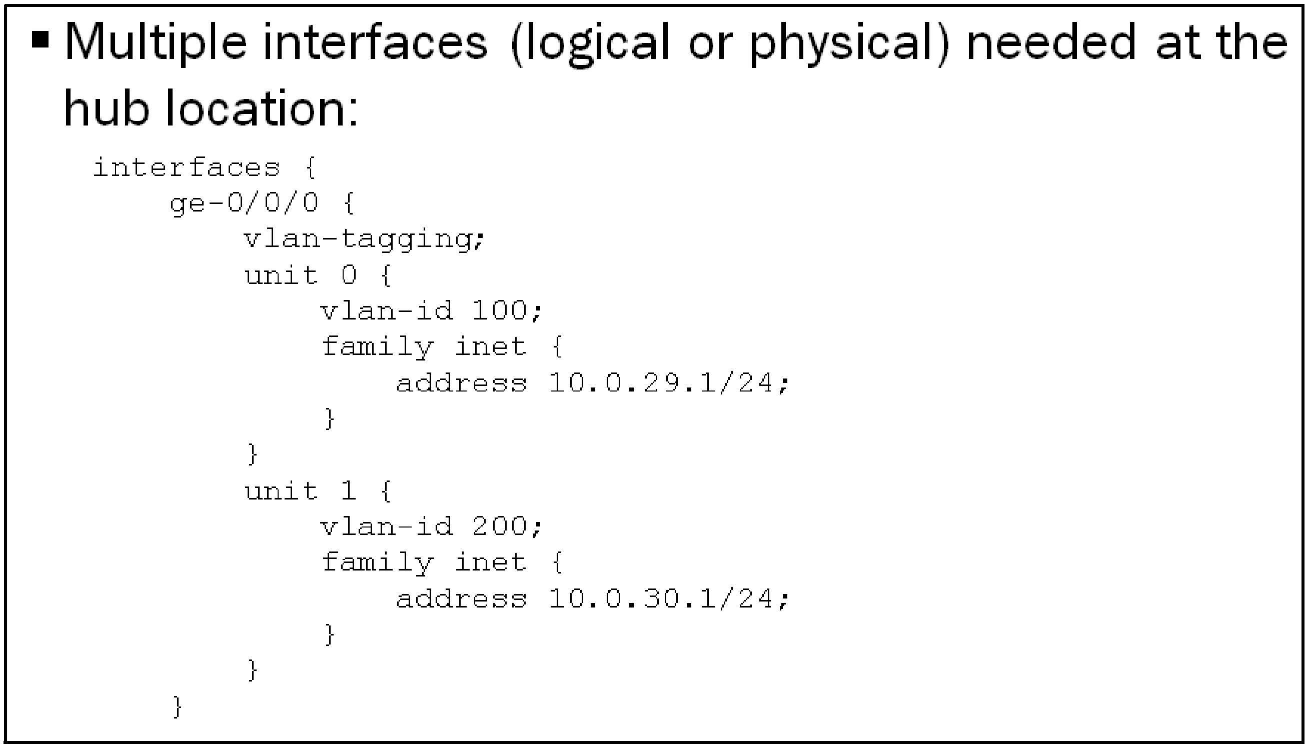
- A spoke site's export policy and community definitions:

```
policy-statement vpna-export {
        term 1 {
            from protocol [bgp static direct ];
            then {
                community add origin-pe1;
                community add spoke;
                accept;
            }
        }
         term 3 {
            then reject;
        }
    }
    community origin-pe1 members origin:192.168.16.1:1;
    community hub members target:65412:100;
    community spoke members target:65412:101;
}
```

This graphic shows that a spoke site's VRF export policy is configured to attach the spoke route target to the advertisements it sends to the hub PE router.

This example also shows the extended community definitions, including both a hub and a spoke route target.

Sample Hub Configuration: VRF Interfaces

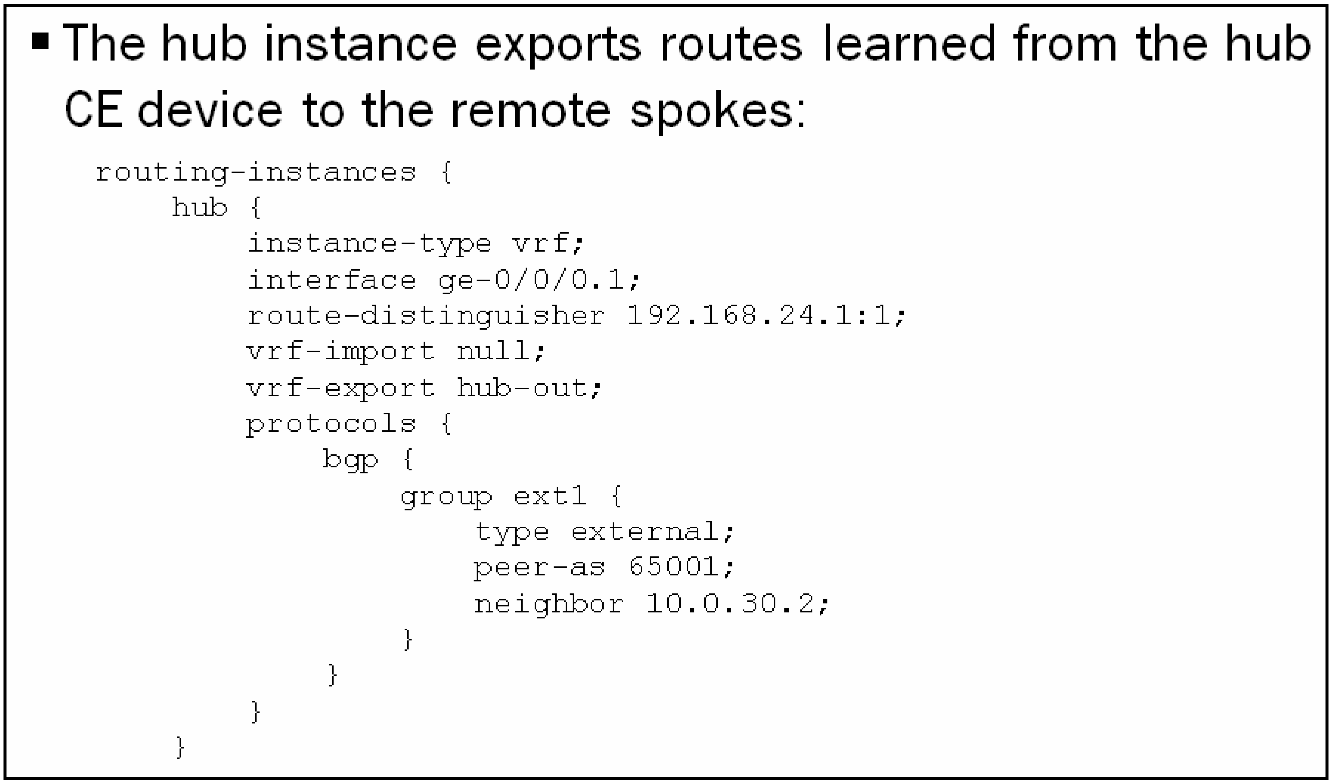


- Multiple interfaces (logical or physical) needed at the hub location:

```
interfaces {
    ge-0/0/0 {
        vlan-tagging;
        unit 0 {
            vlan-id 100;
            family inet {
                address 10.0.29.1/24;
            }
        }
        unit 1 {
            vlan-id 200;
            family inet {
                address 10.0.30.1/24;
            }
        }
    }
```

This portion of the hub PE router's configuration shows that two virtual LAN (VLAN)-tagged logical interfaces are provisioned to support the two routing instances required by the hub PE router.

Sample Hub Configuration: Hub Instance



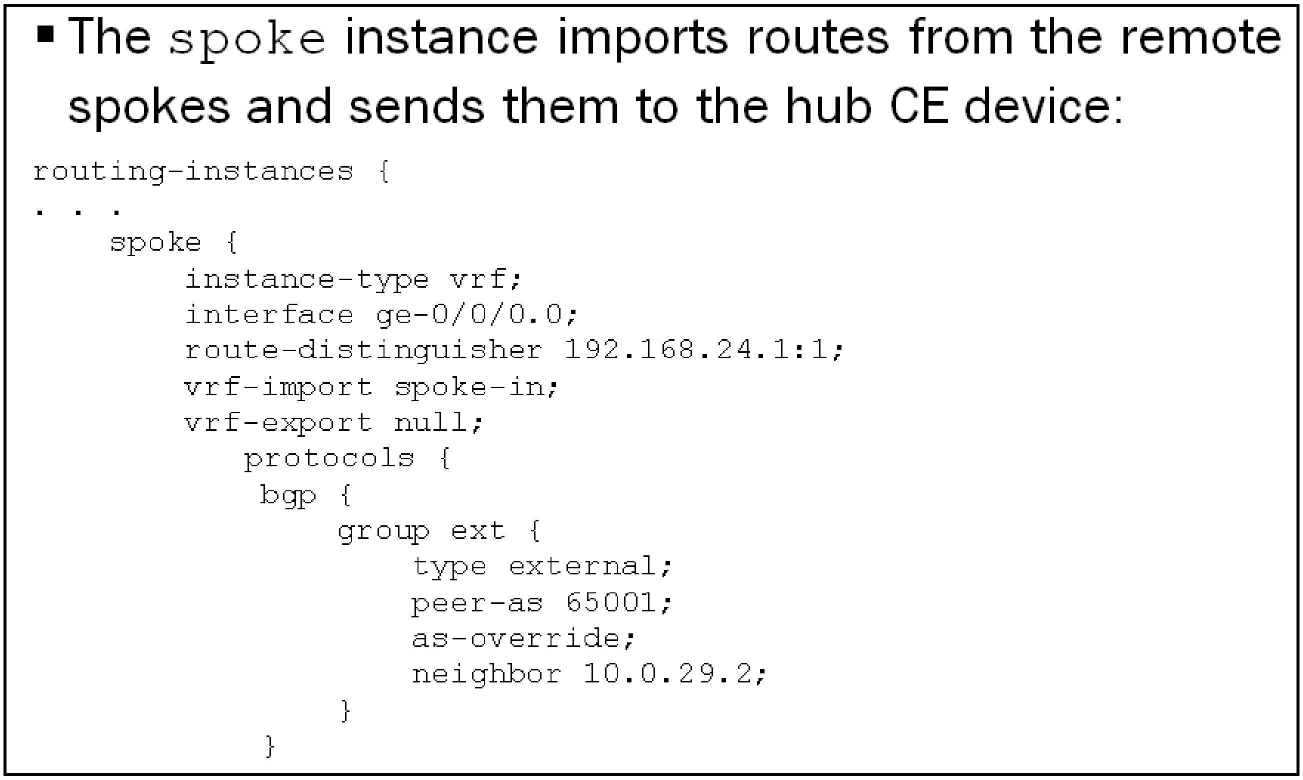
- The hub instance exports routes learned from the hub CE device to the remote spokes:

```
routing-instances {
    hub {
        instance-type vrf;
        interface ge-0/0/0.1;
        route-distinguisher 192.168.24.1:1;
        vrf-import null;
        vrf-export hub-out;
        protocols {
            bgp {
                group ext1 {
                    type external;
                    peer-as 65001;
                    neighbor 10.0.30.2;
                }
            }
        }
    }
```

This graphic displays the hub PE router’s hub VRF configuration. This instance is tied to the hub PE router’s ge-0/0/0.1 VRF interface and is configured for EBGP routing exchange with the hub CE router.

Because spoke routes are learned by the hub site’s spoke VRF instance, the hub instance uses a **`null`** VRF import policy. As shown on subsequent sections, this policy requires that a policy statement named ***`null`*** be configured with a single **`then reject`** statement.

Sample Hub Configuration: Spoke Instance



- The `spoke` instance imports routes from the remote spokes and sends them to the hub CE device:

```
routing-instances {
. . .
    spoke {
        instance-type vrf;
        interface ge-0/0/0.0;
        route-distinguisher 192.168.24.1:1;
        vrf-import spoke-in;
        vrf-export null;
           protocols {
            bgp {
                group ext {
                    type external;
                    peer-as 65001;
                    as-override;
                    neighbor 10.0.29.2;
                }
            }
```

This graphic displays the hub PE router’s `spoke` VRF table configuration. This instance is tied to the hub PE router’s ge-0/0/0.0 VRF interface and also is configured for EBGP routing exchange with the hub CE router.

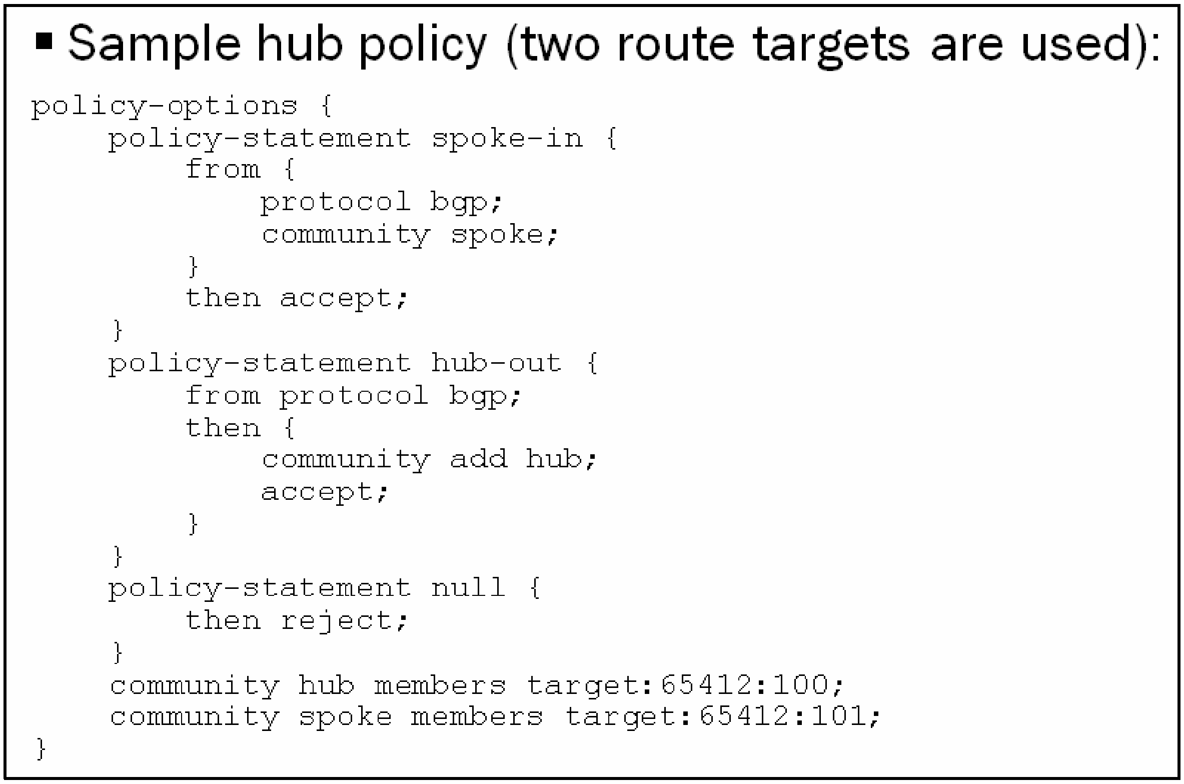
Because the hub site’s hub VRF instance advertises spoke routes, the spoke instance is using a `null` VRF export policy. As shown on subsequent graphics, this policy requires that a policy statement named *`null`* be configured with a single `then reject` statement.

Because EBGP is used on the hub’s PE-CE link, AS-path loop detection is a problem. In this case, the use of the `as-override` knob prevents loop detection problems as the spoke routes are delivered to the hub CE router through the spoke instance. However, because the provider’s AS number is now at the front of the AS path, when the hub CE router readvertises the routes back to the hub PE router’s hub instance, the hub PE router detects an AS loop and discard the routes. Therefore, you should observe the following guideline:

- Do not use EBGP at the hub site.
- Configure AS `loops 2` on the hub PE router’s hub instance.

Configure the hub CE router with static routes (which can be aggregates) redistributed into the hub CE device’s hub instance EBGP session. Because these routes originate at the hub CE router, the provider’s AS number is not present in the AS path.

Sample Hub Configuration: VRF Policy

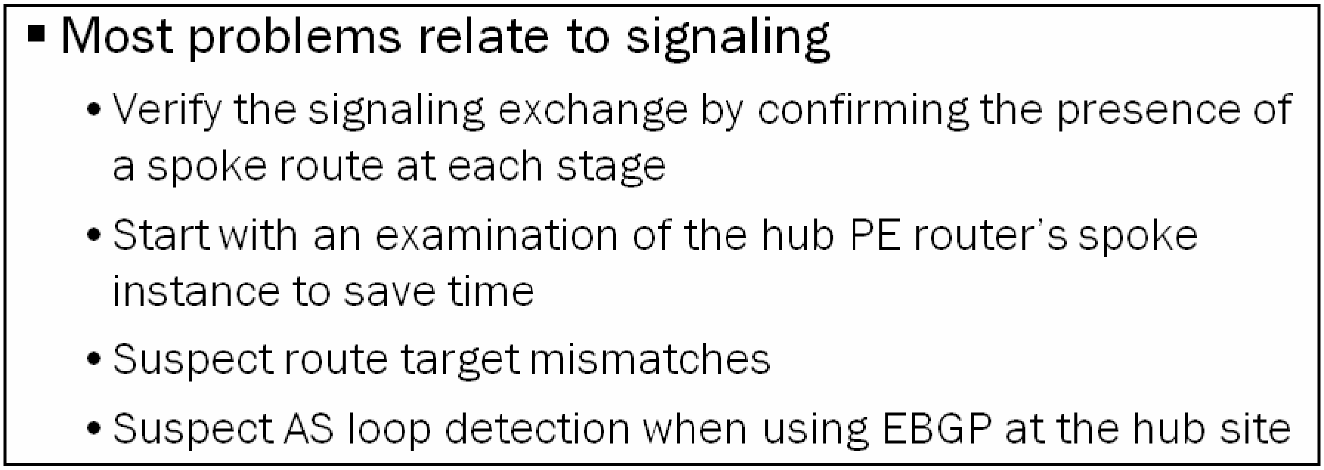


- Sample hub policy (two route targets are used):

```
policy-options {
    policy-statement spoke-in {
        from {
            protocol bgp;
            community spoke;
        }
        then accept;
    }
    policy-statement hub-out {
        from protocol bgp;
        then {
            community add hub;
            accept;
        }
    }
    policy-statement null {
        then reject;
    }
    community hub members target:65412:100;
    community spoke members target:65412:101;
}
```

This graphic displays the hub PE router's VRF policy and extended BGP community definitions. The hub's spoke-in policy matches the routes with the spoke route target, while the hub-out policy adds the hub community. The spoke VRF policy configuration in effect reverses the above policies by attaching the spoke community on advertised routes and matching the routes learned from the hub community for received routes.

Most Problems Relate to Signaling Exchanges



- Most problems relate to signaling
    - Verify the signaling exchange by confirming the presence of a spoke route at each stage
    - Start with an examination of the hub PE router's spoke instance to save time
    - Suspect route target mismatches
    - Suspect AS loop detection when using EBGP at the hub site

Because the signaling plane is more complex than the forwarding plane, and because forwarding cannot work when signaling is broken, you should approach hub-and-spoke troubleshooting by first verifying proper signaling flows.

While complex in its entirety, breaking down the signaling into discrete steps makes signaling verification a manageable task. For example, if the spoke route is in the local spoke CE device's VRF table but not in the hub PE router's spoke instance, the problem must relate to either that spoke's advertisements (VRF export) or the hub PE router's reception (VRF import).

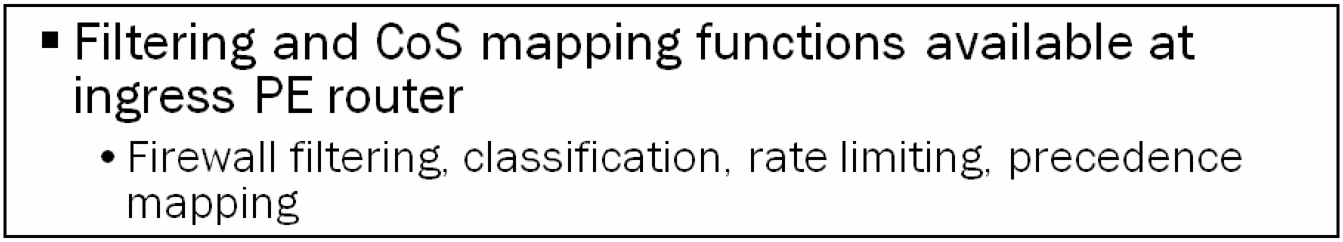
By examining the hub PE router's spoke VRF instance first, you can verify nearly one half of the total signaling exchange in one step. Eliminating half of all possible causes with each test is a prime way of expediting the fault isolation process.

Because of the requirement for two route targets, and the likelihood of AS loop detection when EBGP is provisioned at the hub PE-CE link, you always should suspect these two areas as likely causes for operational problems.

Traceroute from Spoke to Hub First

When a traceroute between two spoke locations fails, it is often difficult to determine the location of the problem. Because spoke-to-spoke communications must transit the hub location, first verify that each spoke location can communicate successfully with the hub site. When two spokes can reach the hub, but not each other, the problem normally lies in the hub CE device operation, as it would relate to the re-advertisement of the spoke routes.

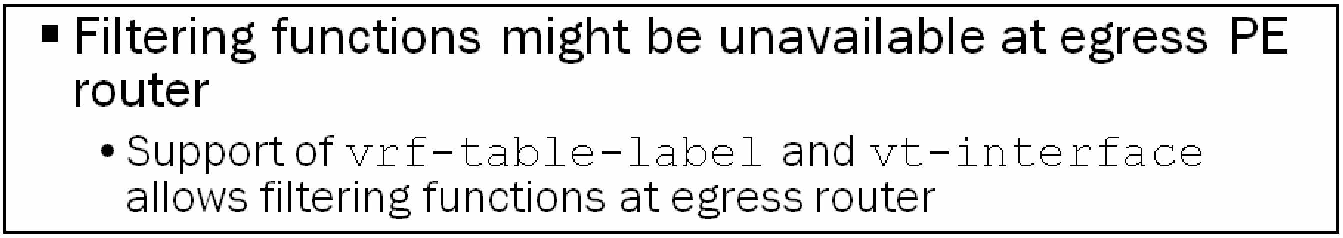
Filtering and CoS Functions Available at Ingress



- Filtering and CoS mapping functions available at ingress PE router
  - Firewall filtering, classification, rate limiting, precedence mapping

The full range of filtering and CoS functions are available at the ingress PE router. The functions include firewall filtering, rate limiting, queue selection, and IP precedence mapping.

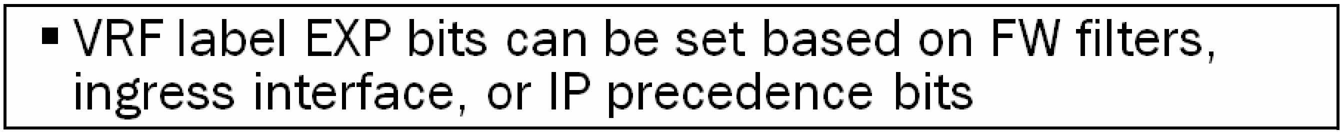
Filtering and CoS Functions Available at Egress



- Filtering functions might be unavailable at egress PE router
  - Support of `vrf-table-label` and `vt-interface` allows filtering functions at egress router

You can also employ filtering and CoS functions at the egress PE router when certain conditions are met. These functions allow for Address Resolution Protocol (ARP) operations, egress rate limiting, and firewall filtering.

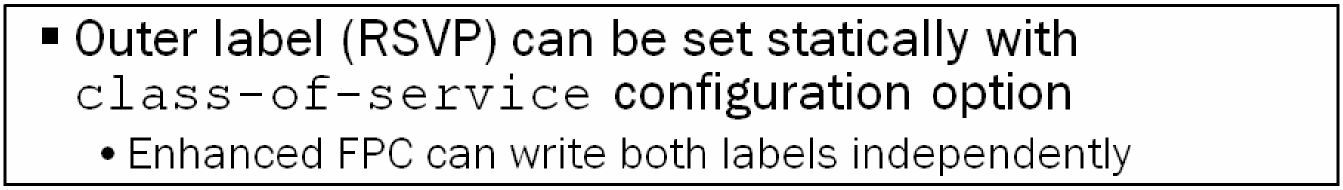
VRF Label Experimental Bits



- VRF label EXP bits can be set based on FW filters, ingress interface, or IP precedence bits

The EXP bits in the VRF label can be set based on firewall classification, IP precedence bits, or ingress interface.

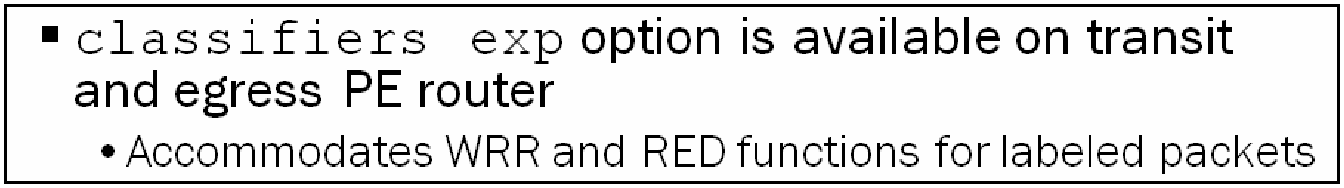
RSVP Label Experimental Bits



- Outer label (RSVP) can be set statically with `class-of-service` configuration option
  - Enhanced FPC can write both labels independently

The EXP bits of the RSVP label can be set with a static CoS value. Or, with the Enhanced Flexible PIC Concentrator (FPC), the RSVP or LDP label can have its EXP field set to the value used by the VRF label.

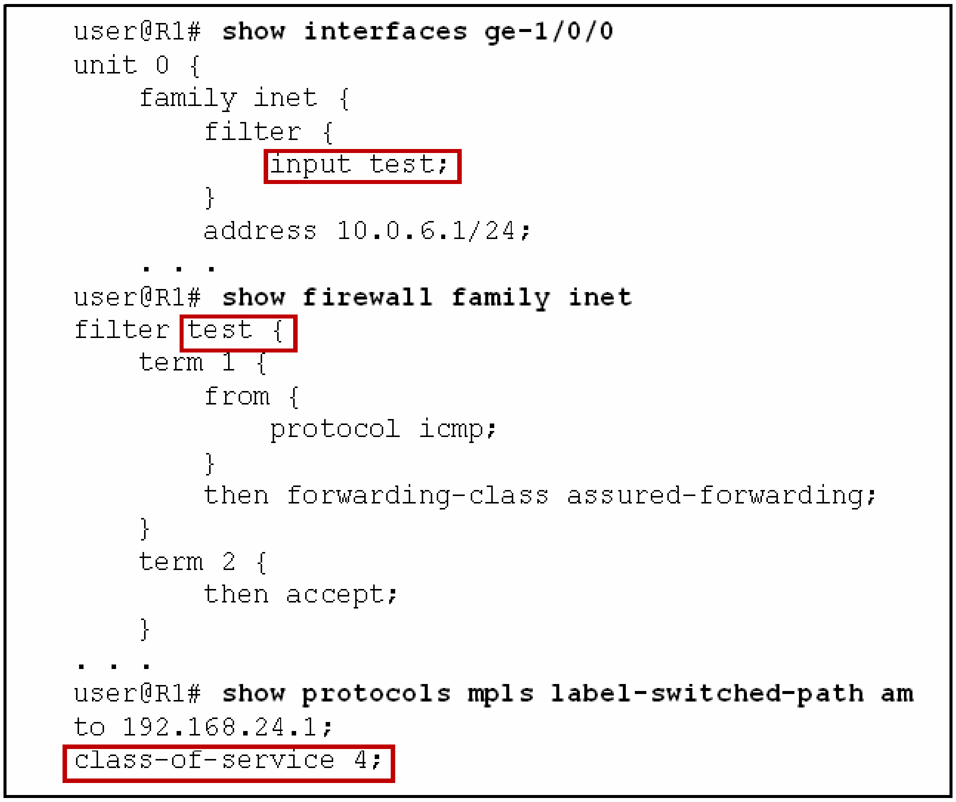
`classifiers exp` Setting on Transit LSRs



- `classifiers exp` option is available on transit and egress PE router
  - Accommodates WRR and RED functions for labeled packets

Setting the **`classifiers exp`** option on transit LSRs makes weighted round-robin (WRR) and random early detection (RED) functionality available for labeled packets. Failing to specify an EXP classifier results in all labeled packets being placed into output queue 0 by default. With Enhanced FPC hardware, you can create custom EXP to output queue mappings, but an **`exp classifiers`** statement is still necessary to effect EXP-based output queue selection for queues 1–3.

Layer 3 VPN CoS Example

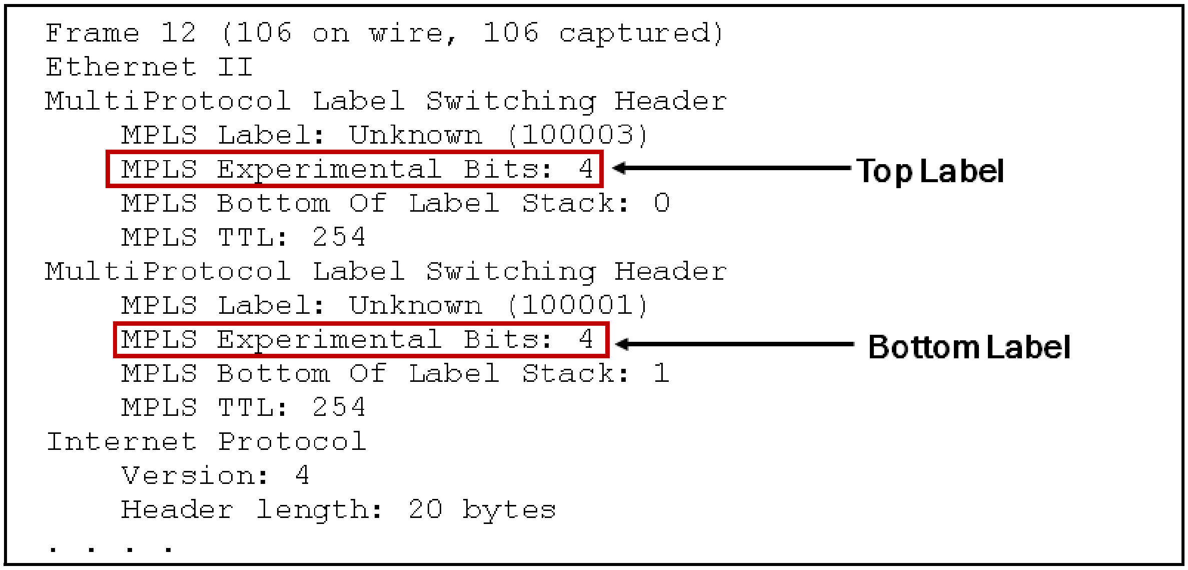


```
user@R1# show interfaces ge-1/0/0
unit 0 {
    family inet {
        filter {
            input test;
        }
        address 10.0.6.1/24;
    . . .
user@R1# show firewall family inet
filter test {
    term 1 {
        from {
            protocol icmp;
        }
        then forwarding-class assured-forwarding;
    }
    term 2 {
        then accept;
    }
. . .
user@R1# show protocols mpls label-switched-path am
to 192.168.24.1;
class-of-service 4;
```

This graphic provides an example of how you can use firewall filters to classify packets for queuing, and how you can configure an RSVP session with a static CoS value. The result of this configuration is that transit LSRs queue the labeled packets in queue number 2 (`assured-forwarding` forwarding class). The ingress PE router places all Internet Control Message Protocol (ICMP) traffic into queue 2 with all other traffic going into queue 0 (the default queue).

With an Enhanced FPC, both labels can be written independently. Thus, the queuing decisions made by the ingress PE router can be mirrored in the transit LSRs and at the egress PE router.

RSVP Label Has Static CoS



```
Frame 12 (106 on wire, 106 captured)
Ethernet II
MultiProtocol Label Switching Header
    MPLS Label: Unknown (100003)
    MPLS Experimental Bits: 4
    MPLS Bottom Of Label Stack: 0
    MPLS TTL: 254
MultiProtocol Label Switching Header
    MPLS Label: Unknown (100001)
    MPLS Experimental Bits: 4
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 254
Internet Protocol
    Version: 4
    Header length: 20 bytes
. . . .
```

This protocol capture shows the results of the CoS configuration shown on the previous page. The top label in this example is carrying the static CoS value associated with the LSP itself.

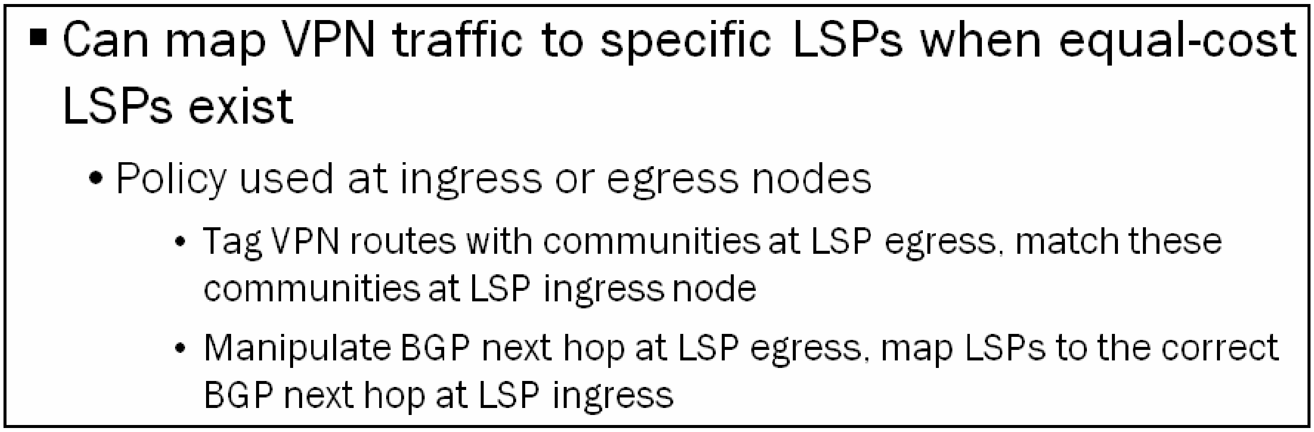
Bottom Label Has Firewall-Based Classification

The bottom (VRF) label in this example is carrying a CoS value set by the firewall-based classification of the packet at ingress. With a B2 FPC, the firewall-based classification is overwritten by the outer label's EXP value. Therefore, differentiated queuing is only possible at the ingress PE router. With the Enhanced FPC, the values are set independently. By default, an Enhanced FPC-equipped router sets the outer label to the value of the inner label such that classification at the ingress PE router sets the EXP field of both labels, thereby allowing transit and egress queuing based on input classification.

Load Balancing

You can load-balance VPN traffic across multiple LSPs by applying a load-balancing policy to the main forwarding instance.

Mapping Traffic to Specific LSPs



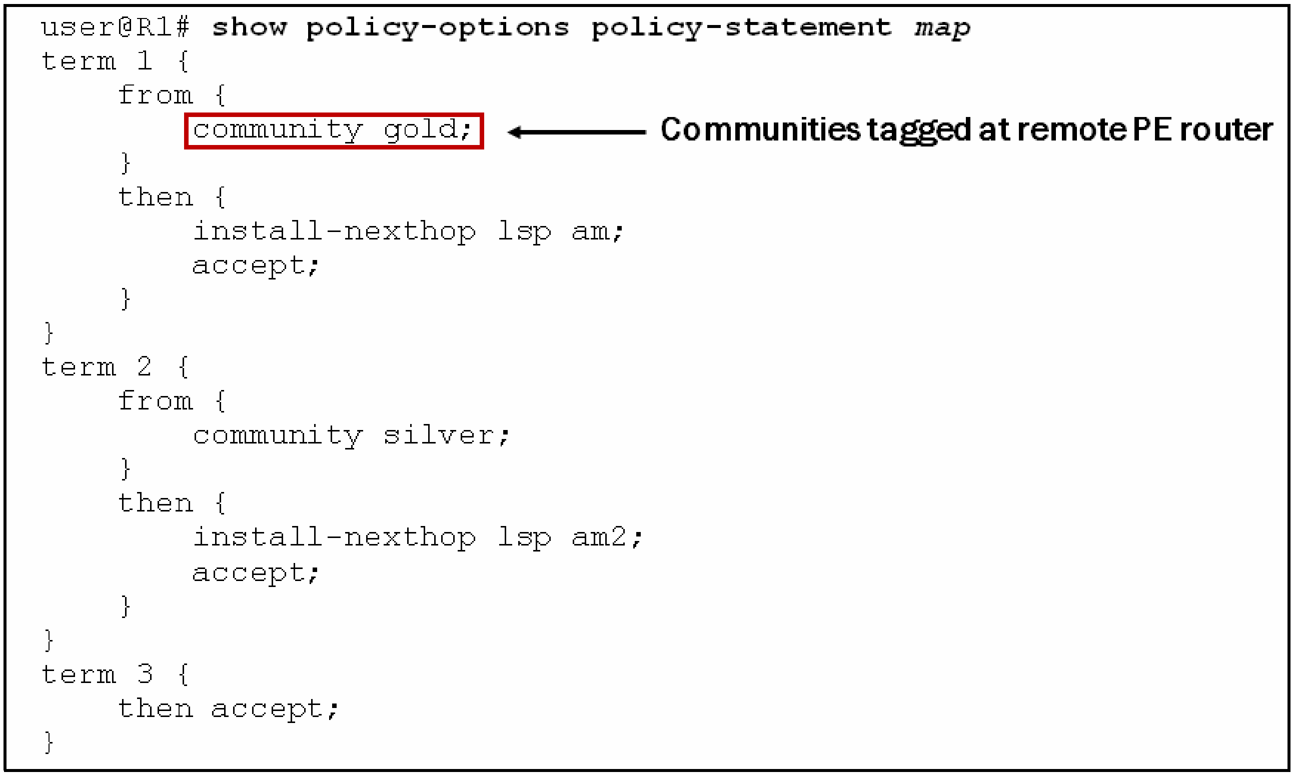
- Can map VPN traffic to specific LSPs when equal-cost LSPs exist
  - Policy used at ingress or egress nodes
    - Tag VPN routes with communities at LSP egress, match these communities at LSP ingress node
    - Manipulate BGP next hop at LSP egress, map LSPs to the correct BGP next hop at LSP ingress

You also can map VPN traffic to a specific LSP when multiple LSPs exist between a pair of PE routers. This mapping allows a service provider to offer a multitier service by deploying LSPs between PE routers having differing performance characteristics.

The most common technique for prefix-to-LSP mapping involves routing policy at the LSP ingress node. This policy maps traffic to a particular LSP using community-based match criteria. This technique assumes that the LSP egress node tags VPN prefixes with the correct community value as the routes are advertised to PE routers using multiprotocol IBGP. Note that this technique currently does not support route filter match conditions at the LSP ingress node.

You can also map prefixes to LSPs by manipulating the BGP next hop at the LSP egress node as the routes are advertised to PE routers. When establishing the two LSPs, you must use care to ensure that each is defined to terminate on the correct IP address at the LSP egress node. The result is that the LSP ingress node resolves some of the VPN routes to one of the BGP next hops and the remaining routes to the other BGP next hop. When the LSP egress node resolves these BGP next hops through its `inet.3` routing table, it selects the LSP that matches the route's BGP next hop for installation in the forwarding table.

Prefix Mapping Example



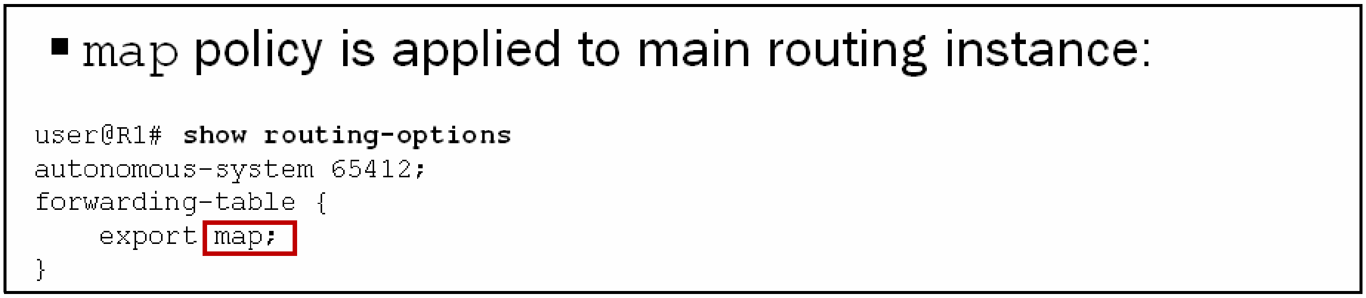
```
user@R1# show policy-options policy-statement map
term 1 {
    from {
        community gold;
    }
    then {
        install-nexthop lsp am;
        accept;
    }
}
term 2 {
    from {
        community silver;
    }
    then {
        install-nexthop lsp am2;
        accept;
    }
}
term 3 {
    then accept;
}
```

Communities tagged at remote PE router

This graphic demonstrates the technique of mapping prefixes to LSPs using routing policy, which matches communities at the LSP ingress node.

The policy uses the **`install-nexthop lsp`** action modifier to direct matching routes to a specific RSVP session. Term 3 accepts all nonmatching routes for the default action of per-prefix load balancing across equal-cost LSPs.

Prefix Mapping Policy

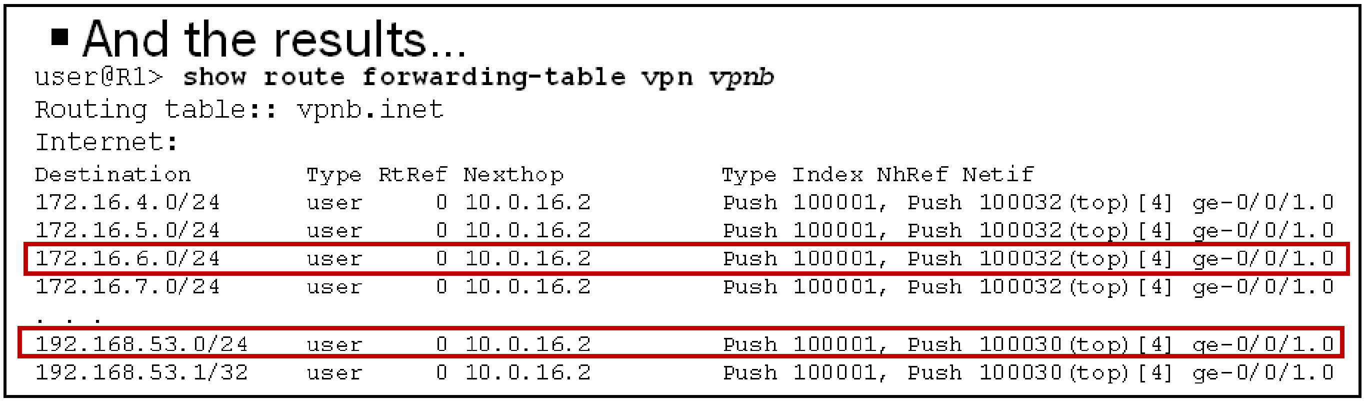


- `map` policy is applied to main routing instance:

```
user@R1# show routing-options
autonomous-system 65412;
forwarding-table {
    export map;
}
```

You must apply the policy shown on the previous page if it is to have any effect. Prefix mapping and load-balancing policies must be applied to the main instance's forwarding table. The graphic shows this application.

The Results...

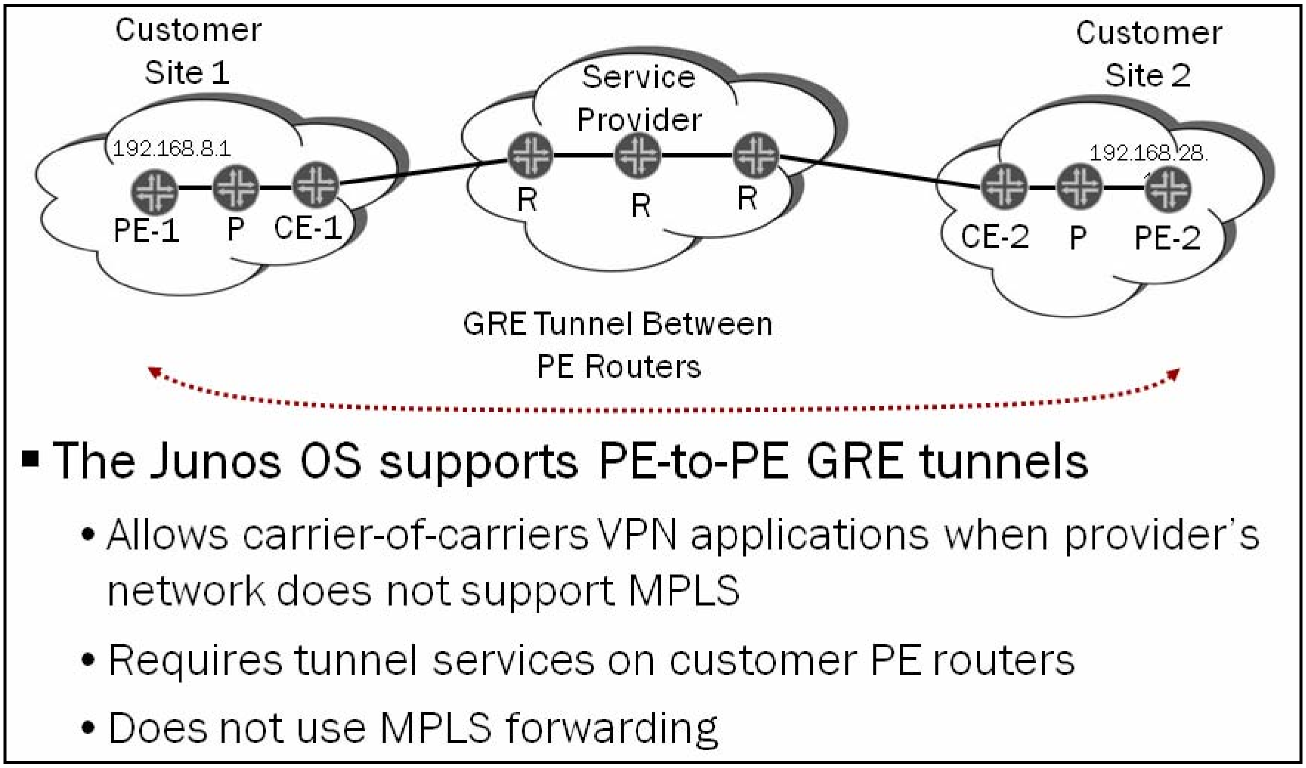


- And the results...

```
user@R1> show route forwarding-table vpn vpnb
Routing table:: vpnb.inet
Internet:
Destination         Type  RtRef Nexthop             Type Index NhRef Netif
172.16.4.0/24       user      0 10.0.16.2           Push 100001, Push 100032(top)[4] ge-0/0/1.0
172.16.5.0/24       user      0 10.0.16.2           Push 100001, Push 100032(top)[4] ge-0/0/1.0
172.16.6.0/24       user      0 10.0.16.2           Push 100001, Push 100032(top)[4] ge-0/0/1.0
172.16.7.0/24       user      0 10.0.16.2           Push 100001, Push 100032(top)[4] ge-0/0/1.0
. . .
192.168.53.0/24     user      0 10.0.16.2           Push 100001, Push 100030(top)[4] ge-0/0/1.0
192.168.53.1/32     user      0 10.0.16.2           Push 100001, Push 100030(top)[4] ge-0/0/1.0
```

After applying and committing the prefix mapping policy, you can verify the results by examining the ***vpnb*** VRF table. The highlighted entries confirm that traffic associated with the 172.16 routes is mapped to one LSP (top label set to 100032), while traffic to the 192.168 routes is mapped to a different LSP (top label set to 100030).

PE-PE GRE Tunnels

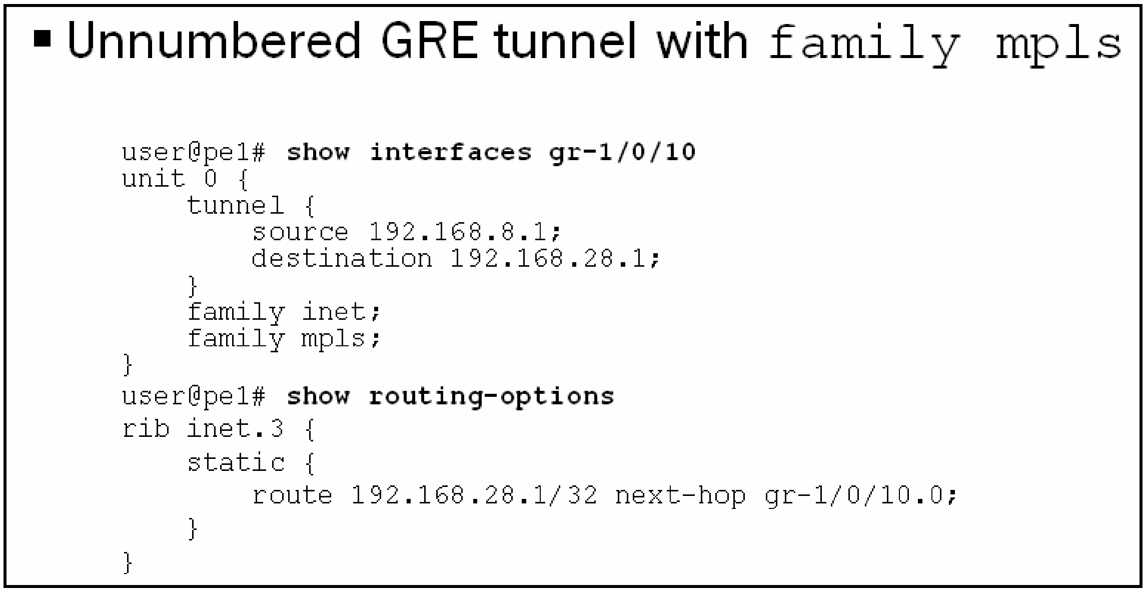


- The Junos OS supports PE-to-PE GRE tunnels
  - Allows carrier-of-carriers VPN applications when provider's network does not support MPLS
  - Requires tunnel services on customer PE routers
  - Does not use MPLS forwarding

The Junos OS supports the GRE tunneling of VPN traffic between PE routers. As shown, this support allows an interprovider VPN application when the provider's backbone does not support MPLS.

To support GRE tunnels, a tunnel services must be enabled as described in previous graphics. GRE-encapsulated packets are not forwarded over MPLS tunnels.

PE-PE GRE Tunnel Configuration



- Unnumbered GRE tunnel with `family mpls`

```
user@pe1# show interfaces gr-1/0/10
unit 0 {
    tunnel {
        source 192.168.8.1;
        destination 192.168.28.1;
    }
    family inet;
    family mpls;
}
user@pe1# show routing-options
rib inet.3 {
    static {
        route 192.168.28.1/32 next-hop gr-1/0/10.0;
    }
}
```

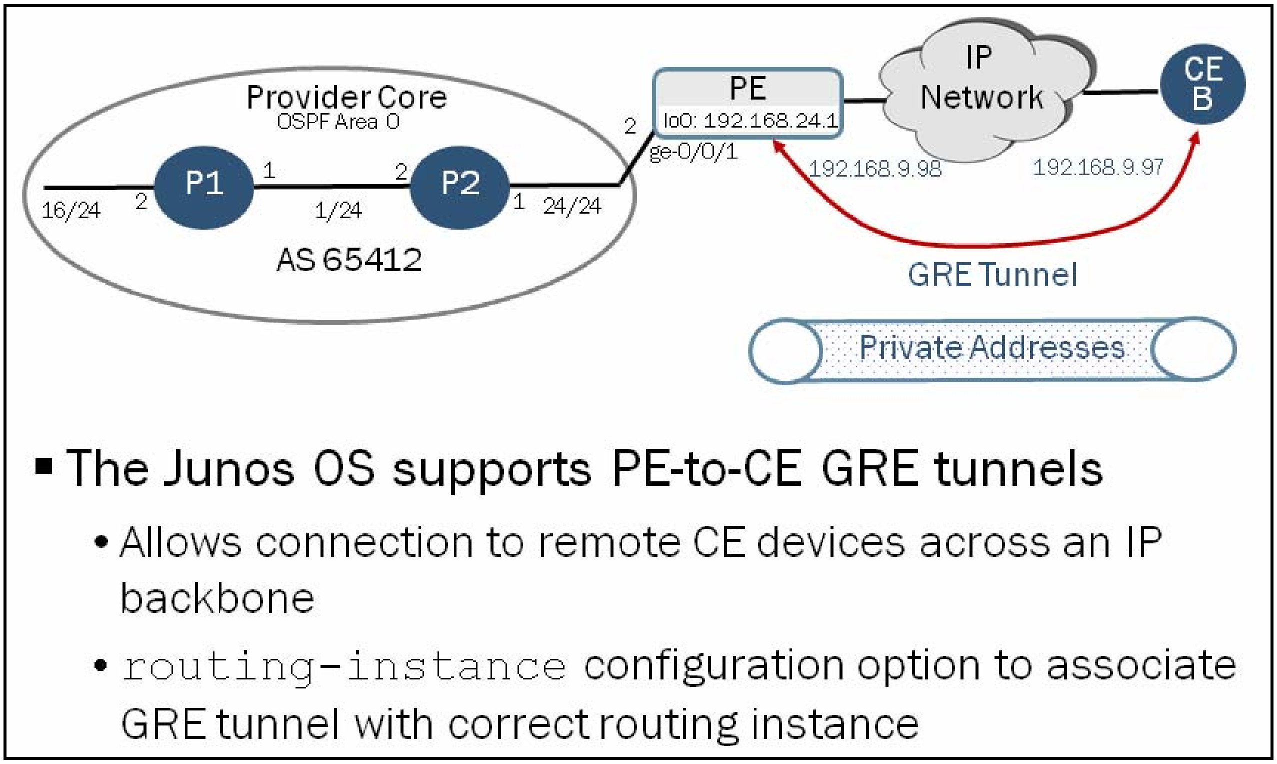
This graphic highlights the key aspects of a PE-to-PE GRE tunnel configuration. Use of a GRE tunnel has no impact on the PE router’s VRF table, VRF policy, or MP-BGP session configuration. Although not shown on the graphic, you should ensure that the customer’s IGP does not run over the GRE tunnel, because this can lead to recursion problems.

In the graphic, `unit 0` of the Tunnel Services interface is configured with tunnel properties such as the tunnel’s source and destination addresses. In this case, the addresses represent the values assigned to the PE router’s loopback interfaces. This example shows an unnumbered GRE tunnel, and therefore no IP address is specified. Because this tunnel will be used to support MPLS, `family mpls` must also be specified.

As illustrated in the graphic, you must configure a static route with the `next-hop` of the GRE interface in the `inet.3` routing table. This is route is configured under the `[edit routing-options rib inet.3]` hierarchy.

Note that you must also include the routing instance destination under the tunnel hierarchy if the GRE-encapsulating interface is also configured under the VRF table. In the example on the graphic, the VRF table does not include the PE router’s encapsulating interface.

PE-CE GRE Tunnels



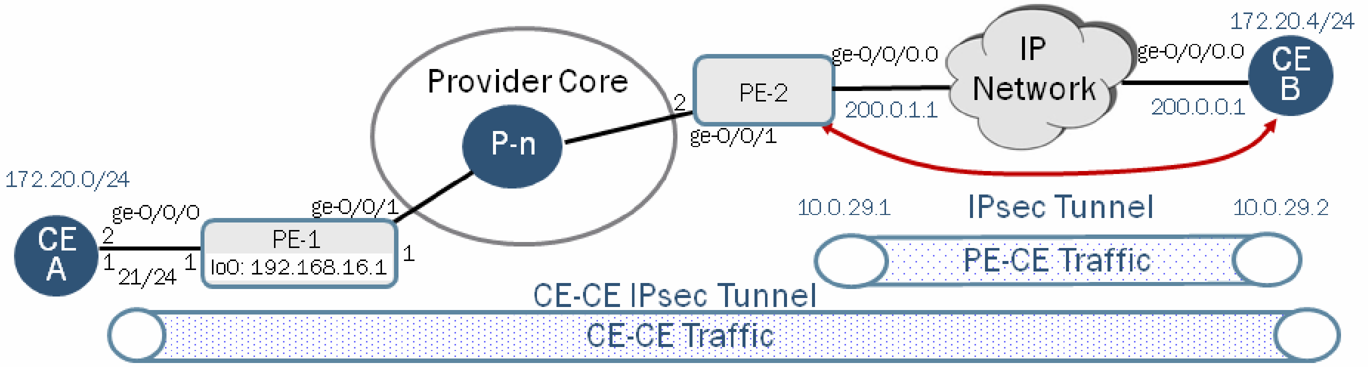
- The Junos OS supports PE-to-CE GRE tunnels
  - Allows connection to remote CE devices across an IP backbone
  - `routing-instance` configuration option to associate GRE tunnel with correct routing instance

The Junos OS supports GRE tunnels for PE-CE connections. As shown, this support allows the interconnection of a remote CE device across an IP network. The use of GRE tunneling allows the use of private and overlapping addresses as the packets are forwarded across the IP network based on the global addressing used for the GRE tunnel.

To support GRE tunnels, tunnel services must be enable on routers running the Junos OS. The new `routing-instance` configuration is used to place a GRE tunnel into the correct routing instance:

```
gr-1/0/0 {
    unit 0 {
        tunnel {
            source 192.168.9.98;
            destination 192.168.9.97;
            routing-instance {
                destination vrf-name;
            }
        }
    }
}
```

Normally, static routing is used to populate the PE router’s VRF table, because running a routing protocol over a GRE tunnel can lead to low speeds or a complete halt.



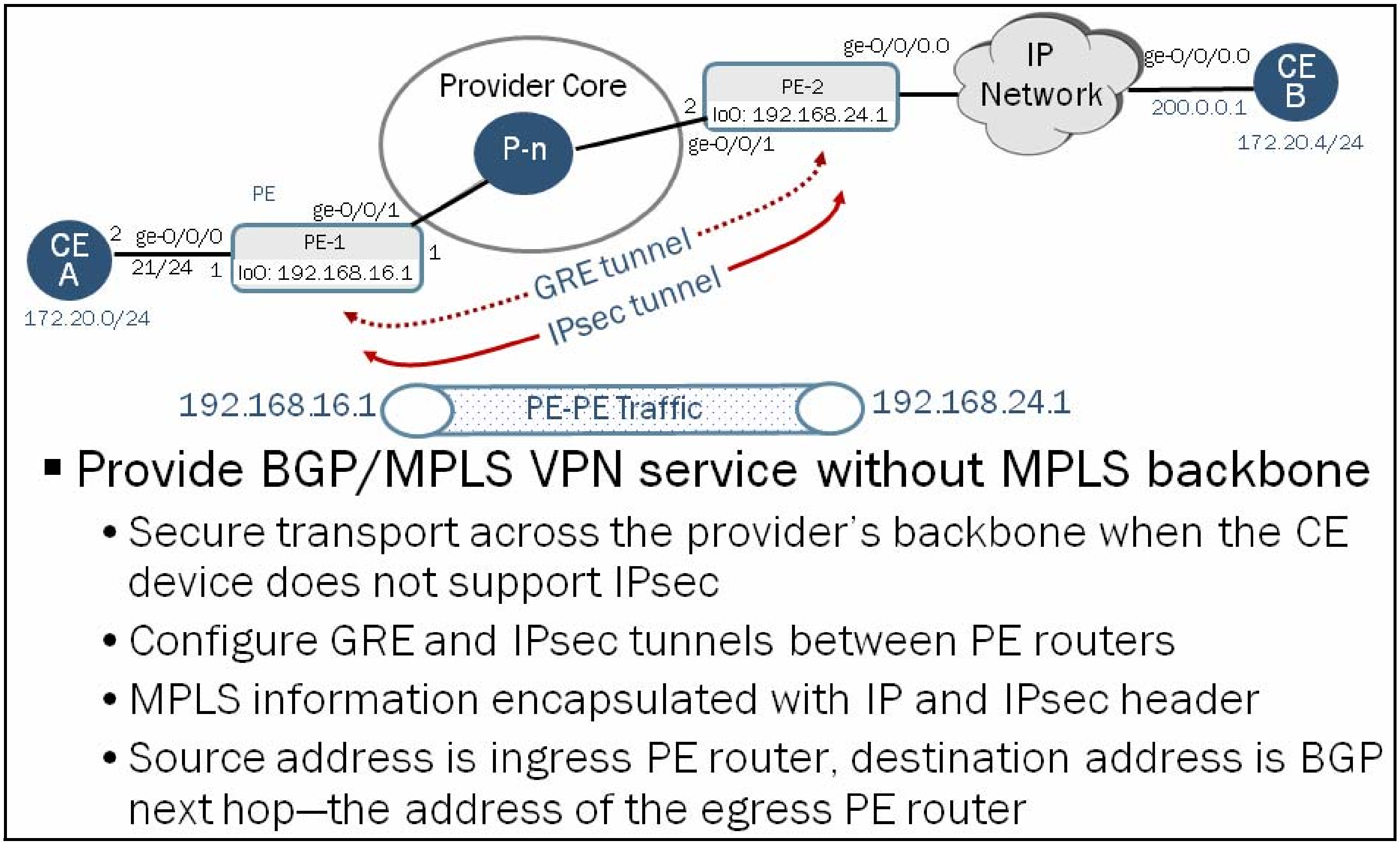
- The Junos OS supports IPsec/Layer 3 VPN integration
  - IPsec tunnels terminate between the PE and CE routers
  - CE-CE IPsec tunnels extend through PE routers
  - IPsec tunnels can use manual or dynamic security associations
  - PE and CE routers both require AS PIC or ES PIC
  - PE-PE configuration requires no change, firewall filter-based classification not used

IPsec and Layer 3 VPN Integration

The Junos OS supports the integration of provider-provisioned Layer 3 VPNs and IPsec protocols. This application most likely will be used to support the secure exchange of information between the local PE router and a CE router that is remotely connected through an IP cloud.

- *PE-CE IPsec tunnel termination*: The Junos OS offers support for the termination of IPsec tunnels between the PE and CE routers.
- *CE-CE tunnels*: As shown, the CE routers establish end-to-end IPsec tunnels, which are passed transparently through the PE routers. These IPsec tunnels provide secure site-to-site communications for data transferred over the provider's backbone.
- *Manual or dynamic SAs*: The PE-CE IPsec tunnel can use either manual or dynamic security associations (SAs). When configuring dynamic SAs, you must ensure that the encapsulating interface is not listed in the PE router's VRF table, because this causes dynamic SAs to fail.
- *Hardware required*: To support PE-CE IPsec tunnels, both the PE and CE routers require the presence of either an AS PIC, ES PIC, or a service Dense Port Concentrator (DPC).
- *PE-PE configuration*: The termination of IPsec tunnels between the PE and CE routers does not affect the PE-PE or P router configuration. The following pages highlight the configuration needed to support PE-CE IPsec tunnels. Because the IPsec tunnel is associated with the control traffic to and from the VRF table, you do not need to use firewall filters to classify traffic for encryption. We also discuss PE-PE configuration over GRE and IPsec.

IPsec Between PE Routers Instead of MPLS

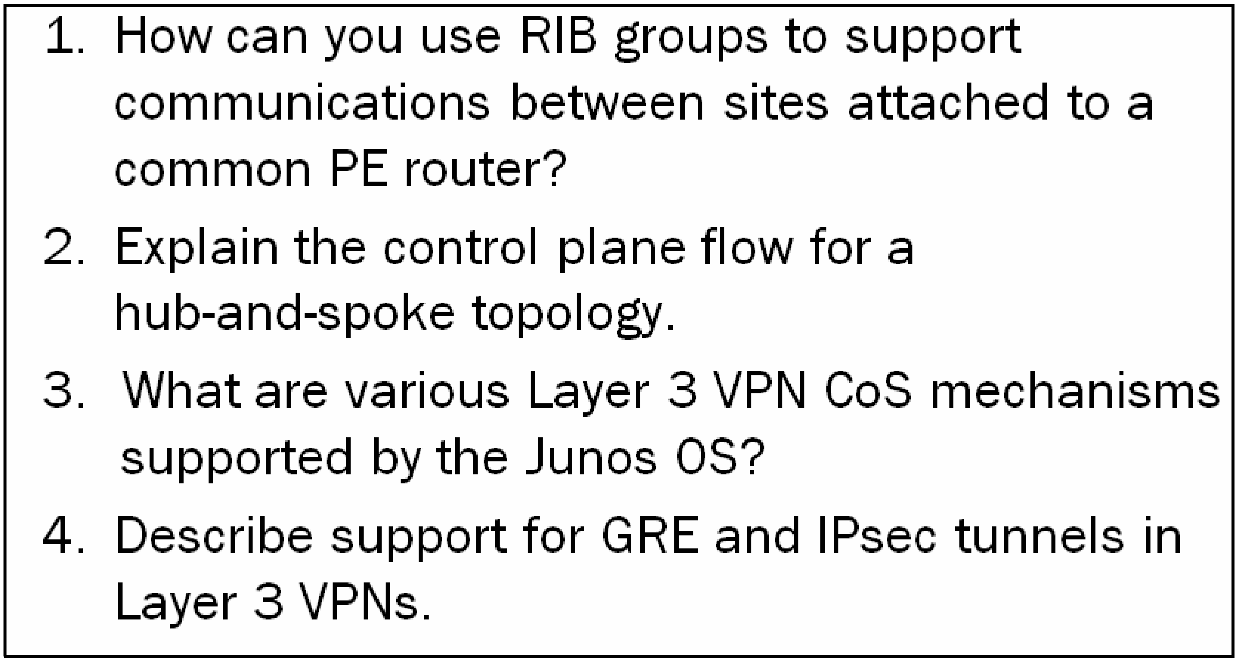


- Provide BGP/MPLS VPN service without MPLS backbone
  - Secure transport across the provider's backbone when the CE device does not support IPsec
  - Configure GRE and IPsec tunnels between PE routers
  - MPLS information encapsulated with IP and IPsec header
  - Source address is ingress PE router, destination address is BGP next hop—the address of the egress PE router

A conventional Layer 3 BGP/MPLS VPN requires the configuration of MPLS LSPs between the PE routers. When a PE router receives a packet from a CE router, it performs a lookup in a specific VRF table for the IP destination address and obtains a corresponding MPLS label stack. The label stack is used to forward the packet to the egress PE router, where the bottom label is removed and the packet is forwarded to the specified CE router.

You can also provide Layer 3 BGP/MPLS VPN service without an MPLS backbone by configuring GRE and IPsec tunnels between the PE routers. The MPLS information for the VPN (the VPN label) is encapsulated within an IP header and an IPsec header. The source address of the IP header is the address of the ingress PE router, while the destination address has the BGP next hop, the address of the egress PE router.

Review Questions



1. How can you use RIB groups to support communications between sites attached to a common PE router?
2. Explain the control plane flow for a hub-and-spoke topology.
3. What are various Layer 3 VPN CoS mechanisms supported by the Junos OS?
4. Describe support for GRE and IPsec tunnels in Layer 3 VPNs.

Answers to Review Questions

1.

To place routes from one routing table into a second routing table, you must first create a routing table-group that lists both routing tables as an import routing table with the primary table listed first. Once the routing table-group is specified, you need to specify which routes will go into the routing table-group. A common set of routes to place in the routing table-group would be interface routes which can be applied to the routing table-group under `[edit routing-options interface-routes]` level of the hierarchy. Apply the routing table-group at this level of the hierarchy will take the local and direct routes found in the primary table (the first table in the list) and ensure they exist in both tables. For routes learned by routing protocols, these routes can be applied to the routing table-group at the `[edit protocols` *`protocol-name`*`]` level of the hierarchy.

2.

Routes from Spoke PEs and CEs are received by and accepted by the Spoke instance on the Hub PE. The HUB PE passes those route to the HUB CE. The HUB CE then advertises those routes to the Hub instance on the Hub PE. The Hub PE then advertises those routes to the Spoke sites.

3.

The Junos OS supports firewall filtering and rate limiting. It also support the setting of the experimental bits on both the inner and outer headers of an MPLS packet.

4.

GRE and IPsec tunnels are support from CE to CE, PE to PE, and CE to PE using the Junos OS.

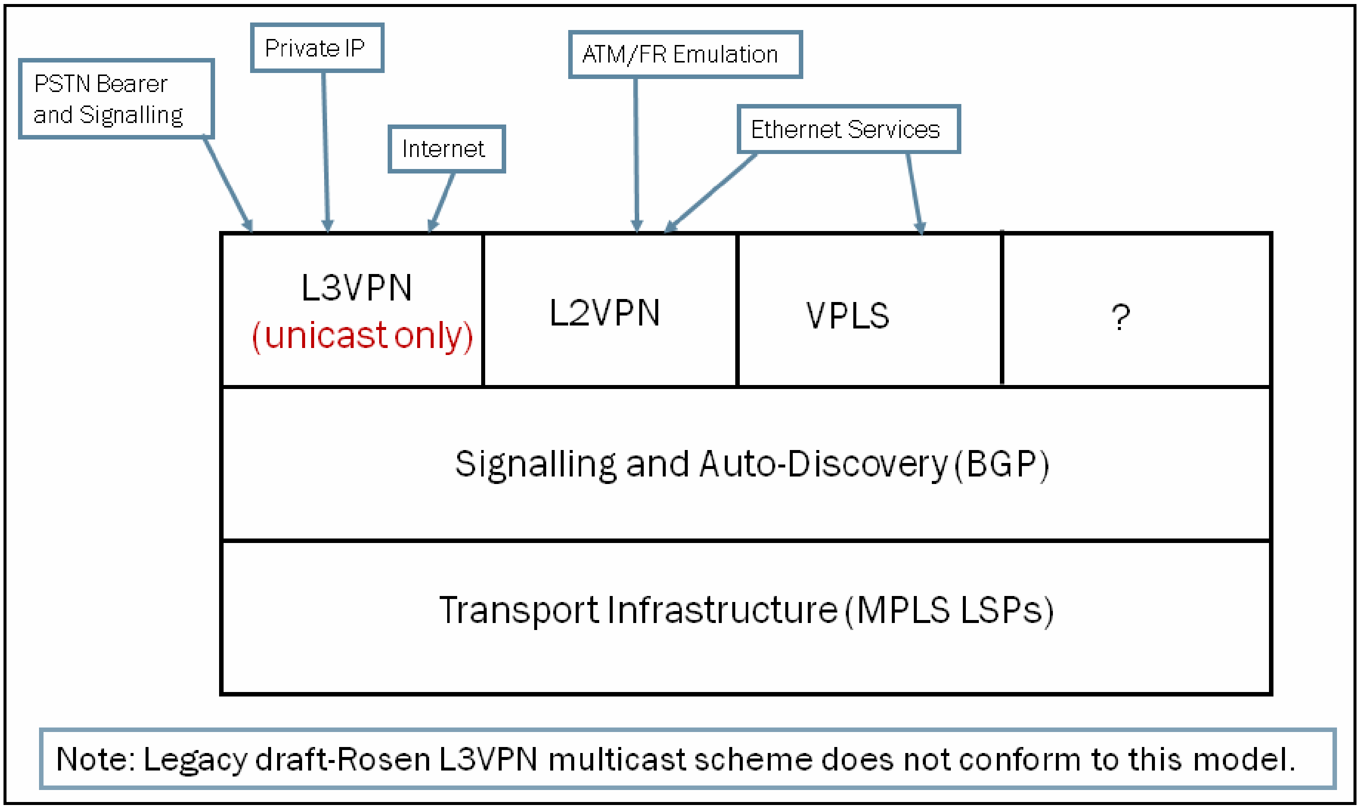
JNCIS-SP Study Guide—Part 3

Chapter 12: Multicast VPNs

This Chapter Discusses:

- The flow of control traffic and data traffic in a next-generation multicast virtual private network (VPN);
- The configuration steps for establishing a next-generation multicast VPN; and
- Monitoring and verifying the operation of next-generation multicast VPNs.

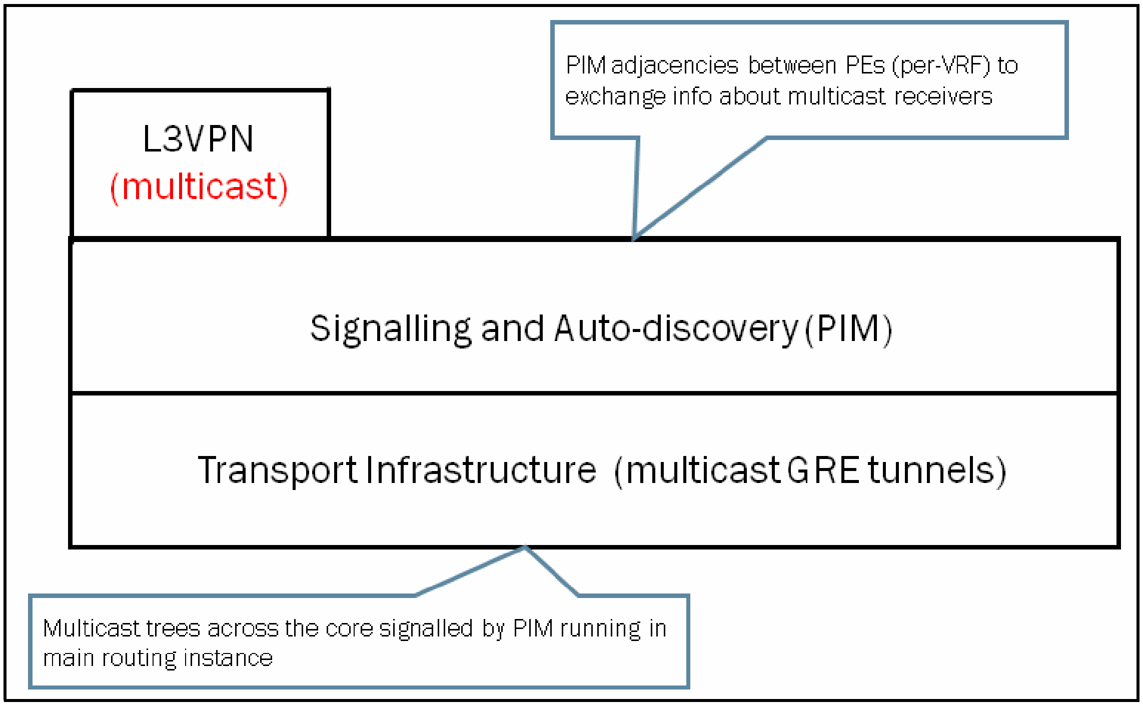
Multiservice Model


PSTN Bearer and Signalling
Private IP
Internet
ATM/FR Emulation
Ethernet Services
L3VPN (unicast only)
L2VPN
VPLS
?
Signalling and Auto-Discovery (BGP)
Transport Infrastructure (MPLS LSPs)
Note: Legacy draft-Rosen L3VPN multicast scheme does not conform to this model.

Service providers of today are moving many of their individual networks to a single IP/MPLS backbone. Today, the services shown on the graphic (Private IP, Internet, Frame Relay, and so on) no longer need a dedicated network to provide these services to customers. Instead, these can be provided to customers transparently over an IP/MPLS network using standards-based Layer 3 VPNs, Layer 2 VPNs, and virtual private LAN service (VPLS). Each of the standards-based features rely on the foundation of MPLS for the transport of the customer data and BGP for signaling and autodiscovery.

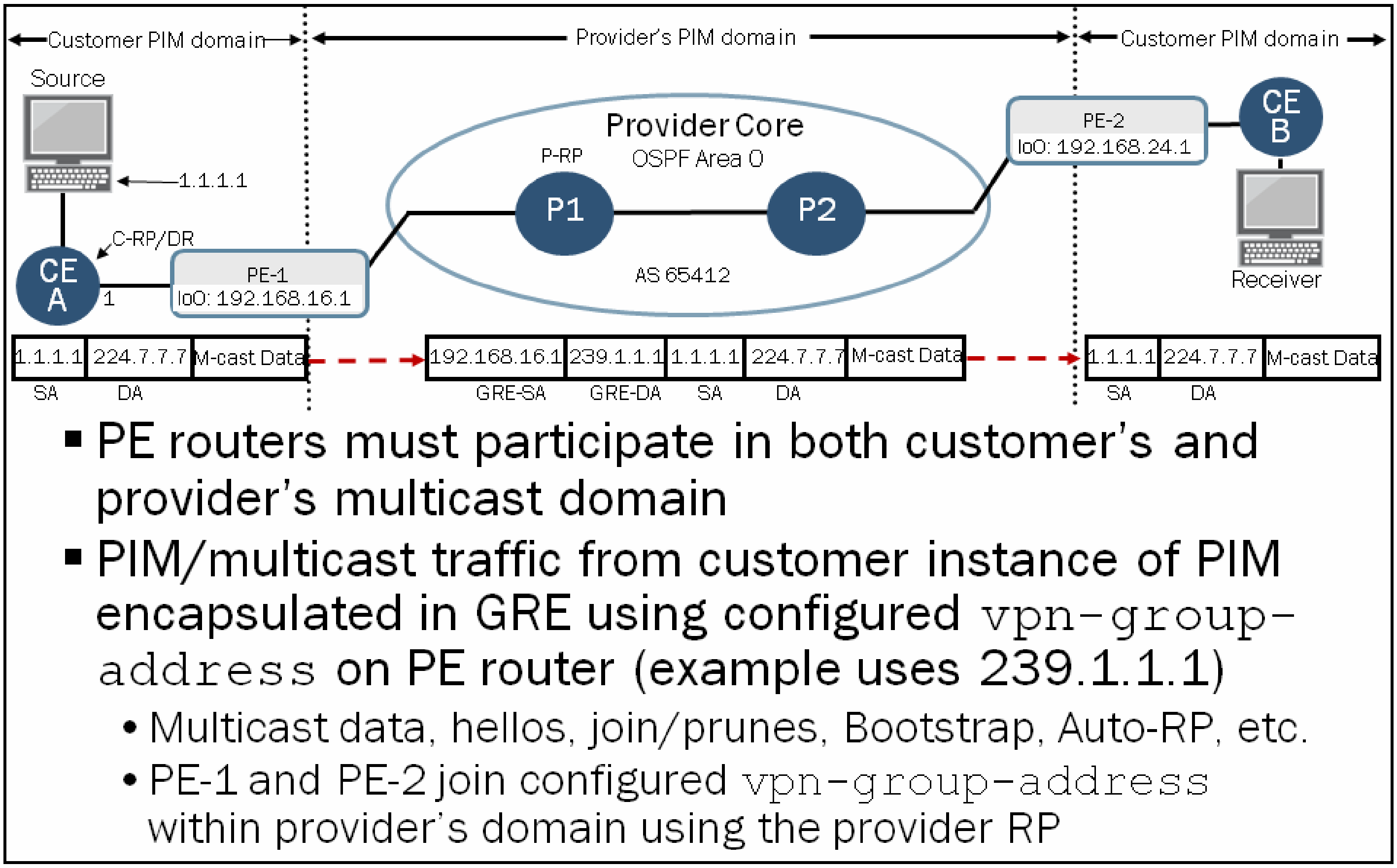
Multicast service over an IP/MPLS network has been evolving over time. The draft-Rosen method of multicast transport as described in subsequent sections does not conform to the model shown on the graphic.

Legacy Model for MVPN—draft-Rosen


L3VPN
(multicast)
PIM adjacencies between PEs (per-VRF) to exchange info about multicast receivers
Signalling and Auto-discovery (PIM)
Transport Infrastructure (multicast GRE tunnels)
Multicast trees across the core signalled by PIM running in main routing instance

For some time, draft-Rosen has been the standard by which multicast is transported between Layer 3 VPN (L3VPN) sites. This method does not rely on either MPLS or BGP. Instead, not only does the customer need to run a multicast routing protocol like Protocol Independent Multicast (PIM) but the service provider network must also use PIM to signal the end to end path of the L3VPN multicast traffic. Also, MPLS is not used to transport the multicast data between sites, instead, multicast generic routing encapsulation (GRE) tunnels are used.

PEs Participate in Customer and Provider Multicast



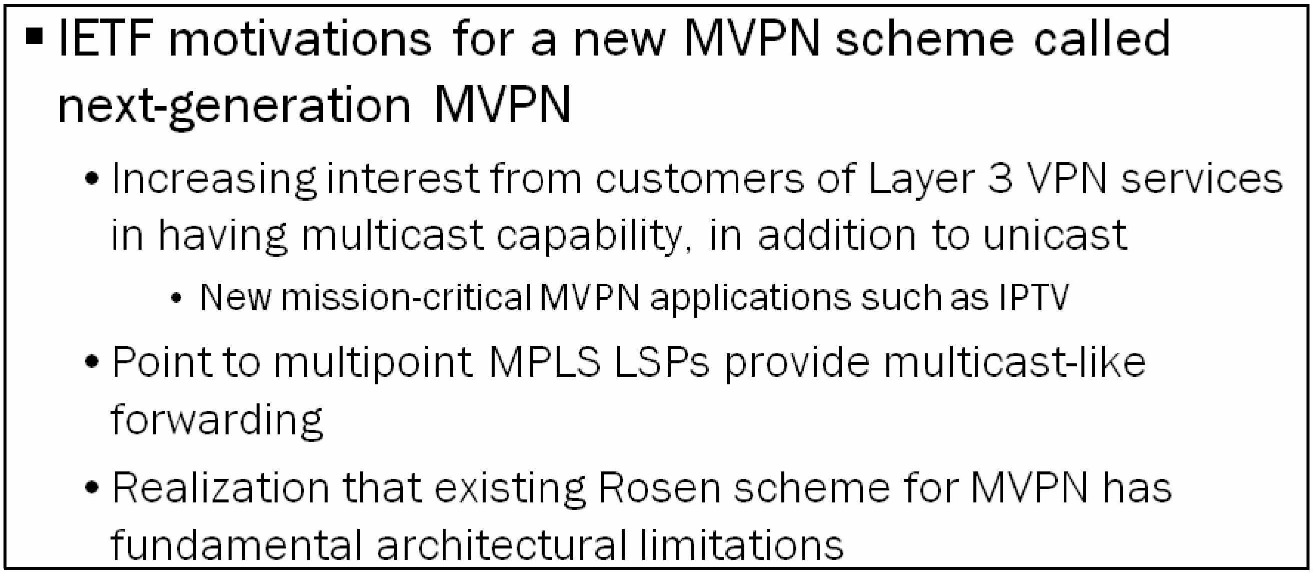
- PE routers must participate in both customer's and provider's multicast domain
- PIM/multicast traffic from customer instance of PIM encapsulated in GRE using configured `vpn-group-address` on PE router (example uses 239.1.1.1)
  - Multicast data, hellos, join/prunes, Bootstrap, Auto-RP, etc.
  - PE-1 and PE-2 join configured `vpn-group-address` within provider's domain using the provider RP

The graphic shows the relationship between the customer sites and provider network in the draft-Rosen model. Within the customer network (VPN routing and forwarding table [VRF]), a provider edge (PE) must participate in the customers PIM domain. Within the provider network (main routing instance), a PE must participate in the providers PIM domain.

PIM and Multicast Traffic Encapsulated in GRE

The provider must dedicate an individual multicast group to each customer that desires multicast service. This dedicated group is specified within the VRF as a `vpn-group-address` on the PE router. The `vpn-group-address` is used as the destination address of the GRE packets which tunnel customer multicast traffic across the provider network.

Motivations for Next-Generation MVPN



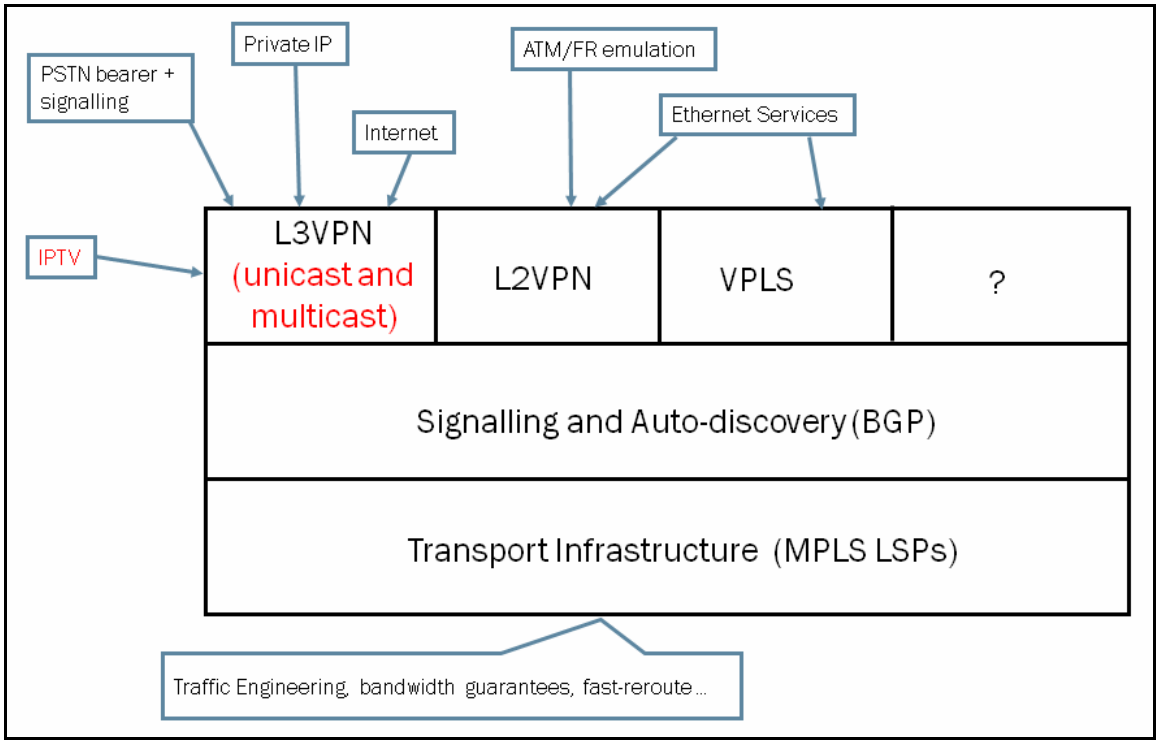
- IETF motivations for a new MVPN scheme called next-generation MVPN
  - Increasing interest from customers of Layer 3 VPN services in having multicast capability, in addition to unicast
    - New mission-critical MVPN applications such as IPTV
  - Point to multipoint MPLS LSPs provide multicast-like forwarding
  - Realization that existing Rosen scheme for MVPN has fundamental architectural limitations

Over the last few years their has been increasing interest in transporting multicast traffic over Layer 3 VPNs along with unicast. For example, multicast is the logical solution for delivering Internet Protocol Television (IPTV). Broadcast television providers have become increasingly interested in looking to the internet to deliver content in a secure environment to their customers.

MPLS forwarding has evolved as well. With the advent of the point-to-multipoint LSP, an MPLS-based network can provide multicast-like forwarding capabilities without the need for running multicast protocols.

The draft-Rosen method of delivering multicast content has some scaling limitations. For example, consider an example where a PE has 1,000 VRFs, and each of these VRFs corresponds to a multicast VPN (MVPN) that is present on 100 PEs. The PE would need to maintain 100,000 PIM adjacencies with other PEs. The rate of PIM Hellos that the PE would need to process is 3,300 per second.

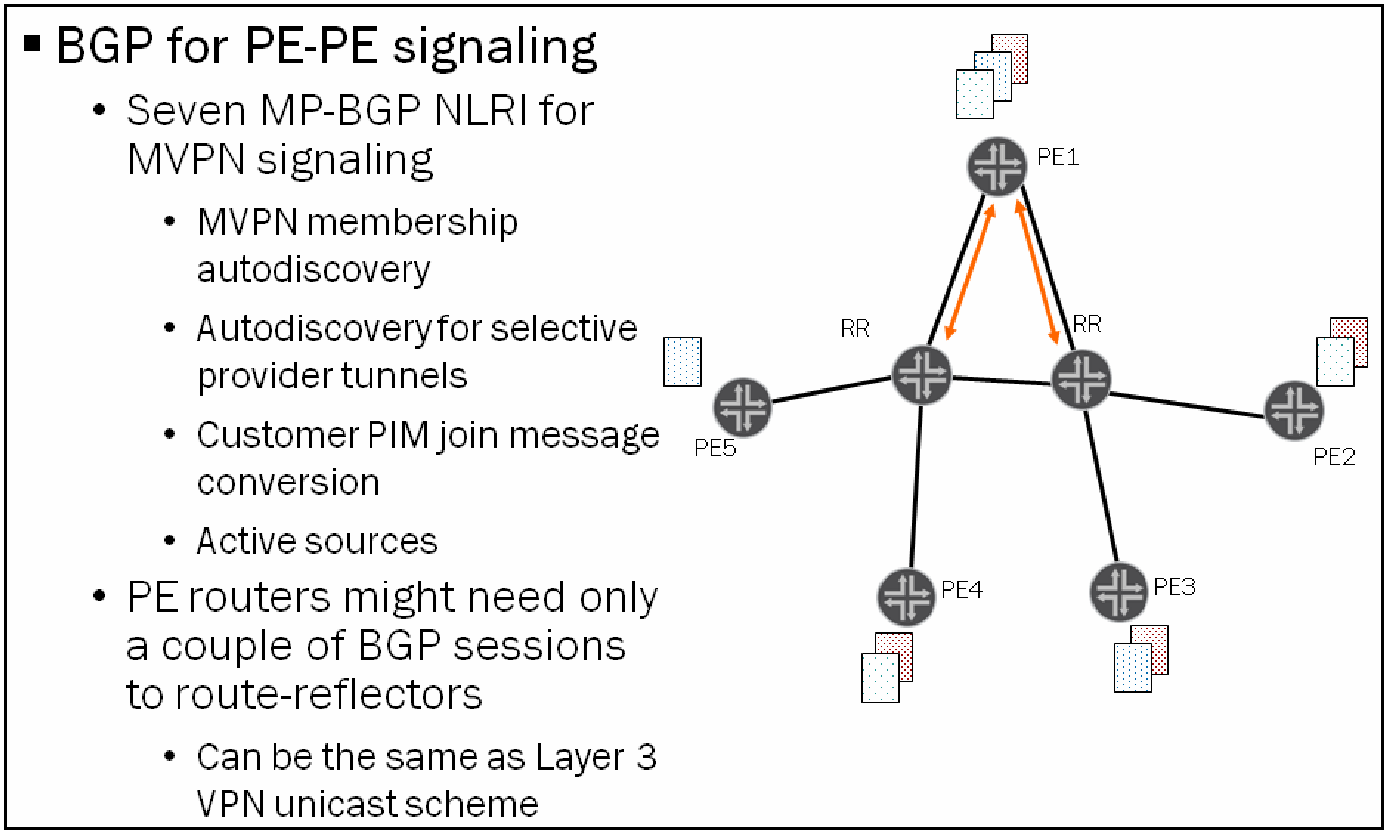
Model for Next-Generation MVPNs



PSTN bearer + signalling
Private IP
Internet
ATM/FR emulation
Ethernet Services
IPTV
L3VPN (unicast and multicast)
L2VPN
VPLS
?
Signalling and Auto-discovery (BGP)
Transport Infrastructure (MPLS LSPs)
Traffic Engineering, bandwidth guarantees, fast-reroute…

The graphic shows the signaling and transport model of next-generation MVPNs. Next-generation MVPNs use the same MPLS and BGP infrastructure as Layer 3 VPNs, Layer 2 VPNs, and VPLS.

BGP for PE to PE Signaling



- BGP for PE-PE signaling
  - Seven MP-BGP NLRI for MVPN signaling
    - MVPN membership autodiscovery
    - Autodiscovery for selective provider tunnels
    - Customer PIM join message conversion
    - Active sources
  - PE routers might need only a couple of BGP sessions to route-reflectors
    - Can be the same as Layer 3 VPN unicast scheme

Next-generation MVPNs call for Multiprotocol Border Gateway Protocol (MP-BGP) as the signaling method for multicast trees. Seven new network layer reachability information (NLRI) types have been standardized in draft form (draft-ietf-l3vpn-2547bis-mcast-bgp). The new NLRI types perform functions like MVPN membership autodiscovery, selective tunnel autodiscovery, PIM join message conversion, and active source advertisement.

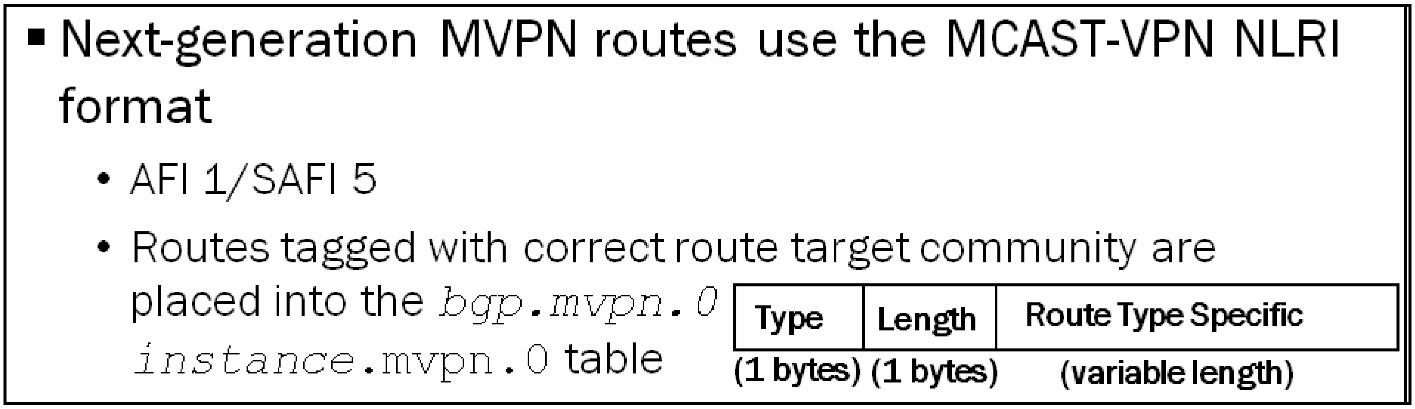
The PIM adjacency problem between PEs that was found in draft-Rosen no longer exists. Instead, a PE router might only need a few BGP neighbor relationships with route-reflectors, which might also be the same route-reflectors used for the L3VPN.

Next-Generation MVPN Terms

Next-generation MVPN terminology includes the following:

- Provider-Multicast Service Interface (PMSI) - Tunnel used to transport multicast data from PE to PE. It is also called a provider tunnel. Provider tunnels can take the form of RSVP-traffic engineered point-to-multipoint label-switched paths (LSPs), provider instance PIM distribution trees, and mLDP (not currently supported on the Junos OS).
- Inclusive-PMSI (I-PMSI) - There are two type of I-PMSIs. A multidirectional I-PMSI allows all PEs of a multicast VPN (MVPN) to transmit multicast data between each other (one point-to-multipoint LSP from all PEs to all other PEs). A unidirectional I-PMSI allows a single PE to transmit multicast data to other PEs (one point-to-multipoint LSP from a single PE to all other PEs).
- Selective-PMSI (S-PMSI) - A PE can transmit multicast packets to only those PEs of an MVPN that have requested to be a part of the multicast forwarding tree.

MCAST-VPN NLRI

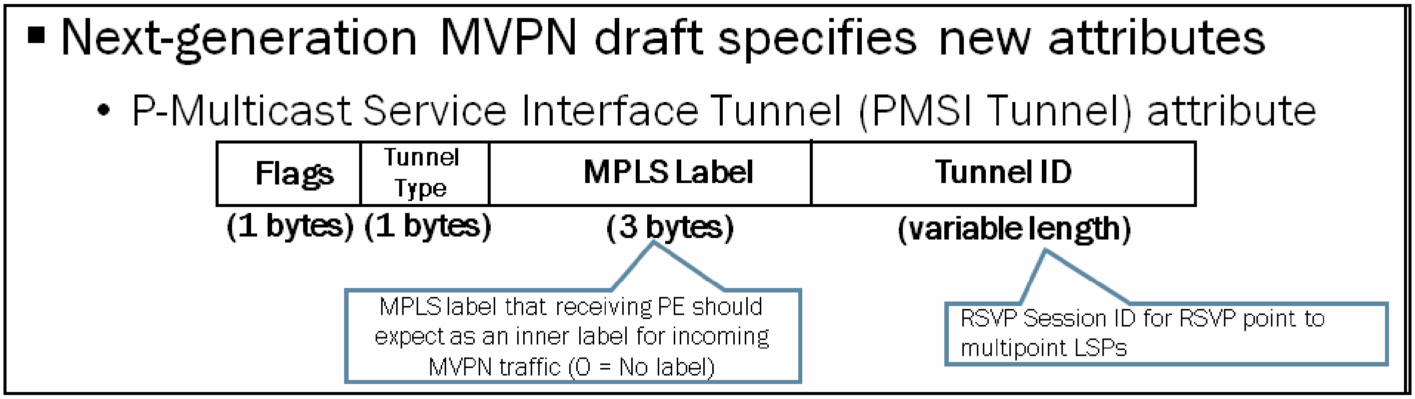


- Next-generation MVPN routes use the MCAST-VPN NLRI format
  - AFI 1/SAFI 5
  - Routes tagged with correct route target community are placed into the *bgp.mvpn.0* *instance*.mvpn.0 table

Type
Length
Route Type Specific
(1 bytes)
(1 bytes)
(variable length)

The NLRI format for next-generation MVPN signaling can be found in draft-ietf-l3vpn-2547bis-mcast-bgp. The MCAST-VPN NLRI is carried in MP-BGP extensions with an AFI of 1 and SAFI of 5. When these type of routes are received from remote PEs and accepted by a policy that matches on the route target community (same as L3VPNs), the receiving PE will place the routes in the MVPN routing table-IN called `bgp.mvpn.0` and then into the corresponding VRFs MVPN routing table, ***routing-instance***`.mvpn.0`.

Next-Generation MVPN Attributes

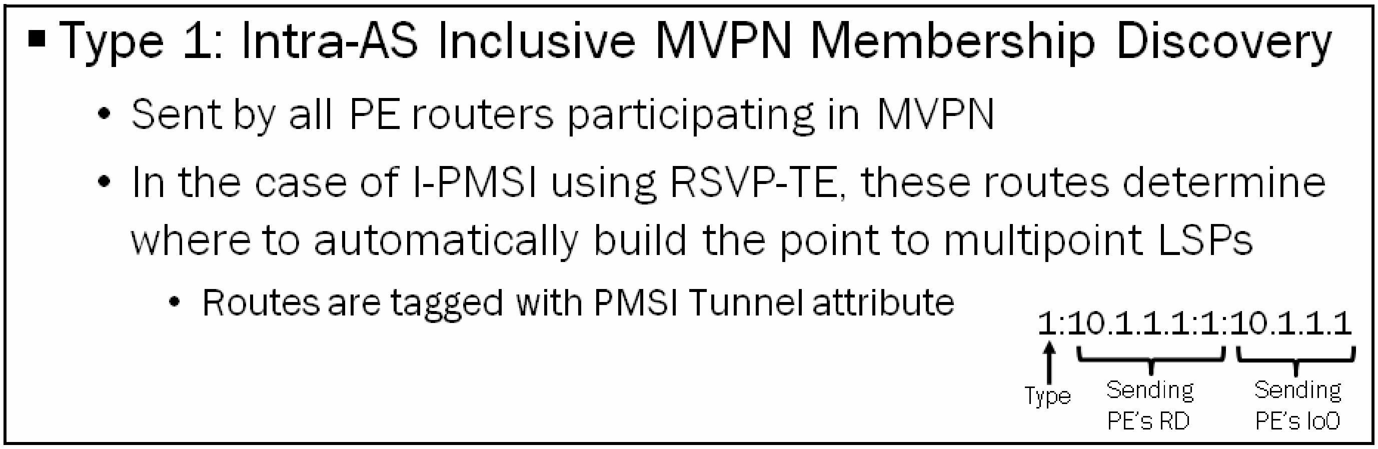


- Next-generation MVPN draft specifies new attributes
  - P-Multicast Service Interface Tunnel (PMSI Tunnel) attribute

Flags
Tunnel Type
MPLS Label
Tunnel ID
(1 bytes)
(1 bytes)
(3 bytes)
(variable length)
MPLS label that receiving PE should expect as an inner label for incoming MVPN traffic (0 = No label)
RSVP Session ID for RSVP point to multipoint LSPs

The next-generation MVPN draft defines a few new attributes. One important attribute is called the PMSI Tunnel attribute. It carries label and tunnel ID information allowing a receiving PE to know what data channel (LSP for example) to expect multicast traffic on. Subsequent sections will describe its usage in more detail.

Type 1 NLRI

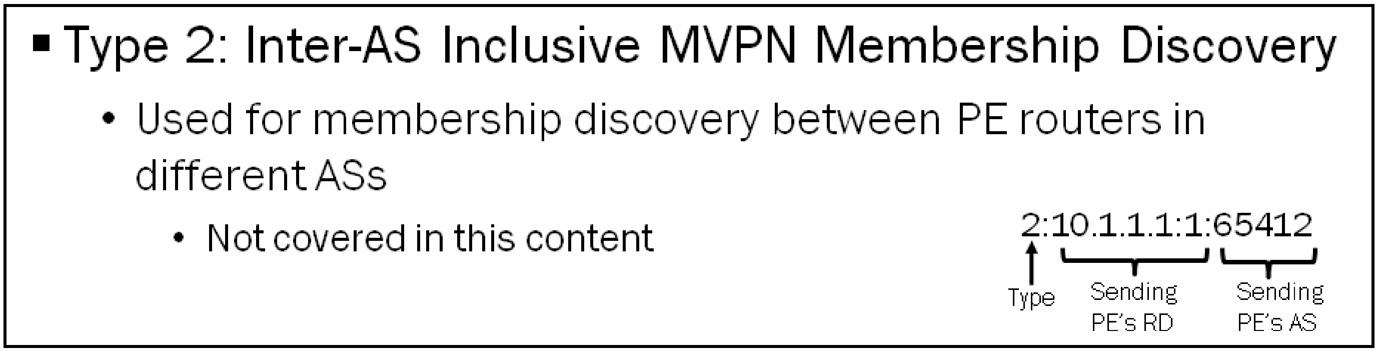


- Type 1: Intra-AS Inclusive MVPN Membership Discovery
  - Sent by all PE routers participating in MVPN
  - In the case of I-PMSI using RSVP-TE, these routes determine where to automatically build the point to multipoint LSPs
    - Routes are tagged with PMSI Tunnel attribute

1:10.1.1.1:1:10.1.1.1
Type
Sending PE's RD
Sending PE's lo0

The Intra-autonomous system (AS) I-PMSI autodiscovery route is the initial route type that is advertised between PEs of the same MVPN allowing them to autodiscover on another. It is distributed to other PEs that attach to sites of the MVPN. The routes carry the sending PE's route distinguisher (RD), the sending PE's loopback address, and a route target community to allow for import into a VRF. In the case of an inclusive provider tunnel, the route will also be tagged with the PMSI Tunnel attribute.

Type 2 NLRI



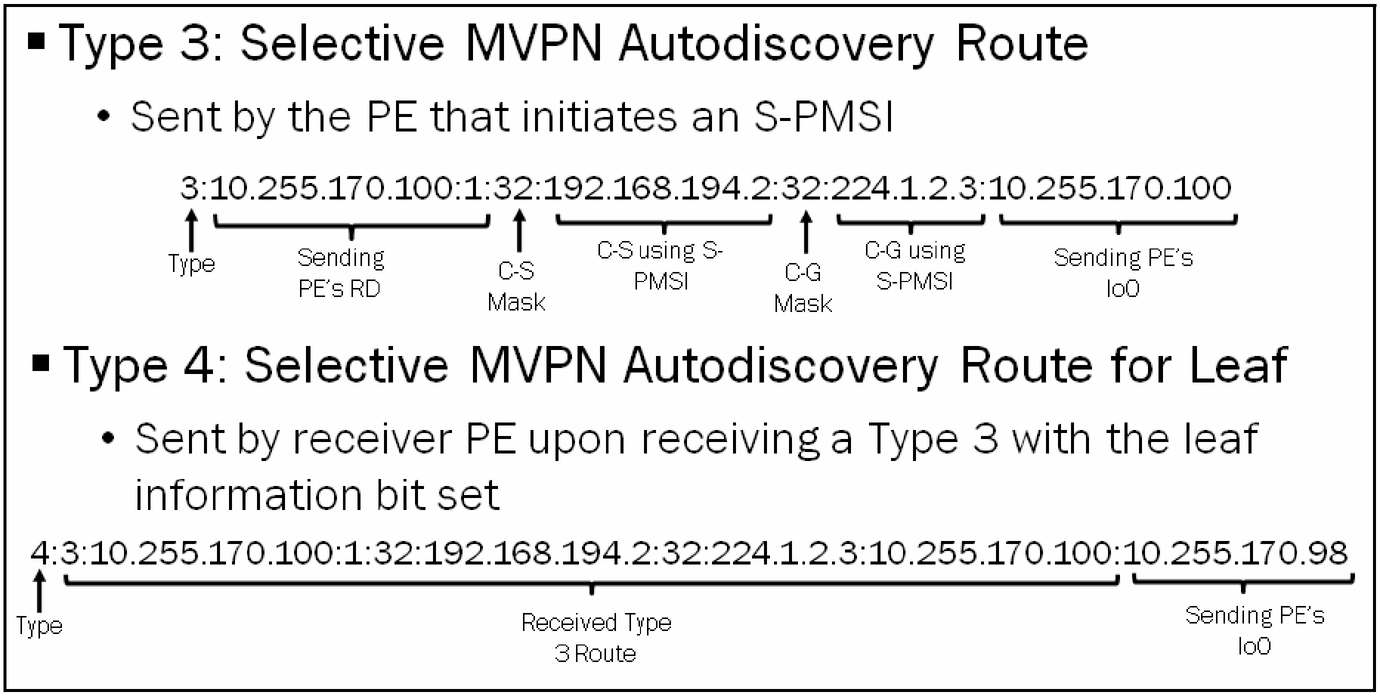
- Type 2: Inter-AS Inclusive MVPN Membership Discovery
  - Used for membership discovery between PE routers in different ASs
    - Not covered in this content

2:10.1.1.1:1:65412

Type
Sending PE's RD
Sending PE's AS

The Inter-AS I-PMSI autodiscovery route is used to discover members of an MVPN in different ASs. Inter-AS MVPNs are outside the scope of this guide.

Type 3 NLRI



- Type 3: Selective MVPN Autodiscovery Route
  - Sent by the PE that initiates an S-PMSI

3:10.255.170.100:1:32:192.168.194.2:32:224.1.2.3:10.255.170.100

Type
Sending PE's RD
C-S Mask
C-S using S-PMSI
C-G Mask
C-G using S-PMSI
Sending PE's lo0

- Type 4: Selective MVPN Autodiscovery Route for Leaf
  - Sent by receiver PE upon receiving a Type 3 with the leaf information bit set

4:3:10.255.170.100:1:32:192.168.194.2:32:224.1.2.3:10.255.170.100:10.255.170.98

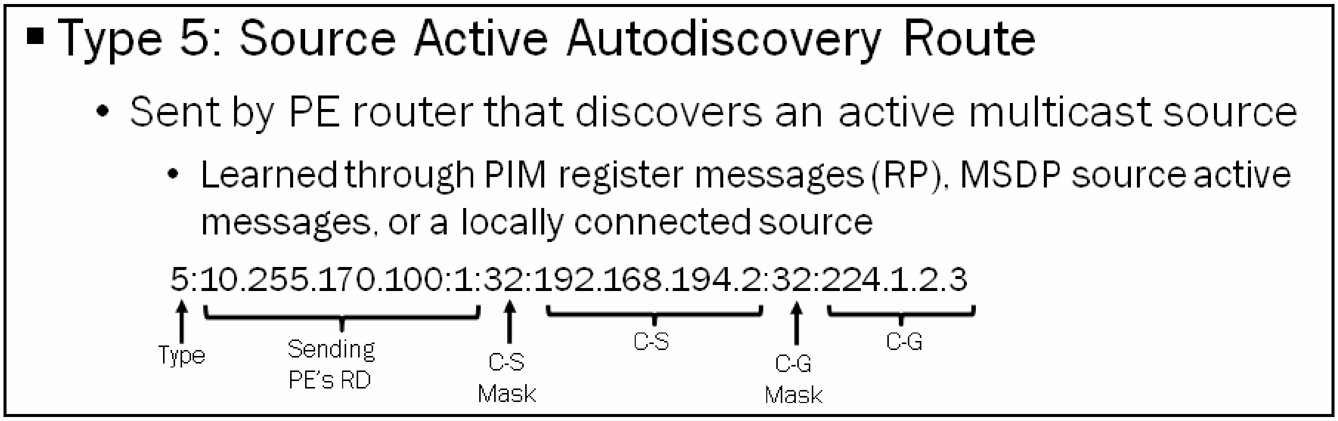
Type
Received Type 3 Route
Sending PE's lo0

Selective MVPN Autodiscovery routes are used to help build an S-PMSI. This route is advertised by the multicast source's PE in response to receiving a Type 6 or Type 7 route (described in subsequent sections) which are essentially requests to join the multicast forwarding tree (BGP version of a PIM join). The graphic shows the details of what is carried in the Type 3 route. Even though the source PE learns that the remote PE wants to receive a particular multicast stream from a type 7 advertisement, the source PE sends the type 3 as a request to the receiver PE to join the S-PMSI. The type 3 is tagged with the PMSI tunnel attribute allowing the receiver PEs to know the details of the provider tunnel.

Type 4 NLRI

The Selective MVPN autodiscovery route is sent by an interested receiver PE in response to receiving a type 3 route from a source PE.

Type 5 NLRI

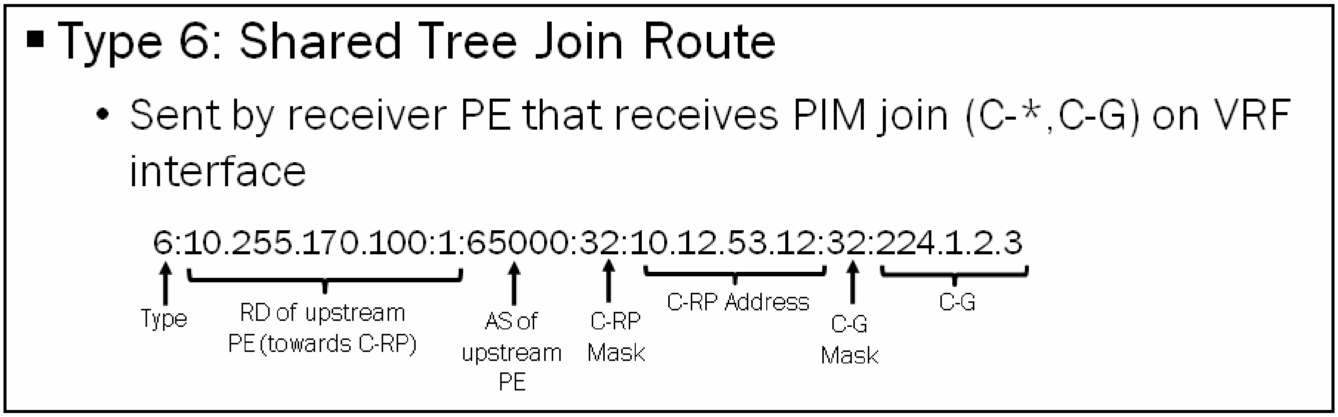


- Type 5: Source Active Autodiscovery Route
  - Sent by PE router that discovers an active multicast source
    - Learned through PIM register messages (RP), MSDP source active messages, or a locally connected source

5:10.255.170.100:1:32:192.168.194.2:32:224.1.2.3

The Source Active autodiscovery route is advertise by a a PE that discovers a source that is attached to a locally connected site. The PE learns of the source either from PIM register messages, Multicast Source Discovery Protocol (MSDP) source active messages, or a locally connected source. The source PE sends this advertisement to all other PEs participating in the MVPN.

Type 6 NLRI

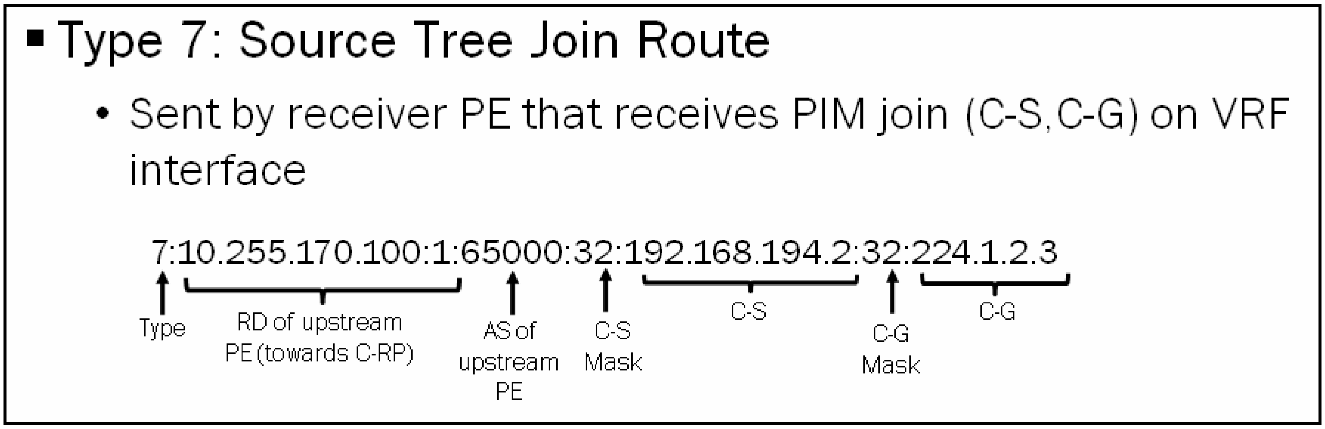


- Type 6: Shared Tree Join Route
  - Sent by receiver PE that receives PIM join (C-*,C-G) on VRF interface

6:10.255.170.100:1:65000:32:10.12.53.12:32:224.1.2.3

The Shared Tree Join route is advertised by a receiver PE to all other PEs participating in the MVPN in response to receiving a PIM (*,G) join from the local CE. It serves a similar purpose to the PIM (*,G) join in that it is a request to join the shared multicast tree.

Type 7 NLRI

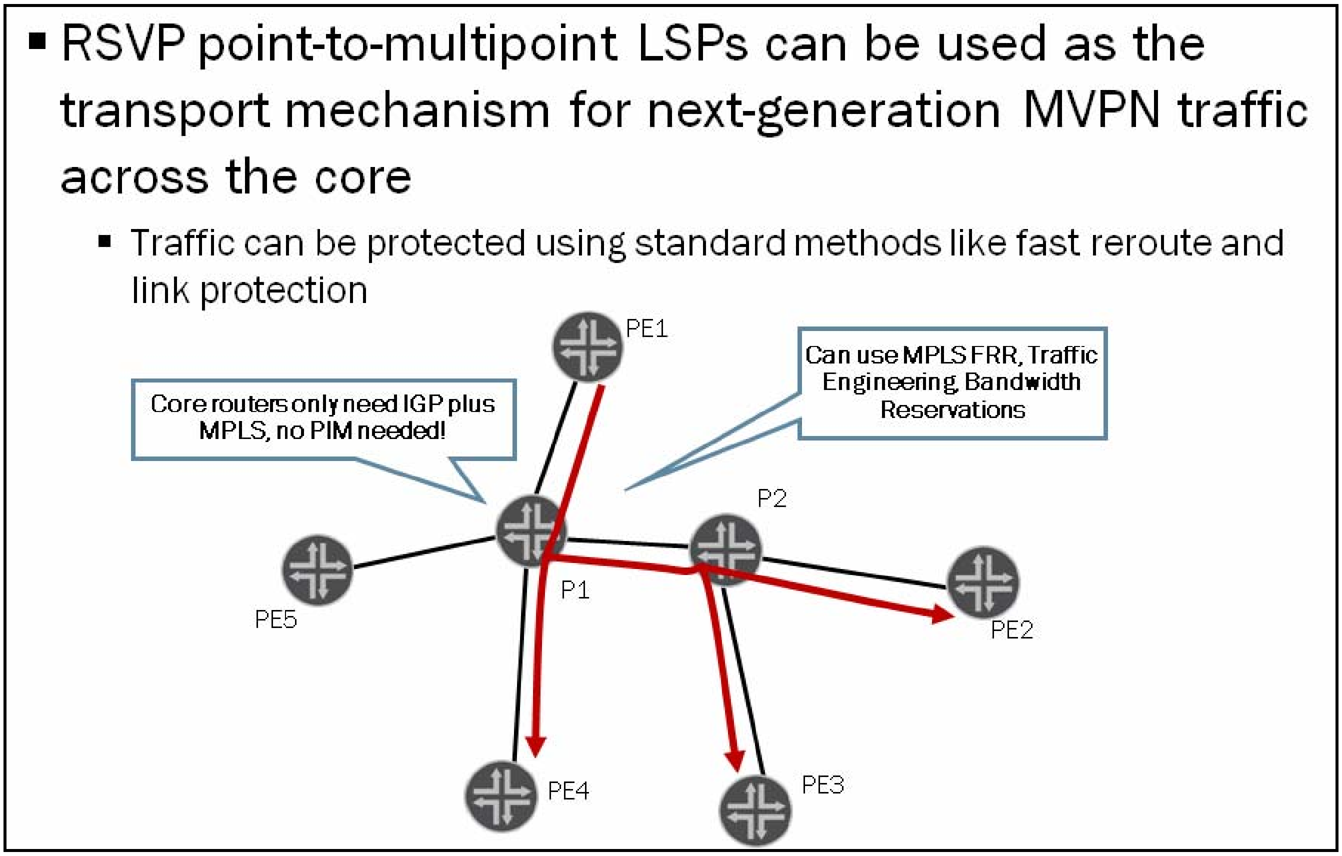


- Type 7: Source Tree Join Route
  - Sent by receiver PE that receives PIM join (C-S,C-G) on VRF interface

7:10.255.170.100:1:65000:32:192.168.194.2:32:224.1.2.3

The Source Tree Join route is advertised by a receiver PE to all other PEs participating in the MVPN in response to receiving a PIM (S,G) join from the local CE. It serves a similar purpose to the PIM (S,G) join in that it is a request to join the source multicast tree.

RSVP Point-to-Multipoint LSPs



- RSVP point-to-multipoint LSPs can be used as the transport mechanism for next-generation MVPN traffic across the core
  - Traffic can be protected using standard methods like fast reroute and link protection

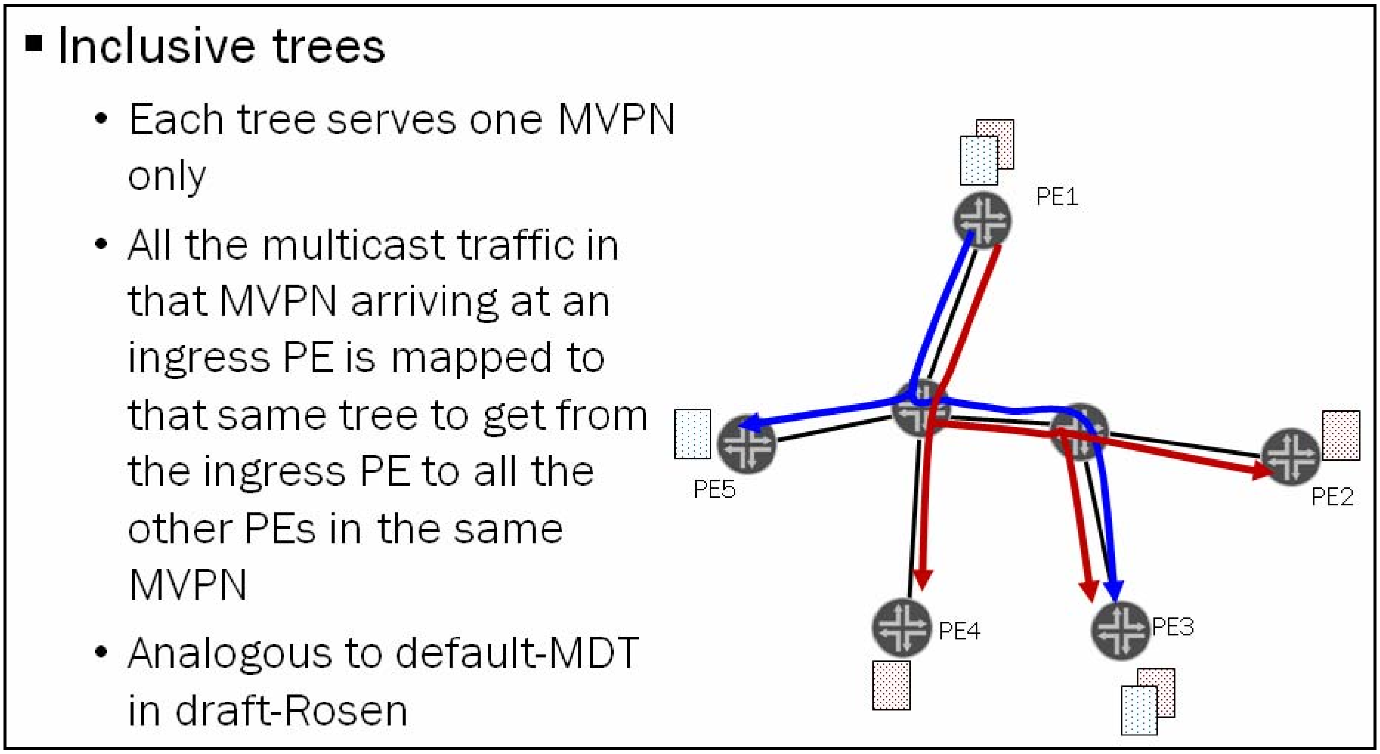
Core routers only need IGP plus MPLS, no PIM needed!

Can use MPLS FRR, Traffic Engineering, Bandwidth Reservations

One transport mechanism that can be used in next-generation MVPN scenario is RSVP-signalled point-to-multipoint LSPs. There are several benefits to using point-to-multipoint LSPs in the service provider network:

1. The burden of data replication is taken off of the ingress PE. Instead, each router along the path of the LSP can help in that responsibility.
2. Multicast traffic can be protected using the standard methods of RSVP protection like fast-reroute and link protection.
3. Certain levels of performance can be guaranteed with the use of traffic engineering and bandwidth reservation.
4. The service provider network does not need to run PIM to support multicast routing. Multicast routing of customer traffic can occur on the same IP/MPLS design that was used to build the unicast L3VPNs.

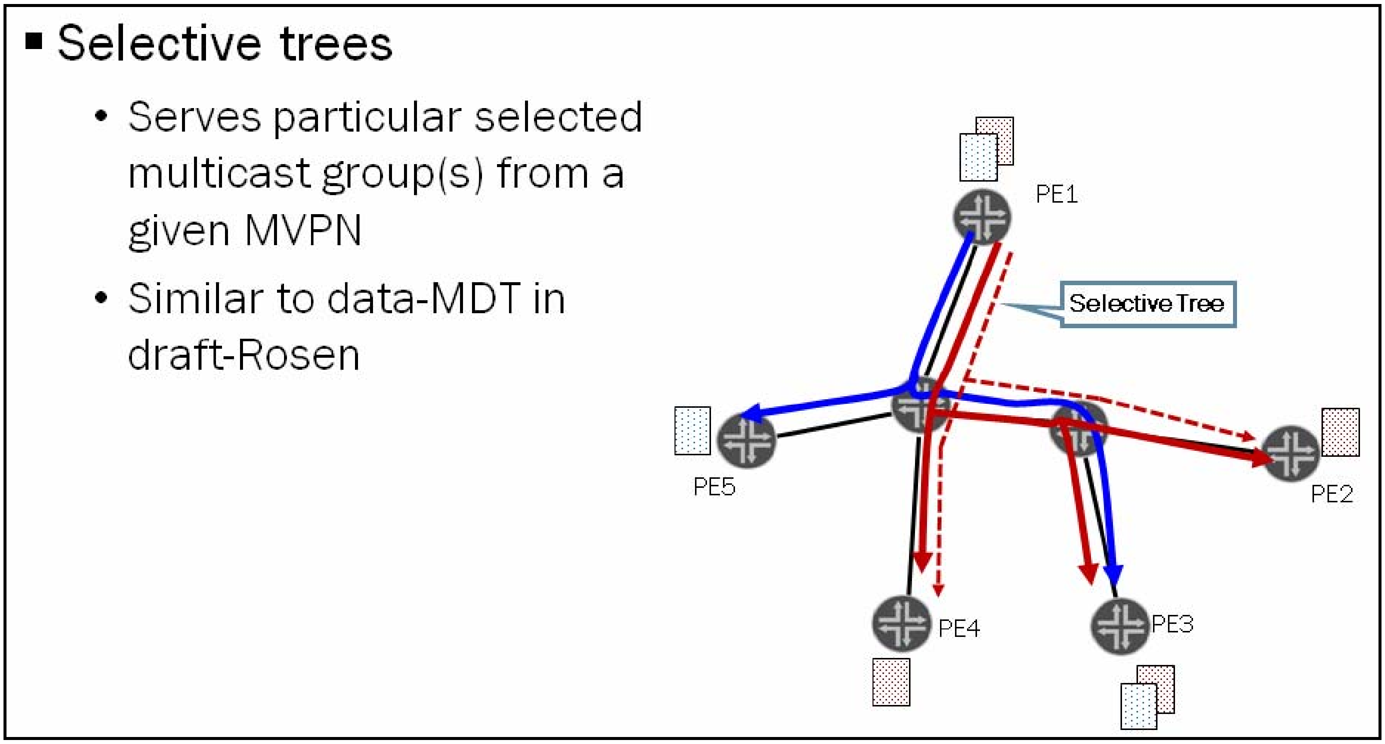
Inclusive Trees



- Inclusive trees
  - Each tree serves one MVPN only
  - All the multicast traffic in that MVPN arriving at an ingress PE is mapped to that same tree to get from the ingress PE to all the other PEs in the same MVPN
  - Analogous to default-MDT in draft-Rosen

The simplest form of provider tunnel is the inclusive tree (I-PMSI). An inclusive tree serves an entire MVPN. In the diagram, there is one inclusive tree that serves the blue VPN and one that serves the red VPN. Any multicast traffic arriving at the source PE (PE-1) will be sent to all other PEs in the same MVPN. This works well when all remote PEs need to receive the multicast traffic but this form of tree can be wasteful of resources (bandwidth, packet processing, and so on) when only a few of the remote PEs need to receive multicast traffic. The solution to this problem is the use of selective trees described in subsequent sections.

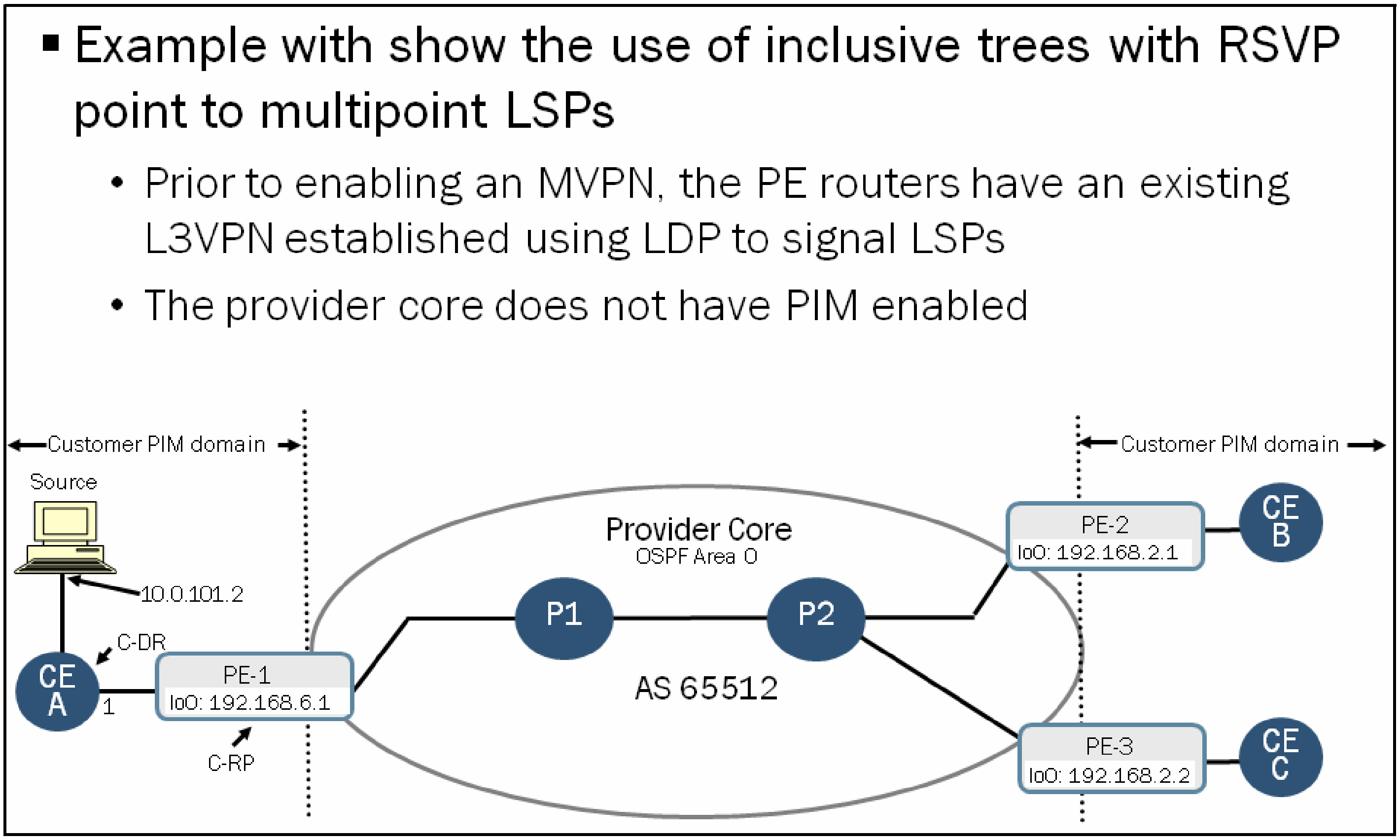
Selective Tree



- Selective trees
  - Serves particular selected multicast group(s) from a given MVPN
  - Similar to data-MDT in draft-Rosen

Selective trees can be used to forward traffic for particular source and group combinations to the remote PE that specifically request to receive that traffic. The dotted line in the diagram shows that a point-to-multipoint LSP has been built to send multicast traffic for the red VPN to PE2 and PE4 only.

Inclusive Tree Example—Initial State



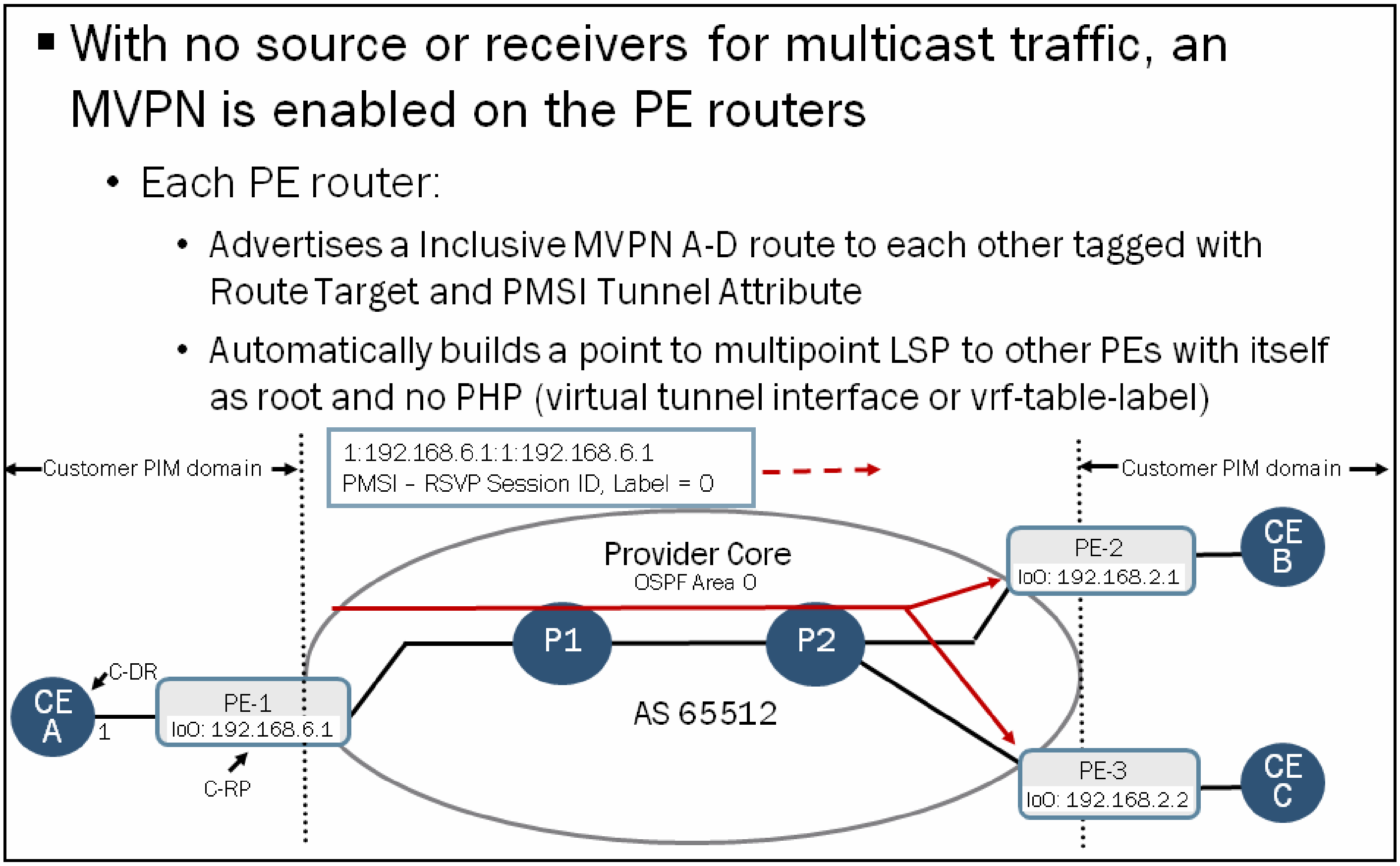
- Example with show the use of inclusive trees with RSVP point to multipoint LSPs
  - Prior to enabling an MVPN, the PE routers have an existing L3VPN established using LDP to signal LSPs
  - The provider core does not have PIM enabled

The graphic shows an example L3VPN prior to enabling next-generation MVPN.

Some things to note are:

1. The provider core is not running PIM;
2. There is an existing L3VPN between all customer sites using LDP to signal the unicast LSPs;
3. PE-1 will be acting as the customer rendezvous point (RP) (within the VRF);
4. CE-A will be acting as the customer designated router (DR) closest to the source; and
5. CE-B and CE-C will eventually have receivers attached.

Inclusive Tree Example—Enabling the MVPN



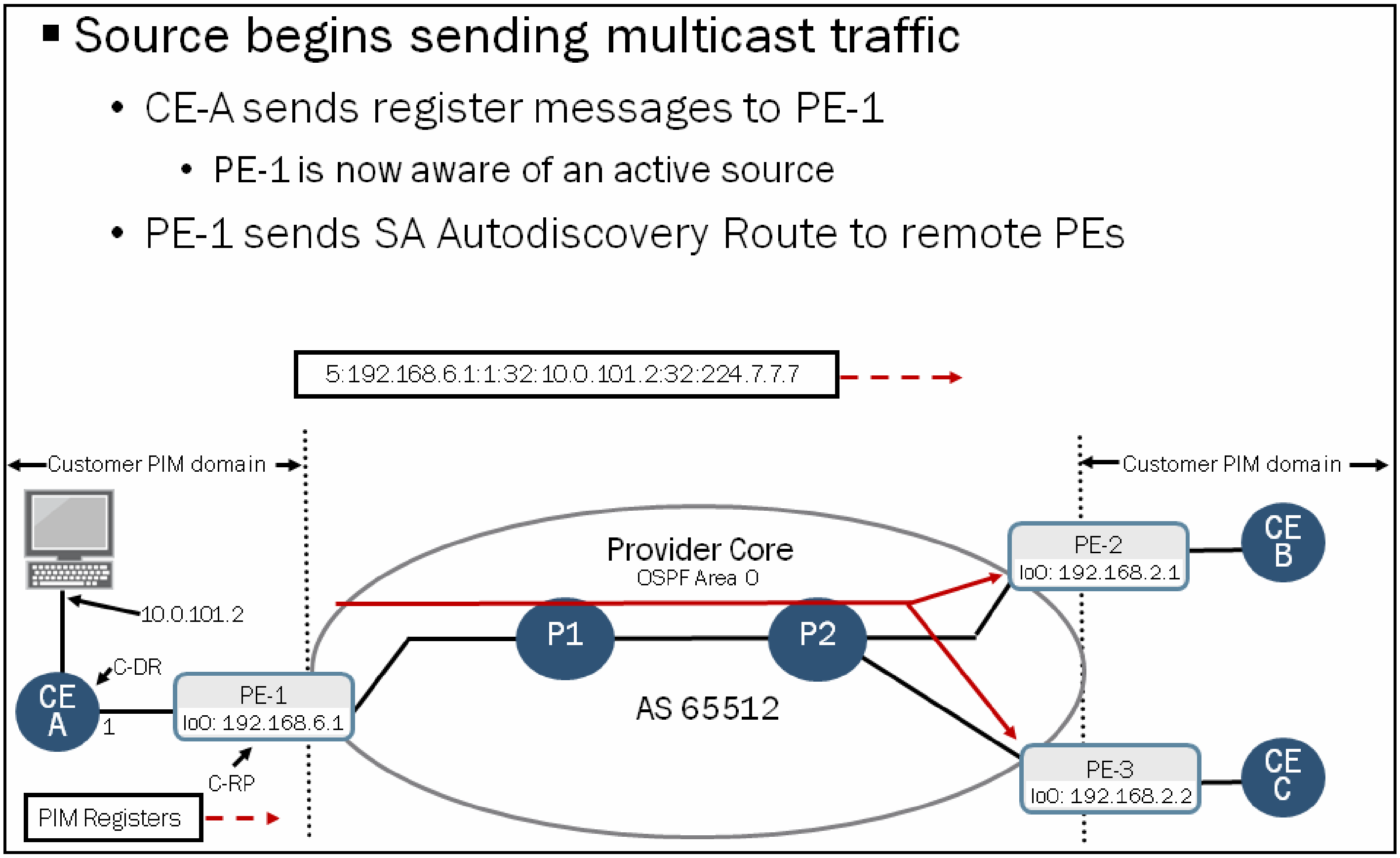
- With no source or receivers for multicast traffic, an MVPN is enabled on the PE routers
  - Each PE router:
    - Advertises a Inclusive MVPN A-D route to each other tagged with Route Target and PMSI Tunnel Attribute
    - Automatically builds a point to multipoint LSP to other PEs with itself as root and no PHP (virtual tunnel interface or vrf-table-label)

1:192.168.6.1:1:192.168.6.1
PMSI - RSVP Session ID, Label = 0

With no source and receivers in the network, an MVPN is enabled on all three PEs. Once enabled, each PE will advertise their membership to the MVPN using a Type 1 route tagged with the PMSI tunnel attribute. Each PE will automatically build a point-to-multipoint LSP to all other PEs. In the network shown on the graphic, there will only ever be a single source attached to PE-1. Because PE-2 and PE-3, will never be attached to a source site, the point-to-multipoint LSP that each of them instantiated as themselves as ingress routers will never be used. It is possible to configure PE-2 and PE-3 as receiver-only sites so that they do not build the unnecessary point-to-multipoint LSPs.

When PE-2 and PE-3 eventually receive multicast traffic from PE-1 using the point-to-multipoint LSP, they will need to use the incoming MPLS label encapsulating the multicast packets to determine which VRF to use for forwarding. Normally a point-to-multipoint LSP is signalled with a label of 3 on the penultimate hops meaning that there would be no label encapsulating the incoming traffic. Therefore, a virtual tunnel interface or vrf-table-label must be configured within the VRF to allow for a non-implicit-null label to be used on the penultimate hops.

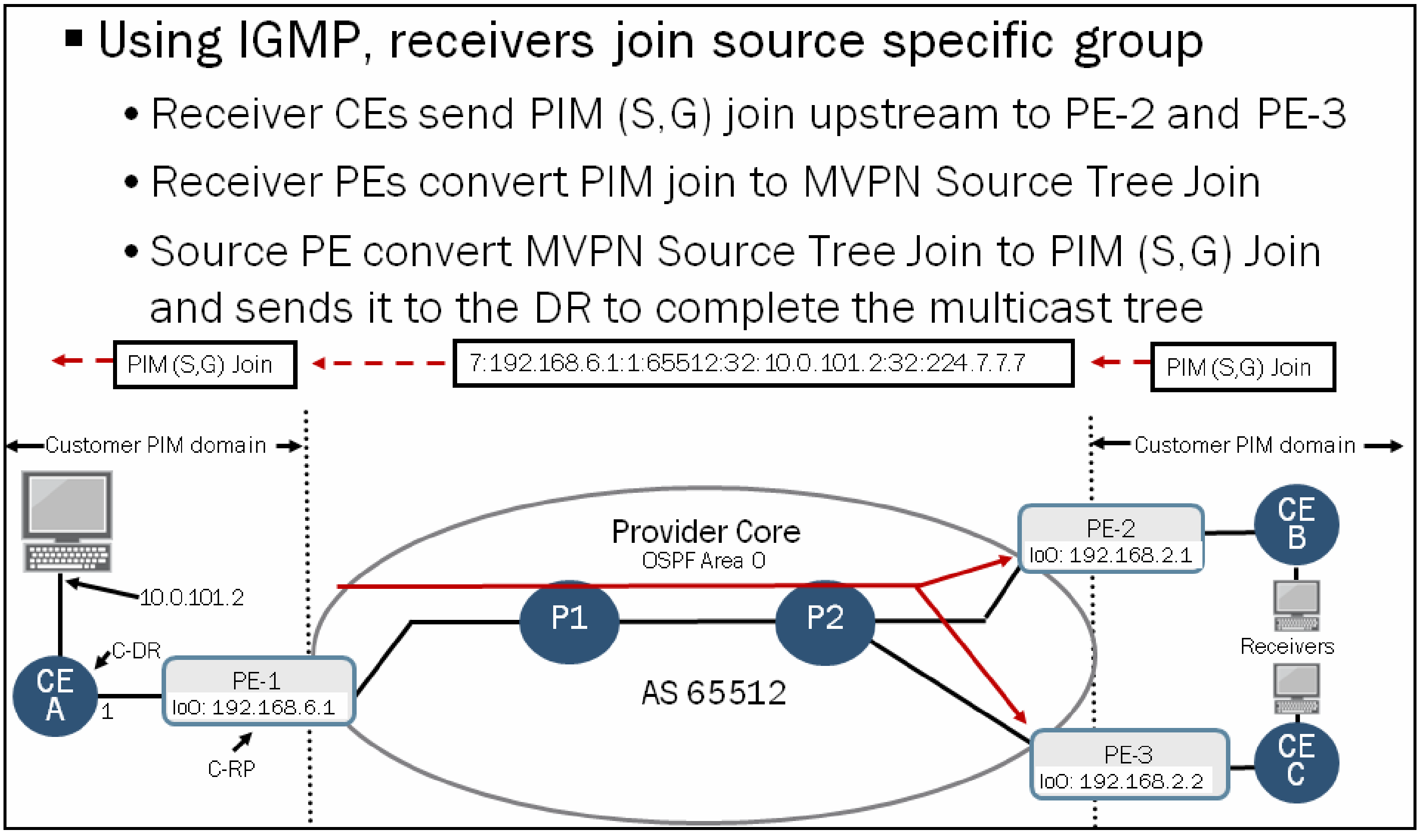
Inclusive Tree Example—Source Begins Sending Traffic



- Source begins sending multicast traffic
  - CE-A sends register messages to PE-1
    - PE-1 is now aware of an active source
  - PE-1 sends SA Autodiscovery Route to remote PEs

With the MVPN now established, the source attached to CE-A begins sending multicast traffic. As the customer PIM DR, CE-A encapsulates the multicast traffic in register messages and unicasts that traffic to the customer RP (C-RP), PE-1. PE-1 learning of a new source in the customer's network advertises that source in the form of the Type 5 Source Active autodiscovery route to all other PEs of the MVPN.

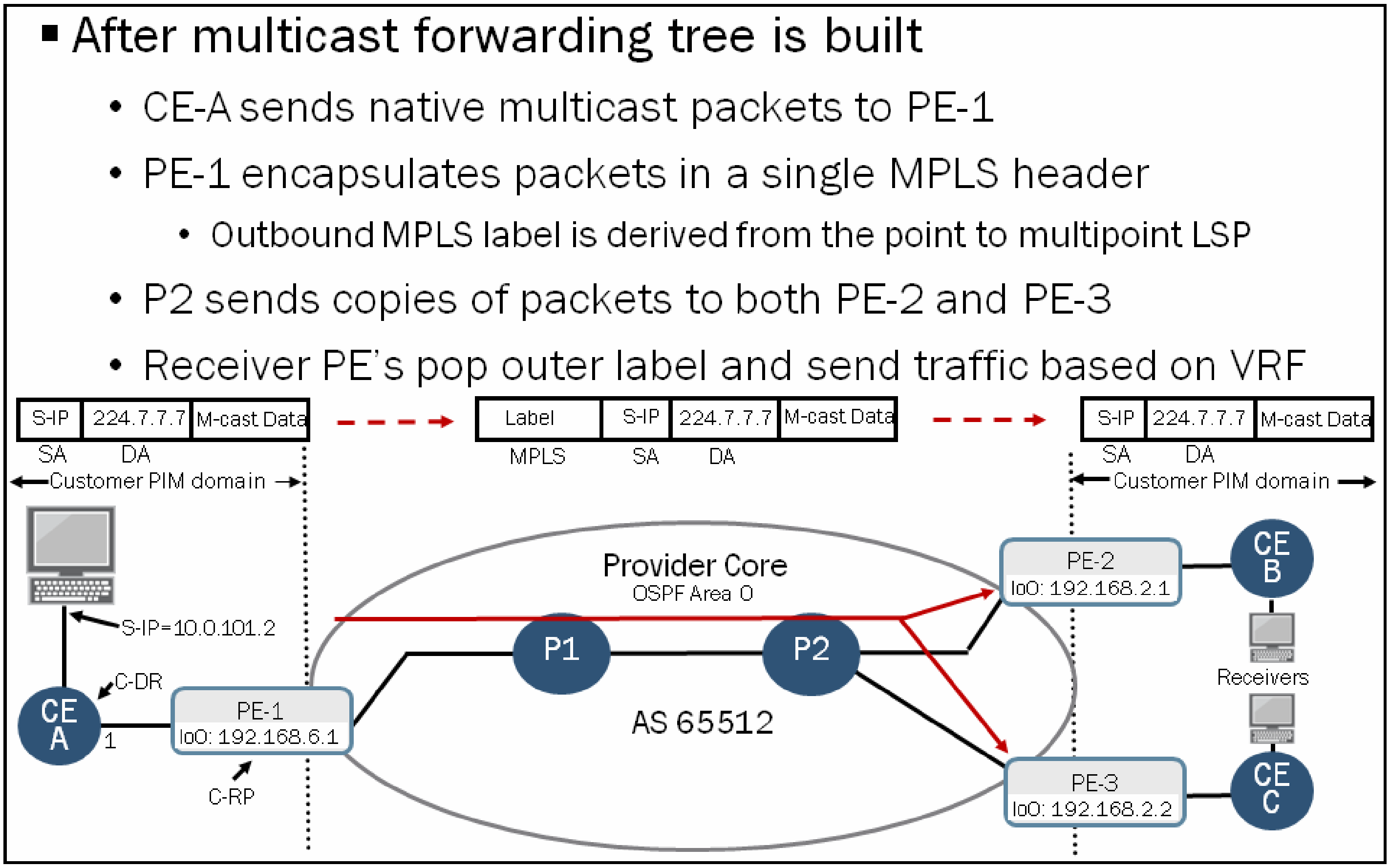
Inclusive Tree Example—Receivers Join



- Using IGMP, receivers join source specific group
  - Receiver CEs send PIM (S,G) join upstream to PE-2 and PE-3
  - Receiver PEs convert PIM join to MVPN Source Tree Join
  - Source PE convert MVPN Source Tree Join to PIM (S,G) Join and sends it to the DR to complete the multicast tree

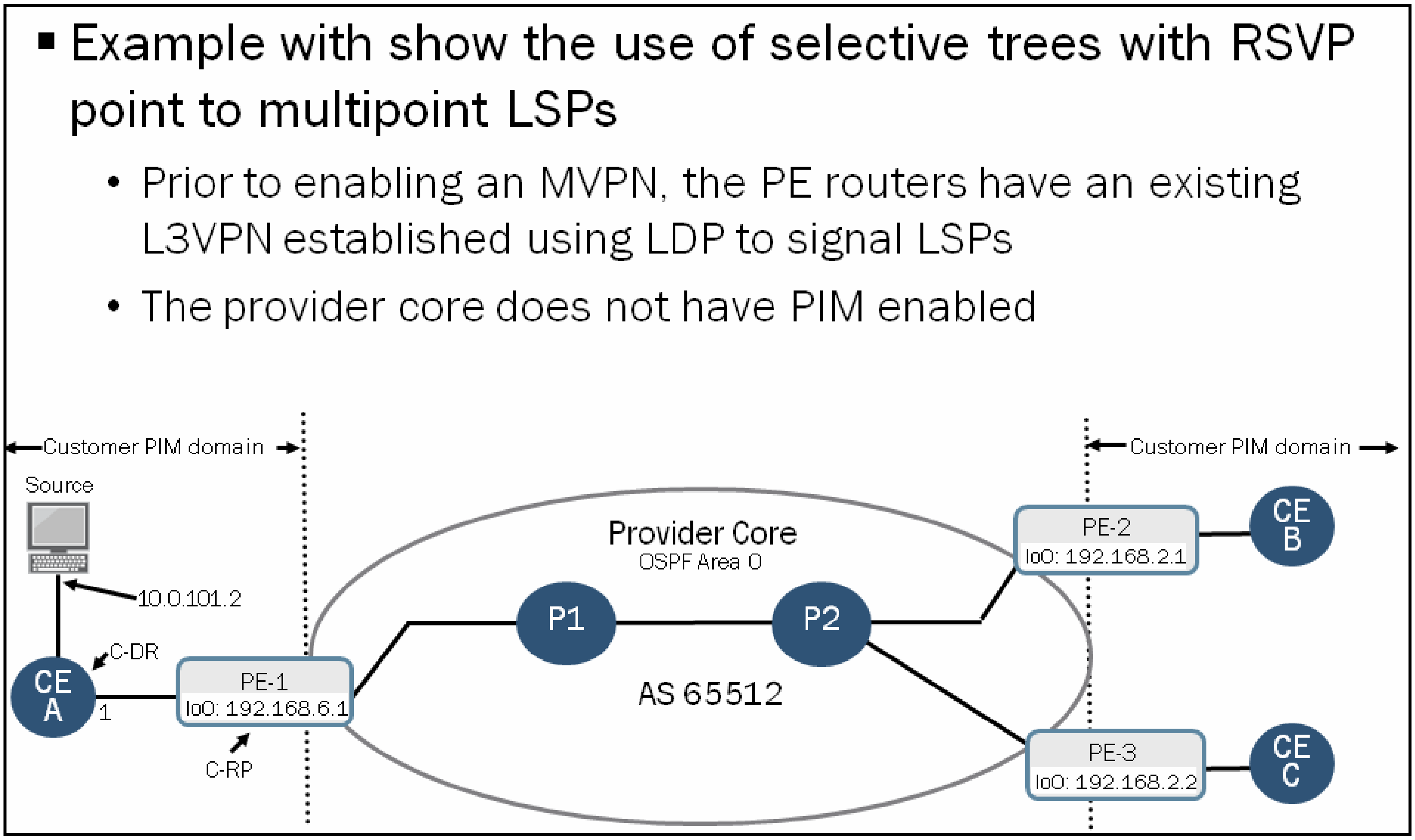
Using Internet Group Management Protocol (IGMP) version 3, the hosts attached to CE-B and CE-C report their membership to a specific multicast source and group. CE-B and CE-C in turn send a PIM (S, G) join upstream towards the source. Upon receiving the PIM joins, PE-2 and PE-3 send Type 7 Source Tree Join routes to the source PE, PE-1. Upon receiving the Type 7 advertisement from the remote PE, PE-1 sends a PIM (S, G) join upstream to the customer's DR router, CE-A. At the point the multicast forwarding tree is complete and multicast traffic forwarded from the source to receivers.

I-PMSI Forwarding



- After multicast forwarding tree is built
  - CE-A sends native multicast packets to PE-1
  - PE-1 encapsulates packets in a single MPLS header
    - Outbound MPLS label is derived from the point to multipoint LSP
  - P2 sends copies of packets to both PE-2 and PE-3
  - Receiver PE's pop outer label and send traffic based on VRF

Now that the multicast forwarding tree is complete, multicast traffic can be sent from end to end. From the source to PE-1, multicast packets are forwarded in their native format. From PE-1 to P1, multicast packets are encapsulated in the MPLS header that is associated with the point-to-multipoint LSP that uses PE-1 as the ingress router. P1 simply performs a label swap. At P2, because of the behavior of point-to-multipoint LSP, the data traffic is replicated, the label is swapped, and then sent to both remote PEs. The receiving PEs pop the incoming label and use the label to determine the VRF to use for forwarding. The receiving PE then send the multicast traffic in its native format towards the receivers.



- Example with show the use of selective trees with RSVP point to multipoint LSPs
  - Prior to enabling an MVPN, the PE routers have an existing L3VPN established using LDP to signal LSPs
  - The provider core does not have PIM enabled

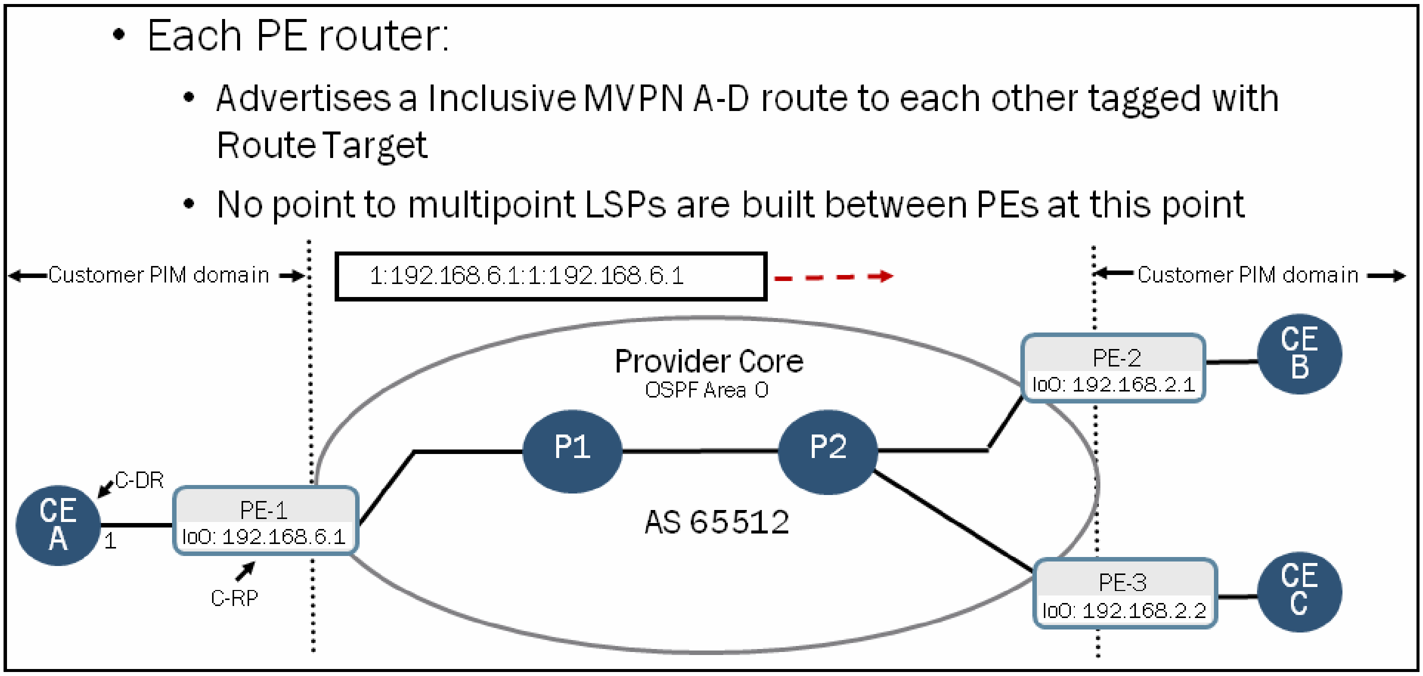
Selective Tree Example—Initial State

The graphic shows an example L3VPN prior to enabling next-generation MVPN.

Some things to note are:

1. The provider core is not running PIM;
2. There is an existing L3VPN between all customer sites using LDP to signal the unicast LSPs;
3. PE-1 will be acting as the customer RP (within the VRF);
4. CE-A will be acting as the customer DR closest to the source; and
5. Only CE-B will eventually have a receiver attached.

Selective Tree Example—Enabling the MVPN

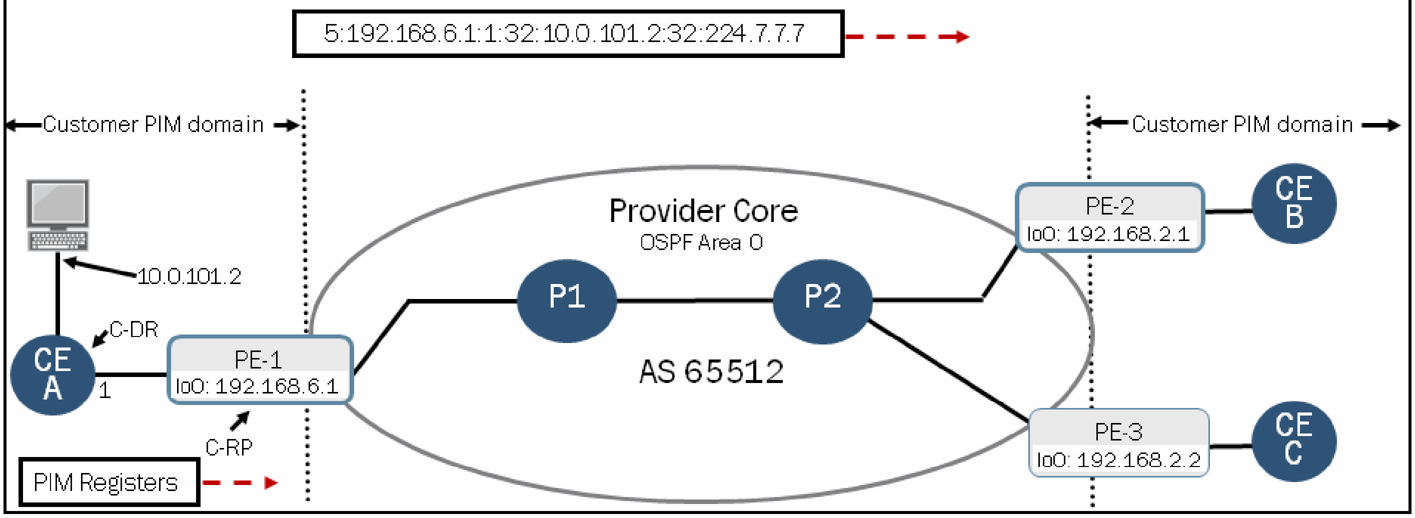


- Each PE router:
  - Advertises a Inclusive MVPN A-D route to each other tagged with Route Target
  - No point to multipoint LSPs are built between PEs at this point

With no source and receivers in the network, an MVPN is enabled on all three PEs. Once enabled, each PE will advertise their membership to the MVPN using a Type 1 route, however the Type 1 routes will not be tagged with the PMSI tunnel attribute. In an S-PMSI scenario, a point-to-multipoint LSP is not built until at least one PE has a receiver attached.

Selective Tree Example—Source Begins Sending Traffic

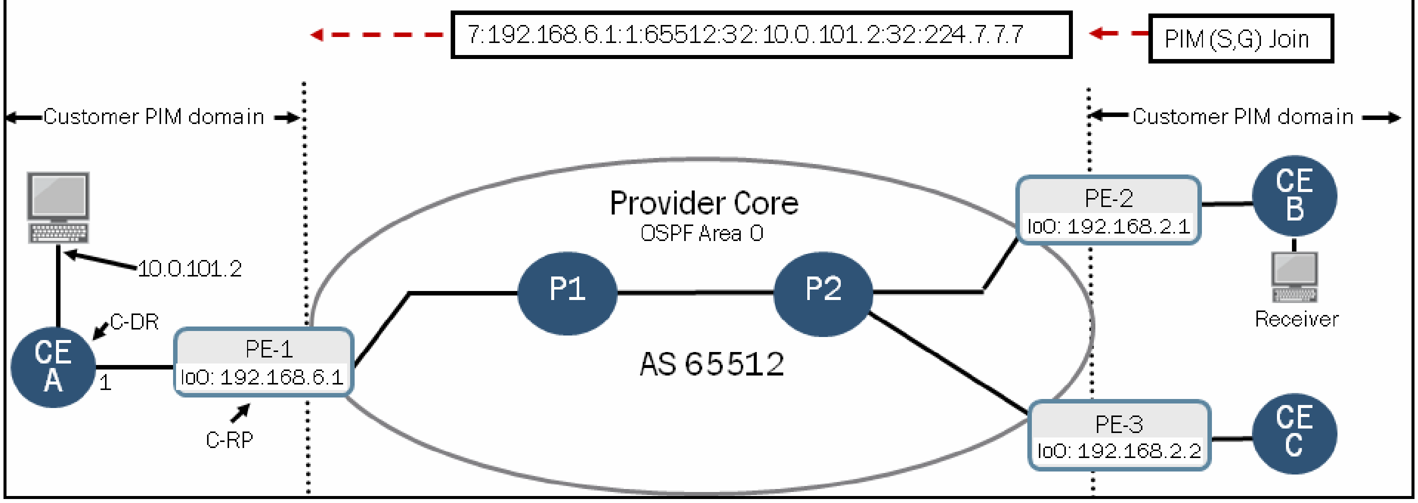
- CE-A sends register messages to PE-1
  - PE-1 is now aware of an active source
- PE-1 sends SA Autodiscovery Route to remote PEs



The source attached to CE-A begins sending multicast traffic. As the customer PIM DR, CE-A encapsulates the multicast traffic in register messages and unicasts that traffic to the customer RP (C-RP), PE-1. PE-1 learning of a new source in the customer's network advertises that source in the form of the Type 5 Source Active autodiscovery route to all other PEs of the MVPN.

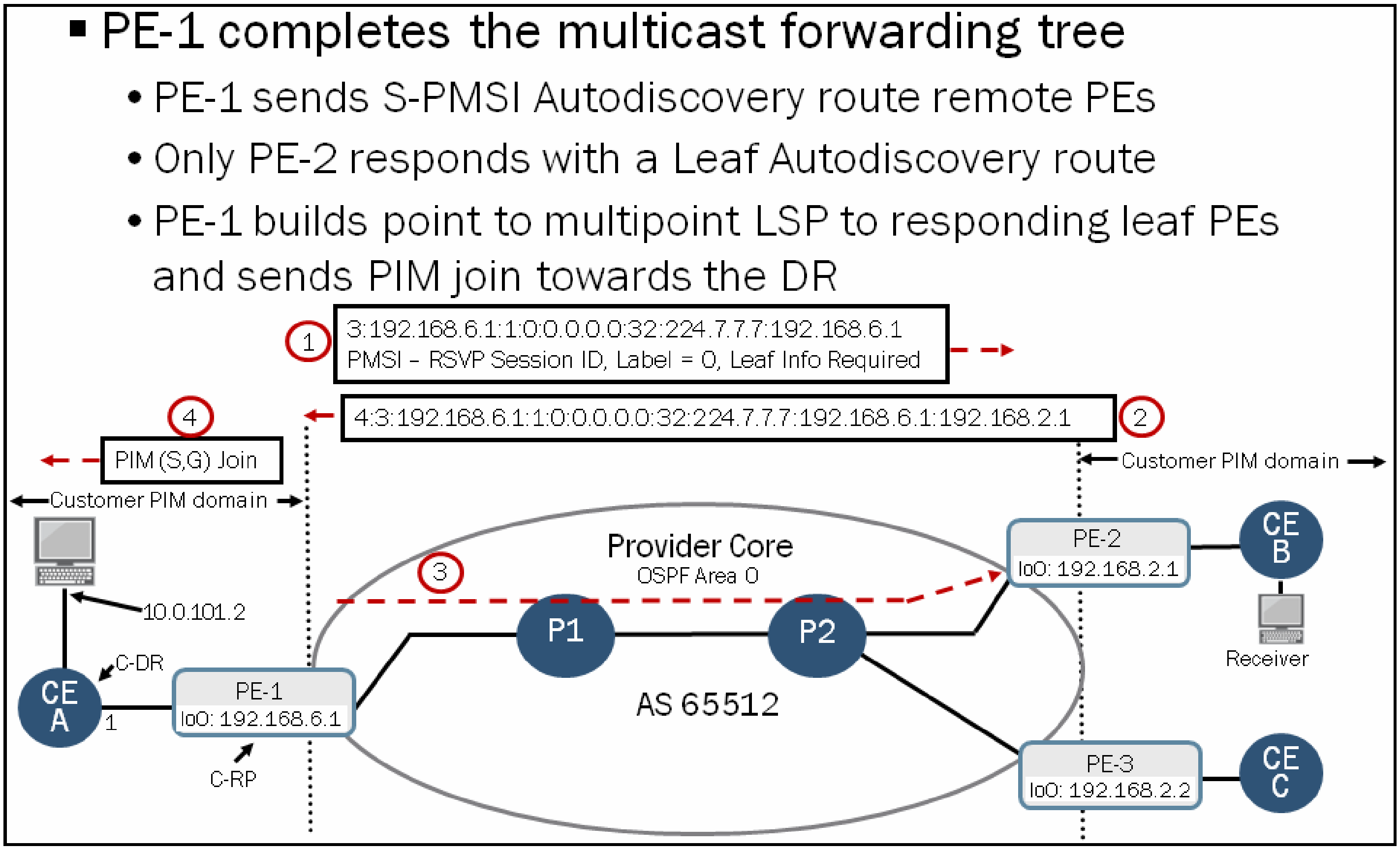
Selective Tree Example—Receivers Join

- Using IGMP, receivers join source specific group
  - Receiver CE-B sends PIM (S,G) join upstream to
  - Receiver PE-2 converts PIM join to MVPN Source Tree Join
  - No receiver attached to CE-C



Using IGMP version 3, the host attached to CE-B reports its membership to a specific multicast source and group. CE-B in turn sends a PIM (S, G) join upstream towards the source. Upon receiving the PIM joins, PE-2 sends a Type 7 Source Tree Join route to the source PE, PE-1.

Selective Tree Example—Completing the Forwarding Tree



- PE-1 completes the multicast forwarding tree
  - PE-1 sends S-PMSI Autodiscovery route remote PEs
  - Only PE-2 responds with a Leaf Autodiscovery route
  - PE-1 builds point to multipoint LSP to responding leaf PEs and sends PIM join towards the DR

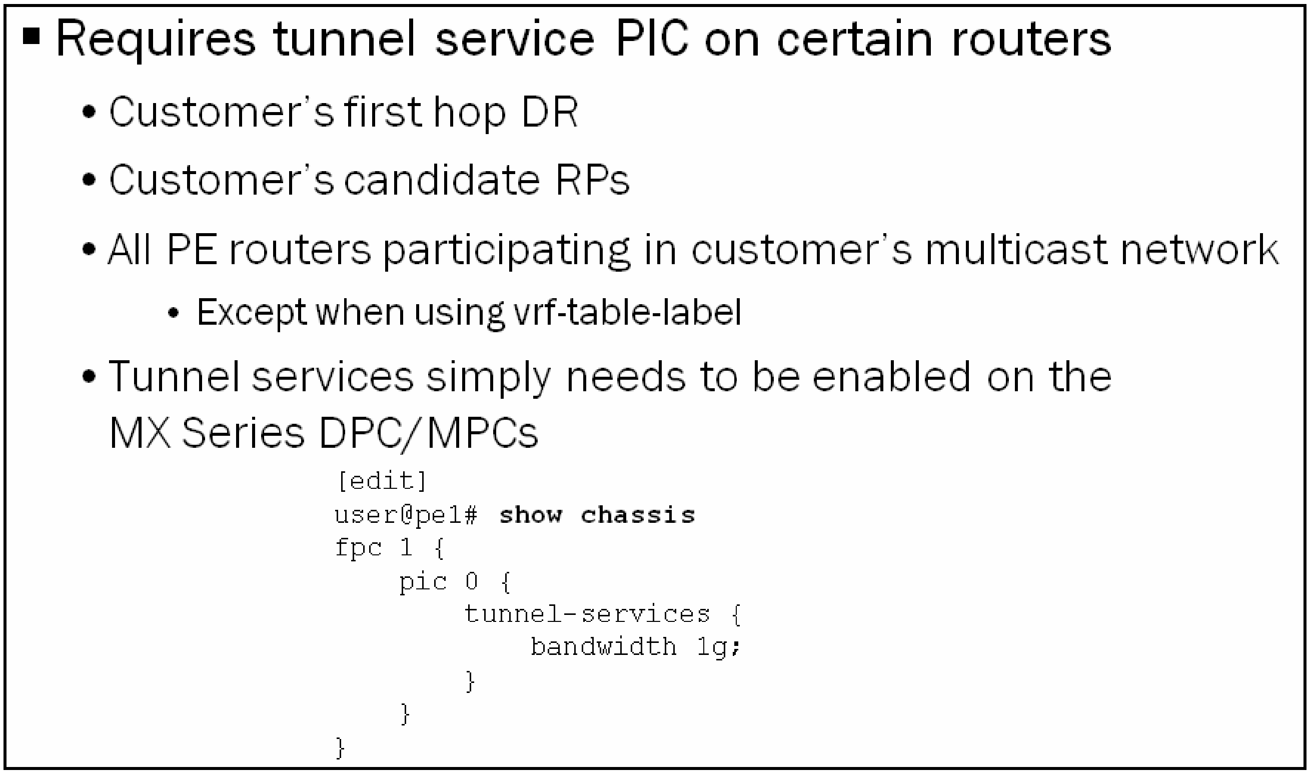
1 3:192.168.6.1:1:0:0.0.0.0:32:224.7.7.7:192.168.6.1
PMSI – RSVP Session ID, Label = 0, Leaf Info Required

2 4:3:192.168.6.1:1:0:0.0.0.0:32:224.7.7.7:192.168.6.1:192.168.2.1

4 PIM (S,G) Join

Upon receiving the Type 7 advertisement from the PE-2, PE-1 sends a Type 3 S-PMSI autodiscovery route tagged with PMSI Tunnel attribute with the leaf information required bit set. PE-2 now knows the RSVP session ID of the point-to-multipoint LSP that will be used for forwarding. PE-2 then responds to PE-1 with a Type 4 Leaf autodiscovery route. PE-1 builds a point-to-multipoint LSP to all PEs that responded with a Type 4. In this case. PE-1 builds a point-to-multipoint LSP to a single end-point, PE-2. Finally, PE-1 sends a PIM (S, G) join upstream to the customer's DR router, CE-A. At the point the multicast forwarding tree is complete and multicast traffic forwarded from the source to receivers.

Tunnel Services



- Requires tunnel service PIC on certain routers
  - Customer's first hop DR
  - Customer's candidate RPs
  - All PE routers participating in customer's multicast network
    - Except when using vrf-table-label
  - Tunnel services simply needs to be enabled on the MX Series DPC/MPCs

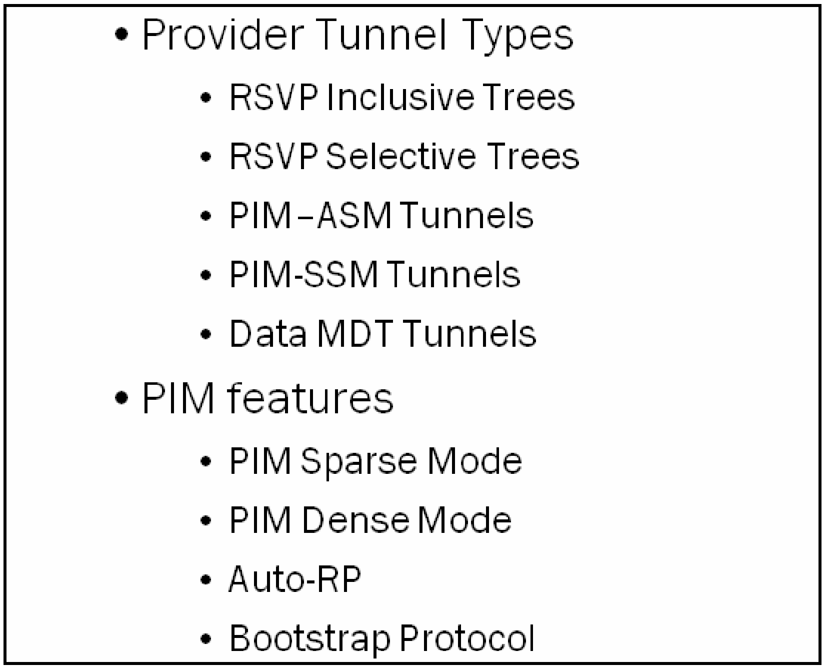
```
[edit]
user@pe1# show chassis
fpc 1 {
    pic 0 {
        tunnel-services {
            bandwidth 1g;
        }
    }
}
```

Assuming every router is running the Junos OS, tunnel services must be enabled on certain routers. Some routers require a Tunnel Service PIC or Adaptive Service PIC to provide tunnel services. In the case of the MX Series device, the feature just needs to be enabled as shown on the graphic.

Router types needing tunnel services:

- DR closest to source - Tunnel services are needed because the DR must encapsulate multicast traffic in unicast messages called register messages;
- Customer's candidate RP - Tunnel services are needed because the RP must de-encapsulate the register messages received from the DR; and
- All MVPN PE routers - Tunnel services are needed because it allows the PE to pop the incoming MPLS header from the incoming multicast traffic, perform an RPF check on the multicast traffic, and then forward the traffic out of the VRF interface. This is assuming that a virtual tunnel interface is used. Optionally, `vrf-table-label` can be configured without the need for tunnel services.

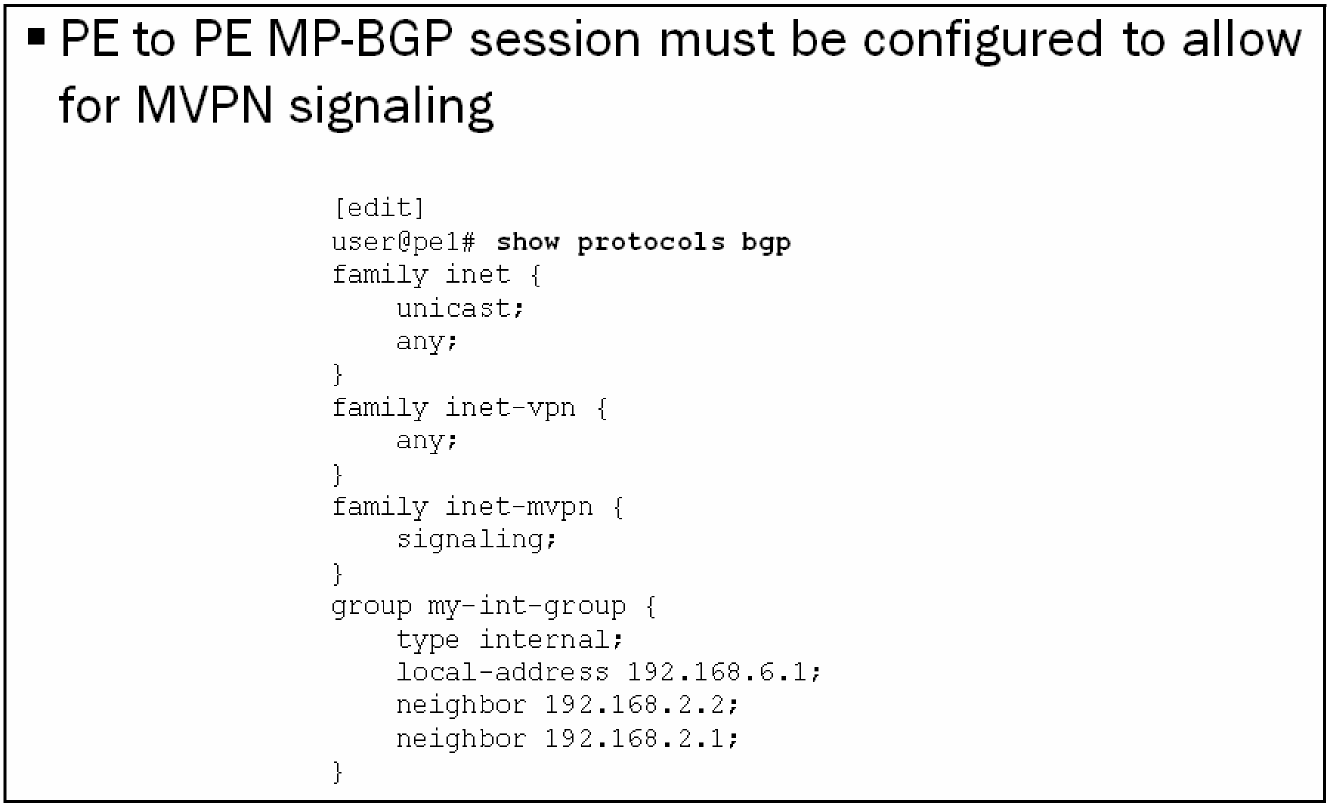
Junos OS Support



- Provider Tunnel Types
  - RSVP Inclusive Trees
  - RSVP Selective Trees
  - PIM –ASM Tunnels
  - PIM-SSM Tunnels
  - Data MDT Tunnels
- PIM features
  - PIM Sparse Mode
  - PIM Dense Mode
  - Auto-RP
  - Bootstrap Protocol

The graphic shows the protocols that are supported when enabling next-generation MVPNs using the Junos OS.

MP-BGP Configuration

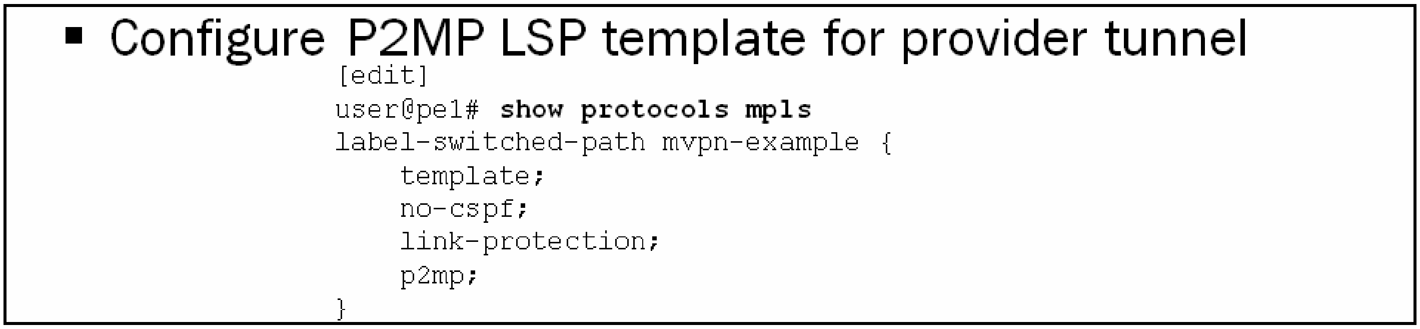


- PE to PE MP-BGP session must be configured to allow for MVPN signaling

```
[edit]
user@pe1# show protocols bgp
family inet {
    unicast;
    any;
}
family inet-vpn {
    any;
}
family inet-mvpn {
    signaling;
}
group my-int-group {
    type internal;
    local-address 192.168.6.1;
    neighbor 192.168.2.2;
    neighbor 192.168.2.1;
}
```

To allow for BGP neighbors to exchange the new MVPN NLRI, `family inet-mvpn signaling` must be enabled on all participating PE routers.

Optional Point-to-Multipoint LSP Template

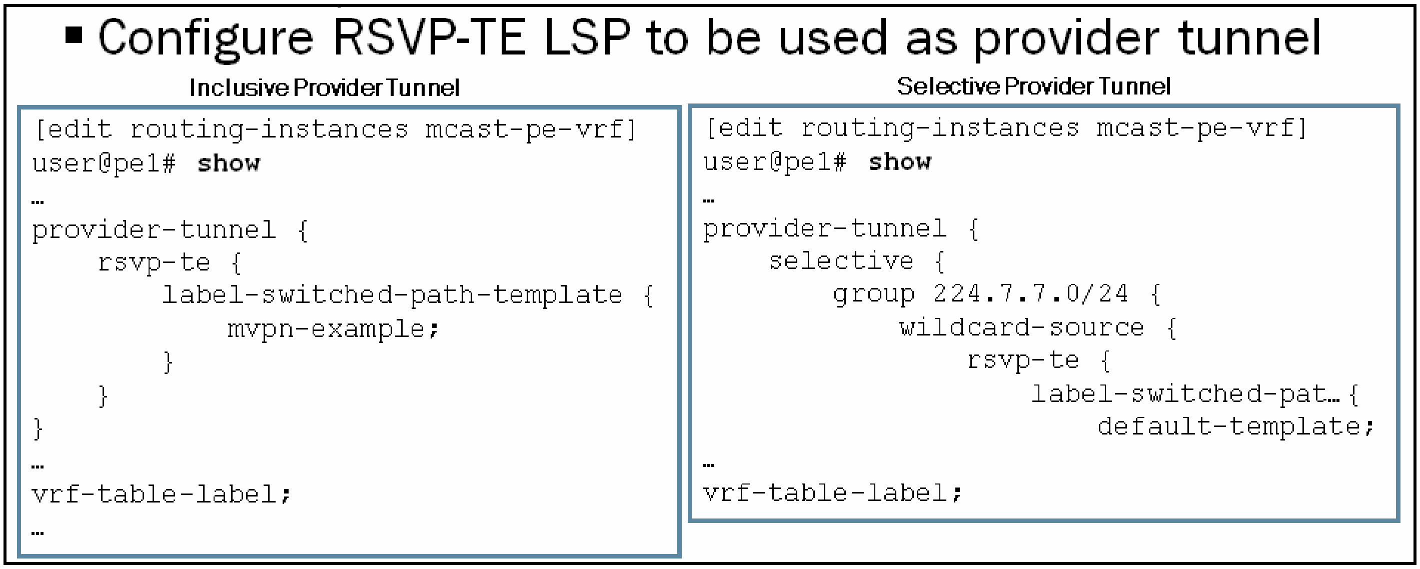


- Configure P2MP LSP template for provider tunnel

```
[edit]
user@pe1# show protocols mpls
label-switched-path mvpn-example {
    template;
    no-cspf;
    link-protection;
    p2mp;
}
```

You can optionally specify the requirements of the point-to-multipoint LSP by creating a template under `[edit protocols mpls]`. You can specify protection requirements, bandwidth requirements, path information, and more.

Provider Tunnel Type



- Configure RSVP-TE LSP to be used as provider tunnel

Inclusive Provider Tunnel

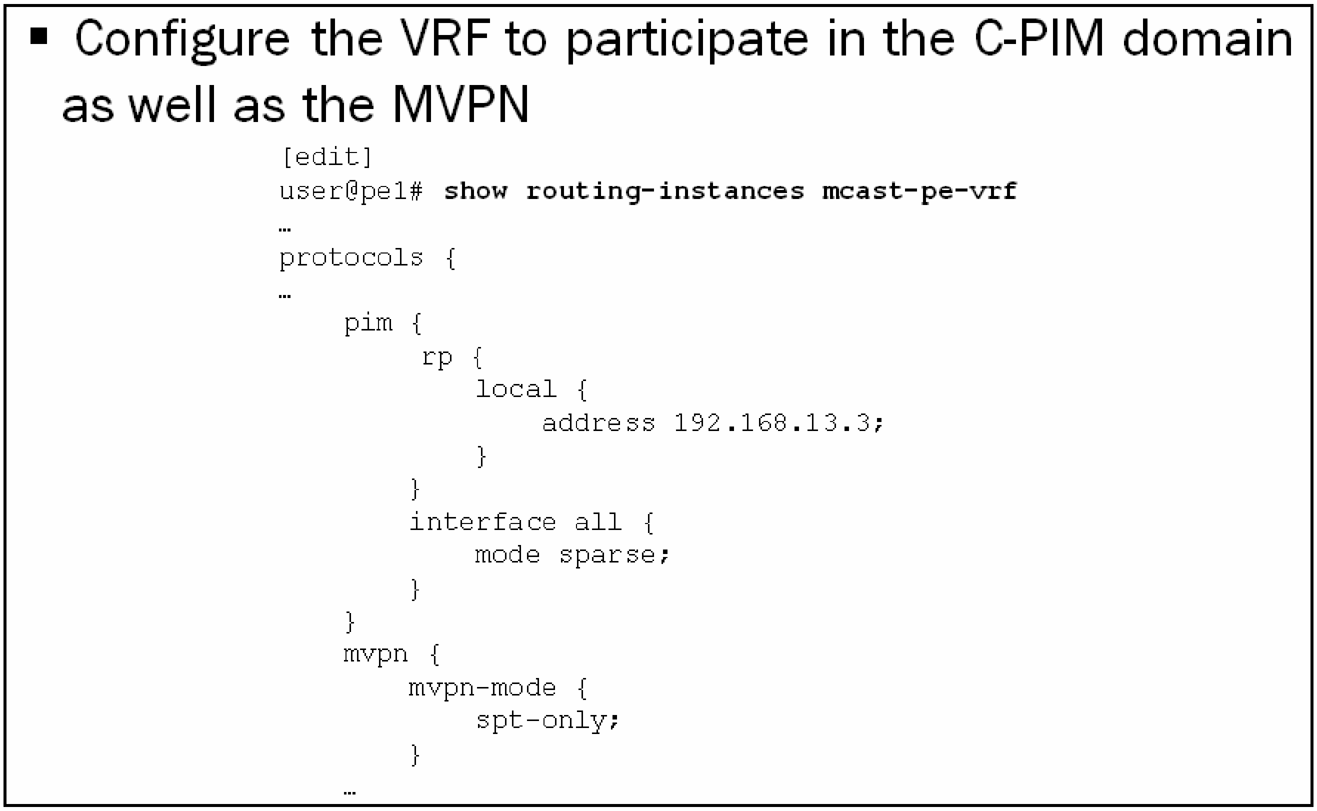
```
[edit routing-instances mcast-pe-vrf]
user@pe1# show
…
provider-tunnel {
    rsvp-te {
        label-switched-path-template {
            mvpn-example;
        }
    }
}
…
vrf-table-label;
…
```

Selective Provider Tunnel

```
[edit routing-instances mcast-pe-vrf]
user@pe1# show
…
provider-tunnel {
    selective {
        group 224.7.7.0/24 {
            wildcard-source {
                rsvp-te {
                    label-switched-pat… {
                        default-template;
…
vrf-table-label;
```

The example in the graphic shows how to configure an RSVP-traffic engineered point-to-multipoint LSP for use as an inclusive provider tunnel and a selective provider tunnel. You can use a configured LSP template or just use the default template. To ensure that penultimate hop popping is not performed along the LSP, the example shows the configuration of `vrf-table-label`. A virtual tunnel interface could also have been used.

Customer PIM Configuration

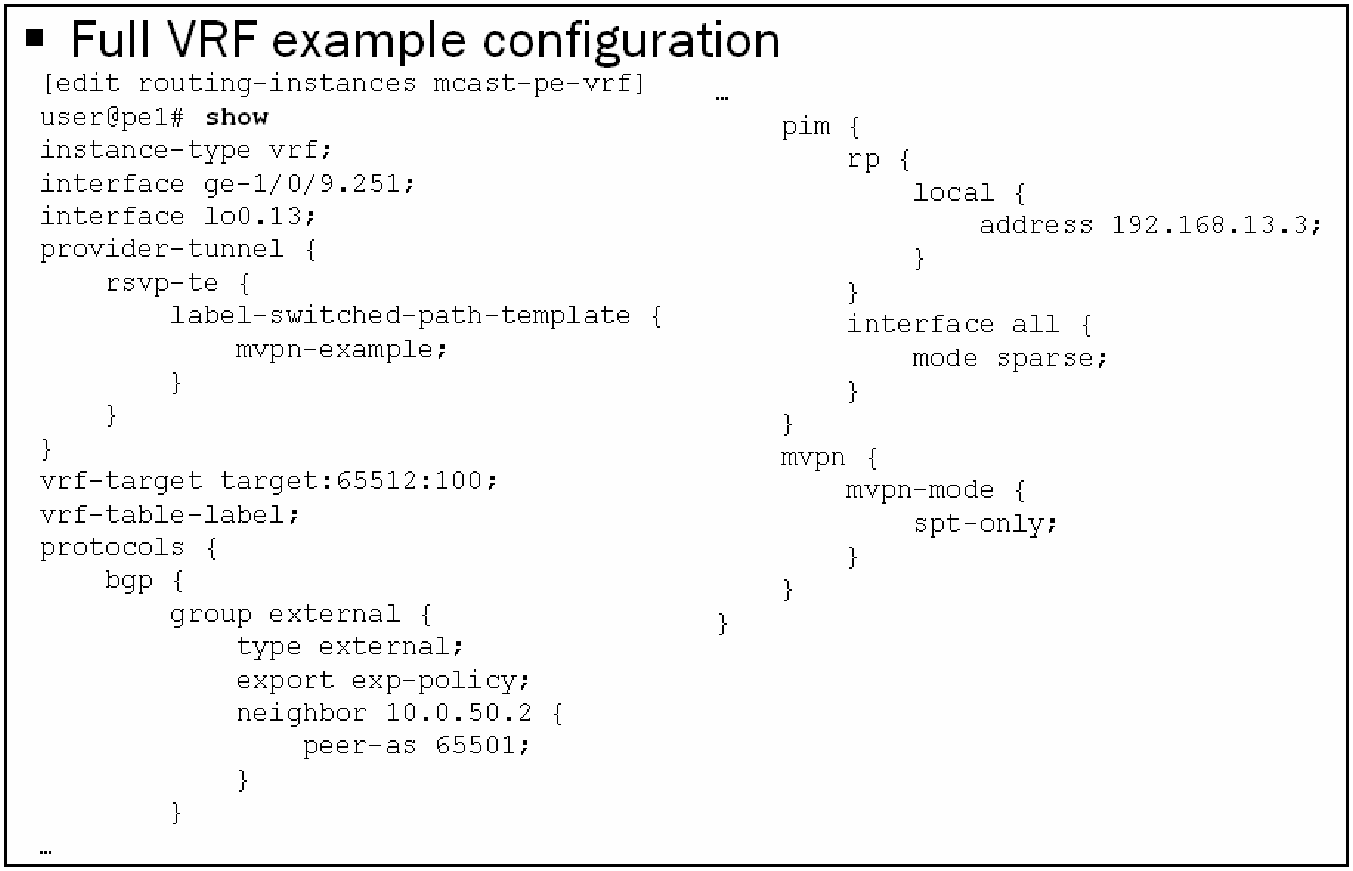


- Configure the VRF to participate in the C-PIM domain as well as the MVPN

```
[edit]
user@pe1# show routing-instances mcast-pe-vrf
…
protocols {
…
    pim {
         rp {
            local {
                address 192.168.13.3;
            }
        }
        interface all {
            mode sparse;
        }
    }
    mvpn {
        mvpn-mode {
            spt-only;
        }
    …
```

Within the VRF, you must configure multicast routing that is specific to the customer's multicast domain. This configuration is shown as the PIM configuration in the graphic. Also, you must enable the mvpn using the `mvpn` statement. There are several settings available under the `mvpn` hierarchy. The graphic shows the configuration of the `mvpn-mode`. There are two options for the mode. First is the `spt-only` mode which allows for only shortest path trees to be built from receiver PEs towards the source (Type 7s only). The second mode is `rpt-spt` mode which allows for both rendezvous point based trees and shortest path trees to be built from receiver PE to source (Type 6s and Type 7s allowed). Subsequent sections will show more options that are available under the `mvpn` hierarchy.

VRF Configuration



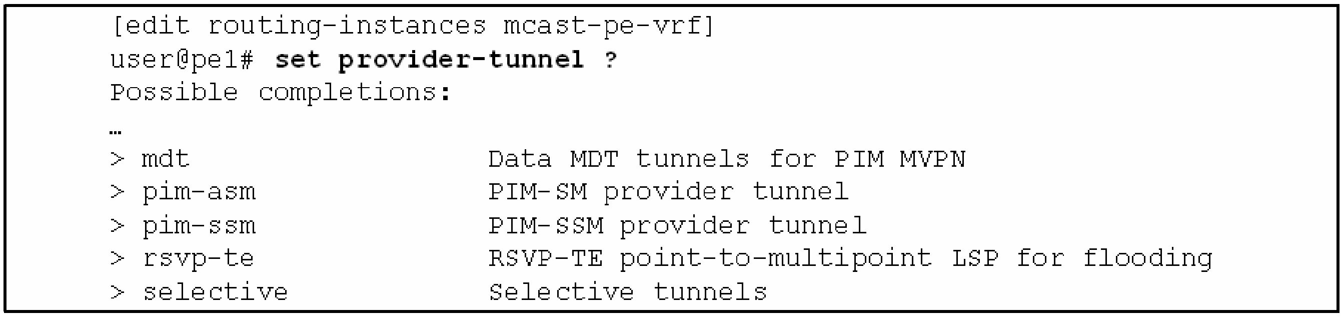
- Full VRF example configuration

```
[edit routing-instances mcast-pe-vrf]
user@pe1# show
instance-type vrf;
interface ge-1/0/9.251;
interface lo0.13;
provider-tunnel {
    rsvp-te {
        label-switched-path-template {
            mvpn-example;
        }
    }
}
vrf-target target:65512:100;
vrf-table-label;
protocols {
    bgp {
        group external {
            type external;
            export exp-policy;
            neighbor 10.0.50.2 {
                peer-as 65501;
            }
        }
…
```

```
…
    pim {
        rp {
            local {
                address 192.168.13.3;
            }
        }
        interface all {
            mode sparse;
        }
    }
    mvpn {
        mvpn-mode {
            spt-only;
        }
    }
}
```

This graphic shows the full, working VRF configuration for PE1.

Provider Tunnels

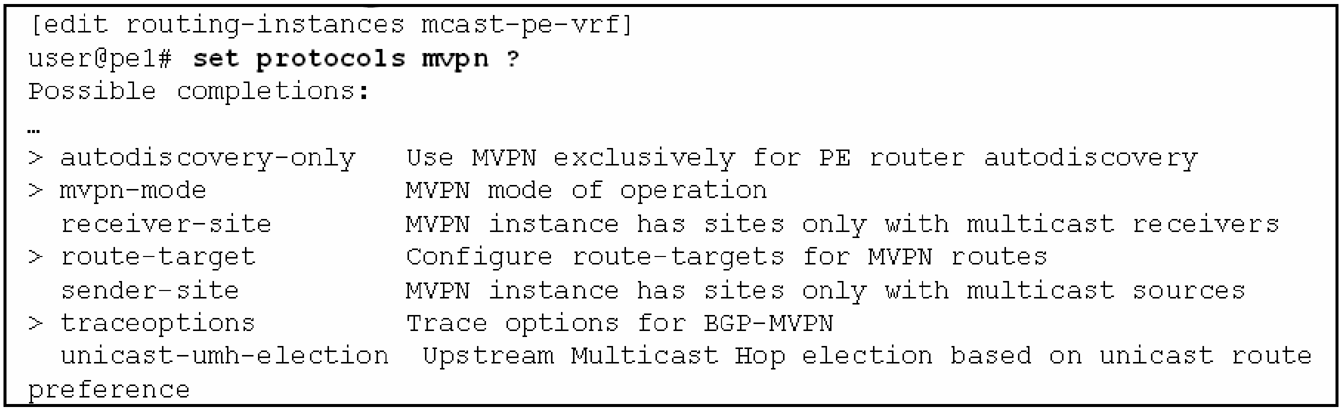


```
[edit routing-instances mcast-pe-vrf]
user@pe1# set provider-tunnel ?
Possible completions:
…
> mdt                     Data MDT tunnels for PIM MVPN
> pim-asm                 PIM-SM provider tunnel
> pim-ssm                 PIM-SSM provider tunnel
> rsvp-te                 RSVP-TE point-to-multipoint LSP for flooding
> selective               Selective tunnels
```

There are several options available for provider tunnels.

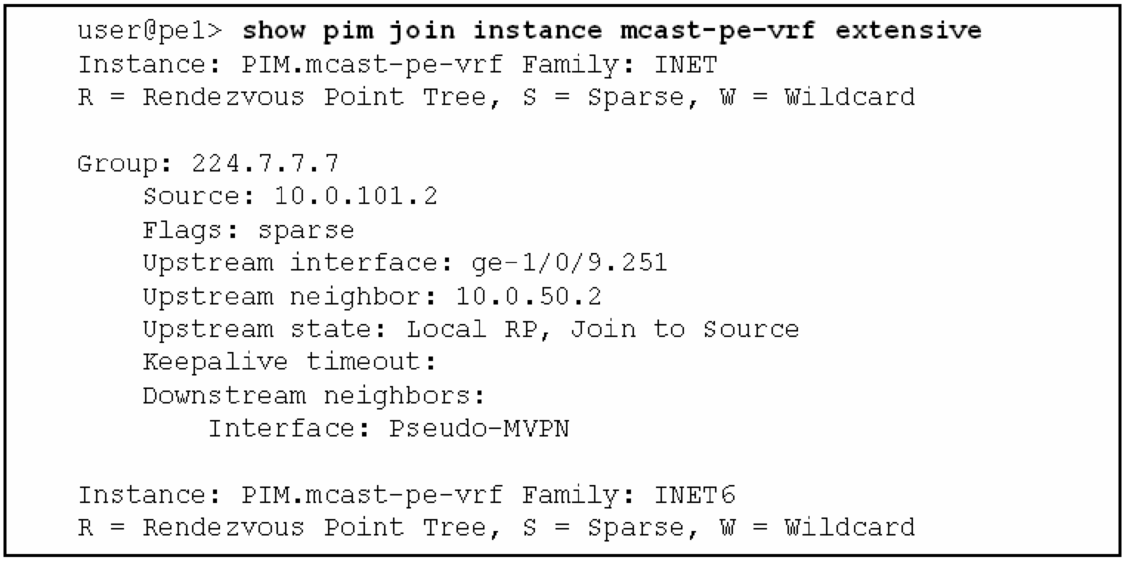
- mdt - Used to configure Multicast Data Tunnels as provider tunnels;
- pim-asm - Used to configure PIM any source provider tunnels;
- pim-ssm - Use to configure PIM source specific provider tunnels;
- rsvp-te - Used to configure an I-PMSI between PEs using RSVP-traffic engineered point-to-multipoint LSPs; and
- selective - Used to configure an S-PMSI between PEs using RSVP-traffic engineered point-to-multipoint LSPs.

MVPN Settings



```
[edit routing-instances mcast-pe-vrf]
user@pe1# set protocols mvpn ?
Possible completions:
…
> autodiscovery-only   Use MVPN exclusively for PE router autodiscovery
> mvpn-mode            MVPN mode of operation
  receiver-site        MVPN instance has sites only with multicast receivers
> route-target         Configure route-targets for MVPN routes
  sender-site          MVPN instance has sites only with multicast sources
> traceoptions         Trace options for BGP-MVPN
  unicast-umh-election  Upstream Multicast Hop election based on unicast route
preference
```

It is under the MVPN settings that you can specify whether a site is a sender-only site or receiver-only site. By default, every site is both and sender and receiver site. You can also configure the MVPN mode, traceoptions, and the upstream multicast hop settings.

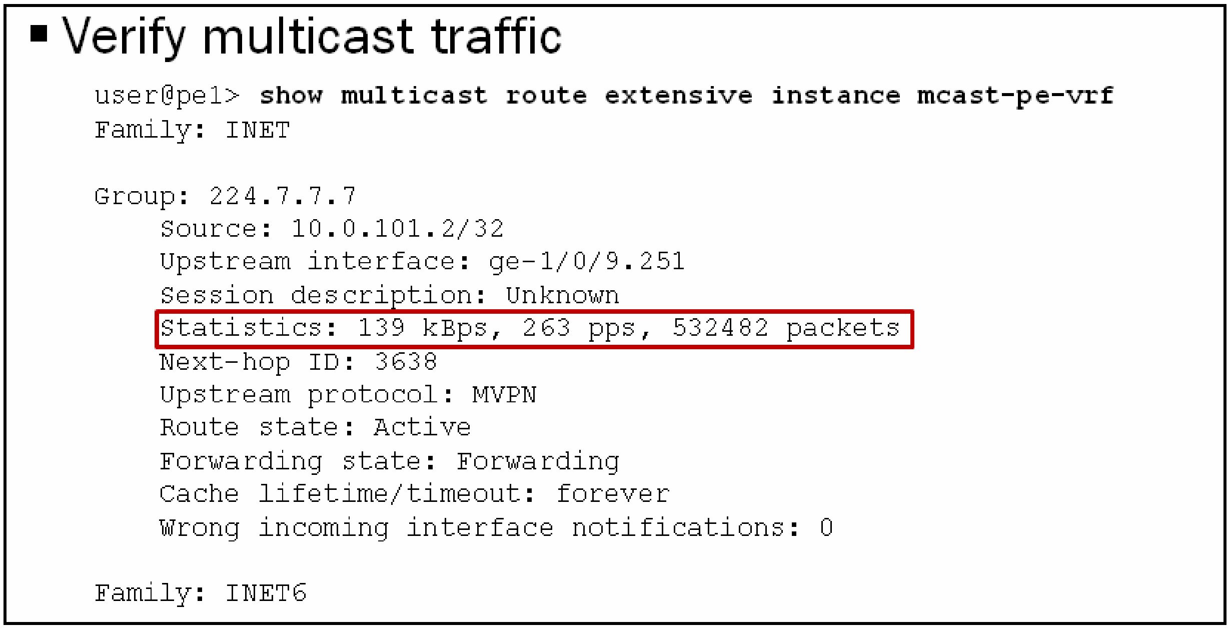


```
user@pe1> show pim join instance mcast-pe-vrf extensive
Instance: PIM.mcast-pe-vrf Family: INET
R = Rendezvous Point Tree, S = Sparse, W = Wildcard

Group: 224.7.7.7
    Source: 10.0.101.2
    Flags: sparse
    Upstream interface: ge-1/0/9.251
    Upstream neighbor: 10.0.50.2
    Upstream state: Local RP, Join to Source
    Keepalive timeout:
    Downstream neighbors:
        Interface: Pseudo-MVPN

Instance: PIM.mcast-pe-vrf Family: INET6
R = Rendezvous Point Tree, S = Sparse, W = Wildcard
```

Verify PIM Status

To verify the status of PIM within the customer network using the `show pim` commands using a modifier of `instance` *`instance-name`*. The command in the graphic shows the (S, G) state of the PE router.

Is Multicast Traffic Flowing?

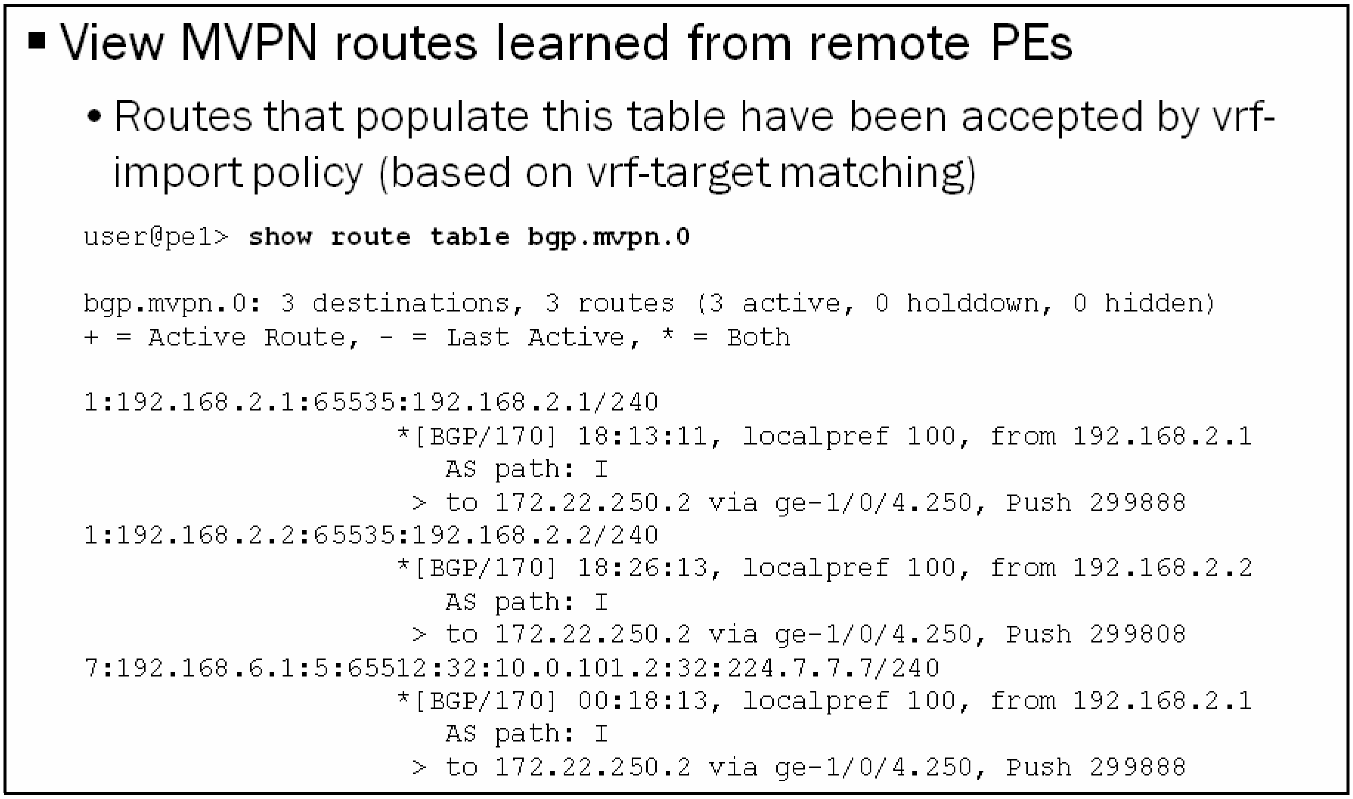


- Verify multicast traffic

```
user@pe1> show multicast route extensive instance mcast-pe-vrf
Family: INET

Group: 224.7.7.7
    Source: 10.0.101.2/32
    Upstream interface: ge-1/0/9.251
    Session description: Unknown
    Statistics: 139 kBps, 263 pps, 532482 packets
    Next-hop ID: 3638
    Upstream protocol: MVPN
    Route state: Active
    Forwarding state: Forwarding
    Cache lifetime/timeout: forever
    Wrong incoming interface notifications: 0

Family: INET6
```

The command in the graphic shows that PE1 is currently forwarding multicast traffic destined for 224.7.7.7 at a rate of 263 packets per second.

Next-Generation MVPN Routing Table-IN

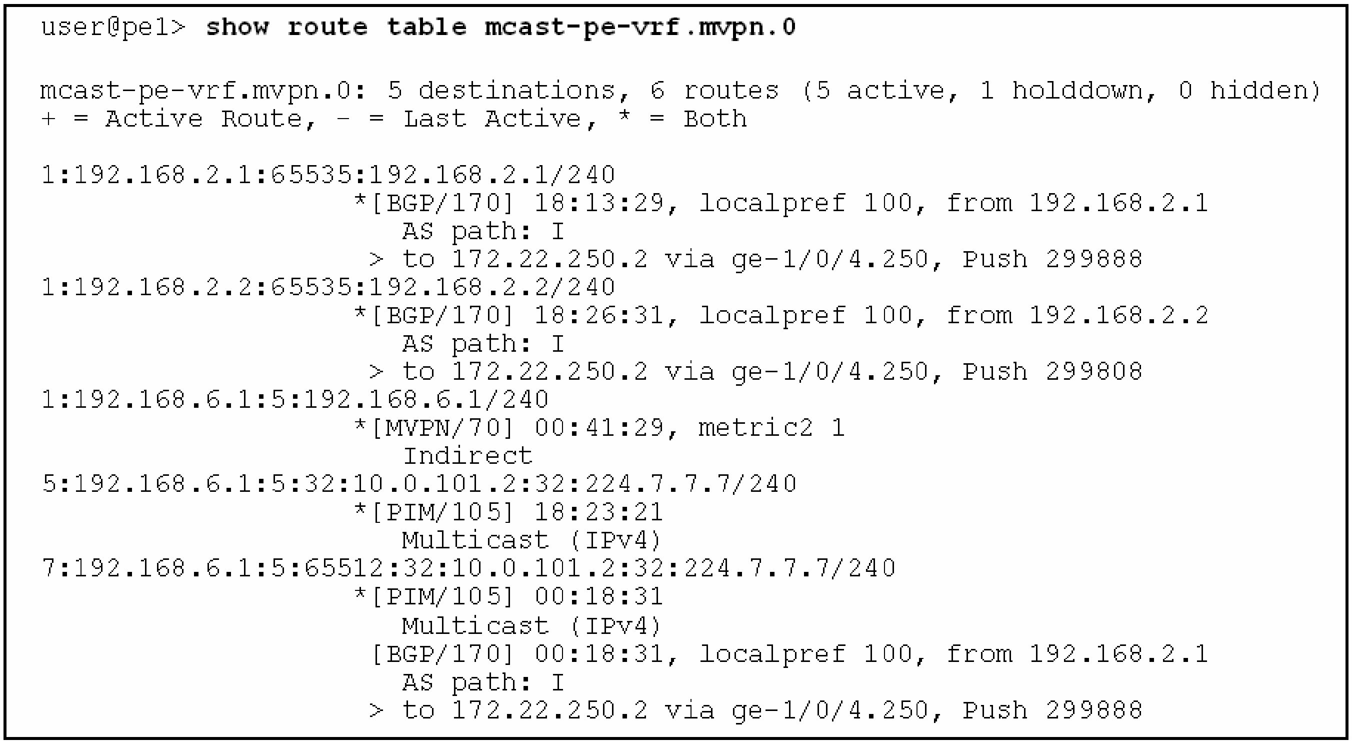


- View MVPN routes learned from remote PEs
  - Routes that populate this table have been accepted by vrf-import policy (based on vrf-target matching)

```
user@pe1> show route table bgp.mvpn.0

bgp.mvpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:192.168.2.1:65535:192.168.2.1/240
                   *[BGP/170] 18:13:11, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 172.22.250.2 via ge-1/0/4.250, Push 299888
1:192.168.2.2:65535:192.168.2.2/240
                   *[BGP/170] 18:26:13, localpref 100, from 192.168.2.2
                      AS path: I
                    > to 172.22.250.2 via ge-1/0/4.250, Push 299808
7:192.168.6.1:5:65512:32:10.0.101.2:32:224.7.7.7/240
                   *[BGP/170] 00:18:13, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 172.22.250.2 via ge-1/0/4.250, Push 299888
```

The `bgp.mvpn.0` table is the routing table-IN for MVPN routes. The command on the graphic shows the routes that are currently populating the `bgp.mvpn.0` table. Routes will only show up in this table if they have been accepted by VRF import policy that matches on the correct target communities.

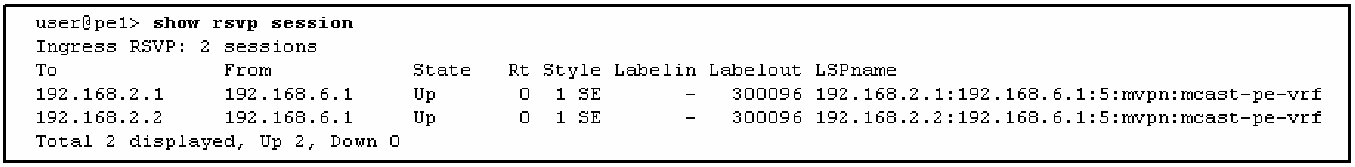
VRF Specific MVPN Routes



```
user@pe1> show route table mcast-pe-vrf.mvpn.0

mcast-pe-vrf.mvpn.0: 5 destinations, 6 routes (5 active, 1 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1:192.168.2.1:65535:192.168.2.1/240
                   *[BGP/170] 18:13:29, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 172.22.250.2 via ge-1/0/4.250, Push 299888
1:192.168.2.2:65535:192.168.2.2/240
                   *[BGP/170] 18:26:31, localpref 100, from 192.168.2.2
                      AS path: I
                    > to 172.22.250.2 via ge-1/0/4.250, Push 299808
1:192.168.6.1:5:192.168.6.1/240
                   *[MVPN/70] 00:41:29, metric2 1
                      Indirect
5:192.168.6.1:5:32:10.0.101.2:32:224.7.7.7/240
                   *[PIM/105] 18:23:21
                      Multicast (IPv4)
7:192.168.6.1:5:65512:32:10.0.101.2:32:224.7.7.7/240
                   *[PIM/105] 00:18:31
                      Multicast (IPv4)
                    [BGP/170] 00:18:31, localpref 100, from 192.168.2.1
                      AS path: I
                    > to 172.22.250.2 via ge-1/0/4.250, Push 299888
```

The command in the graphic shows the MVPN routes that relate to a specific MVPN.

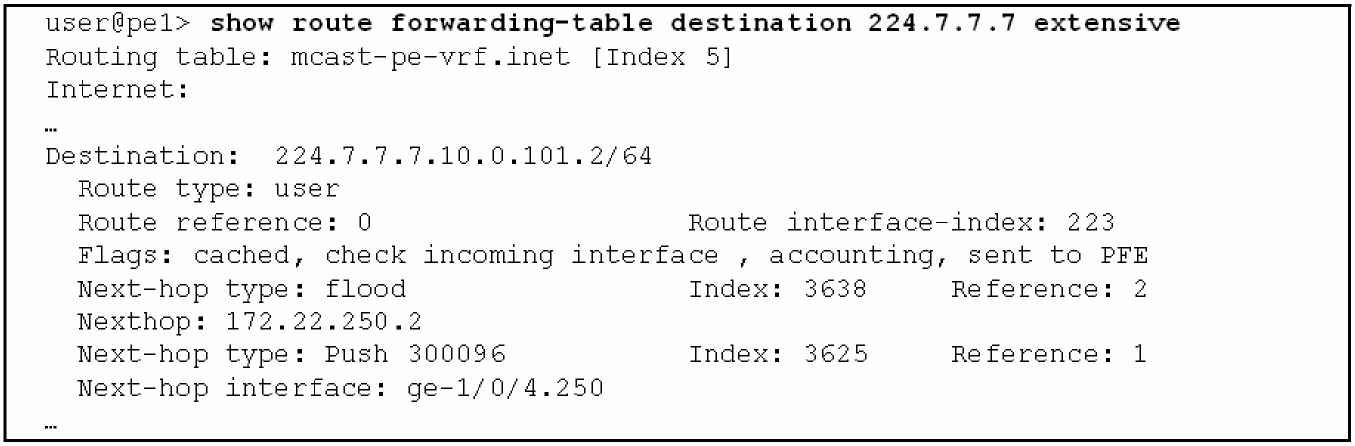
Point-to-Multipoint LSP



```
user@pe1> show rsvp session
Ingress RSVP: 2 sessions
To              From            State   Rt Style Labelin Labelout LSPname
192.168.2.1     192.168.6.1     Up       0  1 SE       -   300096 192.168.2.1:192.168.6.1:5:mvpn:mcast-pe-vrf
192.168.2.2     192.168.6.1     Up       0  1 SE       -   300096 192.168.2.2:192.168.6.1:5:mvpn:mcast-pe-vrf
Total 2 displayed, Up 2, Down 0
```

Use the `show rsvp session` command to determine the status of the point-to-multipoint LSP. In the output, you can see that the outbound label for the point-to-multipoint LSP is 300096.

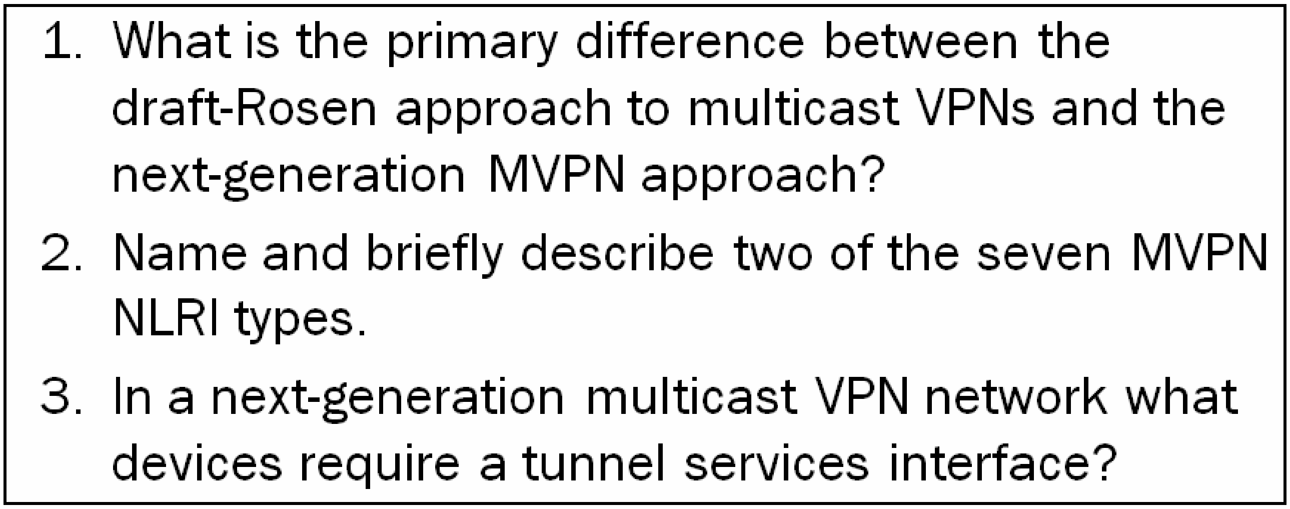
Forwarding Table



```
user@pe1> show route forwarding-table destination 224.7.7.7 extensive
Routing table: mcast-pe-vrf.inet [Index 5]
Internet:
…
Destination:  224.7.7.7.10.0.101.2/64
  Route type: user
  Route reference: 0                   Route interface-index: 223
  Flags: cached, check incoming interface , accounting, sent to PFE
  Next-hop type: flood                 Index: 3638     Reference: 2
  Nexthop: 172.22.250.2
  Next-hop type: Push 300096           Index: 3625     Reference: 1
  Next-hop interface: ge-1/0/4.250
…
```

The command in the graphic shows the routes in PE1's forwarding table that are associated with the multicast group of 224.7.7.7 with a source of 10.0.101.2. Notice that all multicast packets of this type will be sent out of the ge-1/0/4.250 interface with a single MPLS label of 300096.

Review Questions



1. What is the primary difference between the draft-Rosen approach to multicast VPNs and the next-generation MVPN approach?
2. Name and briefly describe two of the seven MVPN NLRI types.
3. In a next-generation multicast VPN network what devices require a tunnel services interface?

Answers to Review Questions

1.

The draft-Rosen required that the provider network be running PIM for signaling. The next-generation approach uses BGP to signal the providers network and does not require PIM be configured in the core.

2.

Type 1: Intra-AS Inclusive MVPN Membership Discovery

Type 2: Inter-AS Inclusive MVPN Membership Discovery

Type 3: Selective MVPN Autodiscovery Route

Type 4: Selective MVPN Autodiscovery Route for Leaf

Type 5: Source Active Autodiscovery Route

Type 6: Shared Tree Join Route

Type 7: Source Tree Join Route

3.

The first hop designated router, the candidate rendezvous points, and all PE routers participating in the multicast network, unless using vrf-table-label option, require the use of a tunnel services interface.

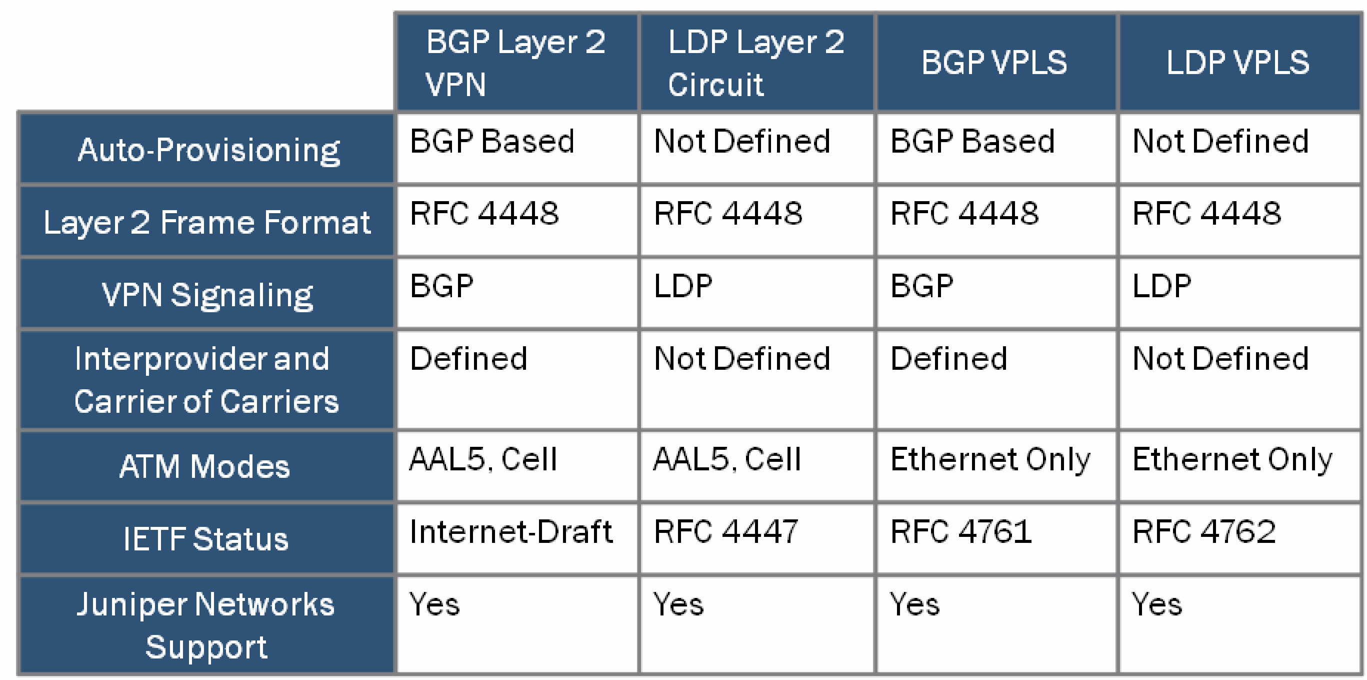
JNCIS-SP Study Guide—Part 3

Chapter 13: BGP Layer 2 VPNs

This Chapter Discusses:

- The purpose and features of a Layer 2 virtual private network (VPN);
- The roles of a customer edge (CE) device, provider edge (PE) router, and provider (P) routers in a Layer 2 VPN;
- The flow of control traffic and data traffic for a BGP Layer 2 VPN;
- Configuring a BGP Layer 2 VPN and describing the benefits and requirements of over-provisioning; and
- Monitoring and troubleshooting a BGP Layer 2 VPN.

BGP and LDP VPN Characteristics



	BGP Layer 2 VPN	LDP Layer 2 Circuit	BGP VPLS	LDP VPLS
Auto-Provisioning	BGP Based	Not Defined	BGP Based	Not Defined
Layer 2 Frame Format	RFC 4448	RFC 4448	RFC 4448	RFC 4448
VPN Signaling	BGP	LDP	BGP	LDP
Interprovider and Carrier of Carriers	Defined	Not Defined	Defined	Not Defined
ATM Modes	AAL5, Cell	AAL5, Cell	Ethernet Only	Ethernet Only
IETF Status	Internet-Draft	RFC 4447	RFC 4761	RFC 4762
Juniper Networks Support	Yes	Yes	Yes	Yes

The BGP Layer 2 VPN (Kompella) draft describes an algorithm used to auto-provision PE routers when a new site is added to a PE router. This algorithm automatically assigns new circuit IDs and labels, notifies other PE routers, and sets up the VPN mesh automatically in all topologies. The BGP Layer 2 VPN has extended this same algorithm of auto-provisioning into the BGP virtual private LAN service (VPLS) RFC. LDP Layer 2 circuits (Martini) require manual provisioning in a manner similar to a traditionally managed Frame Relay network. Both BGP Layer 2 VPNs and LDP Layer 2 circuits call for the use of the Martini-style encapsulation. In most cases, it is not necessary to transport the Layer 2 encapsulation across the network; rather, the Layer 2 header can be stripped at R1, and reproduced at R2. This is done using the information carried in the control word, which is optional, but it is required for Asynchronous Transfer Mode (ATM) and Frame Relay.

In both BGP Layer 2 VPNs and LDP Layer 2 circuits, VPN information must be communicated between PE routers. BGP Layer 2 VPN (Kompella) uses Multiprotocol Border Gateway Protocol (MP-BGP) for this purpose, and LDP Layer 2 circuit (Martini) uses LDP. Because BGP Layer 2 VPN uses MP-BGP, it leverages the interprovider and carrier-of-carriers mechanisms defined in RFC 4364. Both methods (not including VPLS) support the use of ATM AAL5 and ATM cell mode.

The BGP Layer 2 VPN (Kompella) draft remains in draft status; the Internet Engineering Task Force (IETF) would like additional supporting documents prior to its approval as a standard. LDP Layer 2 circuit is defined under RFC 4447.

Most vendors in the Layer 2 VPN space have announced support for LDP Layer 2 circuits. The Junos operating system supports BGP Layer 2 VPNs, LDP Layer 2 circuits, BGP VPLS, and LDP VPLS. The BGP Layer 2 VPN drafts are a second-generation Layer 2 VPN technology that builds on the experience Juniper Networks gained through our own first-generation Layer 2 VPN product: circuit cross-connect (CCC). CCC was designed around early Internet service provider (ISP) customer requests for Layer 2 VPNs and is similar to LDP Layer 2 circuits in many respects.

Customer Sees Standard Layer 2 Circuits

From the user's perspective, there is no obvious difference between a Frame Relay circuit carried over an ATM core versus one transported over an IP core. In either case, the provider's edge equipment delivers one or more Layer 2 circuit identifiers that are used to map traffic to each of the remote sites with which the CE router communicates.

Circuit Identifiers Mapped into LSPs

The provider edge device maps Layer 2 frames received from the CE router into MPLS label-switched paths (LSPs). This mapping can be either one to one or many to one when label stacking is supported. The use of MPLS in the core allows core routers to switch the frame towards its egress point without knowing—or caring—what upper-layer protocols are encapsulated within the labeled packets. The result is that nonroutable and proprietary protocols can now be transported over an IP core.

Customer Manages Its Own Routing

As with a conventional Frame Relay or private line solution, the job of configuring and maintaining the routing between customer sites is the job of the Layer 2 VPN user. Thus, the provider routers are in no way involved with the routing protocols used by the customer. The PE routers no longer carry any customer routes and this can be a big scaling advantage.

Decouple Edge from Core

Service providers want to decouple edge-facing technology from the technology that makes up the core. Such decoupling allows for rapid deployments of new and enhanced services without the requirement of upgrading or modifying edge technology. A core network based on IP and MPLS readily accommodates this separation of edge and core technologies.

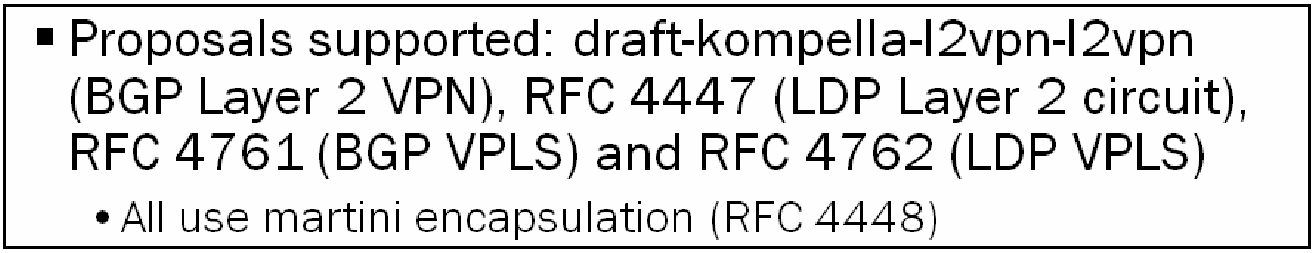
IP-Based Convergence

An IP-based core with MPLS allows the provisioning of a multitude of services—all of which are supported by a common core technology.

Simplified Provisioning

The provisioning of new services is simplified when a common core technology supports all services and when the edge technologies are decoupled from that of the core. For example, consider a service provider currently selling only Internet access. With a multiservice IP backbone, this provider can begin selling Layer 3 VPN, Layer 2 VPN, and valued-added IP services with no changes required in either the core or the user's access technology. Converting an existing customer's Frame Relay link into a multiservice access solution requires only simple software changes. Now some data-link connection identifiers (DLCIs) can terminate on provider routers for Internet access, while other DLCIs are switched across the core to support transparent Frame Relay connectivity between the customer's sites.

Layer 2 VPN Proposals

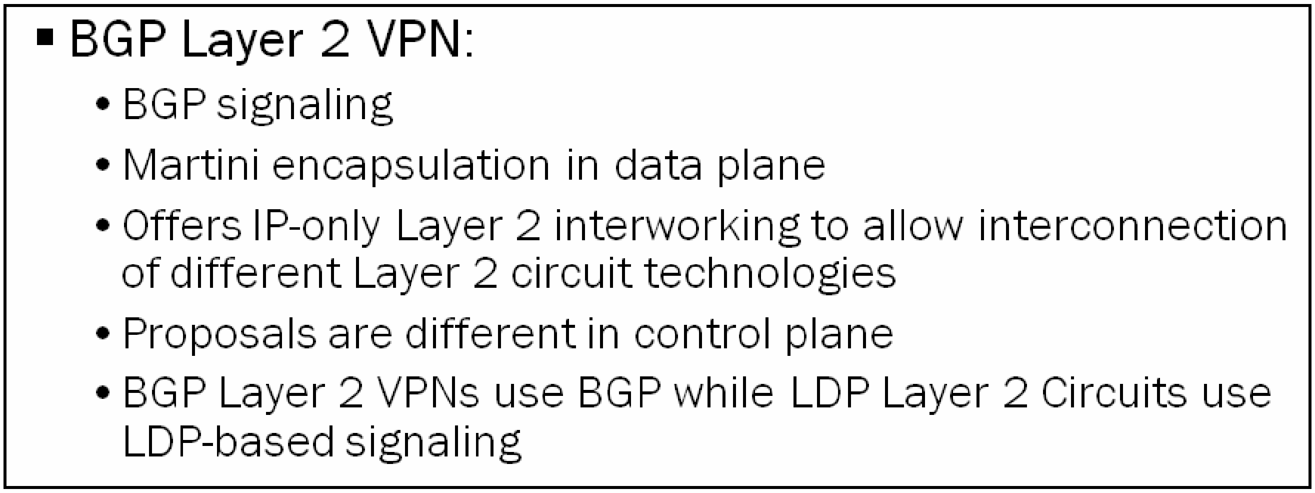


- Proposals supported: draft-kompella-l2vpn-l2vpn (BGP Layer 2 VPN), RFC 4447 (LDP Layer 2 circuit), RFC 4761 (BGP VPLS) and RFC 4762 (LDP VPLS)
  - All use martini encapsulation (RFC 4448)

The Junos OS supports three proposals that specify provider-provisioned Layer 2 VPN solutions. The latest versions of the BGP Layer 2 VPN drafts use the encapsulation approach defined in RFC 4448 while providing the BGP signaling and auto-provisioning benefits of the BGP Layer 2 VPN draft.

The Junos OS offers support of the BGP Layer 2 VPN draft, including support for the IP-only interworking function, which allows the interconnection of dissimilar Layer 2 technologies (such as Frame Relay to ATM). The implementation of draft-BGP Layer 2 VPNs supports the Martini control word, which is used to convey Layer 2 bit indications in the forwarding plane. Currently, only Frame Relay forward explicit congestion notification (FECN), backward explicit congestion notification (BECN), and discard eligible (DE) bits can be signaled using the Martini control word. The Junos OS also offers support for Layer 2 VPNs based on the LDP Layer 2 circuit RFC 4447using LDP-based signaling.

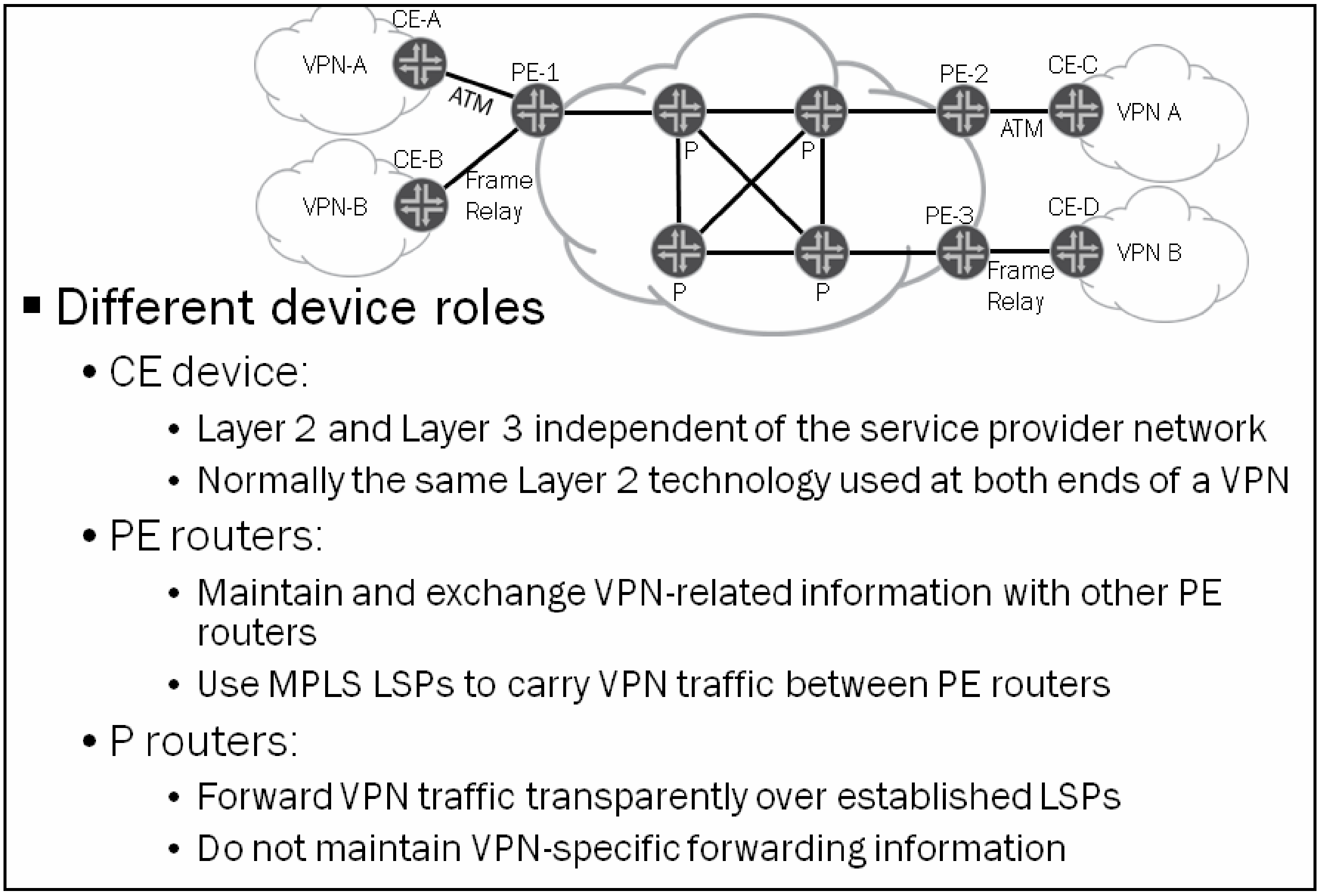
Control Plane Differences



- BGP Layer 2 VPN:
  - BGP signaling
  - Martini encapsulation in data plane
  - Offers IP-only Layer 2 interworking to allow interconnection of different Layer 2 circuit technologies
  - Proposals are different in control plane
  - BGP Layer 2 VPNs use BGP while LDP Layer 2 Circuits use LDP-based signaling

Because the BGP Layer 2 VPN drafts now use Martini encapsulation, the principal differences relate to their signaling approaches. The BGP Layer 2 VPN drafts use only BGP, while the LDP Layer 2 circuit RFC specifies only LDP-based signaling. Another key difference is that the BGP Layer 2 VPN drafts support auto-provisioning of Layer 2 VPNs and BGP VPLS. This capability can simplify significantly the provider's operations when adds and moves occur in a Layer 2 VPN.

Customer Edge Devices



- Different device roles
  - CE device:
    - Layer 2 and Layer 3 independent of the service provider network
    - Normally the same Layer 2 technology used at both ends of a VPN
  - PE routers:
    - Maintain and exchange VPN-related information with other PE routers
    - Use MPLS LSPs to carry VPN traffic between PE routers
  - P routers:
    - Forward VPN traffic transparently over established LSPs
    - Do not maintain VPN-specific forwarding information

The CE device is normally a router or Layer 2 switch that provides access to the provider's edge device. Because the Layer 2 frames generated by the customer are carried across the core using MPLS, there is inherent independence between the Layer 2 technology used at the provider's edge and the technologies used in the core. This independence extends to the upper protocol layers as well, as the provider does not interpret in any way the contents of the Layer 2 frames.

By default, both ends of a Layer 2 VPN must use the same Layer 2 technology unless IP interworking, as outlined in the BGP Layer 2 VPN draft, is configured. While all sites in a given VPN must deploy the same access technology (when IP interworking is not supported), sites belonging to different VPNs have no such restrictions. This fact is shown on the graphic, where VPN A uses ATM technology, while VPN B uses Frame Relay. If VPNs A and B were combined into a single, larger VPN, you could deploy the BGP Layer 2 VPN IP interworking function to provide interworking at the IP layer without having to adjust the Layer 2 technology used at existing customer sites. Note that IP interworking restricts the Layer 2 VPN to the support of the IP protocol only.

As with a conventional Layer 2 service, each remote site must be associated with a unique Layer 2 circuit identifier used to map traffic to a given site. A CE device with full-mesh connectivity to three remote sites therefore requires that at least three Layer 2 circuit identifiers be provisioned on the PE-CE link.

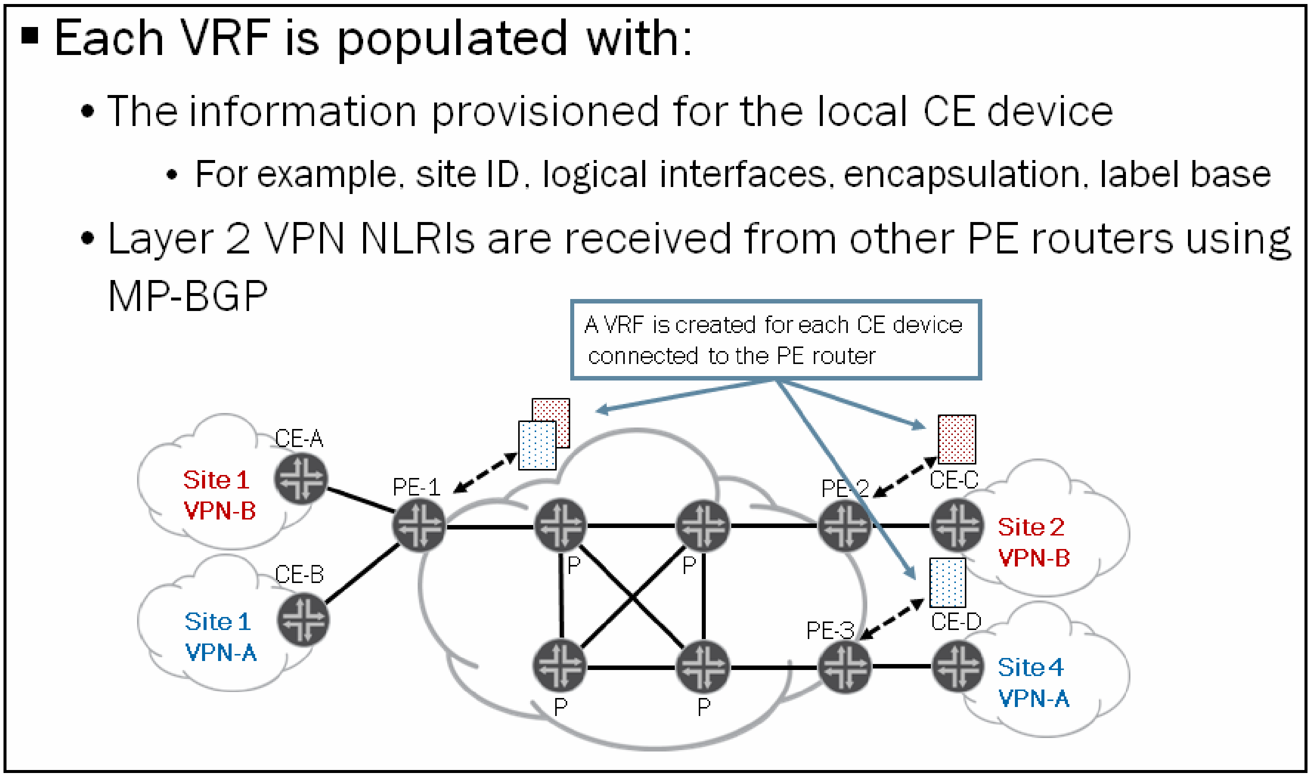
Provider Edge Routers

The PE routers connect to customer sites and maintain Layer 2 VPN-specific information. This VPN information is obtained through local configuration and through signaling exchanges with either BGP or LDP. As with a Layer 3 VPN, the PE routers forward traffic across the provider's core using MPLS LSPs.

Provider Routers

The P routers do not carry any Layer 2 VPN state. They simply provide label-switching router (LSR) services to facilitate the transfer of labeled packets between PE routers.

VPN Forwarding Tables



- Each VRF is populated with:
  - The information provisioned for the local CE device
    - For example, site ID, logical interfaces, encapsulation, label base
  - Layer 2 VPN NLRIs are received from other PE routers using MP-BGP

A VRF is created for each CE device connected to the PE router

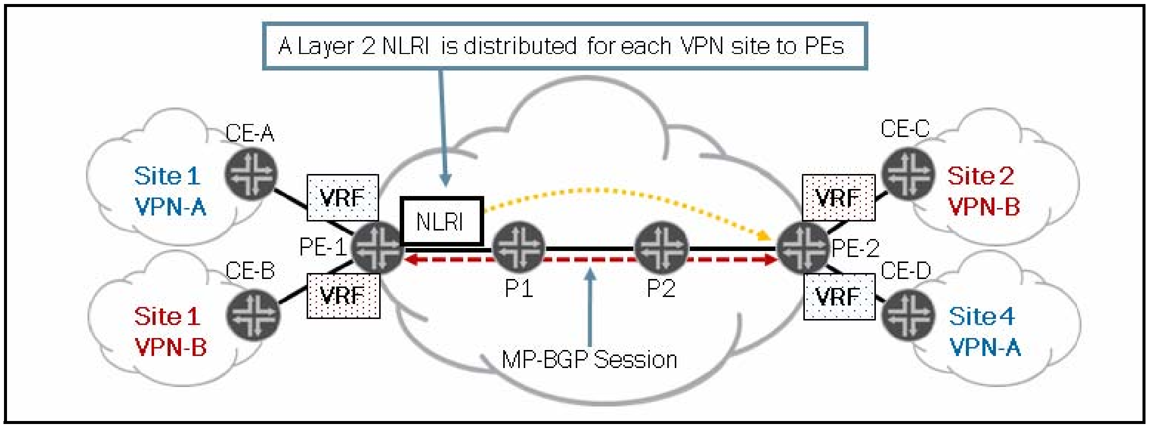
The VRF table is populated with information provisioned for the local CE device and contains:

- The local site ID;
- The site's Layer 2 encapsulation;
- The logical interfaces provisioned to the local CE device; and
- A label base used to associated received traffic with one of the logical interfaces.

The VRF is also populated with information received from other PE routers in MP-BGP updates. These updates contain the remote site's ID, label base, and Layer 2 encapsulation.

The combination of locally provisioned information and Layer 2 VPN network layer reachability information (NLRI) received from remote PE routers results in a Layer 2 VPN forwarding table used to map traffic to and from the LSPs connecting the PE routers.

VPN Connection Tables

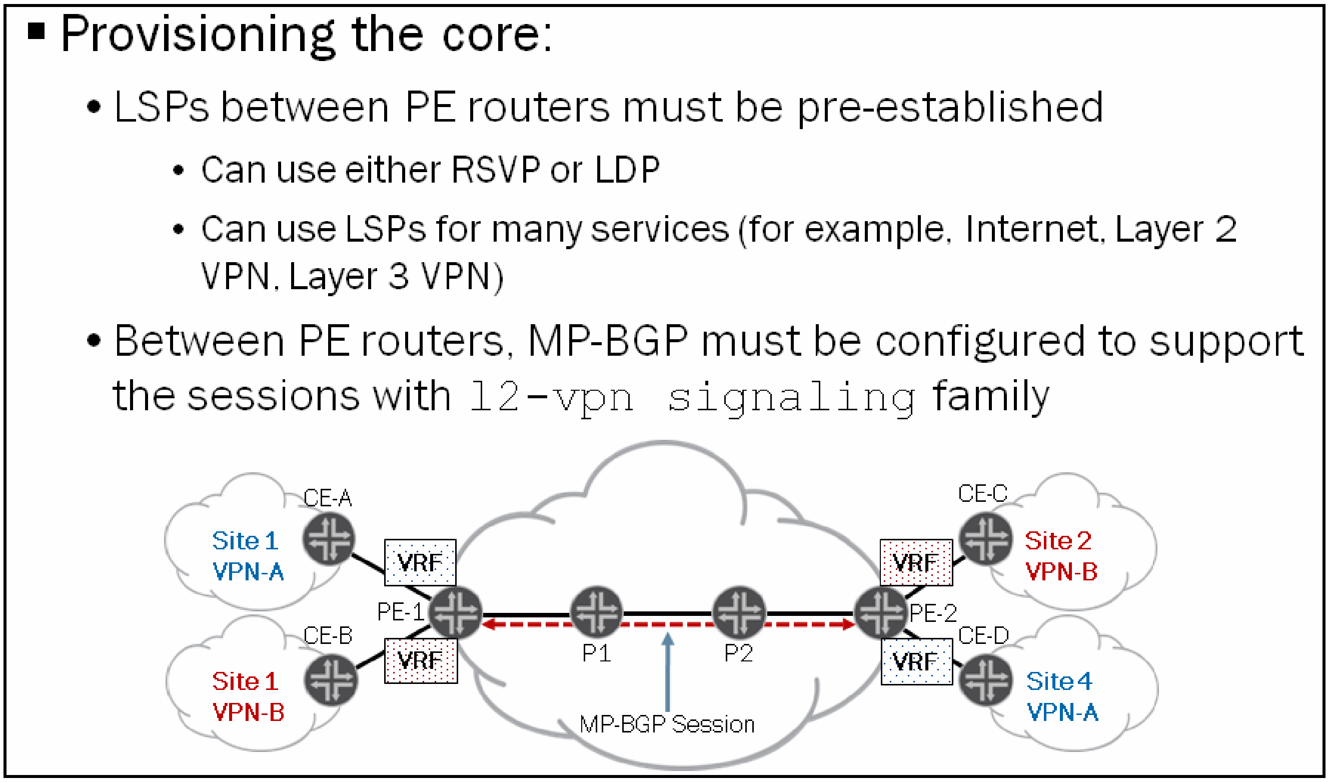


A Layer 2 NLRI is distributed for each VPN site to PEs

The Layer 2 VPN NLRI is a subset of the information held in the PE router's VRF. As a result, one Layer 2 VPN NLRI is associated with each site connected to the PE router.

Layer 2 VPN NLRI Conveys Information Using MP-BGP

The Layer 2 VPN NLRI conveys the local site ID and label blocks to remote PE routers using MP-BGP.



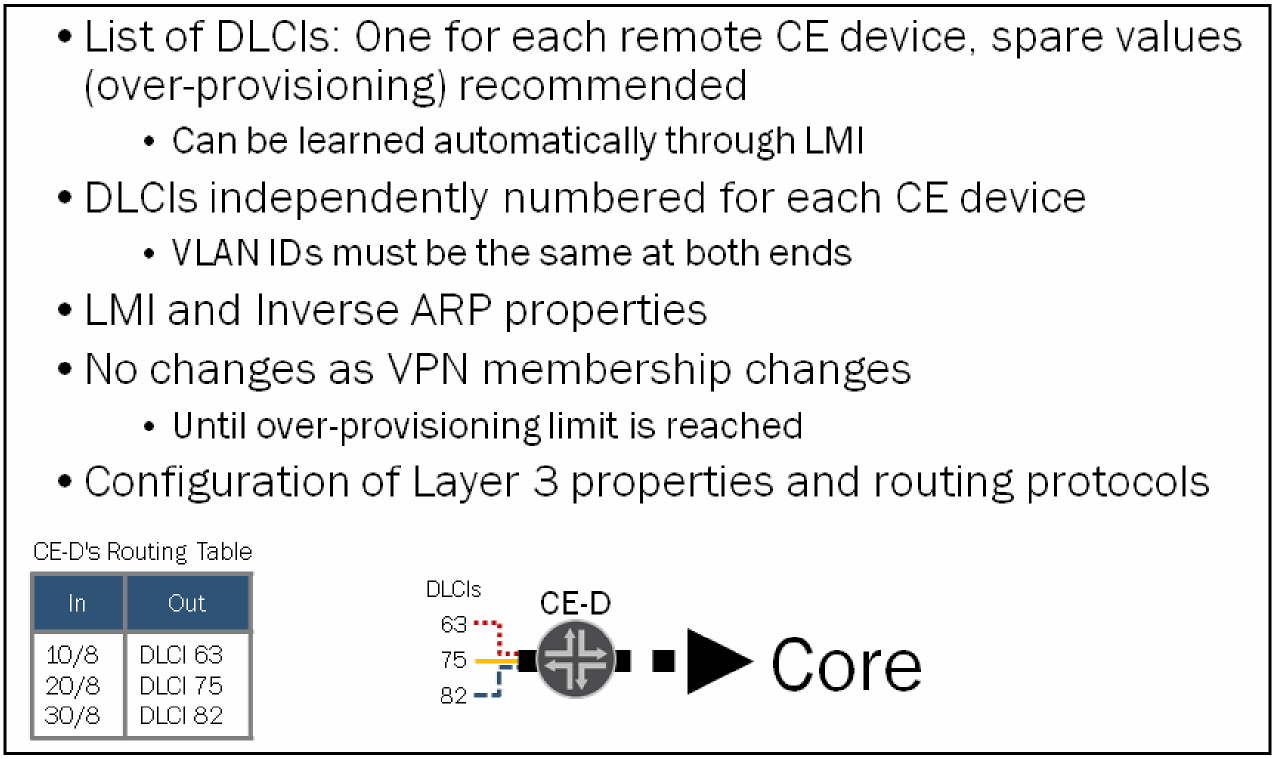
- Provisioning the core:
  - LSPs between PE routers must be pre-established
    - Can use either RSVP or LDP
    - Can use LSPs for many services (for example, Internet, Layer 2 VPN, Layer 3 VPN)
  - Between PE routers, MP-BGP must be configured to support the sessions with `l2-vpn signaling` family

Provisioning the Core

As with a Layer 3 VPN, the provider's core must be provisioned to support the Layer 2 VPN service. Besides a functional interior gateway protocol (IGP), this support normally involves the establishment of MPLS LSPs between PE routers to be used for data forwarding. The PE-PE LSPs are not dedicated to any particular service. With label stacking, the same LSP can be used to support multiple Layer 2 VPN customers while also supporting Layer 3 VPNs and non-VPN traffic.

Each PE router must also be configured with MP-BGP to peer to other PE routers having local sites belonging to the same VPN. These MP-BGP sessions must be configured to support the `l2-vpn signaling` address family so that they can send and receive Layer 2 NLRI updates.

Provisioning the Local CE Device



- List of DLCIs: One for each remote CE device, spare values (over-provisioning) recommended
  - Can be learned automatically through LMI
- DLCIs independently numbered for each CE device
  - VLAN IDs must be the same at both ends
- LMI and Inverse ARP properties
- No changes as VPN membership changes
  - Until over-provisioning limit is reached
- Configuration of Layer 3 properties and routing protocols

CE-D's Routing Table

In	Out
10/8	DLCI 63
20/8	DLCI 75
30/8	DLCI 82

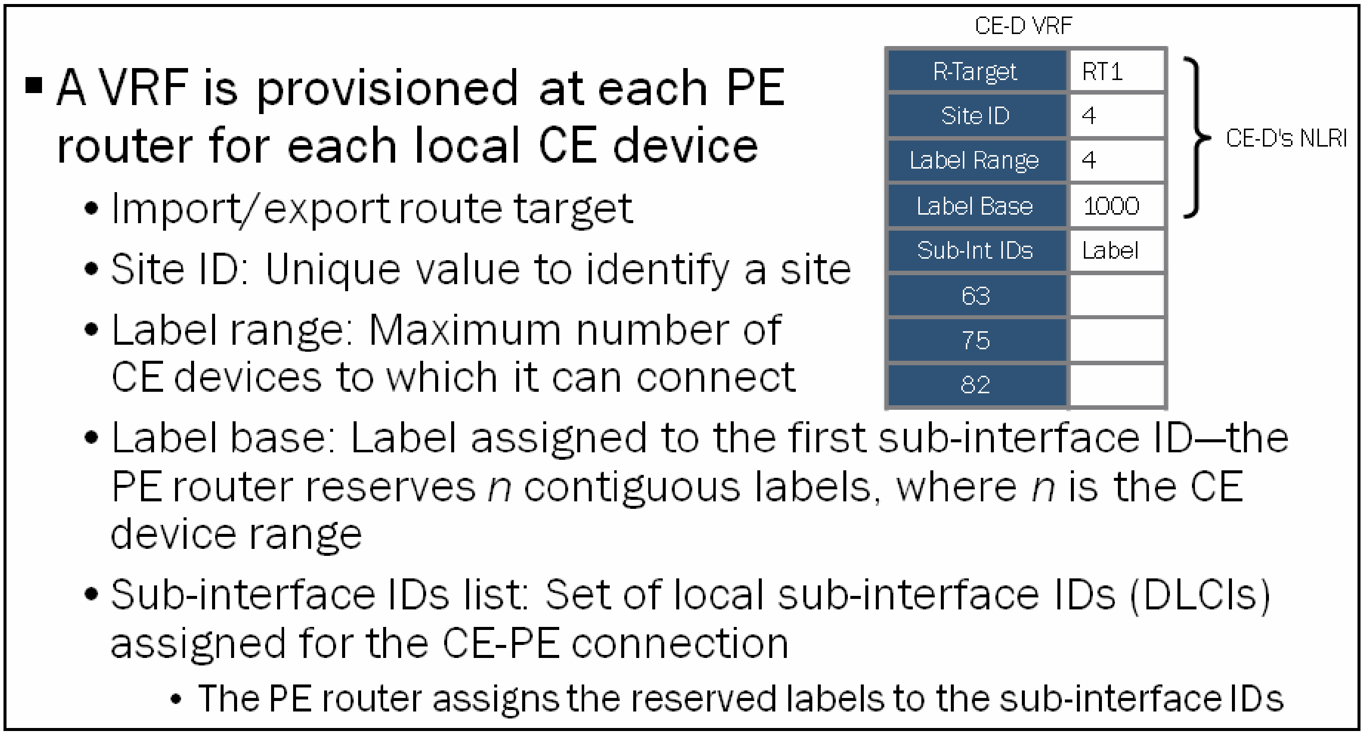
The first step in building a Layer 2 VPN is the configuration of the local CE device. This configuration normally entails assigning a range of Layer 2 circuit identifiers to logical interfaces on the CE device and having the correct encapsulation settings for the Layer 2 protocol being configured.

For Frame Relay, normally, you must configure the permanent virtual circuit (PVC) management protocol and Inverse Address Resolution Protocol (ARP) properties as well as a series of DLCI values, when the CE device cannot learn them automatically through the PVC management protocol. The Junos OS requires that virtual LAN (VLAN) IDs be the same at both ends of a ethernet Layer 2 connection. However, ATM virtual channel identifiers (VCIs) and Frame Relay DLCIs can be the same or can be assigned independently.

The BGP Layer 2 VPN draft allows for the expansion of VPN membership without reconfiguring existing sites when the Layer 2 connection identifiers are over-provisioned.

The CE device also requires the configuration of upper-layer protocols to be compatible with the remote CE router. Unlike a Layer 3 VPN solution, the PE router has no IP or routing protocol configuration, as these functions are configured on the CE routers with end-to-end significance. With Layer 2 VPNs, the CE routers form adjacencies with each other, as opposed to becoming adjacent to the local PE router.

Provisioning the PE Router



- A VRF is provisioned at each PE router for each local CE device
  - Import/export route target
  - Site ID: Unique value to identify a site
  - Label range: Maximum number of CE devices to which it can connect
  - Label base: Label assigned to the first sub-interface ID—the PE router reserves *n* contiguous labels, where *n* is the CE device range
  - Sub-interface IDs list: Set of local sub-interface IDs (DLCIs) assigned for the CE-PE connection
    - The PE router assigns the reserved labels to the sub-interface IDs

CE-D VRF

R-Target	RT1
Site ID	4
Label Range	4
Label Base	1000
Sub-Int IDs	Label
63	
75	
82	

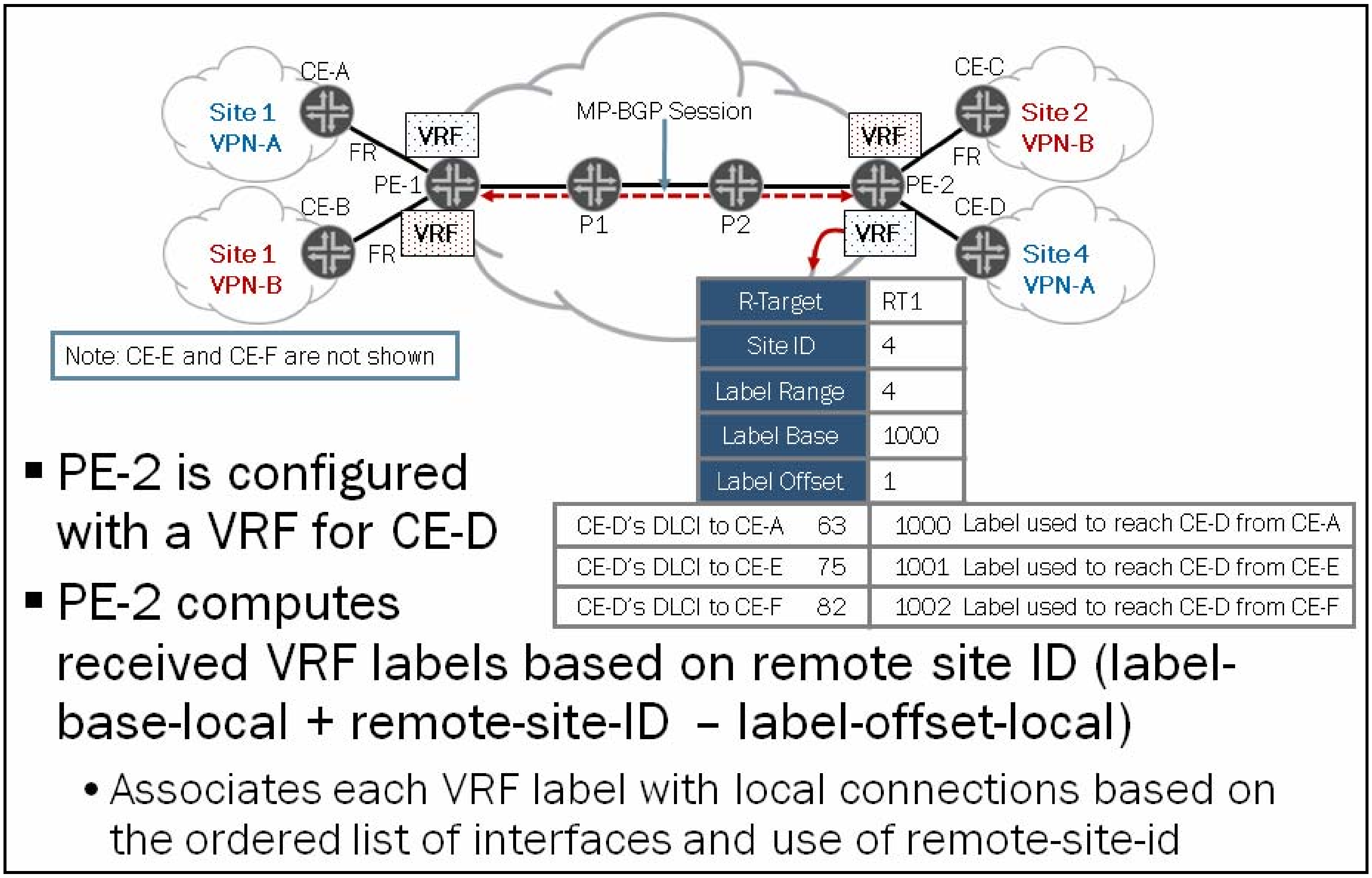
CE-D's NLRI

After configuring the local CE device properties, you must provision the site's VRF on the PE router. The following list shows what is typically involved:

- Specification of route targets and VRF policy.
- CE device identifier (site ID), which must be unique in the context of a specific VPN.
- CE device range (label block), which determines the size of the site's label block and therefore how many remote sites to which it can connect. This range produces a block of *n* contiguous labels beginning with the label base value.
- Logical interfaces associated with this VRF. The PE router assigns each sub-interface listed with a label from the site's label block. This label is used to match received traffic to the correct PE-CE logical interface.

Some of the steps outlined above can occur automatically and therefore do not require explicit configuration. The combined effect of the manual and automatic provisioning is a VRF as shown on the graphic. In operation, a subset of this VRF is sent to remote PE routers to allow them to map a label from the site's label block to traffic received over one of their locally configured PE-CE logical interfaces. The list of interfaces configured under the site's VRF must be backed up by the appropriate configuration of logical interfaces and Layer 2 protocol properties on the PE router. These interfaces must have connection identifiers and Layer 2 settings that are compatible with the configuration in the CE router. For example, a CE device running Frame Relay likely requires the PE router to have its Frame Relay interface set to DCE mode with the appropriate PVC management protocol configured.

PE-2 Has a VRF Configured for CE-D



Note: CE-E and CE-F are not shown

R-Target	RT1
Site ID	4
Label Range	4
Label Base	1000
Label Offset	1

CE-D's DLCI to CE-A	63	1000	Label used to reach CE-D from CE-A
CE-D's DLCI to CE-E	75	1001	Label used to reach CE-D from CE-E
CE-D's DLCI to CE-F	82	1002	Label used to reach CE-D from CE-F

- PE-2 is configured with a VRF for CE-D
- PE-2 computes received VRF labels based on remote site ID (label-base-local + remote-site-ID – label-offset-local)
  - Associates each VRF label with local connections based on the ordered list of interfaces and use of remote-site-id

In this example, PE-2 is configured with a VRF for its local connection to CE-D. This configuration assigns CE-D the site ID of 4 and associates this VPN with a route target of *RT2*. Also, the local site is configured with three DLCI values for use when CE-D communicates with remote sites.

Computes Received Labels Automatically

Based on the Layer 2 VPN NLRI advertisement that results from the information in PE-2's VRF, PE-2 automatically computes the label received when traffic is sent to PE-2 from the remote PE routers. Each of the labels in PE-2's label block in turn is associated with one of the site's logical interfaces, based on the order in which those interfaces are defined in the VRF. The optional use of `remote-site-id` allows local interfaces to be associated with labels in a manner independent from the order in which they are listed.

The result is that PE-2 expects to receive traffic from CE-A with a label value of 1000. This label value is then mapped to DLCI 63 on CE-D's Frame Relay interface

MP-BGP Used for Signaling

The distribution of Layer 2 VPN NLRIs between PE routers is facilitated with MP-BGP using a new Layer 2 VPN address family.

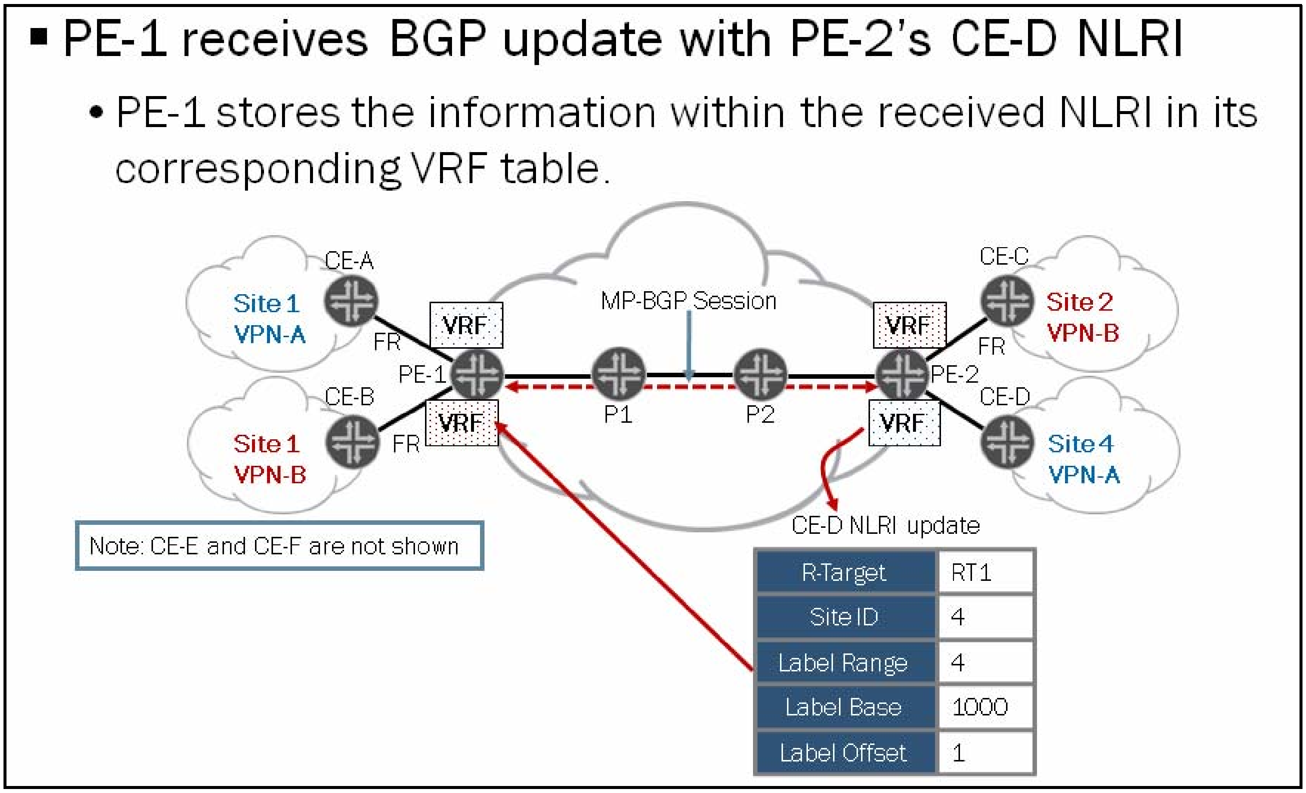
Automatic Connection Mapping

The algorithm defined in the BGP Layer 2 VPN draft allows each PE router to compute automatically the mapping between remote site IDs and the label values used to send and receive traffic from them. The labels advertised by a site are also mapped automatically to logical interfaces on the local PE-CE link. Thus, the connections between sites are created automatically.

VPN Policy

VPN policy using route target communities to filter and accept Layer 2 VPN NLRIs from remote PE routers results in a Layer 2 VPN topology.

PE-1 Receives Layer 2 VPN NLRI Update from PE-2



- PE-1 receives BGP update with PE-2's CE-D NLRI
  - PE-1 stores the information within the received NLRI in its corresponding VRF table.

R-Target	RT1
Site ID	4
Label Range	4
Label Base	1000
Label Offset	1

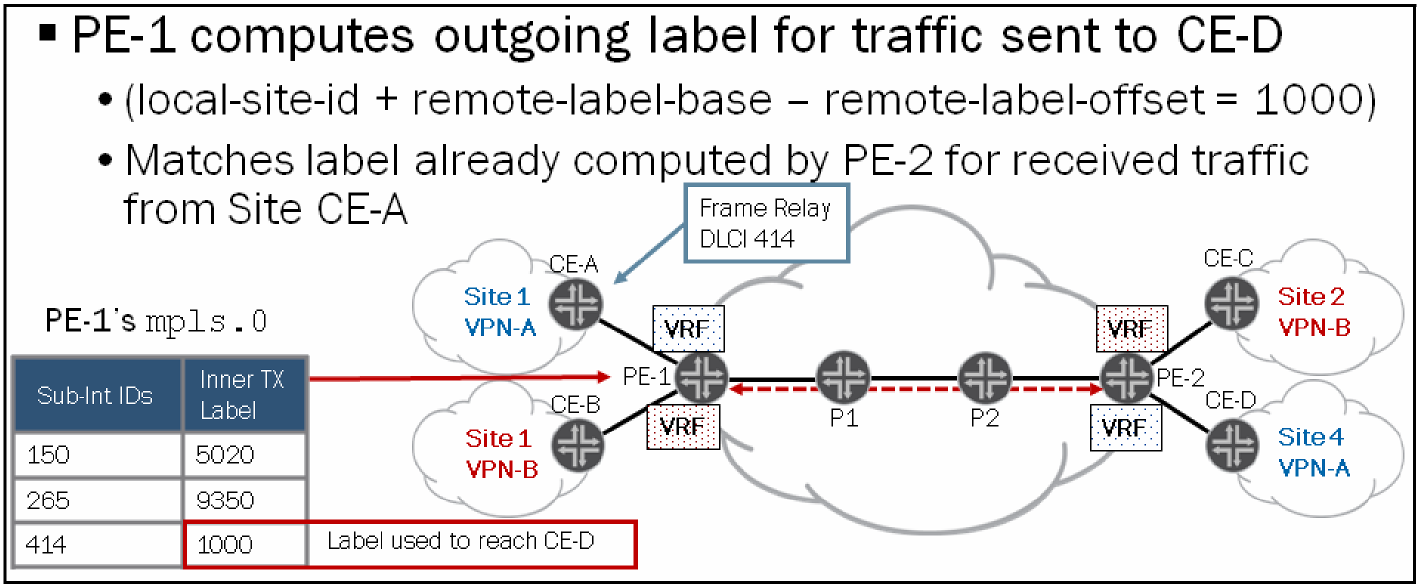
This graphic shows how a portion of PE-2's VRF for Site 4 is advertised to PE-1 using MP-BGP. The Layer 2 VPN NLRI for CE-D contains the site's ID, label block size, label offset, and label base. This update also is associated with the route target extended BGP community.

PE-1 Updates Its VRF

- PE-1 updates its CE-A VRF with CE-D's NLRI information
  - Import route target (RT1) for CE-A VRF matches route target carried by the BGP route
  - NLRI copied into `bgp.l2vpn.0`

PE-1 receives the Layer 2 VPN NLRI update from PE-2 and checks the route target for a match. Because the route target matches, the update is installed in the VRF associated with CE-D.

PE-1 Computes Outgoing Label



- PE-1 computes outgoing label for traffic sent to CE-D
  - (local-site-id + remote-label-base – remote-label-offset = 1000)
  - Matches label already computed by PE-2 for received traffic from Site CE-A

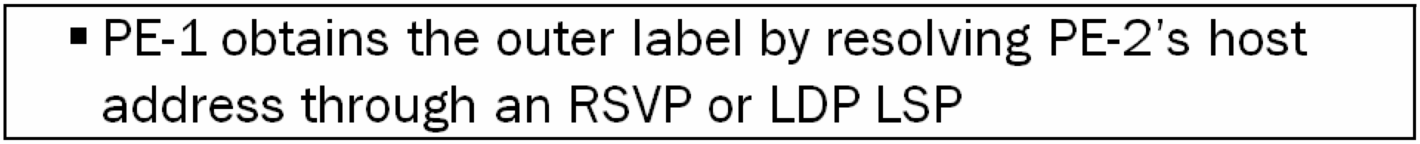
PE-1's mpls.0

Sub-Int IDs	Inner TX Label
150	5020
265	9350
414	1000

Label used to reach CE-D

PE-1 uses the Layer 2 VPN NLRI update from PE-2 to automatically compute the label to be used when sending traffic from CE-A to CE-D. PE-1 uses the algorithm that subtracts the remote PE router's label offset from its local site ID and adds the resulting value to the received label base. In this example, PE-1 computes label 1000 for traffic destined to CE-D (1–1 = 0 + 1000 = 1000). PE-2 computes the same label value (1000) as the label it expects to receive on traffic sent by CE-A.

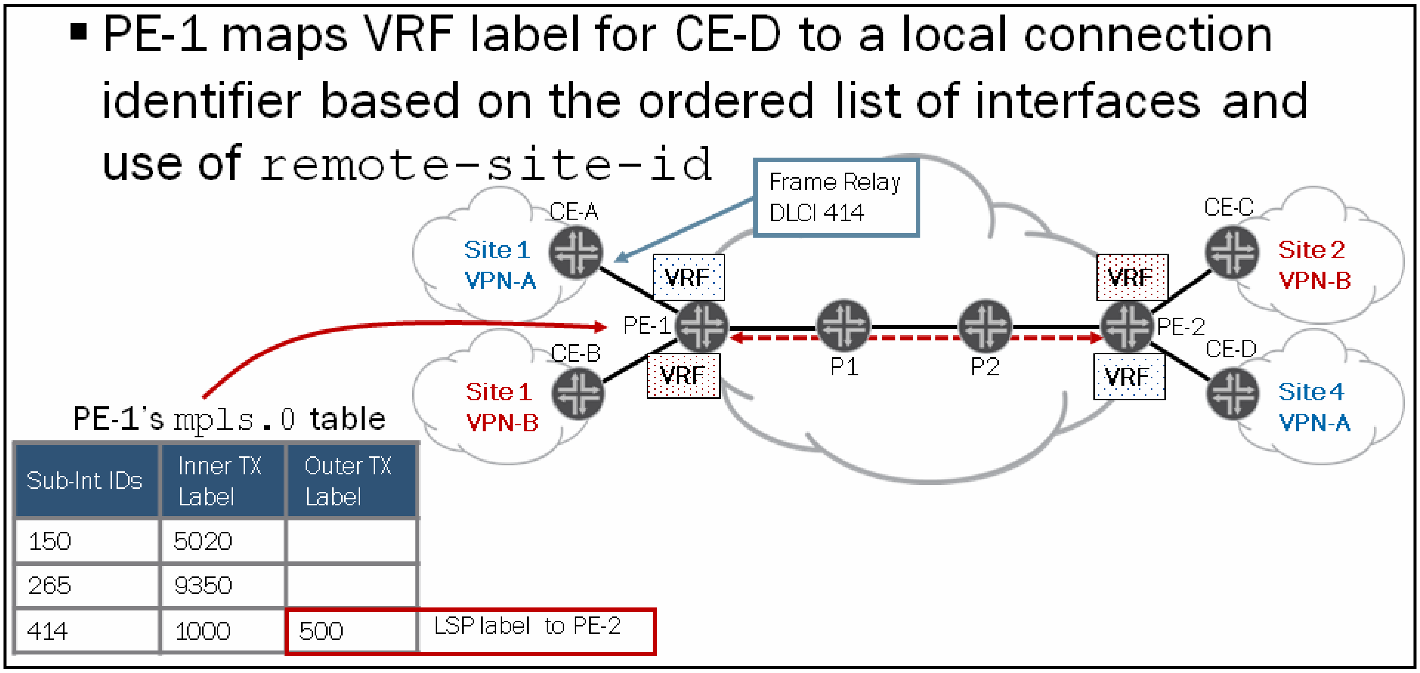
PE-1 Computes the Outer Label



- PE-1 obtains the outer label by resolving PE-2's host address through an RSVP or LDP LSP

PE-1 computes the outer MPLS label by resolving PE-2's router ID to an LSP in the inet.3 routing table. In this example, the LSP from PE-1 to PE-2 is associated with label value 500.

PE-1 Maps VRF Label to Local Connection ID



- PE-1 maps VRF label for CE-D to a local connection identifier based on the ordered list of interfaces and use of `remote-site-id`

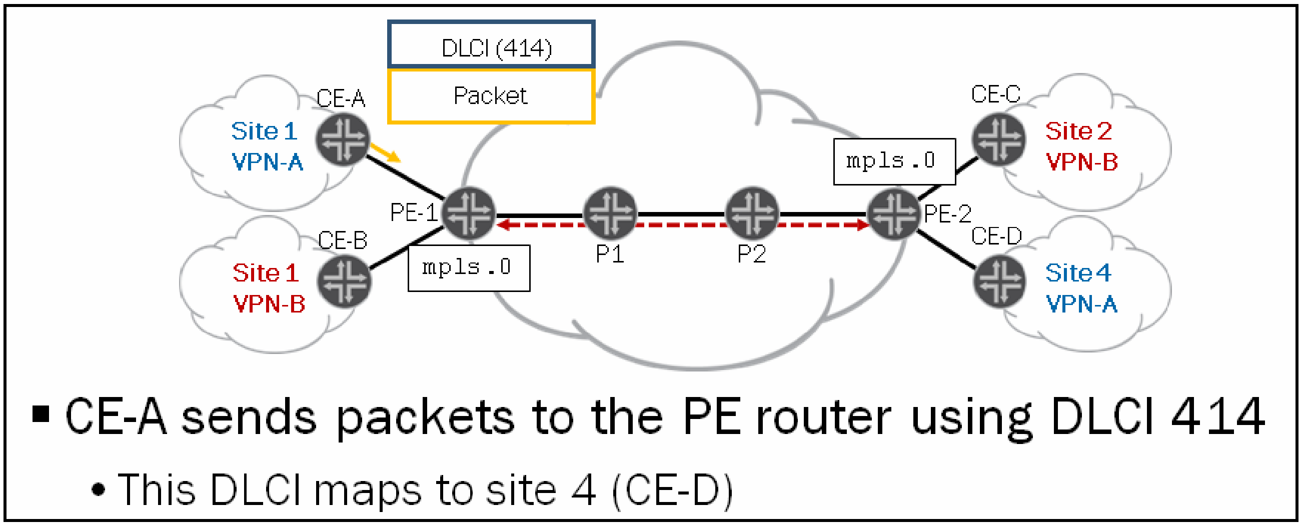
PE-1's `mpls.0` table

Sub-Int IDs	Inner TX Label	Outer TX Label
150	5020	
265	9350	
414	1000	500

As shown in the graphic, PE-1 associates Site 2 with Logical Unit 414 on the local PE-CE interface. This association is the result of either the order in which the logical interfaces are listed in Site 1's VRF, or the use of the **`remote-site-id`** option.

The result is that traffic received from CE-A on DLCI 414 is sent to PE-2 with an inner label of 1000 and a top label of 500. Upon receipt, PE-2 uses the remaining VRF label to map the frame to the logical interface associated with Site 1. Thus, after popping the VRF label, PE-2 delivers the frame to CE-D using DLCI 63.

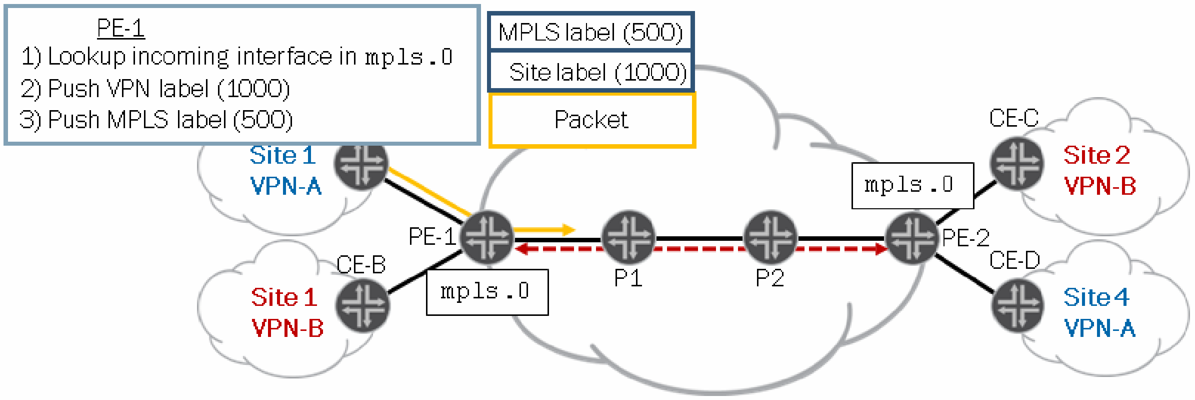
CE-A Sends Traffic to CE-D



- CE-A sends packets to the PE router using DLCI 414
  - This DLCI maps to site 4 (CE-D)

This graphic shows CE-A sending a Frame Relay frame on DLCI 414. This DLCI is associated with CE-D using the mechanisms discussed on previous pages.

PE Router Strips Frame Header



PE-1
1) Lookup incoming interface in `mpls.0`
2) Push VPN label (1000)
3) Push MPLS label (500)

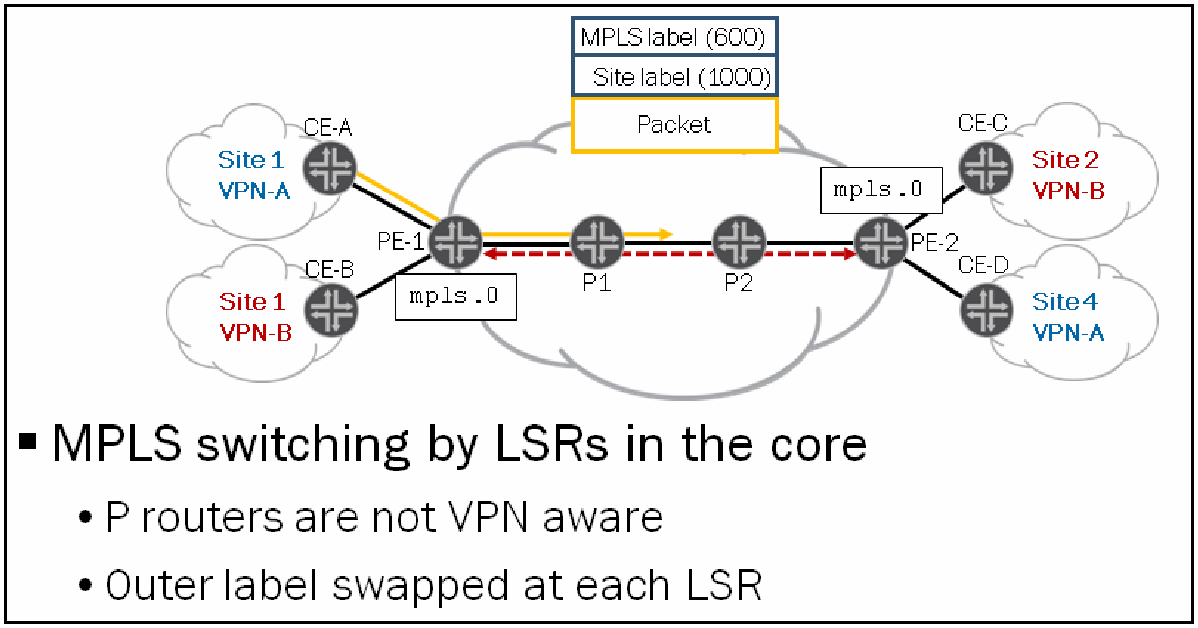
- The ingress PE router removes the DLCI
- VRF lookup derives two labels
  - Outer tunnel (MPLS) label:
    - Identifies the LSP to egress PE router
    - Distributed by RSVP or LDP
  - Inner (site) label:
    - Identifies outgoing subinterface from egress PE router to the CE
    - Derived from BGP Layer 2 NLRI distributed by egress PE router

After receiving the frame on the logical interface associated with CE-D, the PE router removes the frame header and cyclic redundancy check (CRC) fields. The fields are recomputed and added to the frame by PE-2 when it is sent to CE-D.

Double Push Operation at PE-1

PE-1 pushes two labels onto the packet. The inner label is the value computed from the information contained in PE-2's Layer 2 VPN NLRI advertisement. The outer label is derived from the resolution of PE-2's router ID to an LSP that was established using either RSVP or LDP.

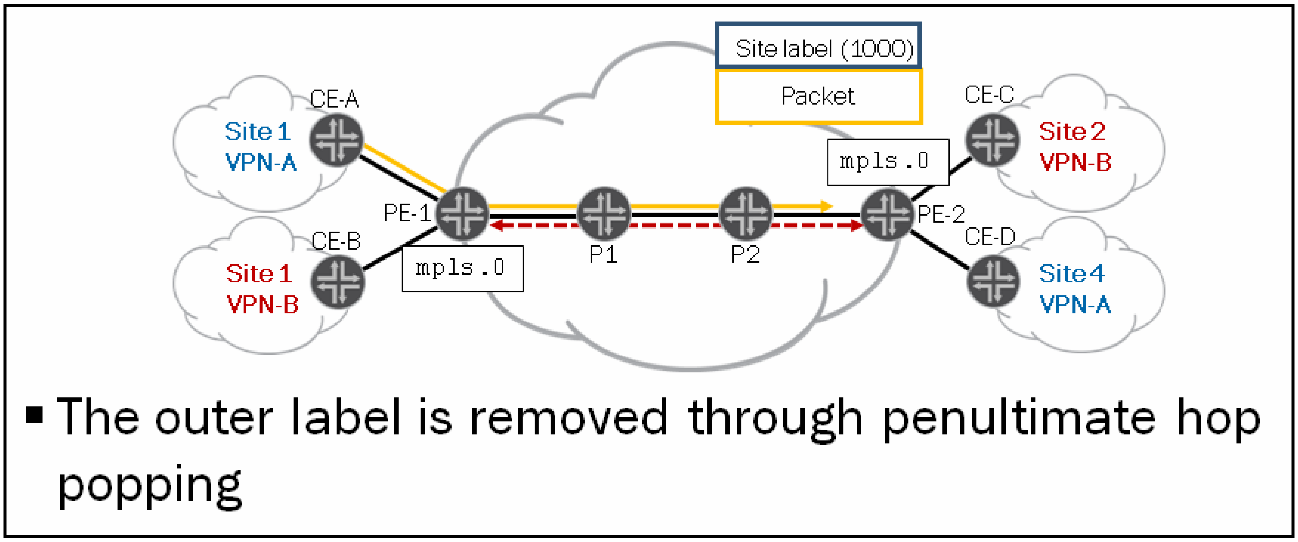
MPLS Switching in Core



- MPLS switching by LSRs in the core
  - P routers are not VPN aware
  - Outer label swapped at each LSR

The labeled packet is forwarded over the LSP connecting the two PE routers. The P routers in the core perform swap operations on the outer label. The P routers are not aware of the inner label, which remains unchanged throughout this process.

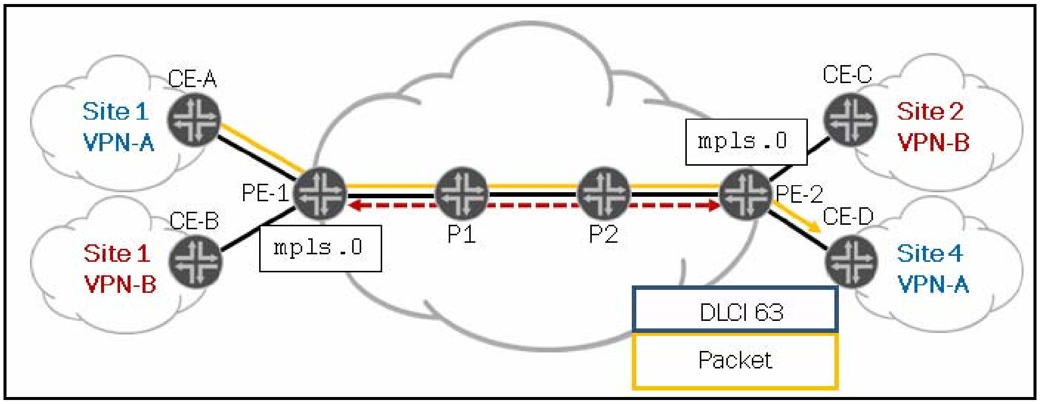
Outer Label Removed



- The outer label is removed through penultimate hop popping

The penultimate router pops the label stack, resulting in PE-2 receiving a packet with a single label.

Egress PE Router Looks Up VRF Label



The egress PE router maps the VRF label to a specific logical interface and DLCI value in the VPN-A VRF.

Egress PE Router Pops VRF Label

The egress PE router pops the label stack.

Egress PE Router Sends Frame to CE-D

The egress PE router adds a new Frame Relay header and CRC to the frame before delivering the frame to CE-D on the logical interface associated with DLCI 63.

Optimized for Common Topologies

- Optimized for common topologies (but also can support arbitrary topologies)

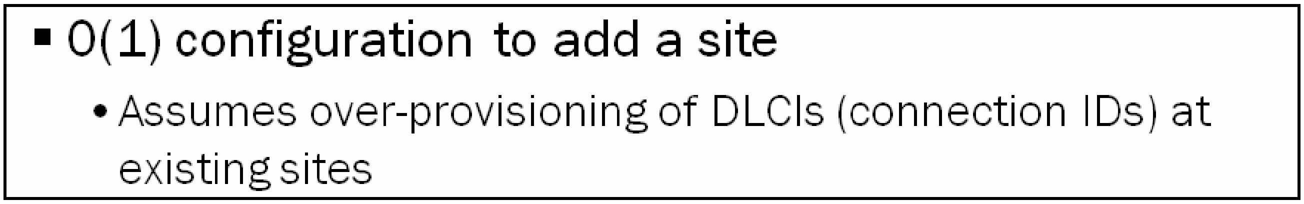
The provisioning and signaling mechanisms defined in the BGP Layer 2 VPN draft are well suited to the deployment of common topologies such as full mesh, hub and spoke, and partial meshes.

0(N) Configuration for Initial VPN

- 0(N) configuration for the whole VPN
  - Could be more for complex topologies

The addition of a new Layer 2 VPN requires the configuration of every PE and CE router involved with the new VPN. Thus, an *n* site VPN requires the configuration of *n* locations.

0(1) Configuration to Add/Remove Sites

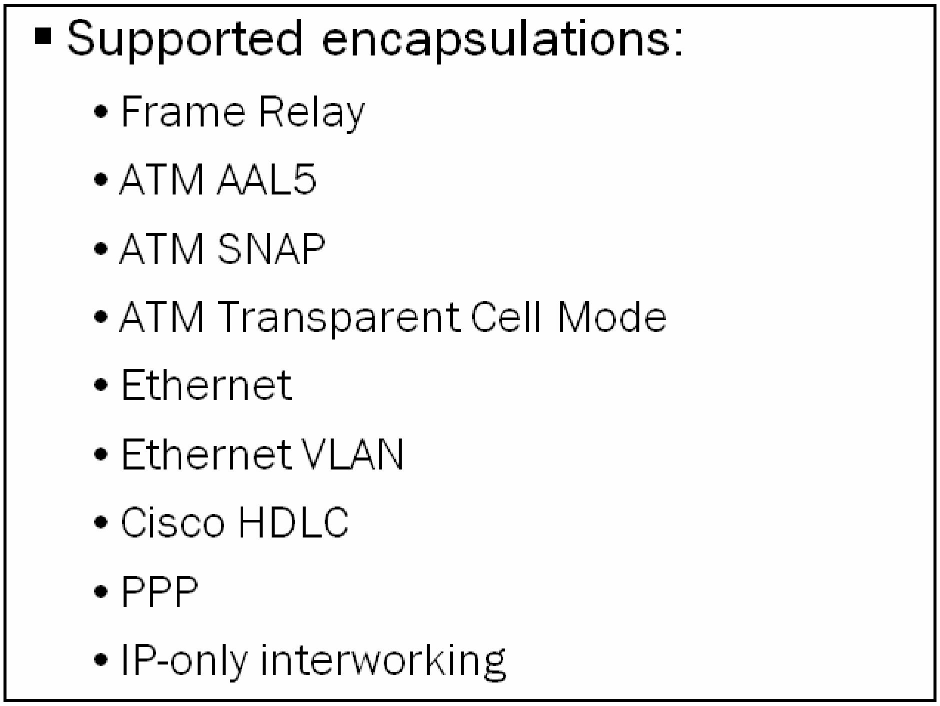


- 0(1) configuration to add a site
  - Assumes over-provisioning of DLCIs (connection IDs) at existing sites

If the Layer 2 VPN is over-provisioned during the initial configuration, the addition of a new site requires the configuration of only the new site. The ability to grow a Layer 2 VPN without having to modify the existing sites of that VPN is a key benefit of the approach outlined in the BGP Layer 2 VPN draft.

A site is considered over-provisioned when the number of logical interfaces configured on the PE-CE link, and in the corresponding VRF, exceeds the requirements dictated by the number of sites currently in use. Because the connection identifiers used to support Layer 2 VPNs are locally significant, there is no waste associated with over-provisioning.

Supported Layer 2 Encapsulations



- Supported encapsulations:
  - Frame Relay
  - ATM AAL5
  - ATM SNAP
  - ATM Transparent Cell Mode
  - Ethernet
  - Ethernet VLAN
  - Cisco HDLC
  - PPP
  - IP-only interworking

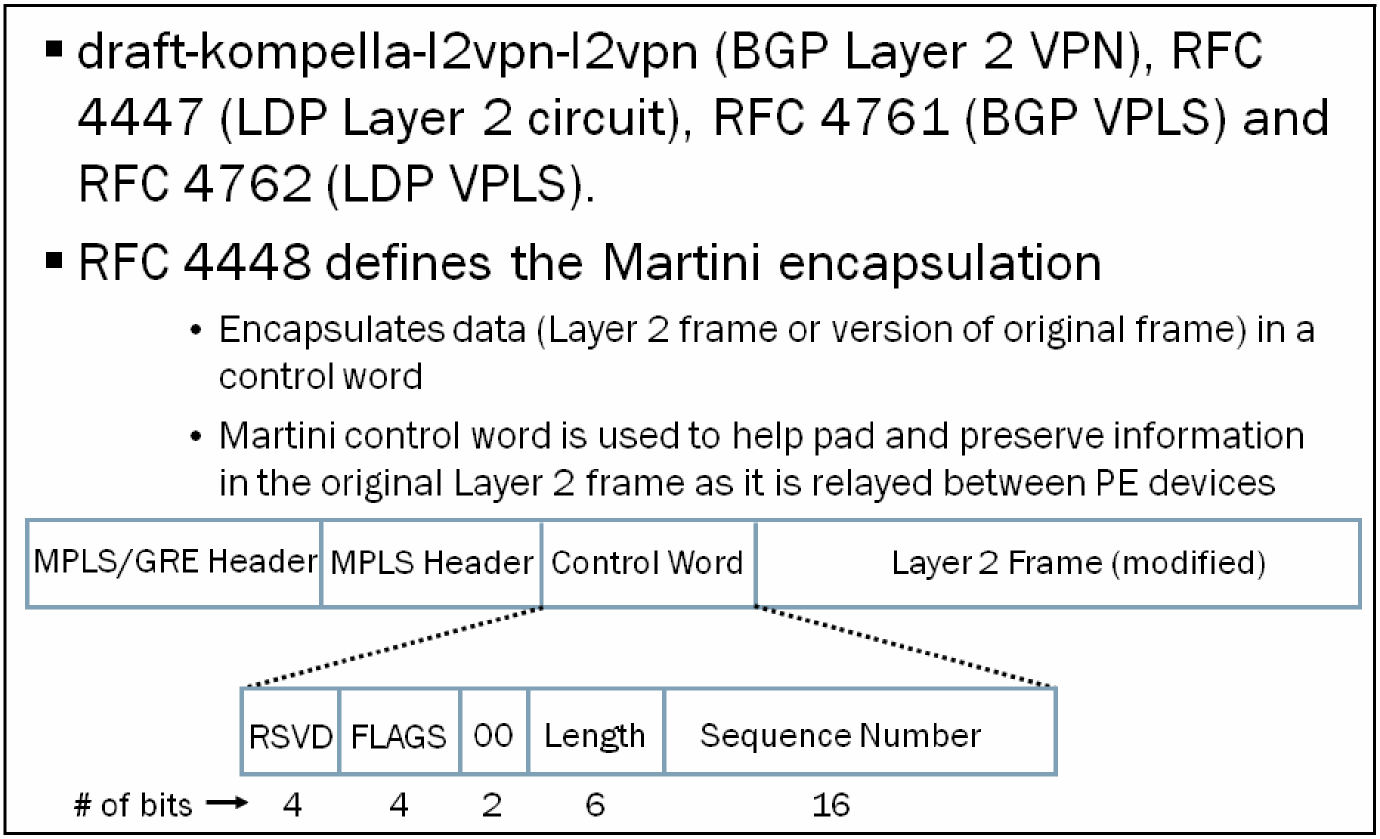
This graphic lists the Layer 2 technologies the Junos OS currently supports.

For ATM connections, the Junos OS supports both ATM Adaptation Layer 5 (AAL5), ATM subnetwork attachment point (SNAP), and cell relay (supports all AALs). All sites of a Layer 2 VPN must be optioned for the same mode.

The Cisco High-Level Data Link Control (Cisco HDLC) and Point-to-Point Protocol (PPP) encapsulation options only permit one logical unit and can therefore only support point-to-point Layer 2 VPNs.

You should consider the BGP Layer 2 VPN IP-only interworking function or a Layer 3 VPN solution when you must interconnect sites with different Layer 2 technologies.

RFC 4448 Encapsulation



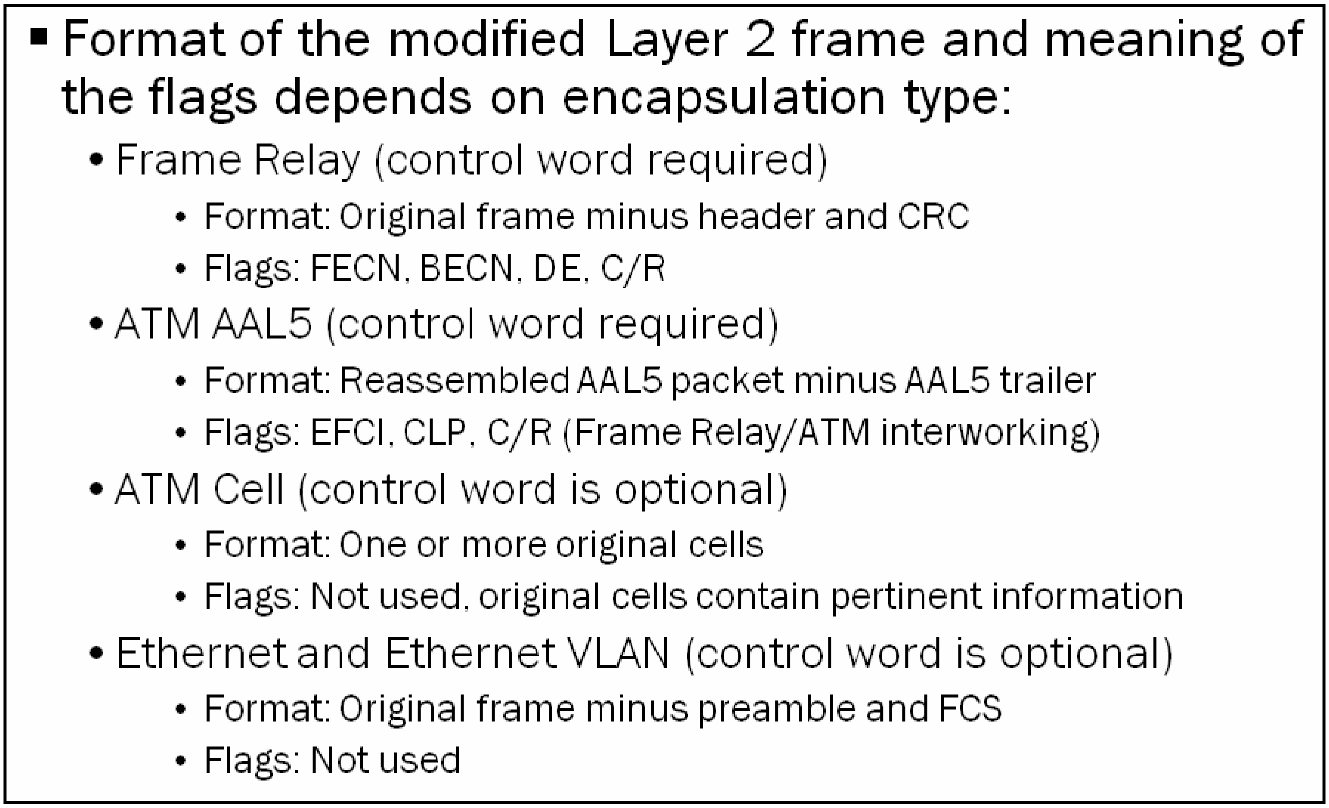
- draft-kompella-l2vpn-l2vpn (BGP Layer 2 VPN), RFC 4447 (LDP Layer 2 circuit), RFC 4761 (BGP VPLS) and RFC 4762 (LDP VPLS).
- RFC 4448 defines the Martini encapsulation
  - Encapsulates data (Layer 2 frame or version of original frame) in a control word
  - Martini control word is used to help pad and preserve information in the original Layer 2 frame as it is relayed between PE devices

MPLS/GRE Header	MPLS Header	Control Word	Layer 2 Frame (modified)

	RSVD	FLAGS	00	Length	Sequence Number
# of bits →	4	4	2	6	16

The data encapsulation method (Martini) is defined in RFC 4448 is used in the forwarding plane for BGP Layer 2 VPNs, LDP Layer 2 circuits, BGP VPLS, and LDP VPLS. The diagram shows how the resulting MPLS-encapsulated packet appears after it leaves the ingress PE router. In general, the Layer 2 frame is slightly modified (described in next few sections) and then encapsulated in a 32-bit control word, followed by two MPLS headers—possibly generic routing encapsulation (GRE) as the final encapsulation. The control word is used for essentially three purposes: to enable the padding of small protocol data units (PDUs) that do not meet minimum maximum transmission unit (MTU) requirements, to preserve Layer 2 bit settings (that is, DE, FECN, BECN for Frame Relay), and to preserve sequencing if sequencing is required.

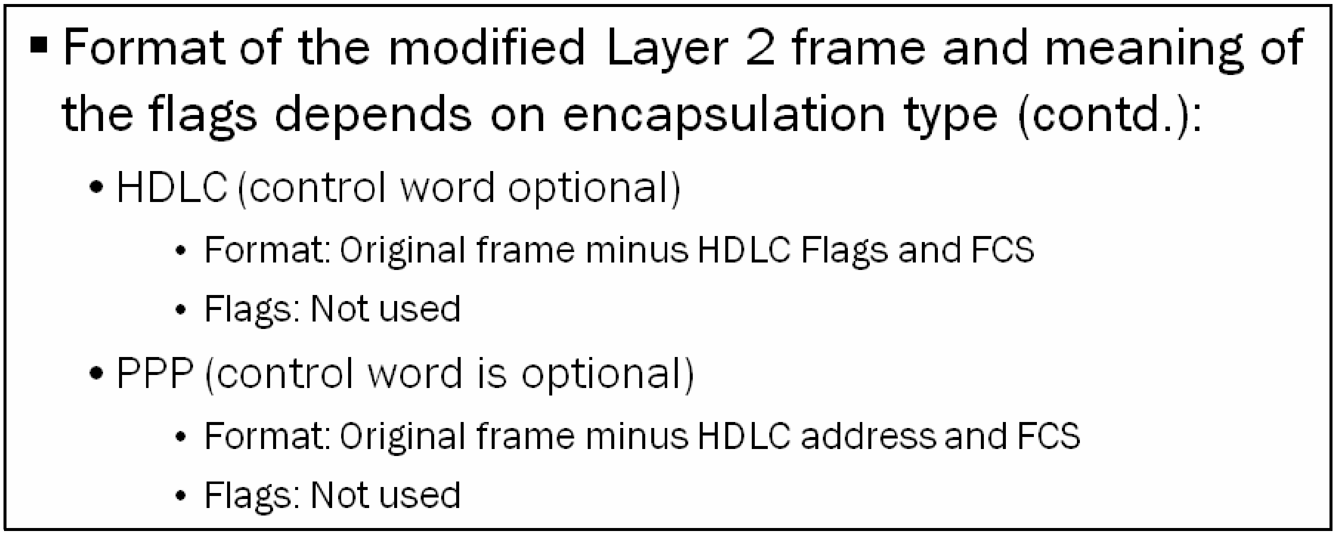
Format of Layer 2 Frame and Meaning of Flags: Part 1



- Format of the modified Layer 2 frame and meaning of the flags depends on encapsulation type:
  - Frame Relay (control word required)
    - Format: Original frame minus header and CRC
    - Flags: FECN, BECN, DE, C/R
  - ATM AAL5 (control word required)
    - Format: Reassembled AAL5 packet minus AAL5 trailer
    - Flags: EFCI, CLP, C/R (Frame Relay/ATM interworking)
  - ATM Cell (control word is optional)
    - Format: One or more original cells
    - Flags: Not used, original cells contain pertinent information
  - Ethernet and Ethernet VLAN (control word is optional)
    - Format: Original frame minus preamble and FCS
    - Flags: Not used

The data that is encapsulated by the control word is generally a modified Layer 2 frame. What is changed from the original Layer 2 frame depends upon the encapsulation type. The same goes for the meaning of the flags in the control word—it depends on the encapsulation type. The graphic describes the changes made to the original Layer 2 frame as well as the meaning of the flags in the control word.

Format of Layer 2 Frame and Meaning of Flags: Part 2



- Format of the modified Layer 2 frame and meaning of the flags depends on encapsulation type (contd.):
  - HDLC (control word optional)
    - Format: Original frame minus HDLC Flags and FCS
    - Flags: Not used
  - PPP (control word is optional)
    - Format: Original frame minus HDLC address and FCS
    - Flags: Not used

The graphic describes the changes made to the original Layer 2 frame as well as the meaning of the flags in the control word when HDLC or PPP are used.

Preliminary Steps

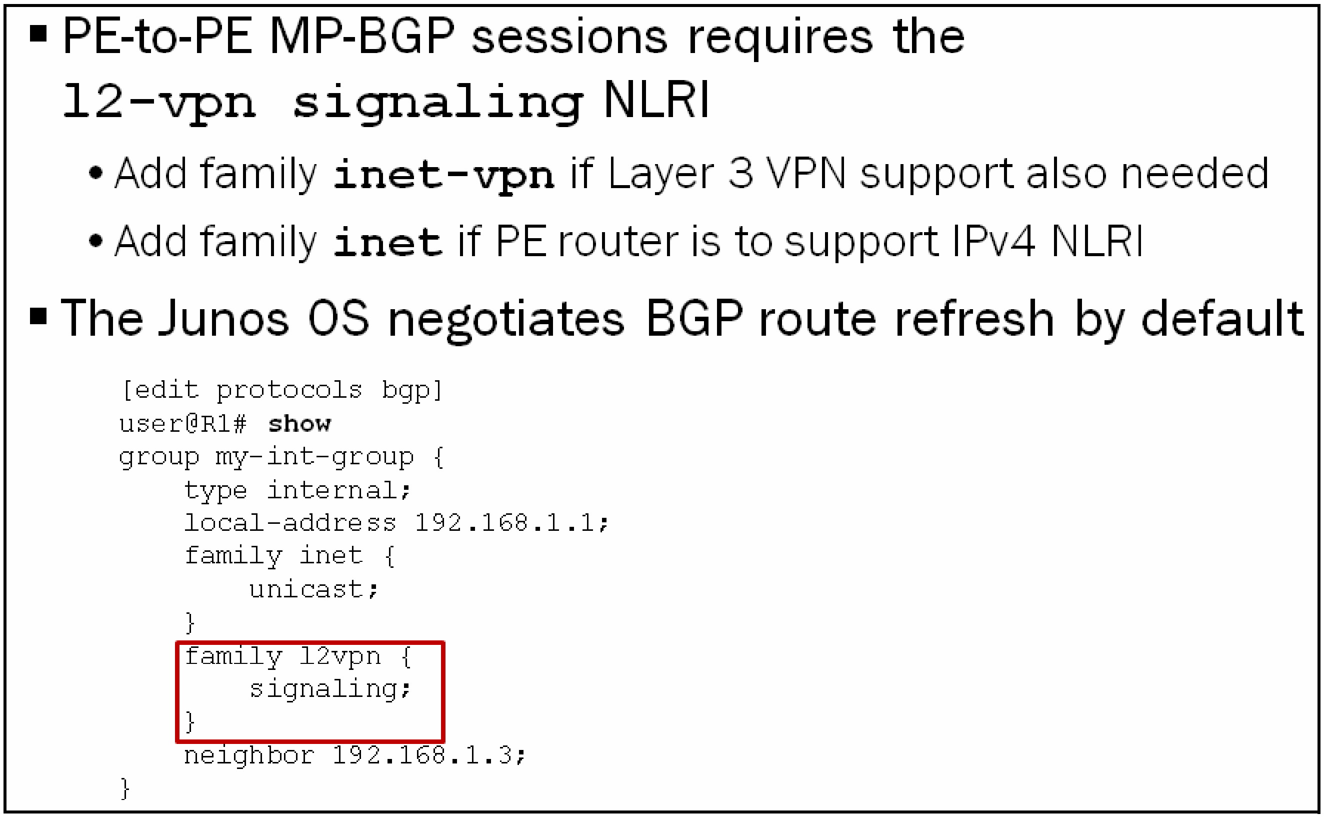
Before a functional Layer 2 VPN can be deployed, the provider's core requires preliminary configuration. These steps include:

1. Choosing and configuring an IGP;

2. Establishing MP-IBGP peering sessions between PE routers sharing VPN membership (these peering sessions must support the `l2-vpn signaling` family);
3. Configuring the PE and P routers to support MPLS and signaling protocols, such as RSVP or LDP; and
4. Establishing LSPs between PE routers sharing VPN membership.

VPN Configuration on PE Routers Only

As with a Layer 3 VPN, PE routers perform all VPN-specific configuration.



- PE-to-PE MP-BGP sessions requires the `l2-vpn signaling` NLRI
  - Add family **`inet-vpn`** if Layer 3 VPN support also needed
  - Add family **`inet`** if PE router is to support IPv4 NLRI
- The Junos OS negotiates BGP route refresh by default

```
[edit protocols bgp]
user@R1# show
group my-int-group {
    type internal;
    local-address 192.168.1.1;
    family inet {
        unicast;
    }
    family l2vpn {
        signaling;
    }
    neighbor 192.168.1.3;
}
```

Layer 2 VPN NLRI Required

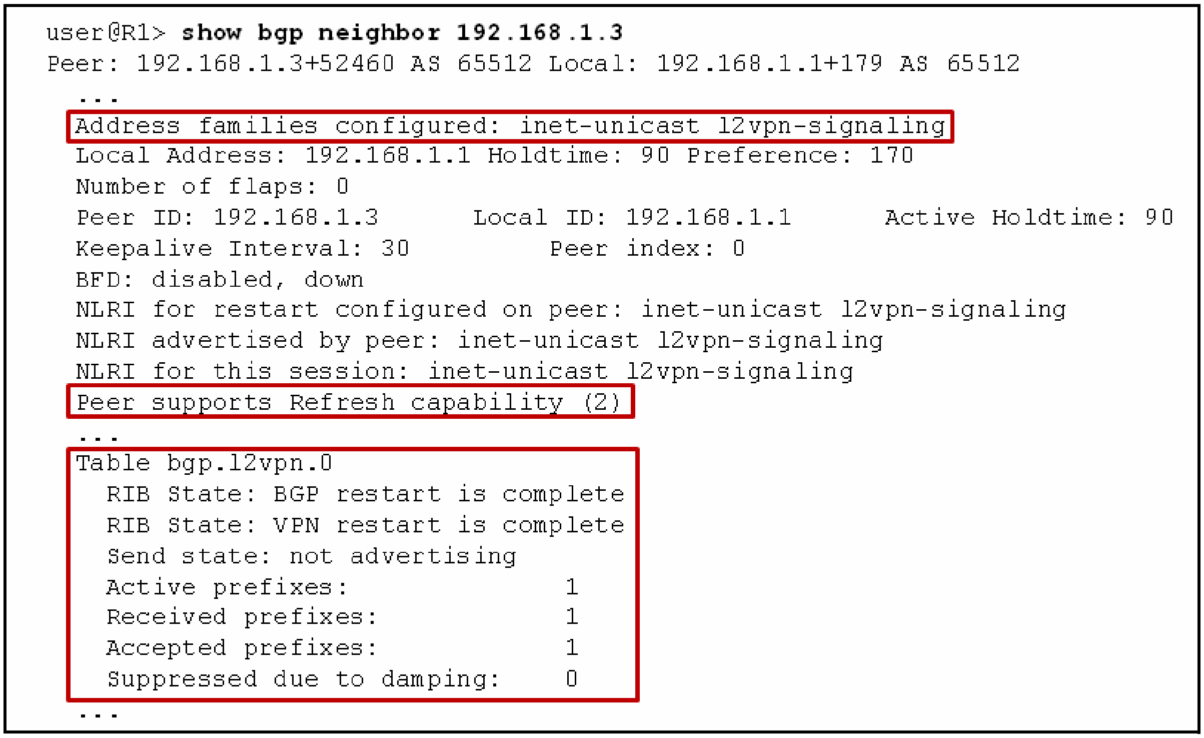
You must configure the PE-PE MP-BGP by specifying the `l2-vpn signaling` family for sessions to support Layer 2 VPN NLRI. If wanted, you also can configure the same MP-BGP session to support Layer 3 VPN signaling and conventional IP version 4 (IPv4) routing by specifying additional address families. You must configure the default address family (IPv4 unicast) explicitly once a nondefault family is configured.

The example on the graphic shows an MP-BGP session configured for Layer 2 VPN and IPv4 unicast support.

BGP Route Refresh Is Automatic

Layer 2 VPNs also require the BGP route refresh capability for non-disruptive moves and changes. The Junos OS automatically negotiates this capability. Thus, explicit configuration is not required.

MP-BGP Peering Example

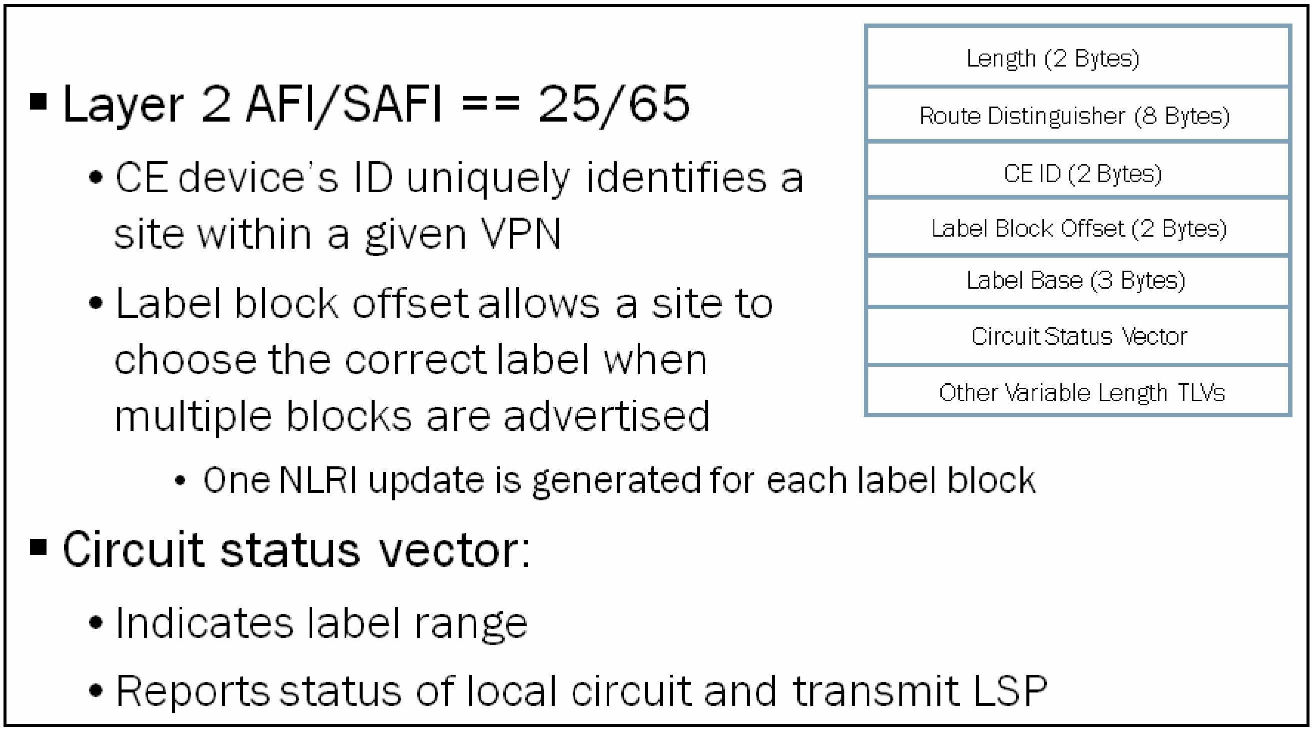


```
user@R1> show bgp neighbor 192.168.1.3
Peer: 192.168.1.3+52460 AS 65512 Local: 192.168.1.1+179 AS 65512
  ...
  Address families configured: inet-unicast l2vpn-signaling
  Local Address: 192.168.1.1 Holdtime: 90 Preference: 170
  Number of flaps: 0
  Peer ID: 192.168.1.3      Local ID: 192.168.1.1      Active Holdtime: 90
  Keepalive Interval: 30         Peer index: 0
  BFD: disabled, down
  NLRI for restart configured on peer: inet-unicast l2vpn-signaling
  NLRI advertised by peer: inet-unicast l2vpn-signaling
  NLRI for this session: inet-unicast l2vpn-signaling
  Peer supports Refresh capability (2)
  ...
  Table bgp.l2vpn.0
    RIB State: BGP restart is complete
    RIB State: VPN restart is complete
    Send state: not advertising
    Active prefixes:              1
    Received prefixes:            1
    Accepted prefixes:            1
    Suppressed due to damping:    0
  ...
```

This graphic shows an MP-BGP session capable of supporting a Layer 2 VPN.

The presence of the `l2-vpn signaling` family on an MP-BGP session results in the automatic creation of the `bgp.l2vpn.0` routing table. This table holds all Layer 2 VPN NLRI received by the PE router that has at least one matching route target. The Layer 2 VPN NLRIs in this table are copied into the matching VRFs.

Layer 2 VPN AFI/SAFIs



- Layer 2 AFI/SAFI == 25/65
  - CE device's ID uniquely identifies a site within a given VPN
  - Label block offset allows a site to choose the correct label when multiple blocks are advertised
    - One NLRI update is generated for each label block
- Circuit status vector:
  - Indicates label range
  - Reports status of local circuit and transmit LSP

Length (2 Bytes)

Route Distinguisher (8 Bytes)

CE ID (2 Bytes)

Label Block Offset (2 Bytes)

Label Base (3 Bytes)

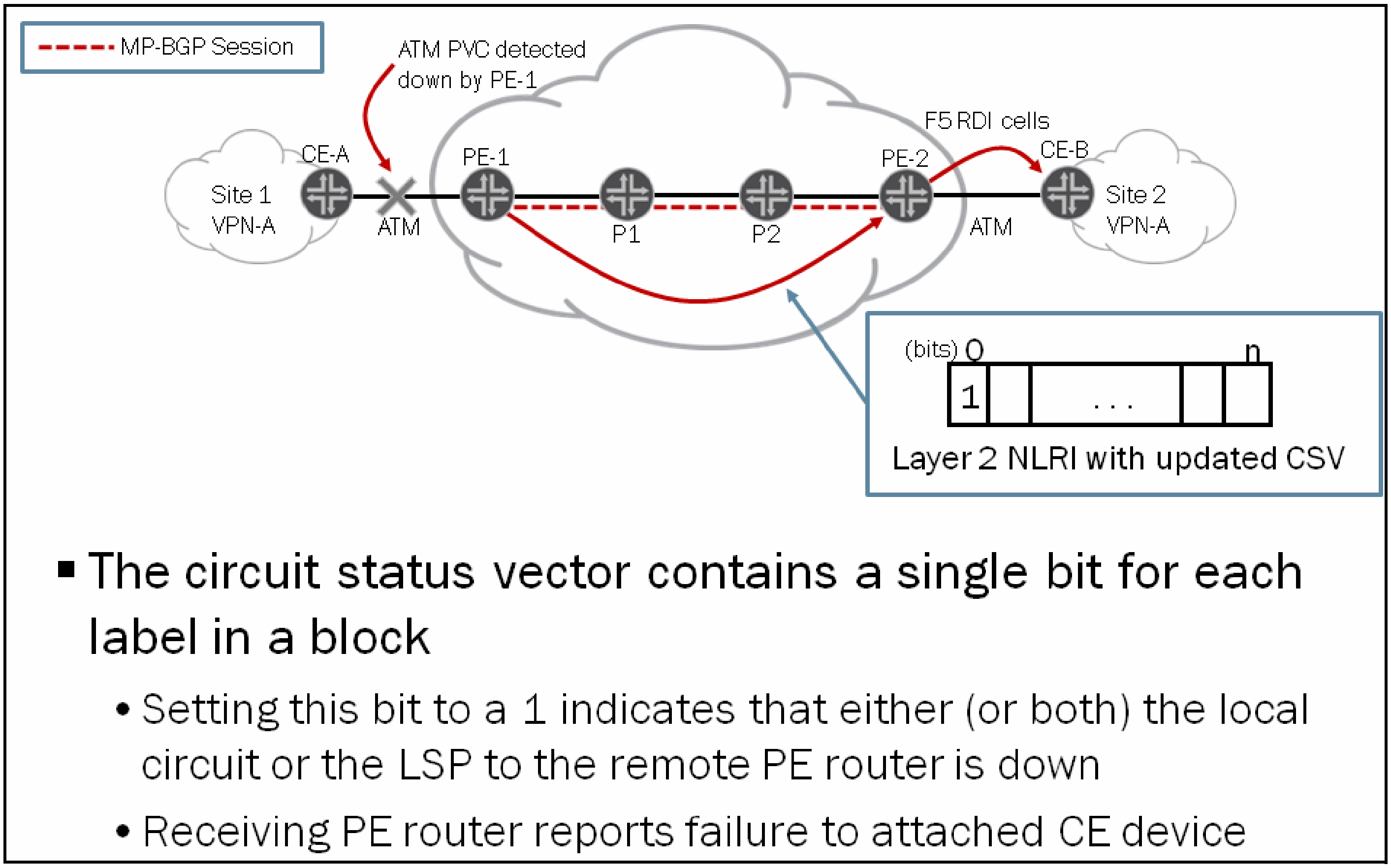
Circuit Status Vector

Other Variable Length TLVs

The graphic displays the structure of Layer 2 VPN NLRI. The address family indicator (AFI) and subsequent address family identifier (SAFI) values of 25 and 65 are shared with BGP VPLS NLRIs.

The NLRI consists of the CE device's ID (site ID), the label base, and the label block offset, which are used when multiple label blocks are generated for a particular site. Each label block is carried as a separate update when multiple blocks exist.

The Circuit Status Vector



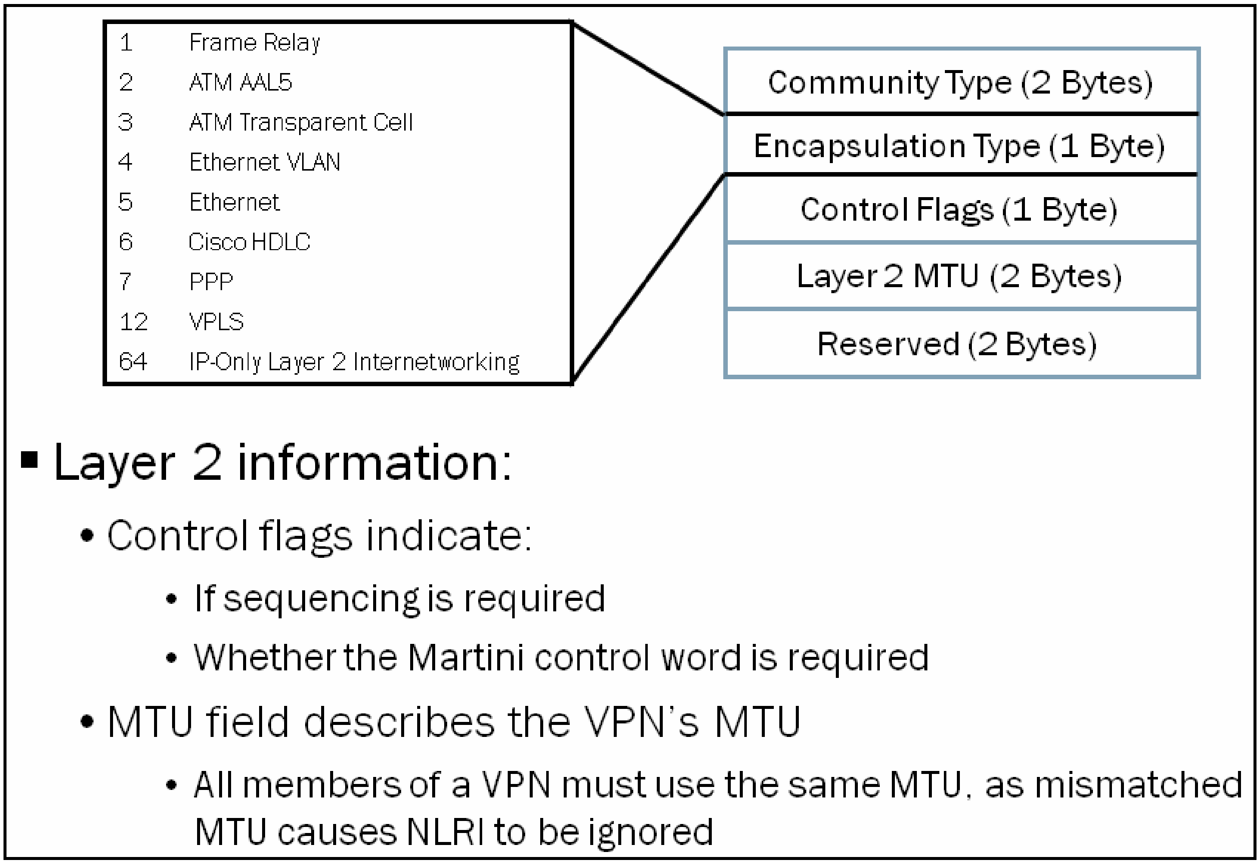
- The circuit status vector contains a single bit for each label in a block
  - Setting this bit to a 1 indicates that either (or both) the local circuit or the LSP to the remote PE router is down
  - Receiving PE router reports failure to attached CE device

The circuit status vector is a bit vector used to indicate the site's label range (that is, block size) and to report failures of a PE router's local circuits.

The Junos OS uses the circuit status vector defined in the BGP Layer 2 VPN draft to report local circuit failures as well as failures of the transmit LSP to the remote PE router. The circuit status vector is a bit vector containing a single bit for each label (circuit ID) in a label block. Therefore, the circuit status vector can be used to convey label block size, as well as to provide an indication of Layer 2 circuit status and transmit LSP status to the remote PE router.

This graphic starts with the PE-1router detecting an ATM circuit failure to Site 1. As a result, the PE-1 router sends an updated Layer 2 NLRI with the corresponding bit in the circuit status vector set to 1. The remote PE router affected by this change carries the failure indication towards the access side using whatever mechanism the Layer 2 protocol supports. In this example, which is based on ATM, F5 remote defect indication (RDI) cells are generated to inform the CE device of the failure. For Frame Relay, the failure results in an inactive PVC status being reported in the PVC management protocol (American National Standards Institute [ANSI] Annex D or International Telecommunication Union [ITU] Annex A).

Layer 2 Information Extended Communities



1	Frame Relay
2	ATM AAL5
3	ATM Transparent Cell
4	Ethernet VLAN
5	Ethernet
6	Cisco HDLC
7	PPP
12	VPLS
64	IP-Only Layer 2 Internetworking

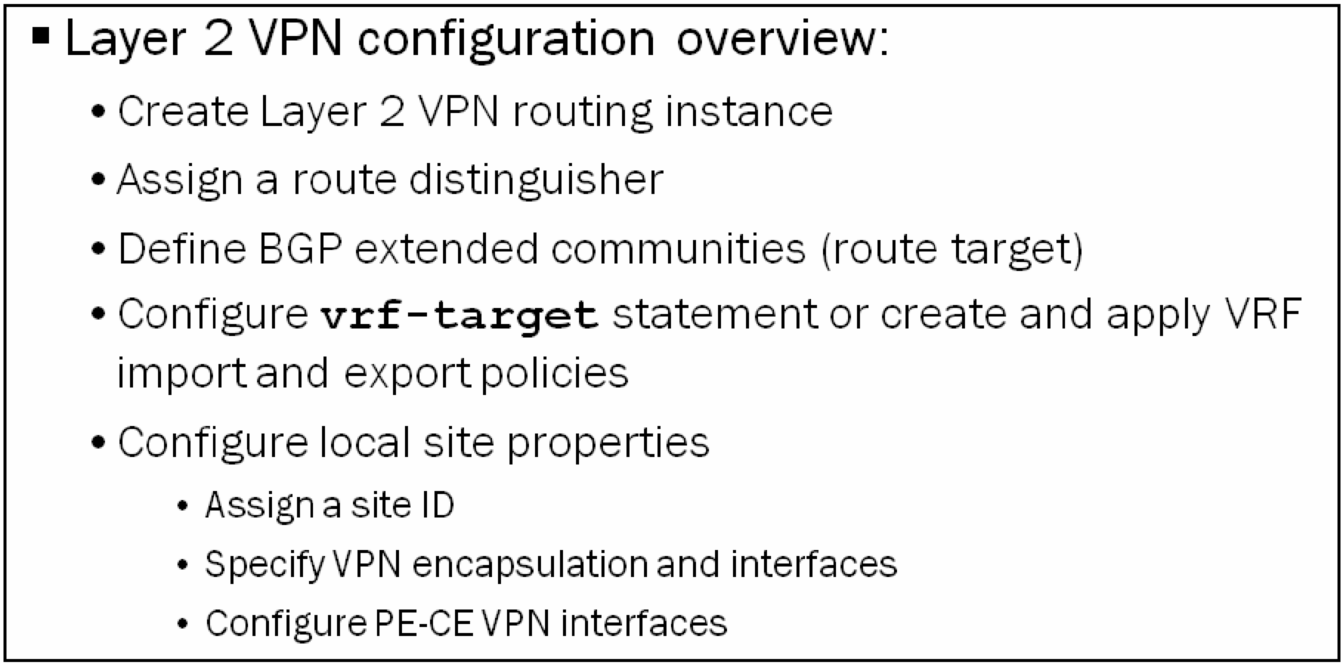
- Layer 2 information:
  - Control flags indicate:
    - If sequencing is required
    - Whether the Martini control word is required
  - MTU field describes the VPN's MTU
    - All members of a VPN must use the same MTU, as mismatched MTU causes NLRI to be ignored

The Layer 2 information extended communities (carried as part of the Layer 2 NLRI) communicate the following information between PE routers:

- The Layer 2 encapsulation type (defined encapsulation types are shown on the graphic);
- The control flags field, which indicates the presence of the optional Martini control word, and whether data sequencing is required; and
- The Layer 2 MTU field, which reports the MTU configured on the sending PE router's PE-CE link (because fragmentation is not supported in a Layer 2 VPN environment, the receiving PE router ignores Layer 2 NLRI with MTU values that differ from the PE router's local VRF interface).

The reserved field is currently undefined.

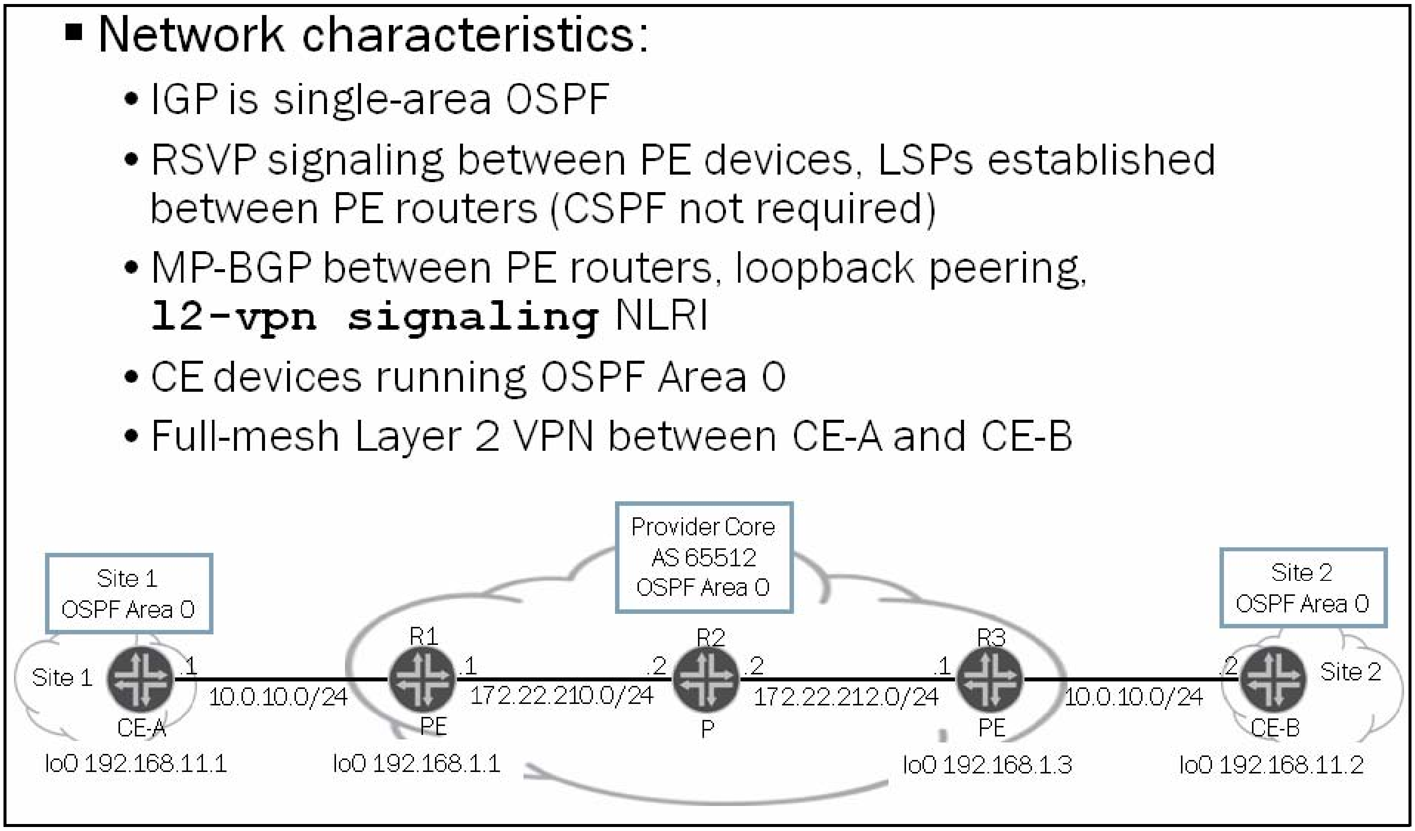
Overview of Layer 2 VPN Configuration



- Layer 2 VPN configuration overview:
  - Create Layer 2 VPN routing instance
  - Assign a route distinguisher
  - Define BGP extended communities (route target)
  - Configure **`vrf-target`** statement or create and apply VRF import and export policies
  - Configure local site properties
    - Assign a site ID
    - Specify VPN encapsulation and interfaces
    - Configure PE-CE VPN interfaces

This graphic provides a summary of the steps required to provision a Layer 2 VPN. We discuss each of these items in detail on subsequent pages.

Example Layer 2 VPN Topology



- Network characteristics:
  - IGP is single-area OSPF
  - RSVP signaling between PE devices, LSPs established between PE routers (CSPF not required)
  - MP-BGP between PE routers, loopback peering, **`l2-vpn signaling`** NLRI
  - CE devices running OSPF Area 0
  - Full-mesh Layer 2 VPN between CE-A and CE-B

The diagram serves as the basis for the various configuration-mode and operational-mode examples that follow.

The IGP is Open Shortest Path First (OSPF), and a single area (Area 0) is configured. Because the examples in this study guide do not rely on the functionality of Constrained Shortest Path First (CSPF), traffic engineering extensions need not be enabled.

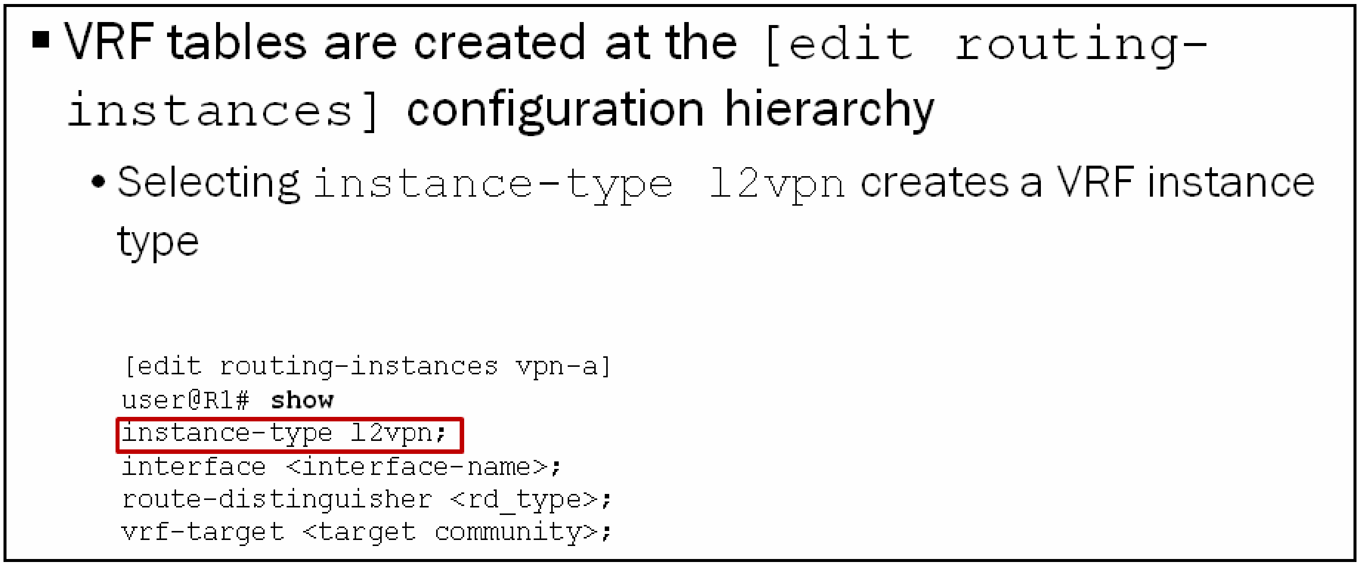
RSVP is deployed as the MPLS signaling protocol, and LSPs are configured between the R1 and R3 PE routers.

An MP-BGP peering session is configured between the loopback addresses of the PE routers. The `l2-vpn signaling` and `inet unicast` address families are configured.

In this example, the CE routers run OSPF with a common IP subnet shared by CE-A and CE-B. The PE routers have no IP addressing on the VRF interfaces.

The goal of this network is to provide full-mesh (which is point-to-point in this case) connectivity between the two CE routers shown. This network is considered a full-mesh application, as the resulting configuration readily accommodates additional sites with any-to-any connectivity.

Layer 2 VPN VRF Table Creation



- VRF tables are created at the `[edit routing-instances]` configuration hierarchy
  - Selecting `instance-type l2vpn` creates a VRF instance type

```
[edit routing-instances vpn-a]
user@R1# show
instance-type l2vpn;
interface <interface-name>;
route-distinguisher <rd_type>;
vrf-target <target community>;
```

Layer 2 VPN routing and VRFs are created at the `[edit routing-instances]` portion of the hierarchy. A Layer 2 instance is specified with arguments applied to the `instance-type` statement. As with a Layer 3 VPN VRF instance, you must assign a route distinguisher, list the VRF interfaces, and link the instance with a `vrf-target` community or VRF import and export policies. You also must configure local site properties under the `protocols` hierarchy on the Layer 2 VPNs.

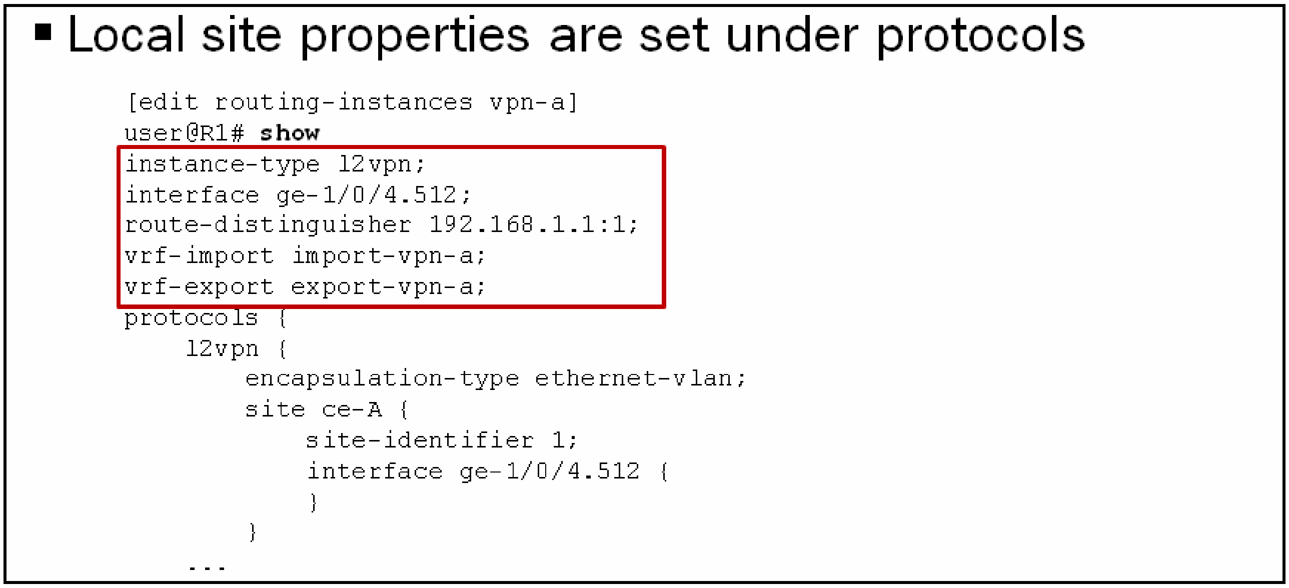
Layer 2 Instance Example

This graphic shows a sample Layer 2 routing instance based on the sample topology. This instance is called *vpn-a*. It is assigned a route distinguisher based on the PE router's loopback address (Type 1 format). The **`instance-type l2vpn`** setting creates a Layer 2 VPN VRF.

This *vpn-a* instance is associated with a single logical interface (`ge-1/0/4.512`). By listing only one VRF interface, the Layer 2 VPN is limited to single site connectivity. It would be common to see additional interfaces listed, even though they might not be required for the VPN's current connectivity, so that the auto-provisioning features of the BGP Layer 2 VPN draft can be realized. This example, however, strives to show a sample configuration with minimal complexity.

The Layer 2 VRF table can be linked to either VRF import and export policies or a **`vrf-target`** statement, which is used to match and add route target communities.

Local Site Properties

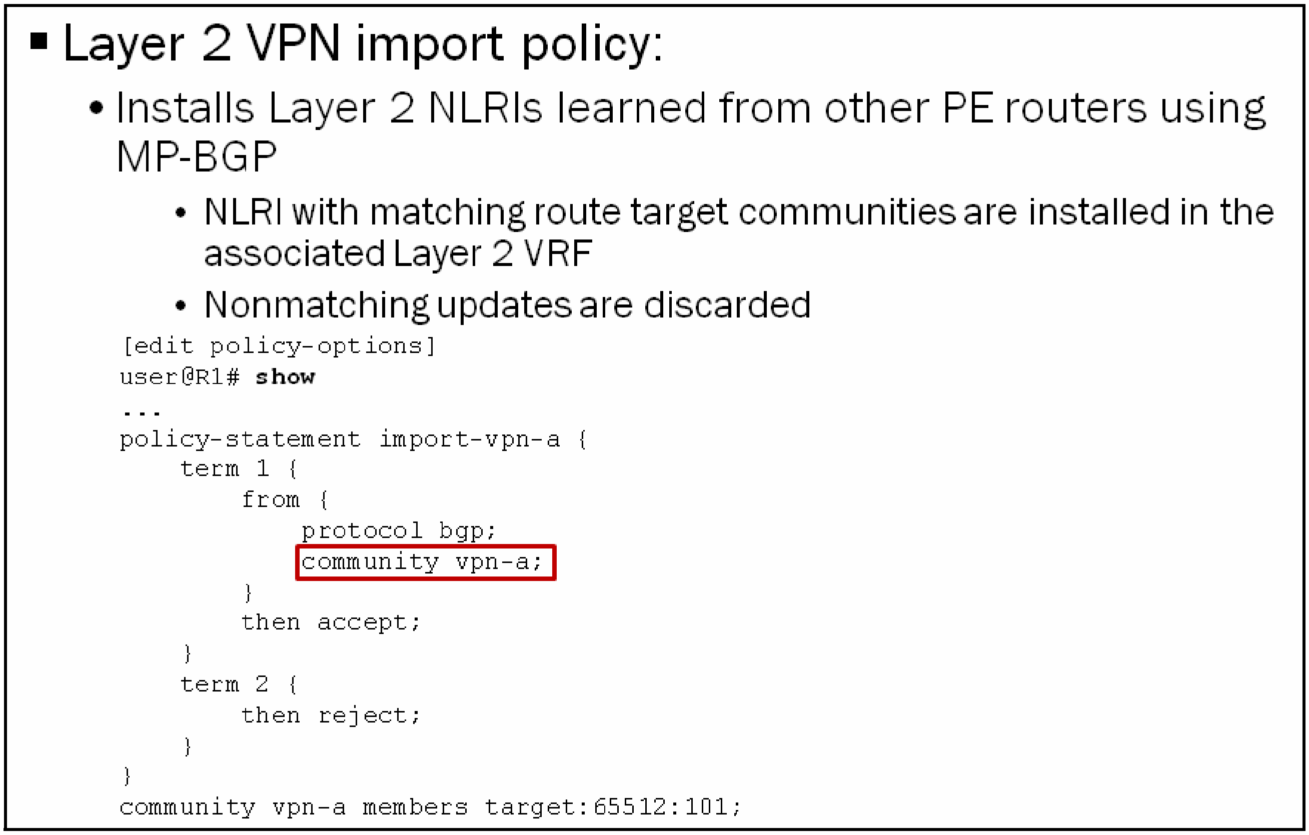


- Local site properties are set under protocols

```
[edit routing-instances vpn-a]
user@R1# show
instance-type l2vpn;
interface ge-1/0/4.512;
route-distinguisher 192.168.1.1:1;
vrf-import import-vpn-a;
vrf-export export-vpn-a;
protocols {
    l2vpn {
        encapsulation-type ethernet-vlan;
        site ce-A {
            site-identifier 1;
            interface ge-1/0/4.512 {
            }
        }
    ...
```

The local site properties are configured under the `protocols` portion of the Layer 2 instance. We discuss these parameters on subsequent pages.

Layer 2 VPN Import Policy Example



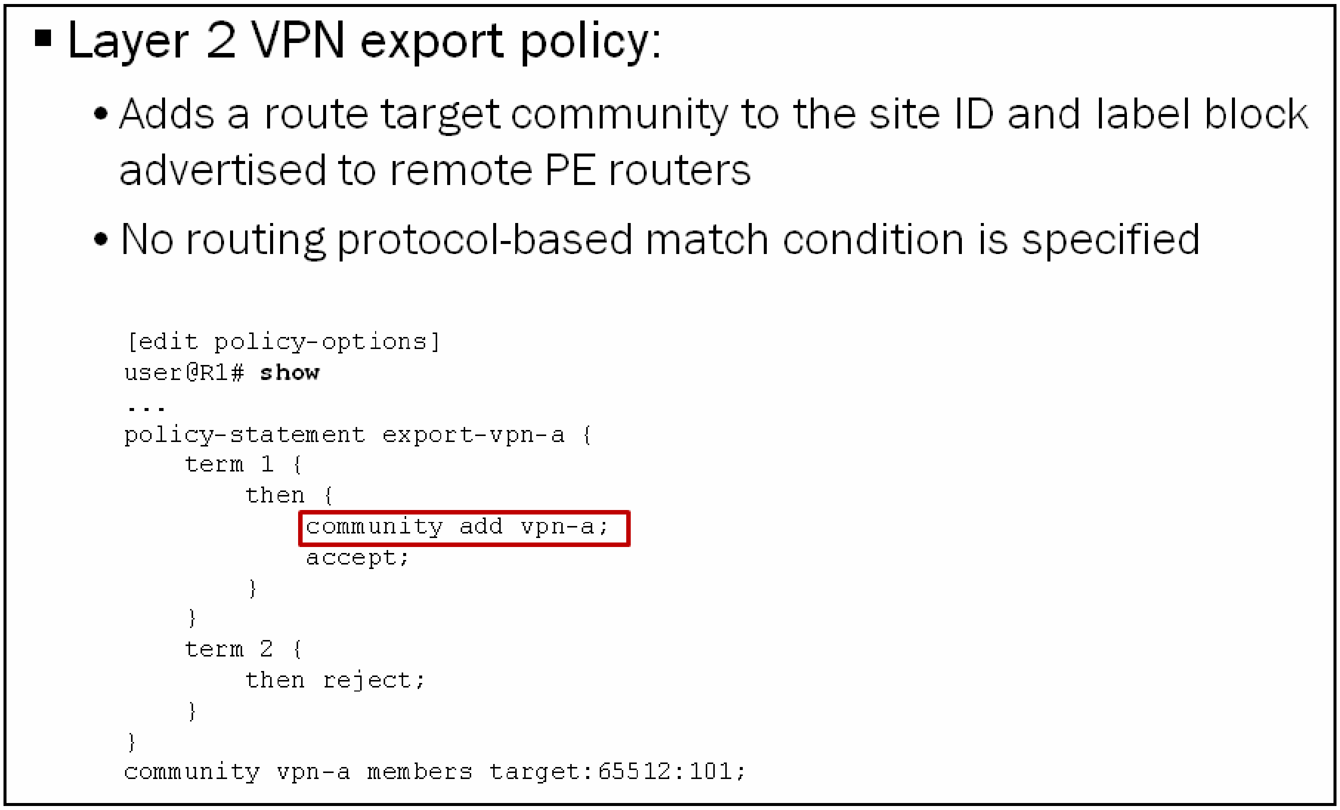
- Layer 2 VPN import policy:
  - Installs Layer 2 NLRIs learned from other PE routers using MP-BGP
    - NLRI with matching route target communities are installed in the associated Layer 2 VRF
    - Nonmatching updates are discarded

```
[edit policy-options]
user@R1# show
...
policy-statement import-vpn-a {
    term 1 {
        from {
            protocol bgp;
            community vpn-a;
        }
        then accept;
    }
    term 2 {
        then reject;
    }
}
community vpn-a members target:65512:101;
```

This graphic shows an example of a Layer 2 VRF import policy. Term 1 matches BGP routes with the *vpn-a* route target while the second term rejects all other routes.

As a result of this policy, Layer 2 VPN NLRIs received from remote PE routers are installed in the `vpn-a` VRF when they contain the *vpn-a* community.

Layer 2 VPN Export Policy Example

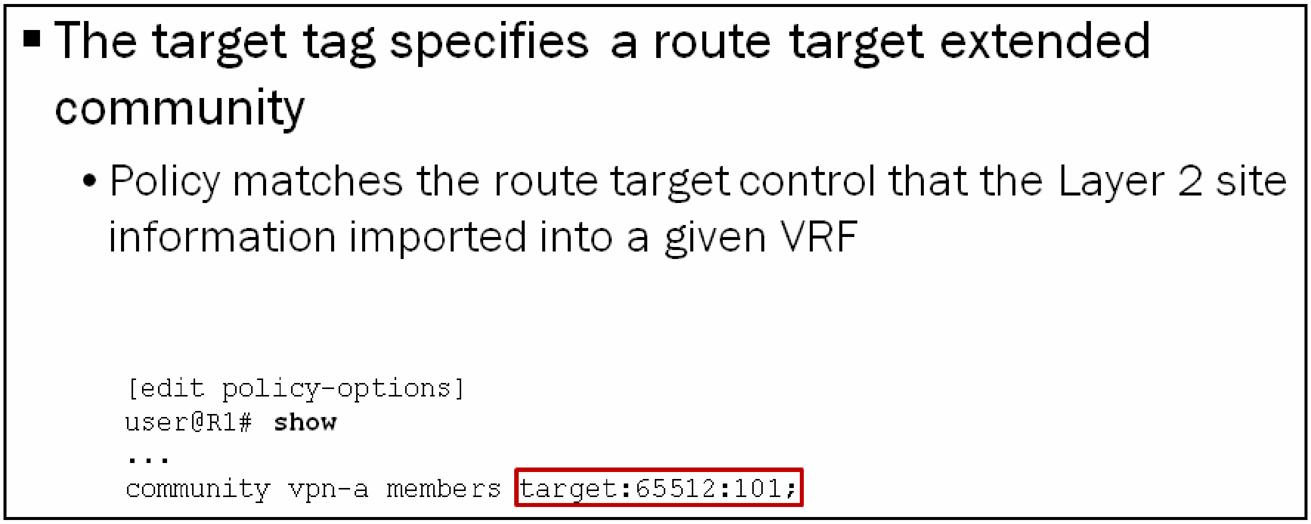


- Layer 2 VPN export policy:
  - Adds a route target community to the site ID and label block advertised to remote PE routers
  - No routing protocol-based match condition is specified

```
[edit policy-options]
user@R1# show
...
policy-statement export-vpn-a {
    term 1 {
        then {
            community add vpn-a;
            accept;
        }
    }
    term 2 {
        then reject;
    }
}
community vpn-a members target:65512:101;
```

This graphic shows an example of a Layer 2 VRF export policy. Term 1 matches the local site's VRF information and adds a route target community. The second term rejects all other sources of information. In contrast to a Layer 3 VPN's export policy, no protocol-based match is used in a Layer 2 VPN's export policy because the PE-CE pairing is not an IP or routing protocol set of peers. Thus, the PE router does not learn any information from the attached site.

Route Target Extended Community



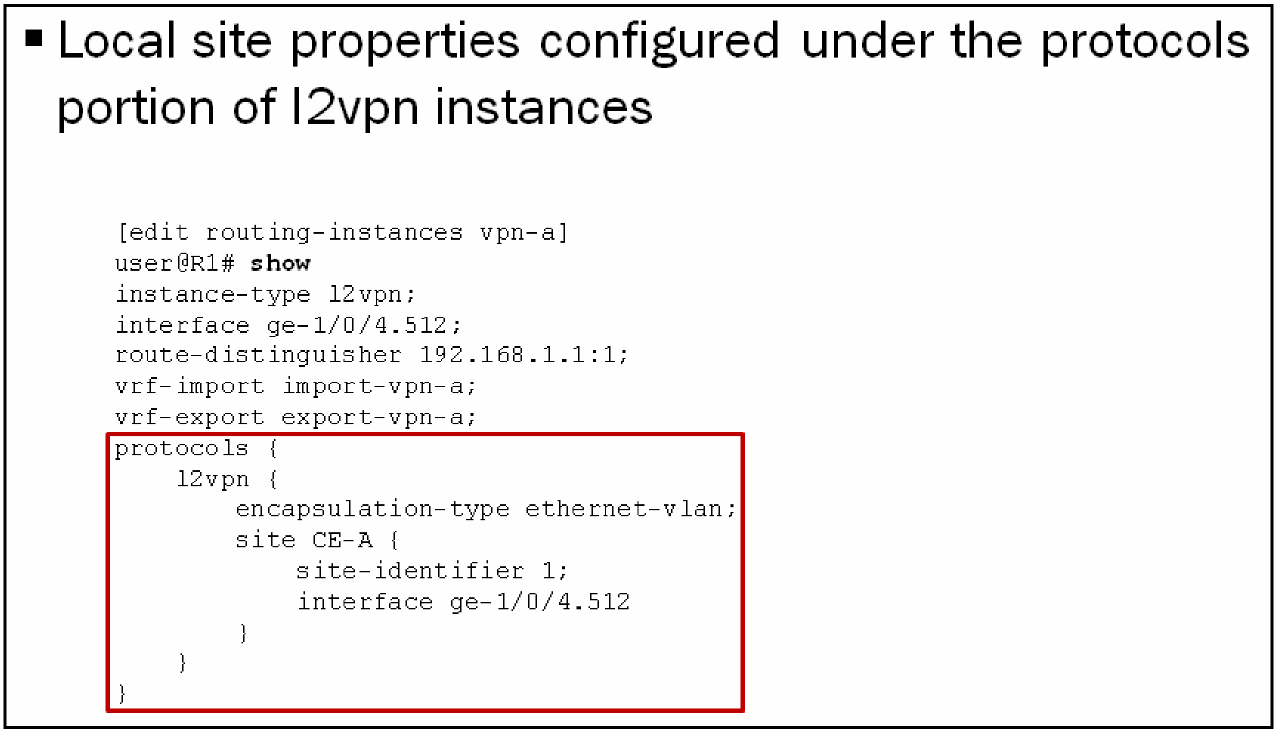
- The target tag specifies a route target extended community
  - Policy matches the route target control that the Layer 2 site information imported into a given VRF

```
[edit policy-options]
user@R1# show
...
community vpn-a members target:65512:101;
```

Layer 2 VPNs use the route target extended BGP community in the same manner as Layer 3 VPNs. The absence or presence of a particular route target in the updates received from remote PE routers causes the receiving PE router to either ignore the update (no route target matches) or install the Layer 2 VPN NLRI into one or more local VRFs (matching route target).

To create a route target community, include the **`target`** tag when defining the members of a named community at the `[edit policy-options]` portion of the hierarchy. The graphic provides an example of a Type 0-formatted route target using a 2-byte administrator field and a 4-byte assigned number value.

Local Site Properties



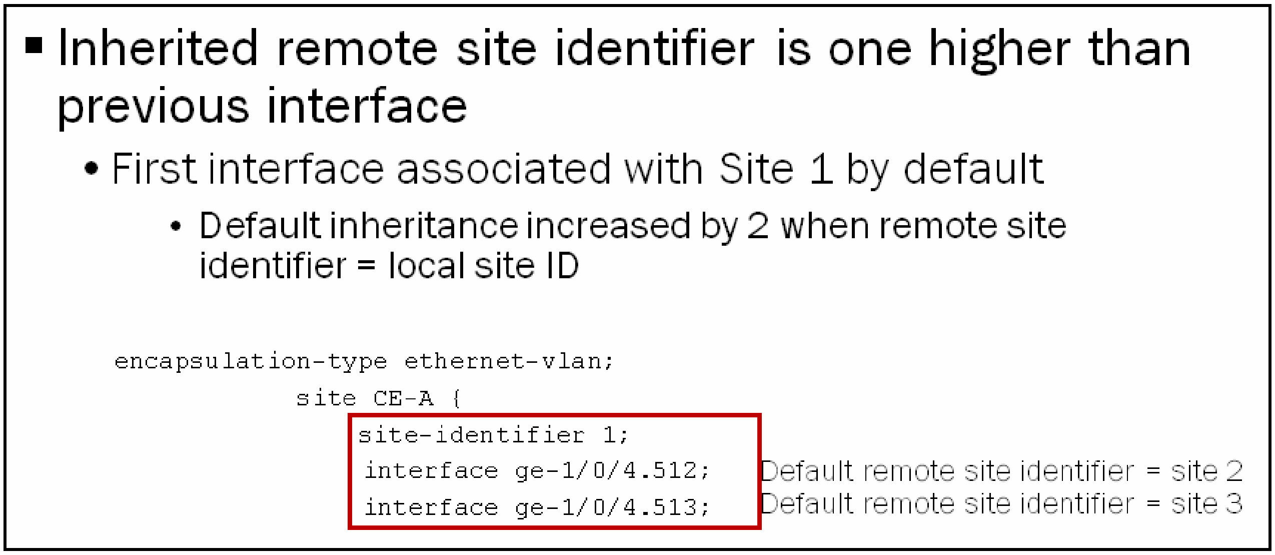
- Local site properties configured under the protocols portion of l2vpn instances

```
[edit routing-instances vpn-a]
user@R1# show
instance-type l2vpn;
interface ge-1/0/4.512;
route-distinguisher 192.168.1.1:1;
vrf-import import-vpn-a;
vrf-export export-vpn-a;
protocols {
    l2vpn {
        encapsulation-type ethernet-vlan;
        site CE-A {
            site-identifier 1;
            interface ge-1/0/4.512
        }
    }
}
```

The local site's properties are configured under the `protocols l2vpn` portion of a Layer 2 VPN routing instance. As shown in the graphic, this portion of the hierarchy specifies the following parameters:

- *Layer 2 encapsulation*: This parameter defines the type of Layer 2 technology supported by the VPN. Options include **`atm-aal5`**, **`atm-cell-port-mode`**, **`atm-cell-vc-mode`**, **`atm-cell-vp-mode`**, **`cisco-hdlc`**, **`ethernet`**, **`ethernet-vlan`**, **`frame-relay`**, **`frame-relay-port-mode`**, **`ppp`**, and **`interworking`**. All sites that are part of the same VPN must use the same encapsulation. All sites must use the **`interworking`** encapsulation type when interconnecting dissimilar Layer 2 technologies to create an IP-only Layer 2.5 VPN.
- *Site*: The site identifier is configured as an argument to the **`site`** statement. The site identifier must be unique among all sites making up a Layer 2 VPN as the site ID is used when PE routers compute the label values for site-to-site communications. The Junos OS does not support a site ID of 0; site identifiers normally are assigned contiguously starting with site ID 1.
- *Interfaces*: The local sites' logical interfaces are listed again under the `protocols l2vpn` hierarchy. The order in which the interfaces are listed has significance in that the first interface listed normally is associated with site ID 1, and so on. If wanted, you can use the **`remote-site-id`** option to alter the default interface to remote site association rules.

Remote Site Inheritance



- Inherited remote site identifier is one higher than previous interface
  - First interface associated with Site 1 by default
    - Default inheritance increased by 2 when remote site identifier = local site ID

```
encapsulation-type ethernet-vlan;
            site CE-A {
                site-identifier 1;
                interface ge-1/0/4.512;   Default remote site identifier = site 2
                interface ge-1/0/4.513;   Default remote site identifier = site 3
```

Each interface listed under the `l2vpn` portion of a Layer 2 VPN VRF is associated with a remote site. Each subsequent interface inherits by default a site association that is one higher than the previous interface. The default inheritance value is increased by two when an interface's default inheritance would cause it to be associated with the PE router's local site identifier.

The first interface listed is therefore associated with Site 1 on all PE routers except the PE router actually attaching to Site 1, as this PE router associates by default the first interface listed with Site 2.

In the example on the graphic, it shows a portion of the Layer 2 VPN configuration from Site CE-A. The default site association rules cause the two interfaces listed to be associated with Sites 2 and 3 respectively. This association is the result of the first interface having two added to the default inheritance to avoid the interface being associated with the local site. Without this algorithm, the operation would either need to begin the interface listing with a place holder interface or use the remote-site-id statement. The default site association algorithm makes these steps unnecessary.

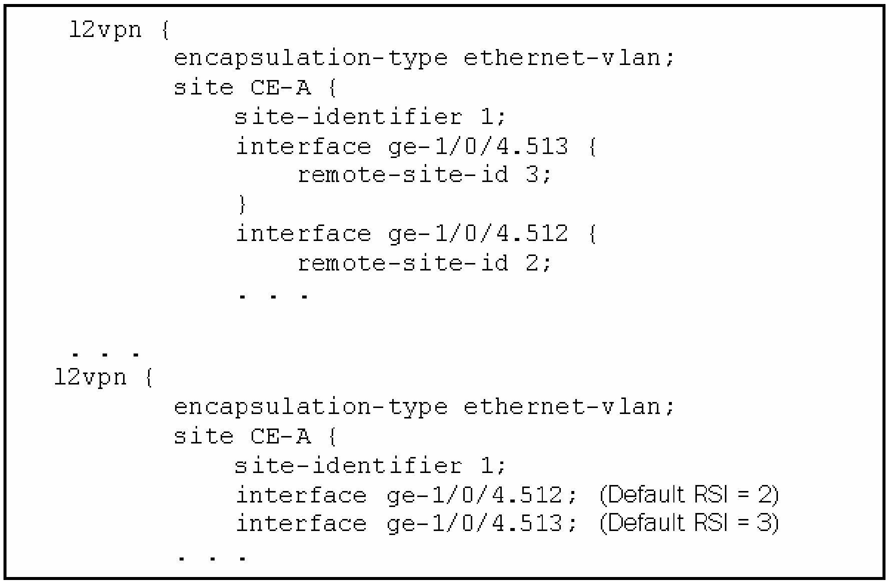
The Configuration

This graphic provides an example of how you can use the `remote-site-id` option to alter the default site association of an interface. In the configuration snippet shown, two interfaces are configured under Site CE-A's local properties. The specification of a `remote-site-id` caused the first interface listed to be associated with Site 3, and the second interface with Site 2.

Both Configurations Produce Identical Connectivity

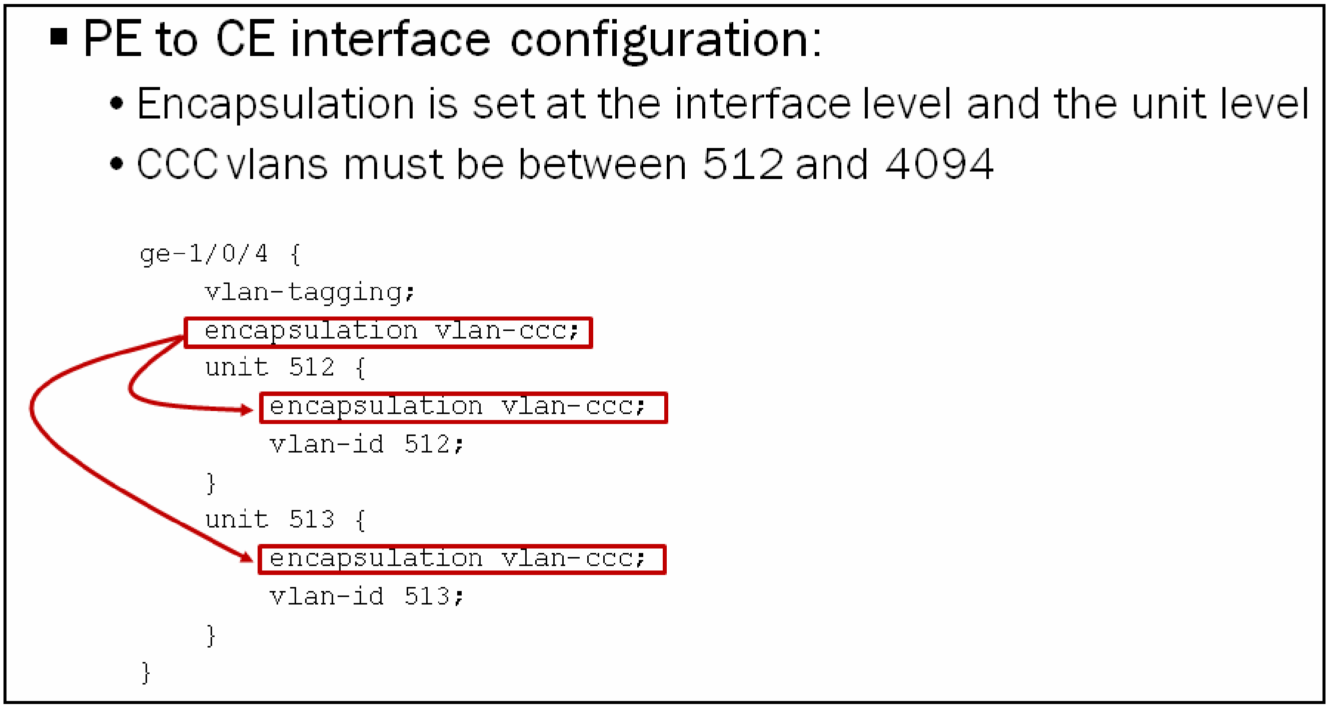
This graphic provides another configuration example without the use of `remote-site-id`. In the example at the bottom, the `ge-1/0/4.512` and `ge-1/0/4.513` interfaces are listed in numeric order so that the default site association rules correctly associate them with Sites 2 and 3.

You can avoid remote site identifier specification by carefully ordering the list of interfaces associated with the local site. Only VPNs with sparse connectivity should require the manual specification of the remote site identifier



```
 l2vpn {
        encapsulation-type ethernet-vlan;
        site CE-A {
            site-identifier 1;
            interface ge-1/0/4.513 {
                remote-site-id 3;
            }
            interface ge-1/0/4.512 {
                remote-site-id 2;
            . . .

 . . .
l2vpn {
        encapsulation-type ethernet-vlan;
        site CE-A {
            site-identifier 1;
            interface ge-1/0/4.512;  (Default RSI = 2)
            interface ge-1/0/4.513;  (Default RSI = 3)
        . . .
```

Interface Configuration Example.



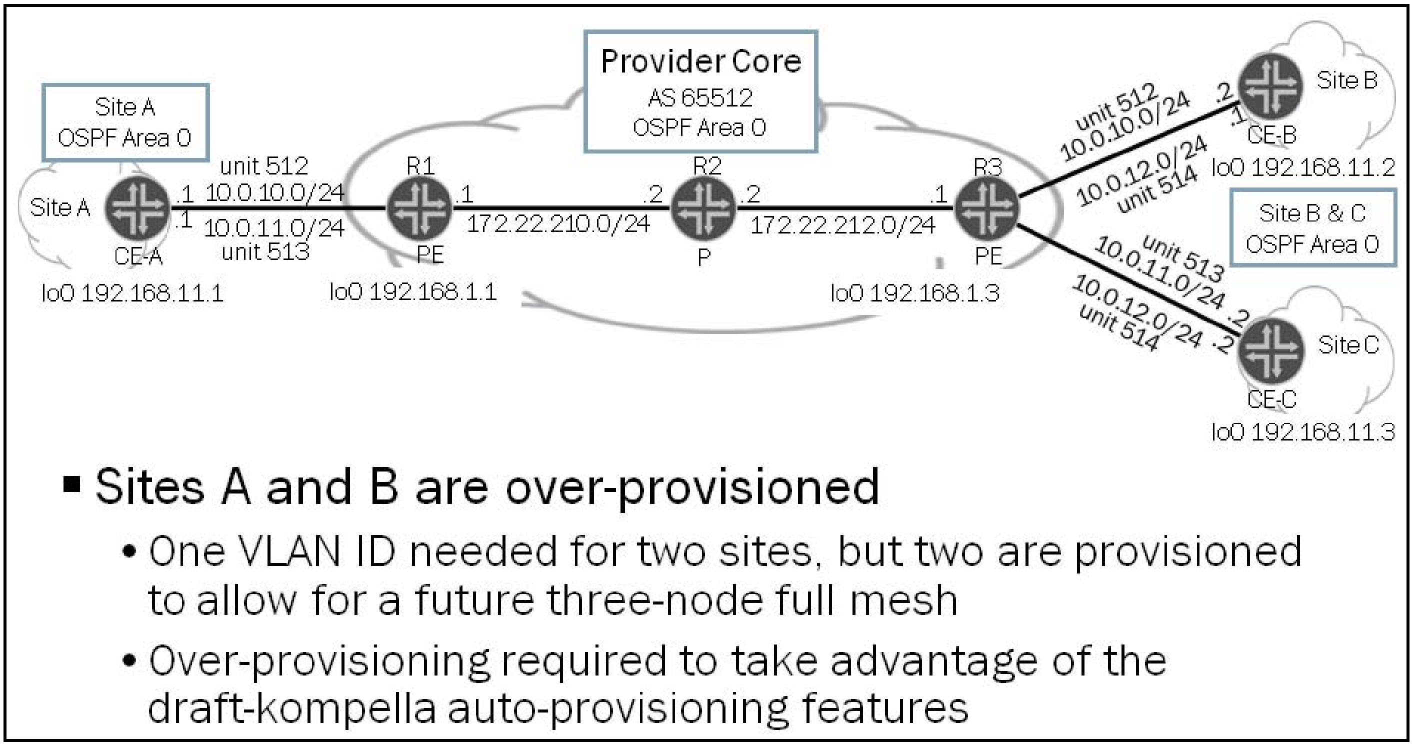
- PE to CE interface configuration:
  - Encapsulation is set at the interface level and the unit level
  - CCC vlans must be between 512 and 4094

```
ge-1/0/4 {
    vlan-tagging;
    encapsulation vlan-ccc;
    unit 512 {
        encapsulation vlan-ccc;
        vlan-id 512;
    }
    unit 513 {
        encapsulation vlan-ccc;
        vlan-id 513;
    }
}
```

This graphic provides an example of a Gigabit Ethernet interface configurations for use with CCC and Layer 2 VPNs.

VLAN tagging is mandatory, and you must specify the use of CCC encapsulation at both the device and logical unit levels. When you enable CCC encapsulation, VLAN IDs from 512 to 4094 are reserved for CCC encapsulation. You can configure VLAN IDs 0 to 511 as normal VLAN-tagged interfaces, if desired.

Site 1 and 2 Are Over-Provisioned



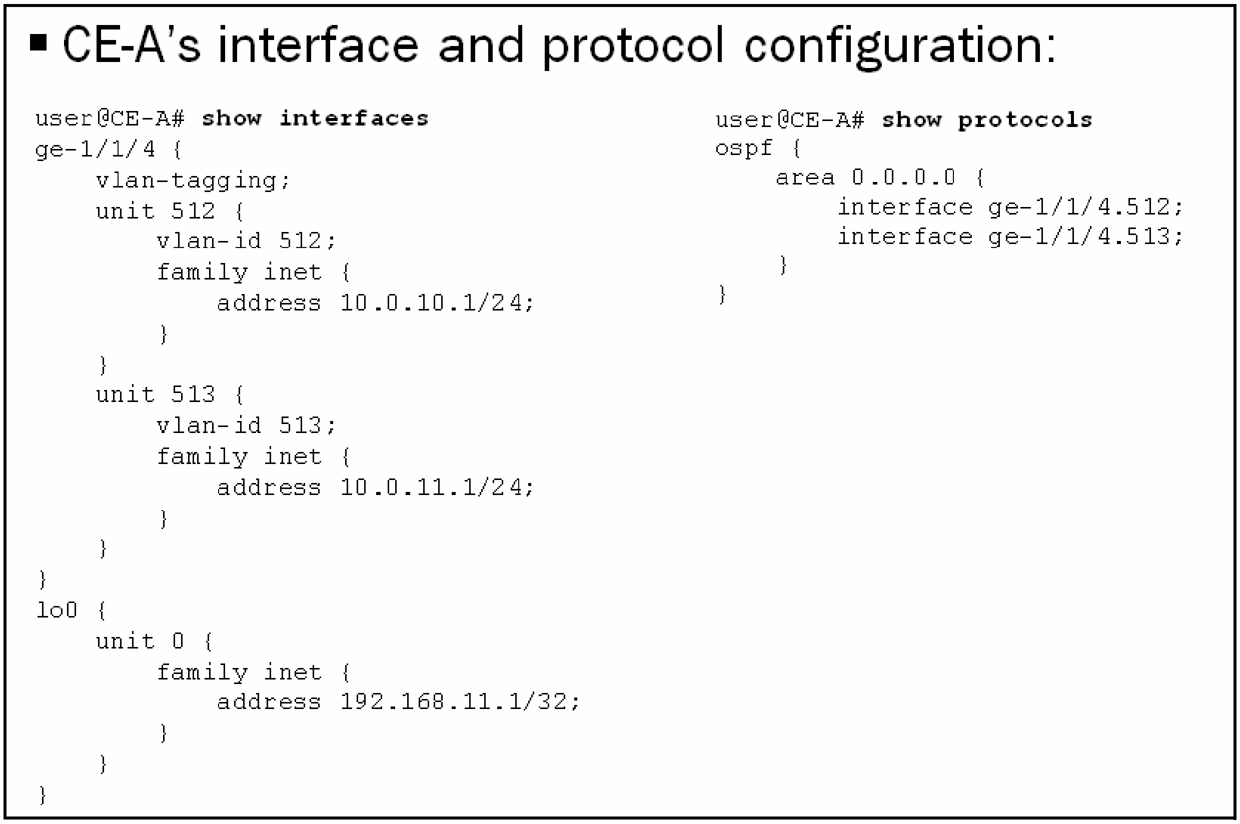
- Sites A and B are over-provisioned
  - One VLAN ID needed for two sites, but two are provisioned to allow for a future three-node full mesh
  - Over-provisioning required to take advantage of the draft-kompella auto-provisioning features

This example demonstrates how the over-provisioning of a Layer 2 VPN allows for easy expansion of the VPN when sites are added later. In this example, two logical interfaces are provisioned at both Sites A and B. Because only one logical interface is needed for connectivity to the remote site, the extra interface configured at each site represents over-provisioning.

Adding a Third Site

With the over-provisioning shown, the addition of a third site only requires modifications to the PE router attaching to the new site. Hence, the R1 PE router requires no modifications in this example.

CE-A’s Configuration



- CE-A’s interface and protocol configuration:

```
user@CE-A# show interfaces
ge-1/1/4 {
    vlan-tagging;
    unit 512 {
        vlan-id 512;
        family inet {
            address 10.0.10.1/24;
        }
    }
    unit 513 {
        vlan-id 513;
        family inet {
            address 10.0.11.1/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 192.168.11.1/32;
        }
    }
}
```

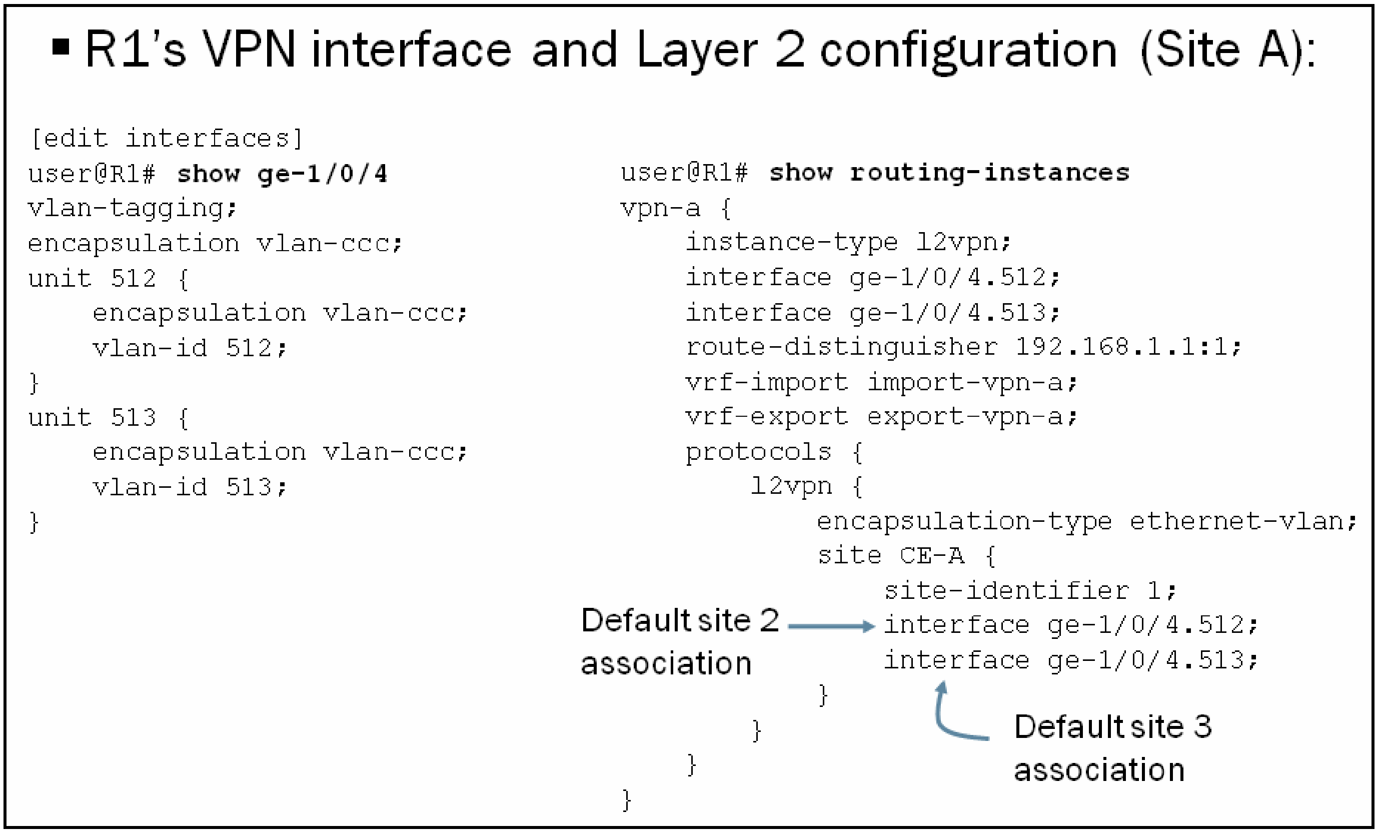
```
user@CE-A# show protocols
ospf {
    area 0.0.0.0 {
        interface ge-1/1/4.512;
        interface ge-1/1/4.513;
    }
}
```

This graphic shows the relevant portions of CE-A’s configuration. The CPE device also must be over-provisioned to avoid modifications when the VPN is expanded.

The CE device has two VLAN-tagged interfaces and IP addresses configured. Because the VLAN ID cannot change across a CCC connection, Unit 512 of the `ge-1/1/4` interface is assigned VLAN ID 512, and this interface must connect to Site B’s `ge-1/0/4.512` interface, which is configured with the same VLAN ID value. Also, the logical IP subnet in CE-A’s `ge-1/1/4.512` interface must be compatible with the subnet configured on CE-B’s `ge-1/0/4.512` interface.

This example also shows the OSPF routing protocol configured to run on both CE-A’s interfaces. Therefore, CE-A ultimately should form adjacencies with both CE-B and CE-C.

R1 Router’s Interface and Layer 2 Configuration



- R1’s VPN interface and Layer 2 configuration (Site A):

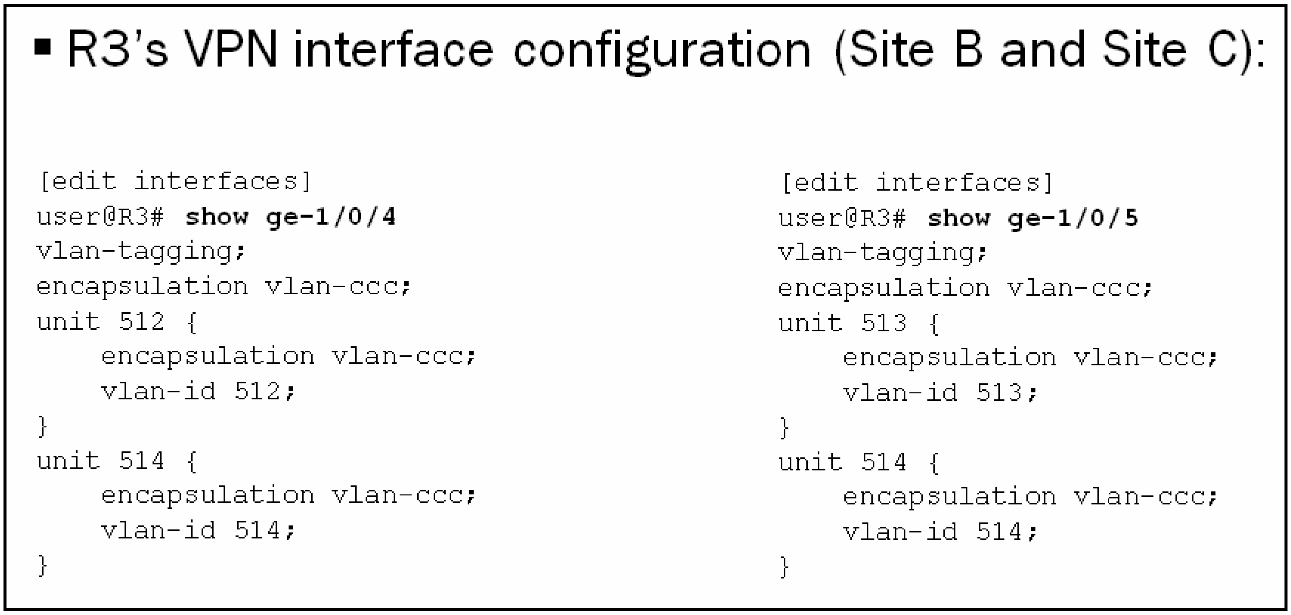
```
[edit interfaces]
user@R1# show ge-1/0/4
vlan-tagging;
encapsulation vlan-ccc;
unit 512 {
    encapsulation vlan-ccc;
    vlan-id 512;
}
unit 513 {
    encapsulation vlan-ccc;
    vlan-id 513;
}
```

```
user@R1# show routing-instances
vpn-a {
    instance-type l2vpn;
    interface ge-1/0/4.512;
    interface ge-1/0/4.513;
    route-distinguisher 192.168.1.1:1;
    vrf-import import-vpn-a;
    vrf-export export-vpn-a;
    protocols {
        l2vpn {
            encapsulation-type ethernet-vlan;
            site CE-A {
                site-identifier 1;
                interface ge-1/0/4.512;
                interface ge-1/0/4.513;
            }
        }
    }
}
```

This graphic shows the relevant portions of the R1 PE router’s configuration. The PE-CE VRF interface is configured with compatible VLAN tagging values and is configured to support `vlan-ccc` encapsulation.

The R1 router’s routing instance lists both of the VRF interfaces under the *vpn-a* instance, and again under the `protocols l2vpn` portion of the configuration. As shown on the graphic, the default site association rules cause the `ge-1/0/4.512` interface to be associated with Site 2, while the `ge-1/0/4.513` interface is mapped to Site C. Because CE-A uses the VLAN tag of 512 for the interface that is compatibly configured for connectivity to Site B, these interface-to-site identifier mappings are correct. Where needed, you can alter the interface order or use `remote-site-id` to ensure that the local site’s interfaces are associated correctly with remote sites.

R3 Interface Configuration



- R3’s VPN interface configuration (Site B and Site C):

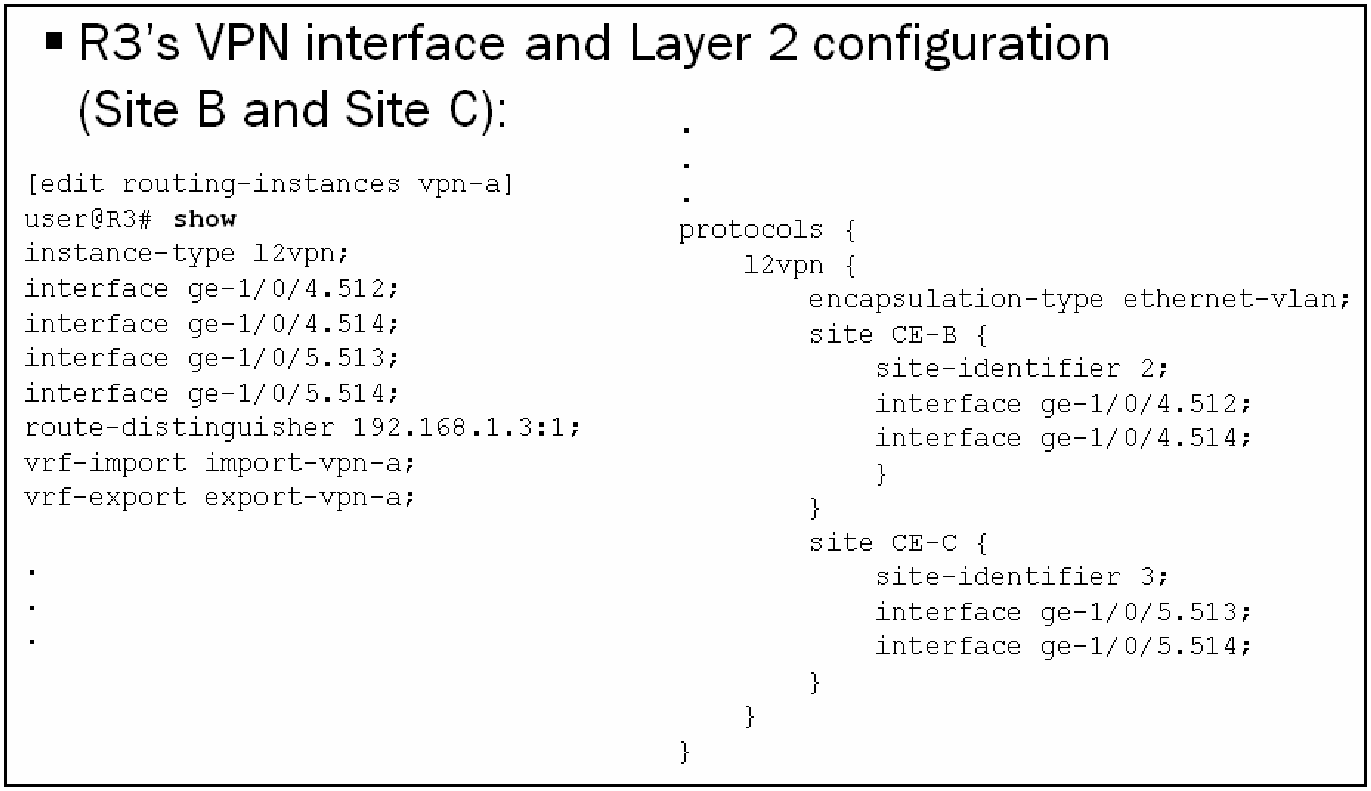
```
[edit interfaces]
user@R3# show ge-1/0/4
vlan-tagging;
encapsulation vlan-ccc;
unit 512 {
    encapsulation vlan-ccc;
    vlan-id 512;
}
unit 514 {
    encapsulation vlan-ccc;
    vlan-id 514;
}
```

```
[edit interfaces]
user@R3# show ge-1/0/5
vlan-tagging;
encapsulation vlan-ccc;
unit 513 {
    encapsulation vlan-ccc;
    vlan-id 513;
}
unit 514 {
    encapsulation vlan-ccc;
    vlan-id 514;
}
```

This graphic shows the configuration of the R3 PE router’s interfaces. The VLAN tagging provisioned on the R3 router is compatible with the tag values used at Site A.

VLAN ID 514 is allocated for the connection between Sites B and C. Because both sites are attached to the R3 PE router, this VLAN tag is configured on both its VRF interfaces. These VLAN assignments make it critical that the R3 router’s ge-1/0/4.514 logical interface is associated with Site B. The next section displays how we accomplish this association.

R3 VPN Configuration



- R3’s VPN interface and Layer 2 configuration (Site B and Site C):

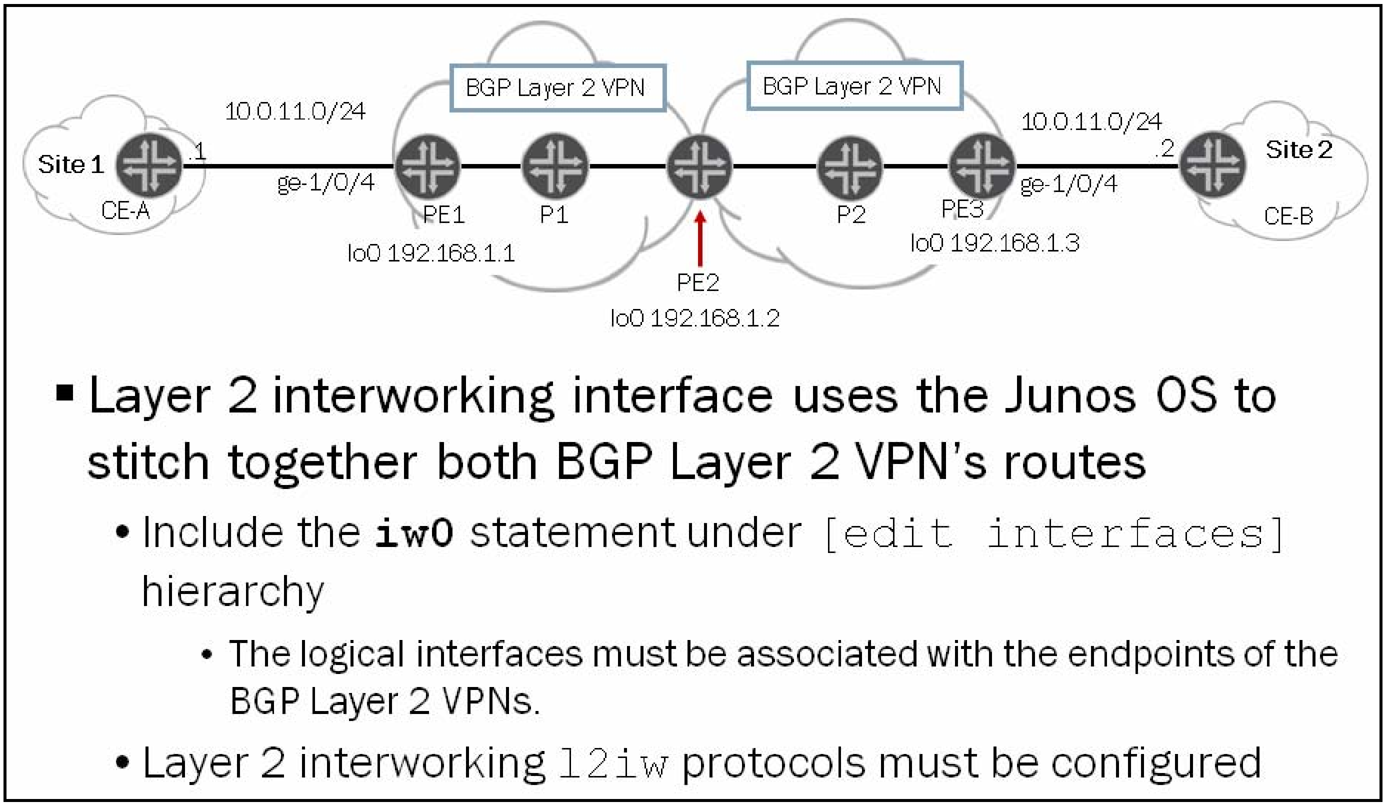
```
[edit routing-instances vpn-a]
user@R3# show
instance-type l2vpn;
interface ge-1/0/4.512;
interface ge-1/0/4.514;
interface ge-1/0/5.513;
interface ge-1/0/5.514;
route-distinguisher 192.168.1.3:1;
vrf-import import-vpn-a;
vrf-export export-vpn-a;

.
.
.
```

```
.
.
.
protocols {
    l2vpn {
        encapsulation-type ethernet-vlan;
        site CE-B {
            site-identifier 2;
            interface ge-1/0/4.512;
            interface ge-1/0/4.514;
            }
        }
        site CE-C {
            site-identifier 3;
            interface ge-1/0/5.513;
            interface ge-1/0/5.514;
        }
    }
}
```

This graphic shows the Layer 2 VPN configuration on the R3 PE router. The Layer 2 instance lists all of the VRF interfaces. Two sites are defined under the `protocols l2vpn` portion of the configuration.

The default site association rules are in use in this example, which require careful ordering of the logical interfaces listed under Site CE-C to avoid the need for explicit declaration of a remote site identifier. If ge-1/0/4.514 were to be listed before ge-1/0/4.513, the use of `remote-site-id` would be required to achieve the correct association between logical interfaces and remote sites.

Layer 2 Interworking



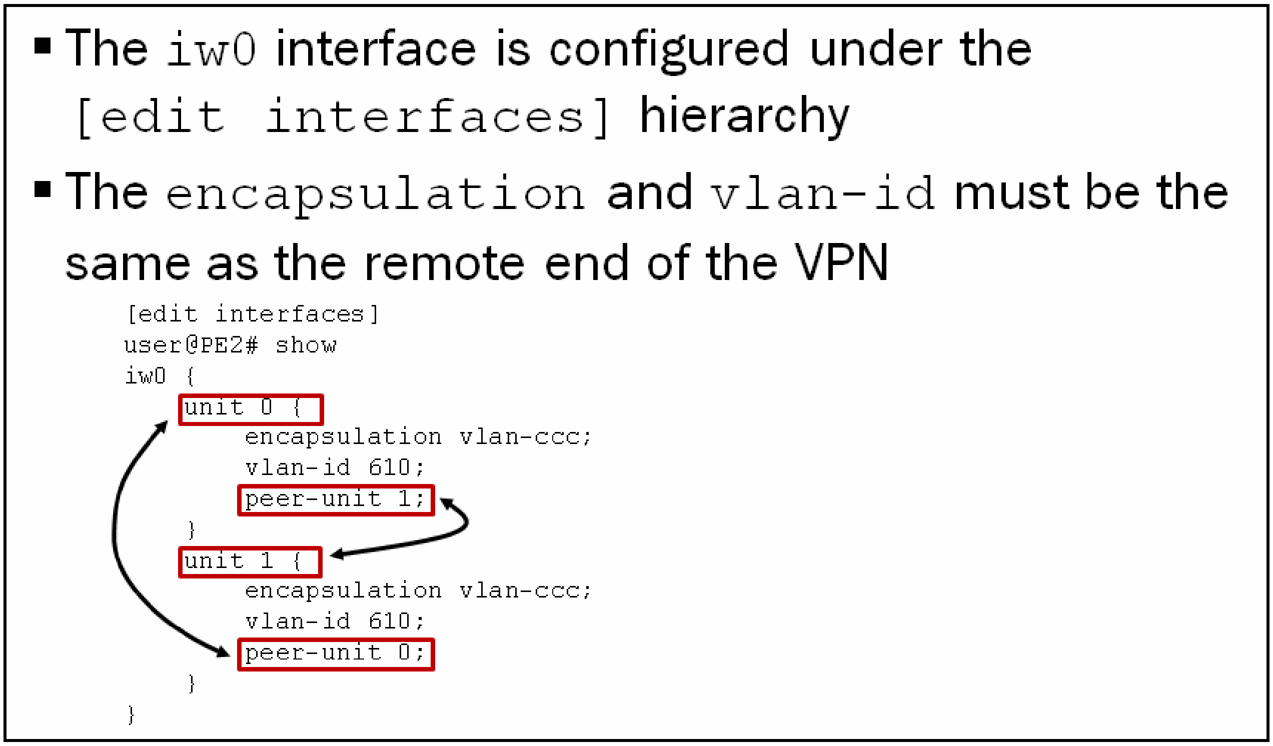
- Layer 2 interworking interface uses the Junos OS to stitch together both BGP Layer 2 VPN's routes
  - Include the **`iw0`** statement under `[edit interfaces]` hierarchy
    - The logical interfaces must be associated with the endpoints of the BGP Layer 2 VPNs.
  - Layer 2 interworking `l2iw` protocols must be configured

As the need to link different Layer 2 services to one another for expanded service offerings grows, Layer 2 MPLS VPN services are increasingly in demand. The next few graphics provide configuration for terminating a Layer 2 VPN into another Layer 2 VPN using the Layer 2 interworking (iw0) interface. Existing Junos OS functionality makes use of a tunnel PIC to loop packets out and back from the Packet Forwarding Engine (PFE), to link together Layer 2 networks. The Layer 2 interworking software interface avoids the need for the Tunnel Services PIC and overcomes the limitation of bandwidth constraints imposed by the Tunnel Services PIC.

The `iw0` statement is configured at the `[edit interfaces]` hierarchy level. This configuration is similar to the configuration for a logical tunnel interface. The logical Interfaces must be associated with the endpoints of both BGP Layer 2 VPNs.

In addition to configuring the interfaces and associating them with the BGP Layer 2 VPNs, the Layer 2 interworking `l2iw` protocol must be configured. Without the `l2iw` configuration, the `l2iw` routes will not be formed, regardless of whether any `iw` interfaces are present. Within the `l2iw` protocols, only trace options can be configured in the standard fashion.

The `iw0` Interface Configuration

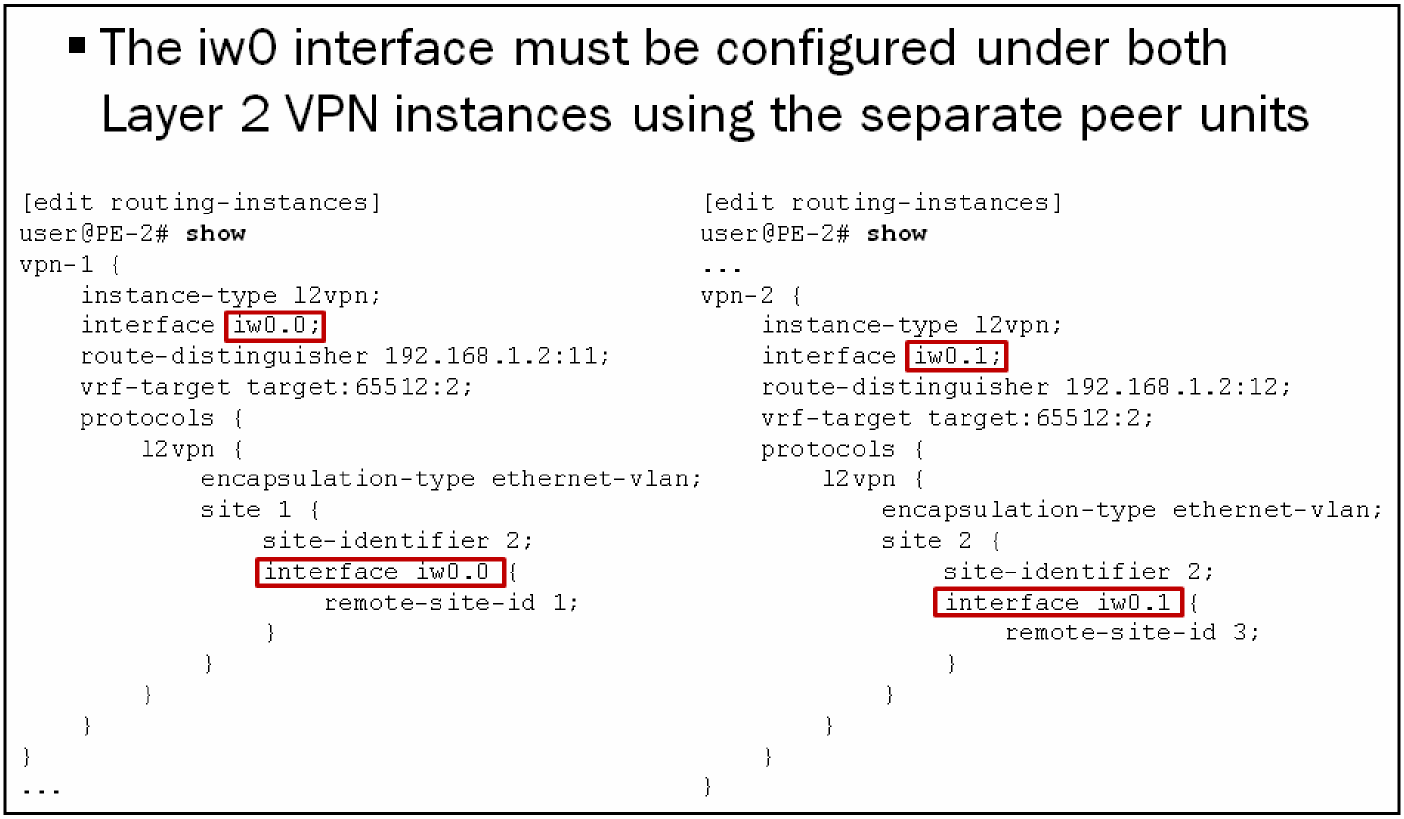


- The `iw0` interface is configured under the `[edit interfaces]` hierarchy
- The `encapsulation` and `vlan-id` must be the same as the remote end of the VPN

```
[edit interfaces]
user@PE2# show
iw0 {
    unit 0 {
        encapsulation vlan-ccc;
        vlan-id 610;
        peer-unit 1;
    }
    unit 1 {
        encapsulation vlan-ccc;
        vlan-id 610;
        peer-unit 0;
    }
}
```

The graphic illustrates a basic `iw0` interface configuration. As indicated on the graphic you need to configure two logical units. The same encapsulation and vlan-id must be configured on the `iw0` units as is configured on the PE to CE interfaces. As pointed out on the graphic, a `peer-unit` must be specified for each unit. This statement associates two units together so that traffic can be stitched between the two Layer 2 VPNs.

BGP Layer 2 VPN Configurations



- The iw0 interface must be configured under both Layer 2 VPN instances using the separate peer units

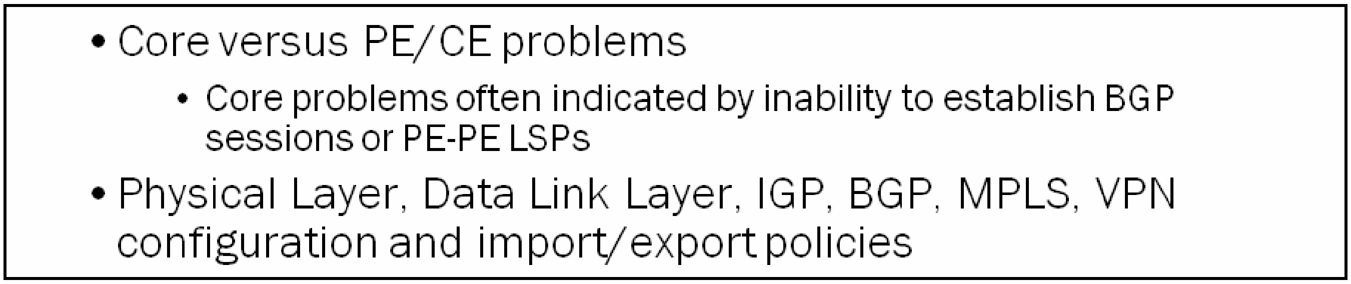
```
[edit routing-instances]
user@PE-2# show
vpn-1 {
    instance-type l2vpn;
    interface iw0.0;
    route-distinguisher 192.168.1.2:11;
    vrf-target target:65512:2;
    protocols {
        l2vpn {
            encapsulation-type ethernet-vlan;
            site 1 {
                site-identifier 2;
                interface iw0.0 {
                    remote-site-id 1;
                }
            }
        }
    }
}
...
```

```
[edit routing-instances]
user@PE-2# show
...
vpn-2 {
    instance-type l2vpn;
    interface iw0.1;
    route-distinguisher 192.168.1.2:12;
    vrf-target target:65512:2;
    protocols {
        l2vpn {
            encapsulation-type ethernet-vlan;
            site 2 {
                site-identifier 2;
                interface iw0.1 {
                    remote-site-id 3;
                }
            }
        }
    }
}
```

The iw0 interface is configured as the CE facing interface for each BGP Layer 2 VPN instance. To configure the Layer 2 VPN protocol, including the **`l2vpn`** statement at the `[edit routing-instances routing-instances-name protocols]` hierarchy level. To configure the `iw0` interface, include the interfaces statement and specify `iw0` as the interface name. In the example provided, the `iw0.0` interface is configured under the Layer 2 VPN protocols for vpn-1 to receive the looped packet from the interface `iw0.1`, which is configured for vpn-2.

In addition to the `iw0` interface configuration, Layer 2 interworking `l2iw` protocols must be configured (not displayed on the graphic). Without the `l2iw` configuration, the `l2iw` routes are not formed, regardless of whether any `iw` interfaces are present.

Take a Layered Approach

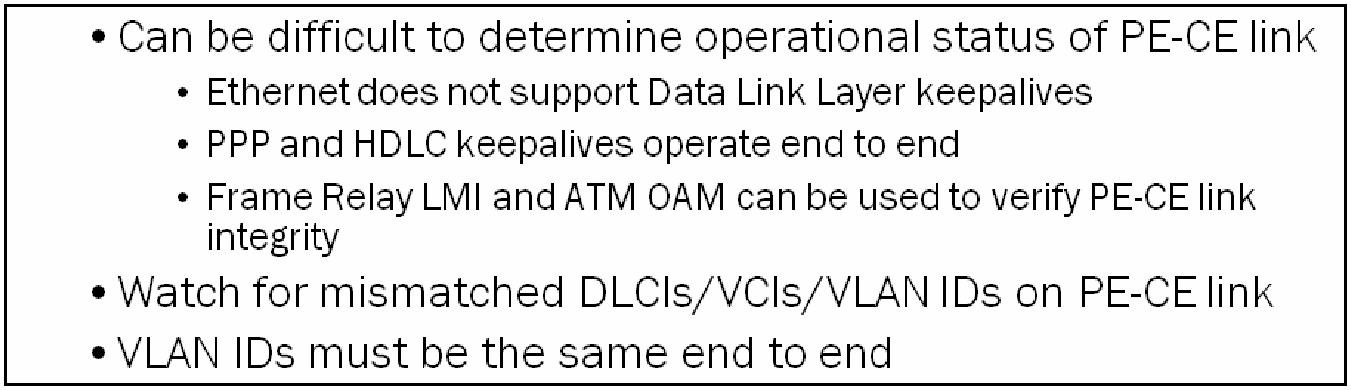


- Core versus PE/CE problems
  - Core problems often indicated by inability to establish BGP sessions or PE-PE LSPs
- Physical Layer, Data Link Layer, IGP, BGP, MPLS, VPN configuration and import/export policies

Any number of configuration and operational problems can result in a dysfunctional VPN. With this much complexity, we encourage you to take a layered approach to the provisioning and troubleshooting of Layer 2 VPN services.

Is the problem core or PE-CE related? and *Are my pings failing because an interface is down, or because a constrained path LSP cannot be established?* are the types of questions that await all who venture here. Fortunately, Layer 2 VPNs have several natural boundaries that allow for expedient problem isolation. As an example, consider a call reporting that three different VPNs on two different PE routers are down. Here, you look for core-related issues (the P routers are common to all VPNs) rather than looking for PE-CE-related problems at the sites reporting problems.

PE-CE Ping Testing No Longer Possible



- Can be difficult to determine operational status of PE-CE link
  - Ethernet does not support Data Link Layer keepalives
  - PPP and HDLC keepalives operate end to end
  - Frame Relay LMI and ATM OAM can be used to verify PE-CE link integrity
- Watch for mismatched DLCIs/VCIs/VLAN IDs on PE-CE link
- VLAN IDs must be the same end to end

Layer 2 VPN troubleshooting differs from Layer 3 VPN troubleshooting in many ways. A significant difference is that the PE router and CE devices do not share IP connectivity, which makes the testing of the local PE-CE VRF link difficult. In some cases you can determine the operational status of the PE-CE link by verifying the correct operation of the data link layer's keepalive function. In the case of PPP or Cisco's HDLC, the keepalive's operation is end to end between CE routers, however.

Mismatches between the connection identifiers configured on the PE-CE link are common sources of problems. VLAN ID must be the same end to end. Sometimes you can provision an out-of-band management interface that permits ping testing and Telnet access to the local CE device. This interface should be another logical unit on the interface also providing Layer 2 VPN connectivity.

Core IGP

A functional core IGP is critical to the operation of LSP signaling protocols and the PE-PE MP-BGP sessions. You always should check the IGP when LSP or BGP session problems are evident. Generally, you verify IGP operation by enforcing such tasks as looking at routing tables and neighbor states (adjacencies) and conducting ping and traceroute testing.

PE-PE IBGP Sessions

Each PE router must have an MP-BGP session established to all other PE routers connecting to sites that form a single VPN. If route reflectors are in use, all PE routers must have sessions established to all route reflectors serving the VPNs for which they have attached members. You must enable the `l2-vpn` family on these sessions.

LSPs

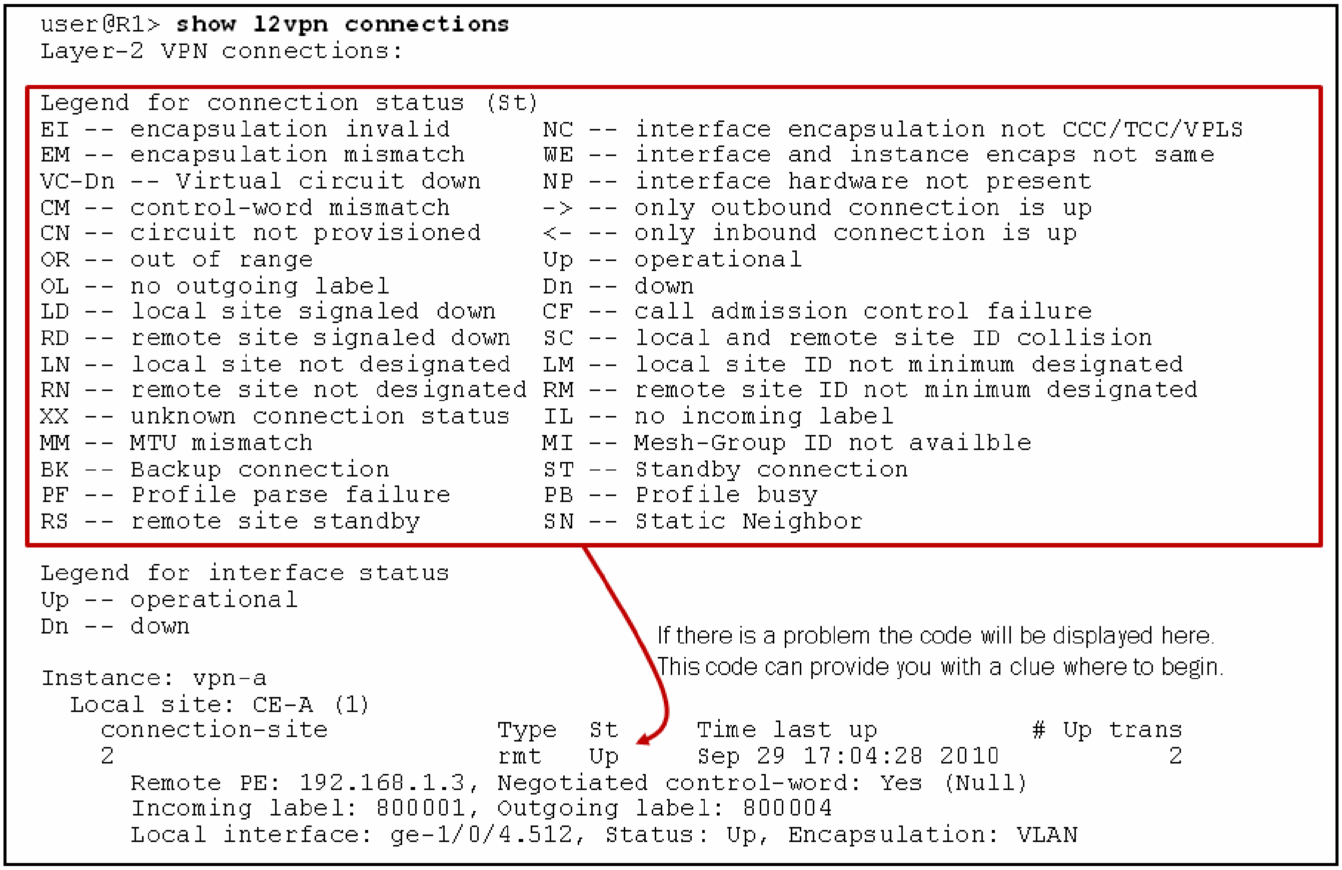
Each pair of PE routers sharing VPN membership must have LSPs established in both directions before traffic can be forwarded over the VPN. Lack of LSPs results in the Layer 2 VPN NLRIs being hidden. When route reflection is in use, LSPs should be established from the route reflector to each PE router that is a client to ensure that hidden routes do not cause failure of the reflection process.

Hidden Routes?

Although sometimes the results of normal BGP route filtering, hidden routes in the context of VPNs generally indicate a problem in the prefix-to-LSP resolution process. VPN routes must resolve to an LSP in either the `inet.3` or `inet.0` routing table, which egresses at the advertising PE router.

While the Junos OS normally keeps all loop-free BGP routes that are received (although kept, they might be hidden), this is not the case with Layer 2 VPN NLRI updates. A PE router receiving VPN updates with no matching route targets acts as if the update never happened. A change in VRF policy triggers BGP route refresh, and the routes appear. When stumped, you can enable the `keep all` option to force the PE router to retain all BGP NLRI updates received. Once you perform fault isolation, you should turn off this option to prevent excessive resource use on the PE router.

Sample **show l2vpn connections** Output



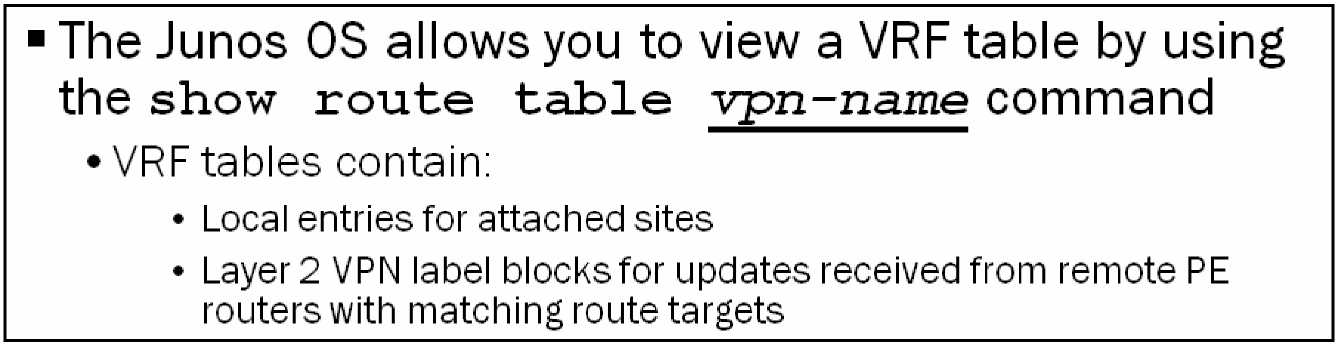
```
user@R1> show l2vpn connections
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not availble
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor

Legend for interface status
Up -- operational
Dn -- down

Instance: vpn-a
  Local site: CE-A (1)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Sep 29 17:04:28 2010           2
      Remote PE: 192.168.1.3, Negotiated control-word: Yes (Null)
      Incoming label: 800001, Outgoing label: 800004
      Local interface: ge-1/0/4.512, Status: Up, Encapsulation: VLAN
```

If there is a problem the code will be displayed here. This code can provide you with a clue where to begin.

This graphic provides a sample of the output generated with the **show l2vpn connections** command.

The top of the display provides a legend for the connection and circuit status portion of each Layer 2 VPN connection. This legend can be very useful when troubleshooting a problem with the VPN not establishing. By identifying the fault you can narrow down where to start you investigation. You can also see the incoming and outgoing labels computed for communications with remote sites.

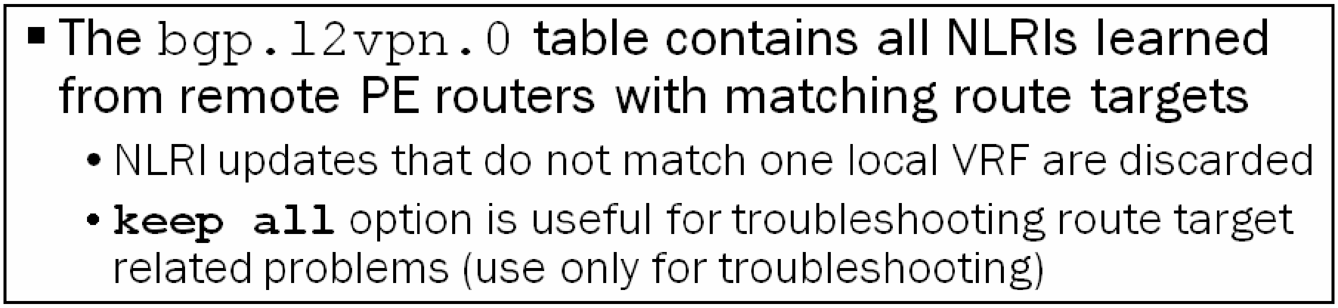
Viewing Layer 2 VPN VRFs



- The Junos OS allows you to view a VRF table by using the **show route table *vpn-name*** command
  - VRF tables contain:
    - Local entries for attached sites
    - Layer 2 VPN label blocks for updates received from remote PE routers with matching route targets

You can view the contents of a specific VRF using the **show route table *vpn-name*** operational command. This table shows configuration associated with the local site as well as Layer 2 VPN NLRIs learned from remote PE routers that have matching route targets.

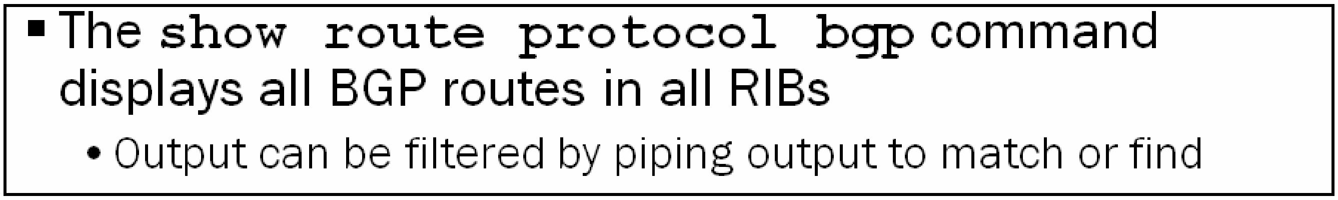
The **bgp.l2vpn.0** Table



- The `bgp.l2vpn.0` table contains all NLRIs learned from remote PE routers with matching route targets
  - NLRI updates that do not match one local VRF are discarded
  - **`keep all`** option is useful for troubleshooting route target related problems (use only for troubleshooting)

The `bgp.l2vpn.0` table houses all Layer 2 VPN NLRIs learned from remote PE routers having at least one matching route target. This table functions as a RIB-in for VPN routes matching at least one local route target. When troubleshooting route target-related problems, you should enable the `keep all` option under the BGP configuration stanza. This option places all received Layer 2 VPN NLRIs into the `bgp.l2vpn.0` table, whether or not matching route targets are present. You should not leave this option enabled in a production PE router due to the increased memory and processing requirements that can result. In normal operation, a PE router should only house Layer 2 VPN NLRIs that relate to its directly connected sites.

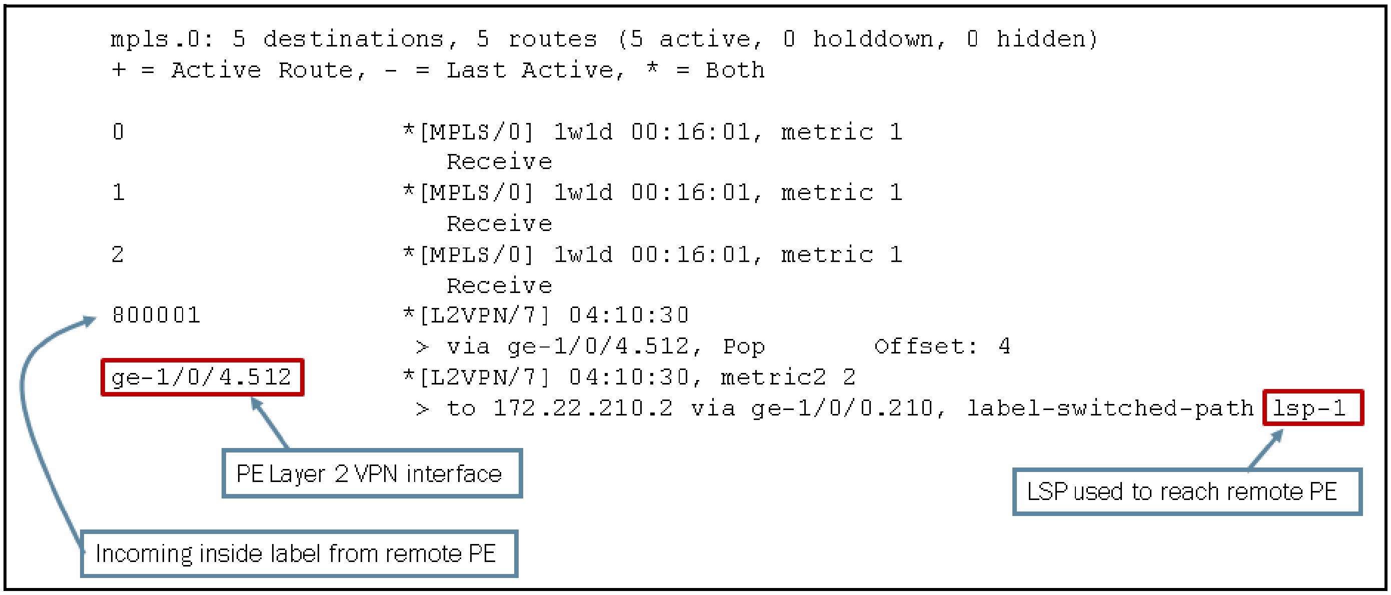
A Shortcut



- The **`show route protocol bgp`** command displays all BGP routes in all RIBs
  - Output can be filtered by piping output to match or find

By issuing a **`show route protocol bgp`** command, you can view all BGP routes, irrespective of the routing tables in which they are stored. This approach is helpful when you cannot recall the exact name of a particular VPN's routing instance. You can use the `match` or `find` arguments to this command to help filter the commands output.

Examining the PE mpls.0 Forwarding Table

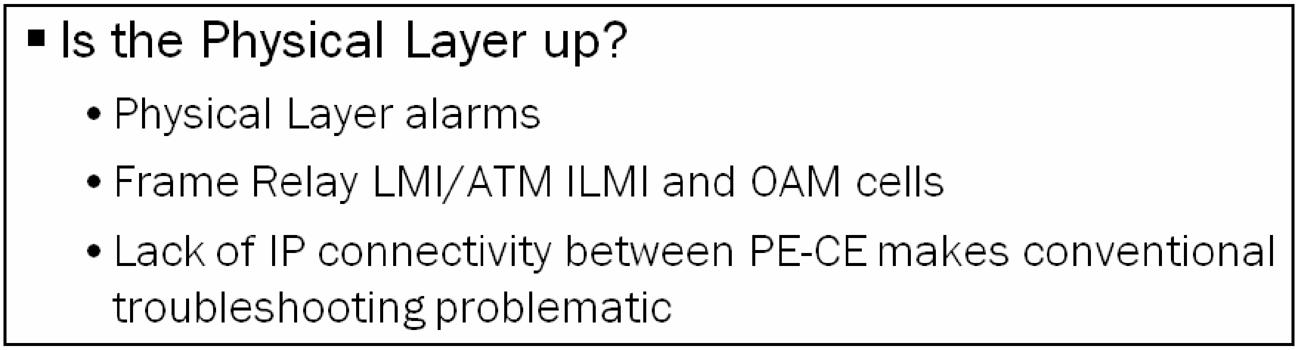


```
mpls.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 1w1d 00:16:01, metric 1
                      Receive
1                  *[MPLS/0] 1w1d 00:16:01, metric 1
                      Receive
2                  *[MPLS/0] 1w1d 00:16:01, metric 1
                      Receive
800001             *[L2VPN/7] 04:10:30
                    > via ge-1/0/4.512, Pop       Offset: 4
ge-1/0/4.512       *[L2VPN/7] 04:10:30, metric2 2
                    > to 172.22.210.2 via ge-1/0/0.210, label-switched-path lsp-1
```

PE Layer 2 VPN interface
LSP used to reach remote PE
Incoming inside label from remote PE

The PE mpls.0 forwarding table shows incoming Layer 2 VPN interface to LSP mapping and incoming inside label value from remote PE.

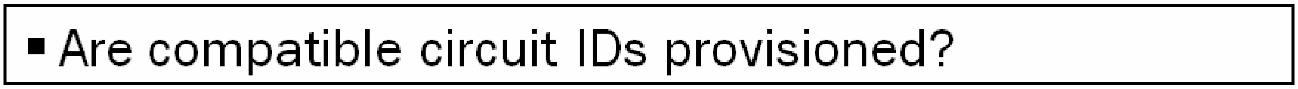
Is the PE-CE Interface Up?



- Is the Physical Layer up?
  - Physical Layer alarms
  - Frame Relay LMI/ATM ILMI and OAM cells
  - Lack of IP connectivity between PE-CE makes conventional troubleshooting problematic

The lack of inherent IP connectivity between the PE and CE routers can make PE-CE VRF interface troubleshooting problematic. Without the ability to conduct ping testing, you must rely on the absence or presence of physical layer and data-link layer alarms and status indications. For example, a loss of light (`LoL`) indication of a SONET link is a sure indication that physical layer problems are present on the PE-CE link. You can monitor ATM and Frame Relay links for proper PVC management protocol operation. In the case of ATM, you can issue **`ping atm`** to validate VC level connectivity to the attached CE device.

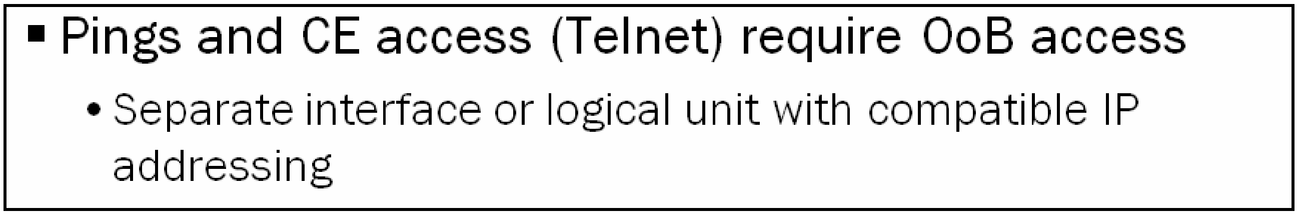
Compatible Circuit IDs



- Are compatible circuit IDs provisioned?

When the physical layer and data link layer operation of the PE-CE links appears normal, you should confirm that compatible connection identifiers are configured on the local PE-CE link. With VLAN tagging, you must ensure that the same VLAN ID values are configured on the remote PE-CE interface as well.

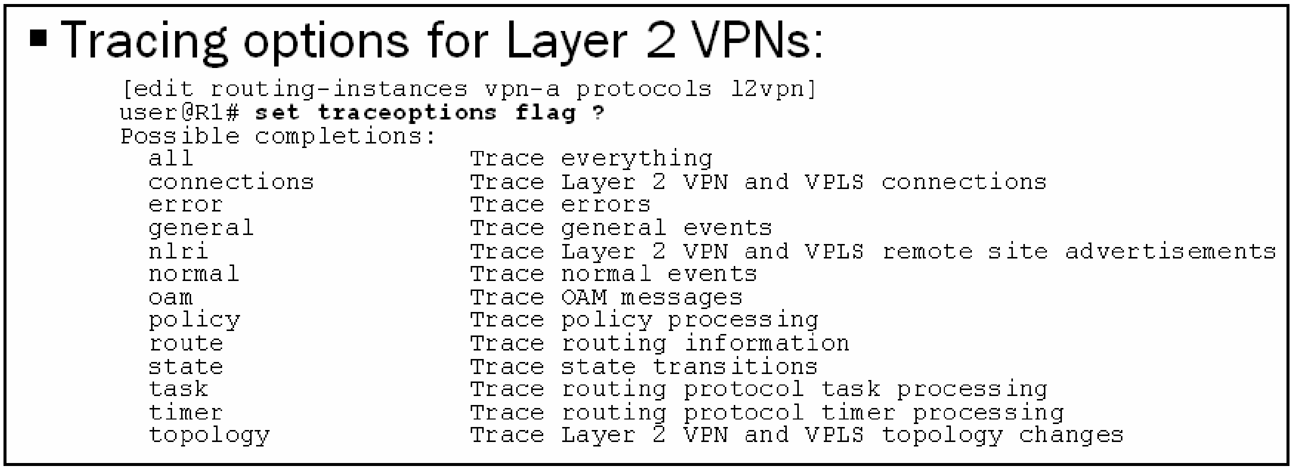
Out-of-Band Management



- Pings and CE access (Telnet) require OoB access
  - Separate interface or logical unit with compatible IP addressing

We recommend that the service provider provision a non-Layer 2 VPN connection between the PE and CE routers to simplify troubleshooting. This connection is normally just another logical unit on the existing PE-CE interface. However, it has the family `inet` and compatible IP addressing configured. You can use the resulting logical IP subnet to verify PE-CE VRF interface operation and to enable Ping, Telnet, FTP, and other such services between the PE and CE routers.

Tracing Options for Layer 2 VPNs

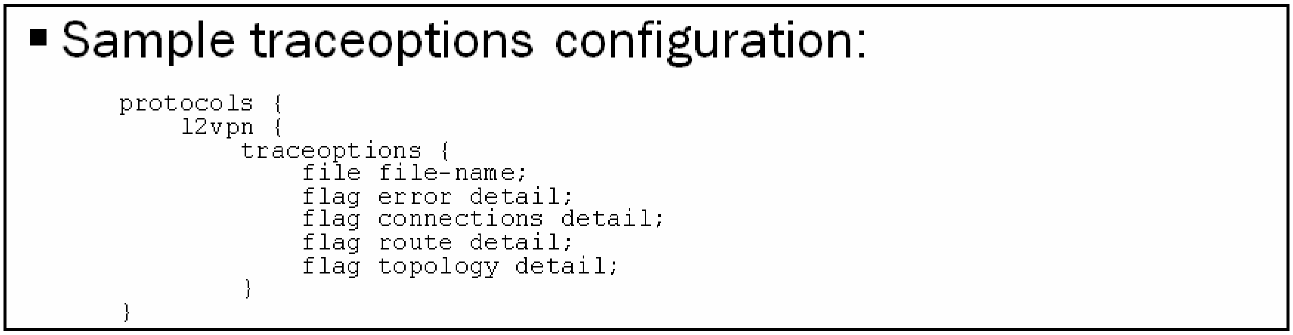


- Tracing options for Layer 2 VPNs:

```
[edit routing-instances vpn-a protocols l2vpn]
user@R1# set traceoptions flag ?
Possible completions:
  all                    Trace everything
  connections            Trace Layer 2 VPN and VPLS connections
  error                  Trace errors
  general                Trace general events
  nlri                   Trace Layer 2 VPN and VPLS remote site advertisements
  normal                 Trace normal events
  oam                    Trace OAM messages
  policy                 Trace policy processing
  route                  Trace routing information
  state                  Trace state transitions
  task                   Trace routing protocol task processing
  timer                  Trace routing protocol timer processing
  topology               Trace Layer 2 VPN and VPLS topology changes
```

You can trace a variety of Layer 2 VPN signaling and state transitions by configuring `traceoptions` under the `l2vpn` portion of a Layer 2 VPN routing instance. The graphic lists the options currently available for Layer 2 VPN tracing.

Sample Tracing Configuration



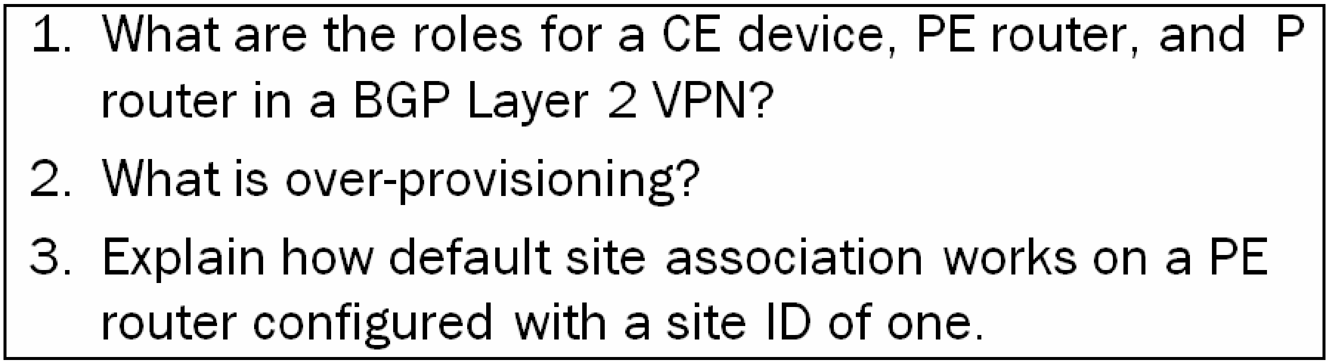
- Sample traceoptions configuration:

```
protocols {
    l2vpn {
        traceoptions {
            file file-name;
            flag error detail;
            flag connections detail;
            flag route detail;
            flag topology detail;
        }
}
```

The graphic provides a sample Layer 2 VPN tracing configuration set to write output to the named file with additional detail for each of the flags specified.

Layer 2 VPN tracing can provide invaluable assistance when troubleshooting Layer 2 VPN operational problems.

Review Questions



1. What are the roles for a CE device, PE router, and P router in a BGP Layer 2 VPN?
2. What is over-provisioning?
3. Explain how default site association works on a PE router configured with a site ID of one.

Answers to Review Questions

1.

The CE device handles all the customer's routing to the remote sites. The CE device serves as the gateway to other customer sites within their network. The PE router maintains and exchanges VPN-related information with other PE routers. The PE router also delivers Layer 2 circuits to the customer and maps these circuits to the LSPs used to connect to the remote site. The P routers forward VPN traffic transparently over established LSPs.

2.

Over-provisioning is when you configure more logical connection to a site than are needed for current site connections. This allows you to easily and quickly add additional sites to the network.

3.

On a PE router with a site ID of one, the first interface configured will be associated with the remote site ID of two. The next interface configured will be associated with three. Each additional interface configured will add one on to the previous site association.

JNCIS-SP Study Guide—Part 3

Chapter 14: Layer 2 VPN Scaling and CoS

This Chapter Discusses:

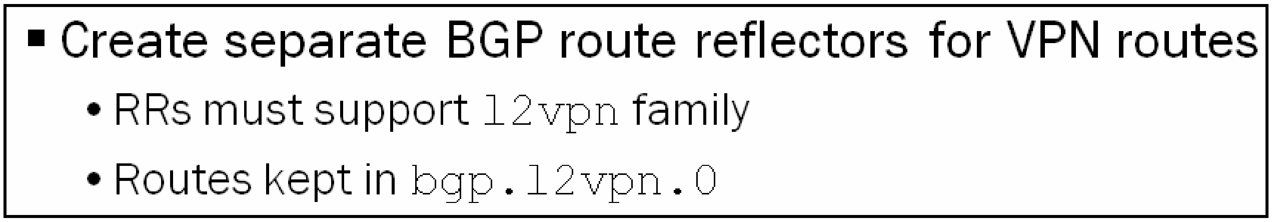
- BGP Layer 2 VPN scaling mechanisms and route reflection; and
- Junos operating system BGP Layer 2 VPN class-of-service (CoS) support.

Observe Vendor-Specific PE Router Limits

Determining how many virtual private networks (VPNs) a given provider edge (PE) router can support is a somewhat intractable question. There are many variables that come into play when factoring the VPN load on a PE router. Current Juniper Networks Layer 2 VPN scaling limits for Junos OS routers are provided on a subsequent page.

Additional PE router scaling factors include memory, processing power, limits on total numbers of labels, and limits on logical interface counts.

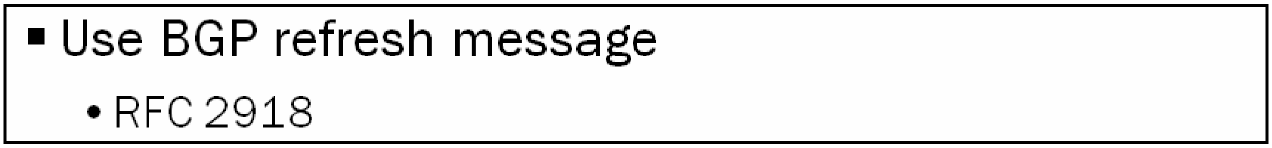
Route Reflection



- Create separate BGP route reflectors for VPN routes
  - RRs must support `l2vpn` family
  - Routes kept in `bgp.l2vpn.0`

A key aspect of the BGP Layer 2 VPN is that no single PE router has to carry all Layer 2 VPN state for the provider's network. This concept can be extended to route reflection by deploying multiple route reflectors responsible for different pieces of the total VPN customer base. Route reflection has the added advantage of minimizing the number of MP-BGP peering sessions in the provider's network, which amounts to a true *win-win* situation.

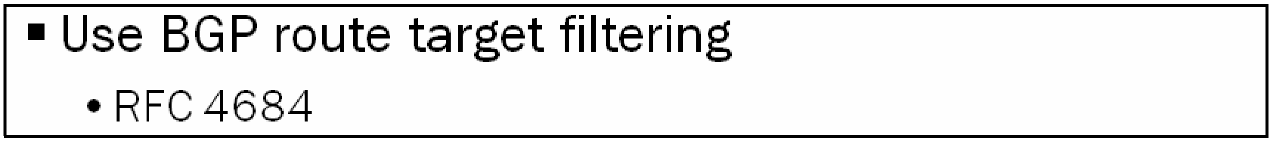
BGP Route Refresh



- Use BGP refresh message
  - RFC 2918

The use of BGP route refresh allows for nondisruptive adds, moves, and changes, which, in turn, reduces routing churn by not forcing the termination of PE-PE MP-BGP sessions when changes are made to the VPN topology or membership.

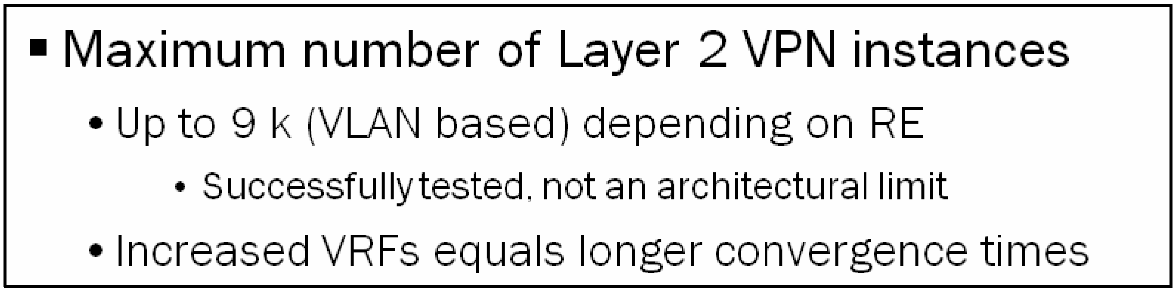
Outbound Route Filters



- Use BGP route target filtering
  - RFC 4684

Route target filtering can improve efficiency, because it allows a route reflector to reflect only those routes a particular client PE router cares about.

Number of VRFs

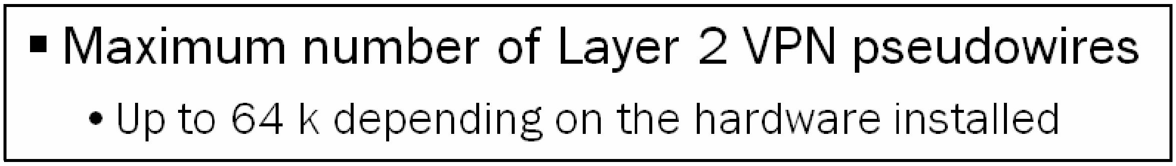


- Maximum number of Layer 2 VPN instances
  - Up to 9 k (VLAN based) depending on RE
    - Successfully tested, not an architectural limit
  - Increased VRFs equals longer convergence times

Currently, Juniper Networks has tested Layer 2 VPN scaling with as many as 9000 instances on a single PE router. While Layer 2 VPN scalability will likely continue to improve as Junos OS evolves, the ultimate limiting factor will be the remote site identifier. The current release uses a maximum of 65,534.

We suggest that you limit the number of Layer 2 VPN sites on a given PE router to 9,000 or less. This value is not an architectural limit, but it does represent the current extent of scalability testing conducted by Juniper Networks. Note that large numbers of VRFs can result in increased convergence time. Increased converge times can impact service-level agreements (SLAs), so each operator must make a compromise between the increased functionality of more Layer 2 VPN sites on each PE router versus the corresponding increase in convergence times.

Number of VRF Interfaces



- Maximum number of Layer 2 VPN pseudowires
  - Up to 64 k depending on the hardware installed

Junos OS support has been confirmed to support up to 64k Layer 2 VPN pseudowires depending on the hardware used. For example, a T Series Core Router will be able to support more pseudowires than an MX80 3D Universal Edge Router. Check with your Juniper Networks account team to determine the scaling limits of your system.

Similar to Layer 3 VPNs

Although Layer 2 VPNs are still emerging, we assume that CoS mechanisms for Layer 2 VPNs will be similar to those offered for Layer 3 VPNs.

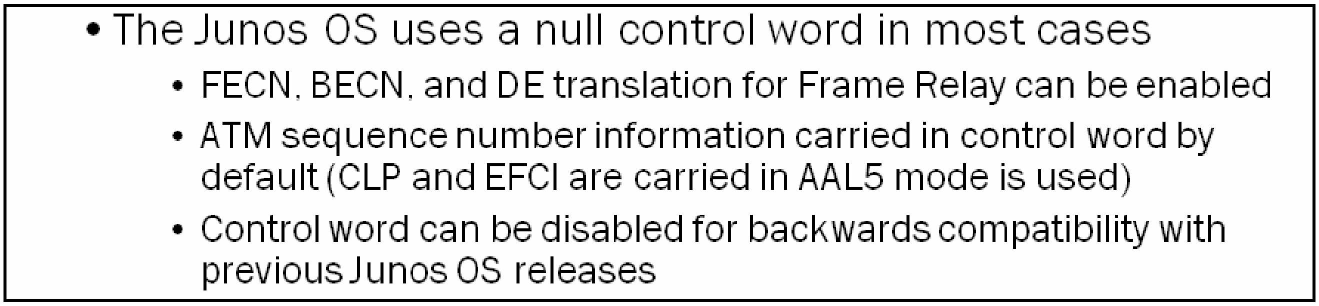
Existing Service Models

It is likely that the existing service models used to define Layer 2 service level agreements will be extended to Layer 2 VPNs. The following list provides examples of existing service models:

- Rate-based controls guarantee minimum transfer rates while seeking to protect the network from noncompliant sources.
- Loss-based parameters define the probability of data loss when the source is compliant with the negotiated traffic parameters.

- CoS-based models provide differentiated services based on the settings of Layer 2 indicators, such as Frame Relay's discard eligibility bit or the 802.1P Ethernet prioritization mechanisms.

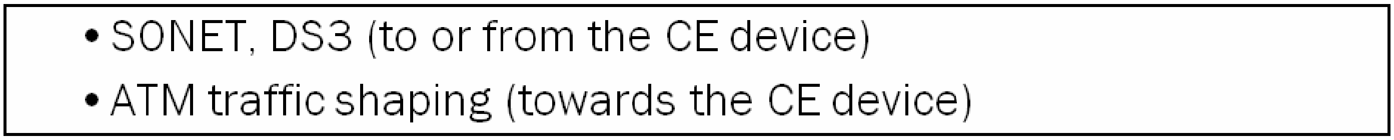
Control Word



- The Junos OS uses a null control word in most cases
  - FECN, BECN, and DE translation for Frame Relay can be enabled
  - ATM sequence number information carried in control word by default (CLP and EFCI are carried in AAL5 mode is used)
  - Control word can be disabled for backwards compatibility with previous Junos OS releases

Generally set to all zeros (null control word) by default in Junos OS, the control word supports the end-to-end conveyance of Layer 2 indicators such as discard eligibility, cell loss priority, and 802.1P priority settings. These capabilities can allow service providers to offer end-to-end significance for these indicators, which, in turn, enables the ability to offer end-to-end service level agreements for Layer 2 VPN services. Currently Junos OS will automatically use the control word to convey ATM Asynchronous Transfer Mode (ATM) sequence number, CLP, and EFCI when used for ATM pseudowires. Also, forward explicit congestion notification (FECN), backward explicit congestion notification (BECN), and discard eligibility (DE) translation can be enabled in the control word in the case of Frame Relay. You can also disable inclusion of the control word as needed for backwards compatibility with previous Junos OS releases that did not support it.

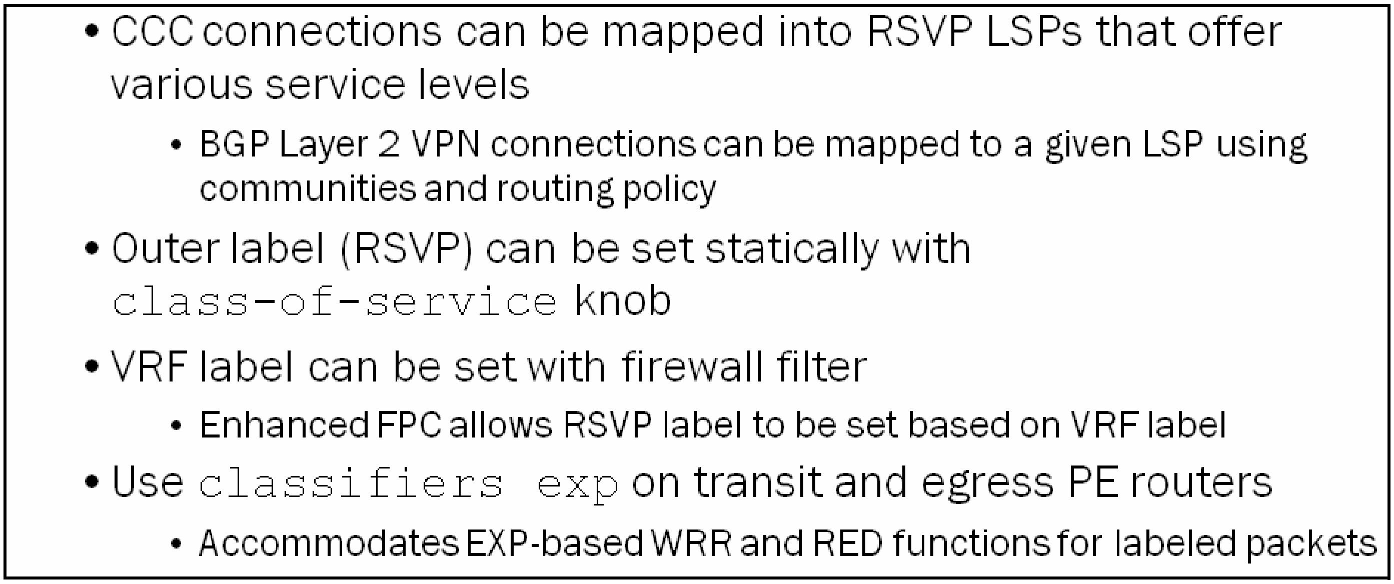
Interface Rate Limiting



- SONET, DS3 (to or from the CE device)
- ATM traffic shaping (towards the CE device)

Providers can use interface-based rate limiting to control the amount of traffic sent or received over interfaces used for PE-CE Layer 2 VPN connectivity.

Traffic Engineering

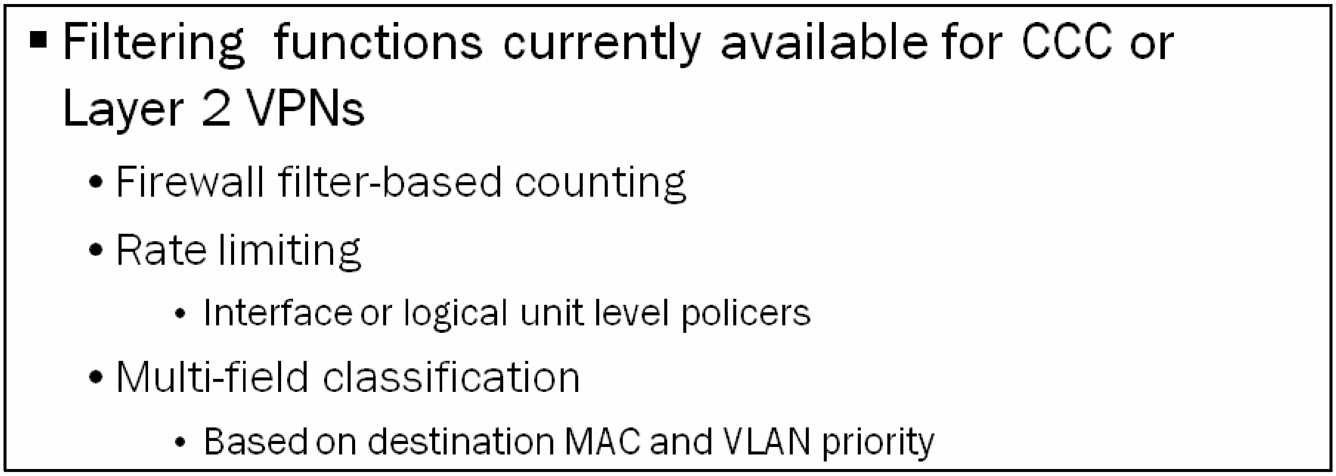


- CCC connections can be mapped into RSVP LSPs that offer various service levels
  - BGP Layer 2 VPN connections can be mapped to a given LSP using communities and routing policy
- Outer label (RSVP) can be set statically with `class-of-service` knob
- VRF label can be set with firewall filter
  - Enhanced FPC allows RSVP label to be set based on VRF label
- Use `classifiers exp` on transit and egress PE routers
  - Accommodates EXP-based WRR and RED functions for labeled packets

Various MPLS traffic engineering and CoS functions can be brought to bear in an effort to provide differentiated Layer 2 VPN services. The following list describes several of these functions:

- *Circuit cross-connect (CCC) connections*: You can map these connections manually to RSVP-signaled label-switched paths (LSPs) having particular routing and resource reservations. BGP Layer 2 VPNs normally use a shared LSP. You can map Layer 2 VPN connections to specific LSPs using policy-based and community-based matching.
- *RSVP EXP bits*: You can set these bits statically, or, with the Enhanced Flexible PIC Concentrator (FPC) hardware, you can set them dynamically, based on the EXP bits in the VPN routing and forwarding table (VRF) label.
- *The VRF label*: You can set this label's EXP bits based on interface-to-queue mapping configurations on the ingress router. The EXP setting in the VRF label can be copied into the RSVP label when Enhanced FPC hardware is present. This setting allows differential treatment of different VPN sites.
- *The* `classifiers exp` *option*: This option allows transit label-switching routers (LSRs) to act on the EXP bit settings to provide differential weighted round-robin (WRR)-related and random early detection (RED)-related actions on transit MPLS traffic. Failing to specify an EXP classifier results in all MPLS packets being placed into queue 0 by default. With Enhanced FPCs, you can alter the default EXP-to-queue mapping when wanted, but a classifier is still needed to alter the default behavior of placing all MPLS packets into queue 0.

Firewall Filtering Functions Available at Ingress

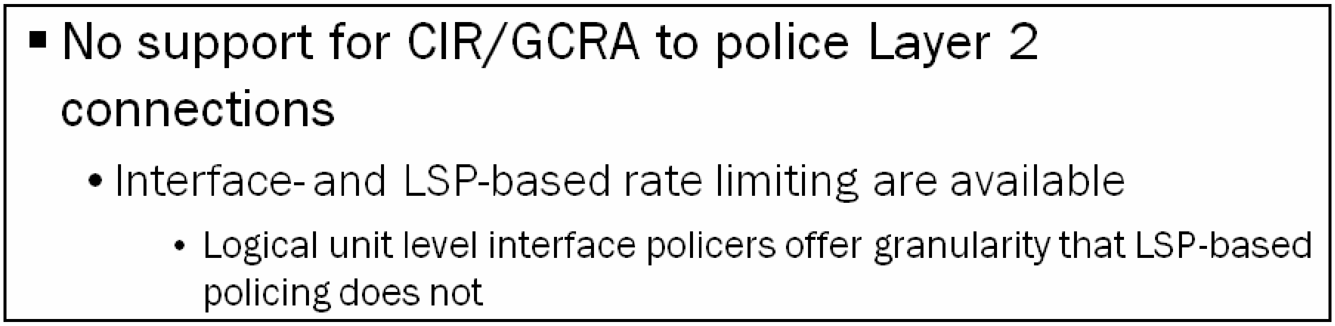


- Filtering functions currently available for CCC or Layer 2 VPNs
  - Firewall filter-based counting
  - Rate limiting
    - Interface or logical unit level policers
  - Multi-field classification
    - Based on destination MAC and VLAN priority

Firewall-based counting, LSP rate limiting, and multi-field classification are available at the PE router for CCC and Layer 2 VPN connections. You can place all traffic associated with an interface device into a specific outgoing queue using a firewall filter.

This capability also can be extended to individual logical units on a VPN interface. The latter can provide differentiated services on individual Layer 2 connections, while the former can be used for differentiated services among VPN sites.

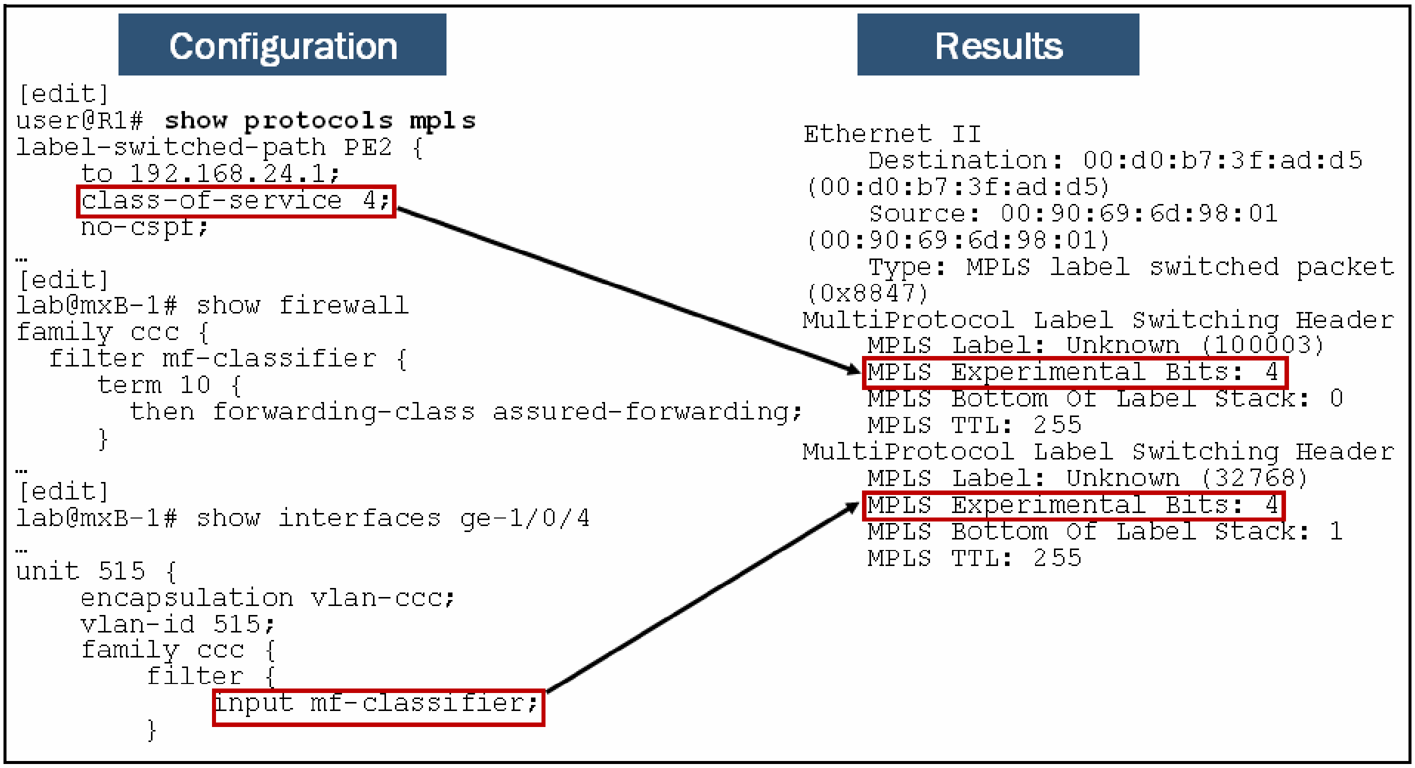
Layer 2 Connection Policing



- No support for CIR/GCRA to police Layer 2 connections
  - Interface- and LSP-based rate limiting are available
    - Logical unit level interface policers offer granularity that LSP-based policing does not

You can perform rate limiting at the interface, logical unit, and LSP levels to help enforce your VPN SLAs. Frame Relay's committed information rate (CIR) and ATM's Generic Cell Rate Algorithm (GCRA) are not directly supported, but you can achieve similar functionality by policing traffic at the Layer 2 VPN connection level. Police at the LSP level to limit the aggregate flow of Layer 2 VPN connections when wanted.

VRF Interface and LSP Mapped to Queue 2



Configuration

```
[edit]
user@R1# show protocols mpls
label-switched-path PE2 {
    to 192.168.24.1;
    class-of-service 4;
    no-cspf;
…
[edit]
lab@mxB-1# show firewall
family ccc {
  filter mf-classifier {
     term 10 {
       then forwarding-class assured-forwarding;
     }
…
[edit]
lab@mxB-1# show interfaces ge-1/0/4
…
unit 515 {
    encapsulation vlan-ccc;
    vlan-id 515;
    family ccc {
        filter {
            input mf-classifier;
        }
```

Results

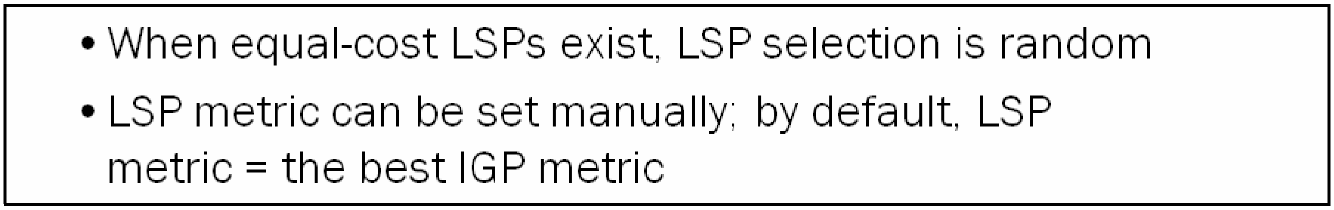
```
Ethernet II
    Destination: 00:d0:b7:3f:ad:d5
(00:d0:b7:3f:ad:d5)
    Source: 00:90:69:6d:98:01
(00:90:69:6d:98:01)
    Type: MPLS label switched packet
(0x8847)
MultiProtocol Label Switching Header
    MPLS Label: Unknown (100003)
    MPLS Experimental Bits: 4
    MPLS Bottom Of Label Stack: 0
    MPLS TTL: 255
MultiProtocol Label Switching Header
    MPLS Label: Unknown (32768)
    MPLS Experimental Bits: 4
    MPLS Bottom Of Label Stack: 1
    MPLS TTL: 255
```

This graphic provides a sample configuration showing how an RSVP session can have a static CoS setting. It also shows how a logical unit on a Layer 2 VPN interface can be mapped to a specific outgoing queue number. A protocol capture on the right of the graphic shows the results of the configuration.

This configuration places traffic received over the ge-1/0/4.515 interface into the assured-forwarding class (queue 2) at the ingress router. The static RSVP CoS setting causes transit LSRs to queue the traffic in queue 2 also.

Note that CoS settings for RSVP LSPs allow the full range of values from 0 to 7. In this case, the two most significant EXP bits are used to convey the queue number, while the least significant bit functions as a packet loss priority (PLP) indicator. The configured CoS value of 4 breaks down to a binary pattern of 1 0 0, which codes to forwarding class assured-forwarding or queue number 2, with PLP = 0. Therefore, the static CoS setting of 4 identifies queue 2, as does the selection of assured-forwarding under the CoS configuration.

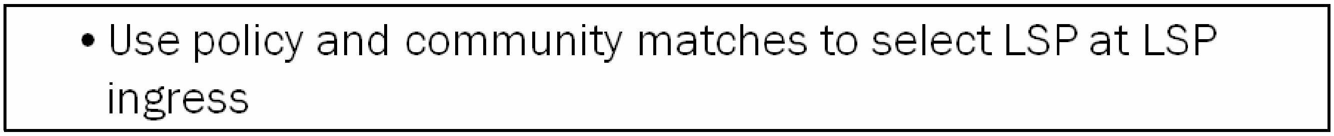
Layer 2 VPN Traffic Mapped to LSP with Lowest Metric



- When equal-cost LSPs exist, LSP selection is random
- LSP metric can be set manually; by default, LSP metric = the best IGP metric

When multiple LSPs exist between PE routers, the Layer 2 VPN traffic is mapped to the LSP with the lowest metric. The metric associated with an LSP is the lowest interior gateway protocol (IGP) metric from ingress to egress router by default (not the metric along the path of the LSP), but Junos OS supports manual metric setting of an LSP. When multiple, equal-cost LSPs exist, the VPN traffic is mapped to one of the LSPs using a random selection algorithm.

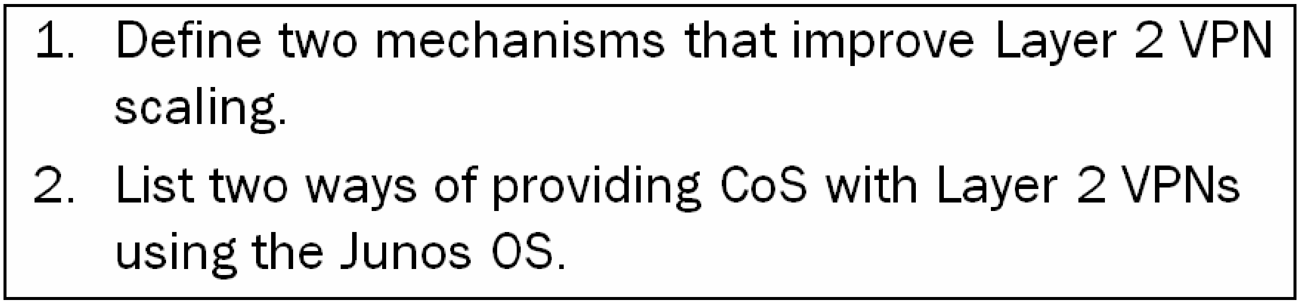
LSP Selection



- Use policy and community matches to select LSP at LSP ingress

When needed, you can map BGP Layer 2 VPN traffic to one of several equal-cost RSVP-signaled LSPs. In most cases, you perform the mapping of Layer 2 VPN traffic to a given LSP based on community tags and a corresponding forwarding table export policy on the LSP ingress node that serves to select a given LSP next hop based on community matches.

Review Questions



1. Define two mechanisms that improve Layer 2 VPN scaling.
2. List two ways of providing CoS with Layer 2 VPNs using the Junos OS.

Answers to Review Questions

1.

The use of route reflectors and route target filtering are recommended methods of improving Layer 2 VPN scaling.

2.

The EXP bits of the VRF label can be set by using an input firewall filter. The EXP bits of the outer RSVP-signalled label can be set with the **`class-of-service`** setting on the LSP definition.

JNCIS-SP Study Guide—Part 3

Chapter 15: LDP Layer 2 Circuits

This Chapter Discusses:

- The flow of control and data traffic for a LDP Layer 2 circuit;
- Configuring a LDP Layer 2 circuit;
- Monitoring and troubleshooting a LDP Layer 2 circuit; and
- Configuring circuit cross-connect (CCC) MPLS interface tunneling.

RFC 4447 Support

The Junos operating system offers support for Layer 2 circuits based on the signaling techniques defined in RFC 4447. Only remote provider edge (PE)-to-PE connections are supported; you cannot use the LDP Layer 2 circuits to establish connections between customer edge (CE) devices that attach to the same PE router.

LDP Signaling

RFC 4447 specifies the use of LDP for exchanging virtual circuit (VC) labels between PE routers. As a result, PE routers no longer require BGP signaling between them. LDP Layer 2 circuits based on RFC 4447 do not use site identifiers, route distinguishers, or VPN routing and forwarding (VRF) policy.

RFC 4447 makes use of LDP extended neighbor relationships (as is used for LDP-over-RSVP tunnels) such that the PE routers establish extended LDP sessions as needed, despite their not being directly connected neighbors. If wanted, the LDP session between PE routers can be tunneled over a traffic engineered RSVP path.

CCC or TCC Encapsulation

Configuring a LDP Layer 2 circuit is very similar to configuring CCC or translational cross-connect (TCC) connections. When CCC encapsulation is used, the Layer 2 technology must be the same at both ends of the connection. RFC 4447 connections are referred to as `l2circuits` in the Junos OS.

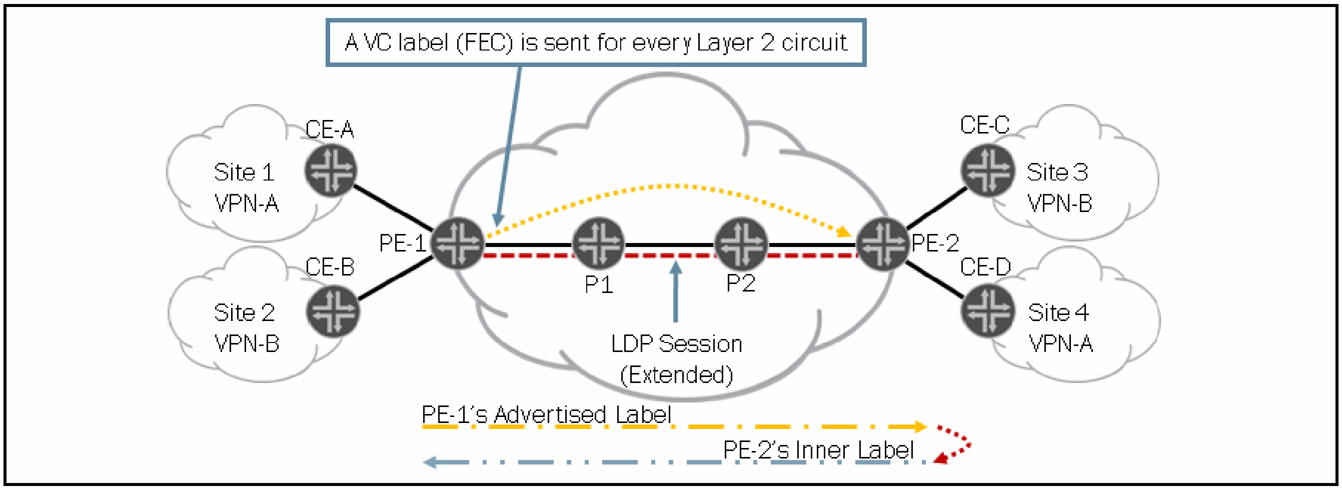
Defines a VC Label

In the LDP Layer 2 circuit approach, a VC label is assigned to each interface connection. This label functions similar to the BGP Layer 2 NLRI label in a BGP Layer 2 VPN solution.

PE Routers Advertise Labels

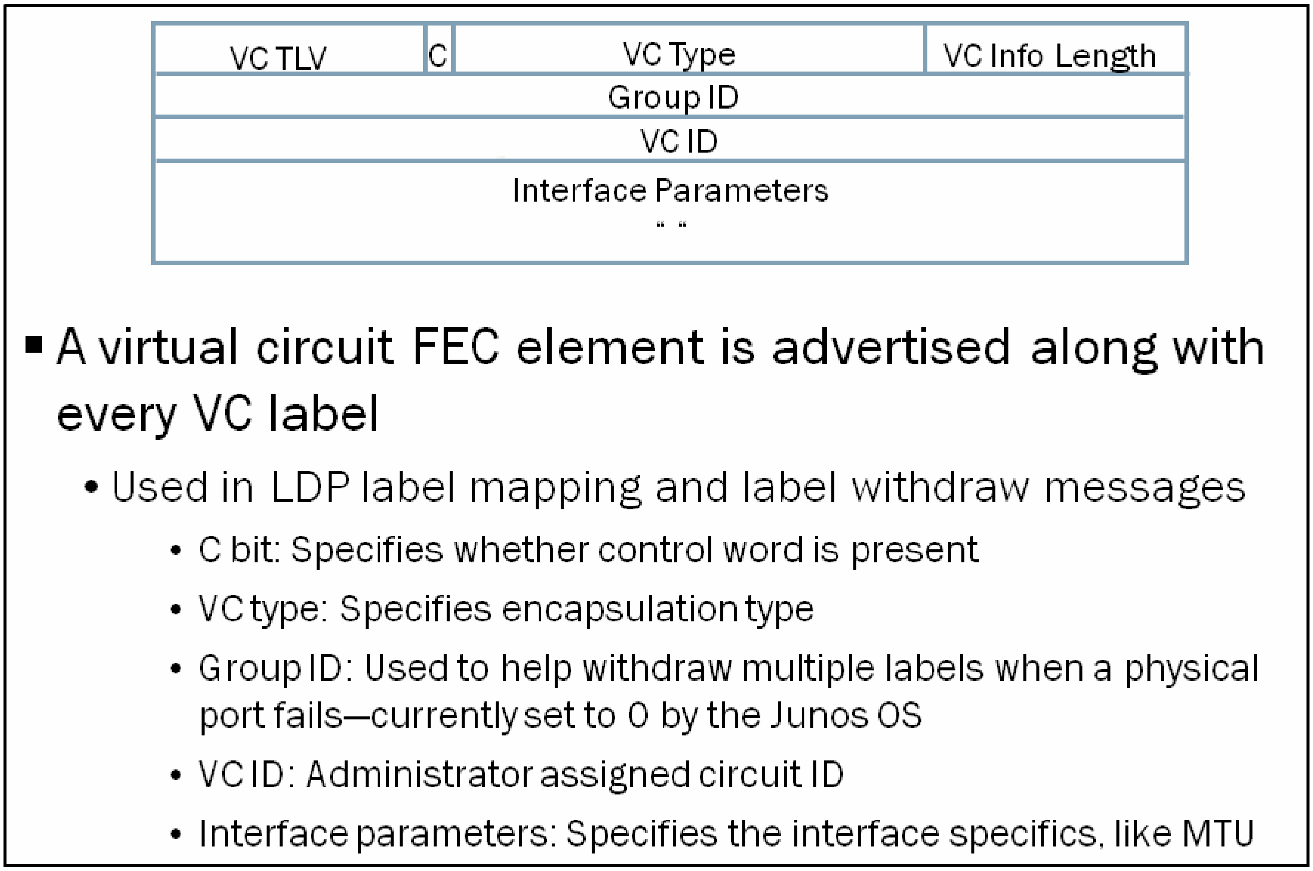
In operation, a PE router advertises a label for each LDP Layer 2 circuit configured. To LDP, this is just another forwarding equivalence class (FEC). These labels are advertised to targeted peers using extended LDP sessions.

Input and Output Labels



As shown on the diagram, the remote PE router (PE-2) uses the input label value advertised by PE-1 as its output label when forwarding traffic associated with this FEC to PE-1. Although not shown here, you can assume that PE-2 has also advertised an input label to PE-1, and that PE-1 pushes this label when sending traffic (for this connection) to PE-2.

Virtual Circuit FEC Element



VC TLV	C	VC Type	VC Info Length
Group ID			
VC ID			
Interface Parameters “ “			

- A virtual circuit FEC element is advertised along with every VC label
  - Used in LDP label mapping and label withdraw messages
    - C bit: Specifies whether control word is present
    - VC type: Specifies encapsulation type
    - Group ID: Used to help withdraw multiple labels when a physical port fails—currently set to 0 by the Junos OS
    - VC ID: Administrator assigned circuit ID
    - Interface parameters: Specifies the interface specifics, like MTU

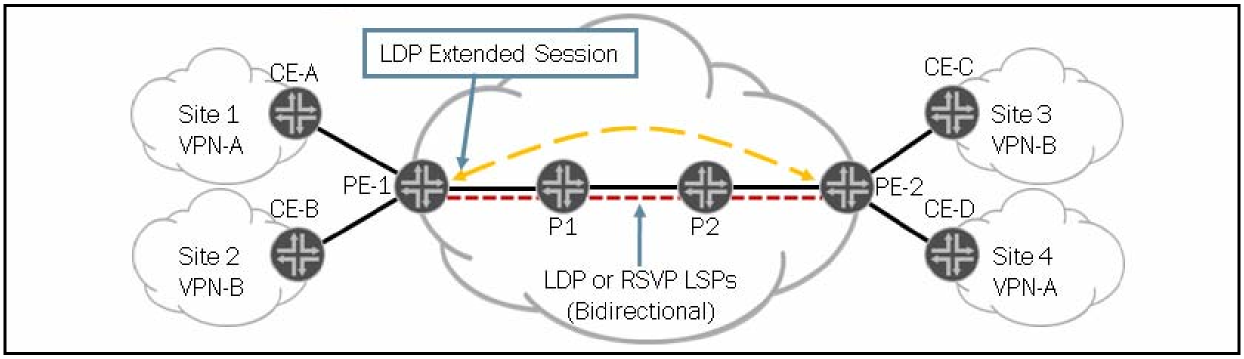
Using the LDP extended neighbor relationship, PE routers can exchange the virtual circuit labels associated with the VPN's interfaces. Along with each label, an associated VC FEC element is also advertised. This FEC element is used to describe the parameters of a PE router to the remote LDP neighbor.

The fields in this FEC element are described as follows:

- *C bit*: Specifies whether the Martini control word is present. This bit is set by default (control word present) in the Junos OS.
- *VC type*: Layer 2 encapsulation on VPN interface.

- *Group ID (optional)*: Used to group a set of labels together that relate to a particular port or tunnel. Makes withdrawal of labels easier when there is a failure of a port and there are many VPN labels associated with that same port.
- *VC ID*: An administrator-configurable value that represents the Layer 2 circuit.
- *Interface parameters*: Used to validate interoperability between ingress and egress ports. Possible parameter can be maximum transmission unit (MTU), maximum number of concatenated Asynchronous Transfer Mode (ATM) cells, interface description string, and other circuit emulation parameters.

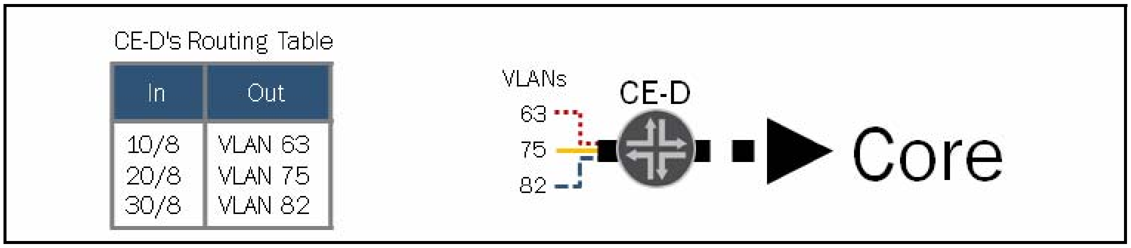
Provisioning the Core



As with a Layer 3 VPN, the provider's core must be provisioned to support a Layer 2 VPN service. To support a LDP Layer 2 circuit, the following requirements must be met:

- *LDP*: The PE and provider (P) routers must be configured to run LDP on their core and core-facing interfaces if the provider chooses to use LDP-signaled label-switched path (LSPs) for forwarding. Otherwise, RSVP-signaled LSPs must be established between PE routers. PE routers must also enable LDP on their loopback interfaces to support extended LDP sessions with remote PE routers.
- *Interior gateway protocol (IGP)*: The PE and P routers must have a functional IGP, and they must share a single routing domain.
- *MPLS*: The P and PE routers must have the MPLS family configured on their core and core-facing interfaces, and they must have MPLS processing enabled by listing each such interface under the `protocol mpls` configuration hierarchy. The PE-CE interface should not be configured with the MPLS family due to its use of CCC encapsulation, which prohibits the declaration of protocol families on affected logical units.

Provisioning the CE Device


CE-D's Routing Table

In	Out
10/8	VLAN 63
20/8	VLAN 75
30/8	VLAN 82

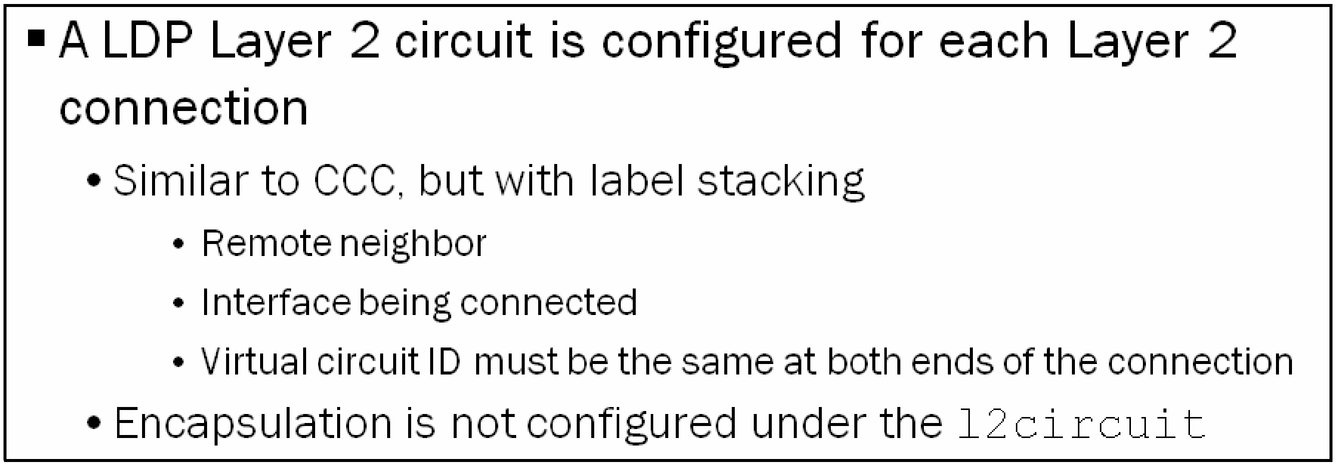
The first step in building a Layer 2 VPN is the configuration of the CE device. This configuration normally involves the assignment of a range of Layer 2 circuit identifiers to logical interfaces on the CE device (one for each remote connection) and the specification of the correct encapsulation settings for the Layer 2 protocol being configured. Other aspects of CE device configuration are:

- *Circuit IDs*: The Junos OS requires that virtual LAN (VLAN) IDs be the same at both ends of a Layer 2 connection. Frame Relay and ATM VC identifiers can be different at the remote sites. You can have different VLAN IDs if you are using TCC encapsulation.
- *Layer 2 parameters*: The CE device might also require the configuration of Layer 2 protocol keepalive functions, MTUs, and payload encapsulation options, such as Institute of Electrical and Electronics Engineers (IEEE) 802.3 versus Ethernet V2 encapsulation, all of which operate end to end in a Layer 2 VPN. You should ensure that the

MTU supported by the CE devices will not cause problems with fragmentation in the PE or core routers, as incompatible MTUs result in silent discards.

- *Layer 3 configuration*: The CE device's upper layers must be compatible with the remote CE device, as these parameters are configured with end-to-end significance when deploying a Layer 2 VPN.

Provisioning the PE Router



- A LDP Layer 2 circuit is configured for each Layer 2 connection
  - Similar to CCC, but with label stacking
    - Remote neighbor
    - Interface being connected
    - Virtual circuit ID must be the same at both ends of the connection
  - Encapsulation is not configured under the `l2circuit`

After configuring the local CE device properties, you should provision the site's PE router. The list of what must be configured for the `l2circuit` portion of the PE configuration includes:

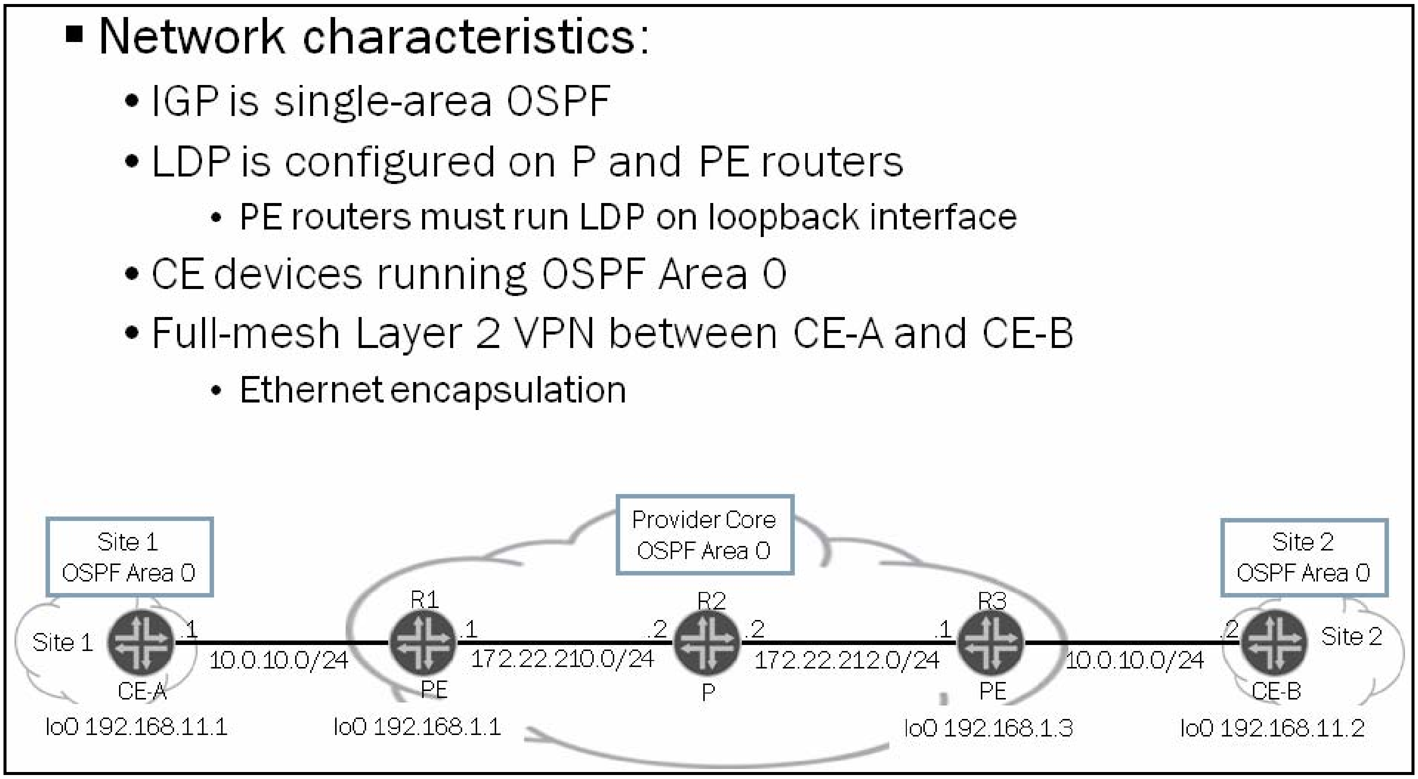
- Specification of the remote PE router using the **`neighbor`** statement;
- The interface (including the logical unit) being connected; and
- A VC identifier using the **`virtual-circuit-id`** statement.

Configuring the Interfaces

Once the site's PE router is provisioned, you should configure its Layer 2 VPN interfaces. This operation is identical to that of CCC or TCC, in that the interface and logical unit must be set to the appropriate form of CCC or TCC encapsulation. The PE router's interfaces must be configured so that they are compatible with the encapsulation and interface type being used by the attached CE device.

The inability to test the local PE-CE link with utilities such as ping, or by observing routing protocol operation, tends to make Layer 2 VPN PE-CE interface troubleshooting difficult.

Example LDP Layer 2 Circuit: Topology



- Network characteristics:
  - IGP is single-area OSPF
  - LDP is configured on P and PE routers
    - PE routers must run LDP on loopback interface
  - CE devices running OSPF Area 0
  - Full-mesh Layer 2 VPN between CE-A and CE-B
    - Ethernet encapsulation

The diagram serves as the basis for the various configuration mode and operational mode examples that follow.

The IGP is OSPF, and a single area (Area 0) is configured.

LDP is deployed as the MPLS signaling protocol and is configured to run on the core-facing and loopback interfaces of the PE routers.

BGP is not configured in this example, as it is not required for a LDP Layer 2 circuit.

In this example, the CE routers run OSPF with a common IP subnet shared by CE-1 and CE-2. The PE routers have no IP addressing on the PE-CE interfaces.

The goal of this network is to provide point-to-point connectivity between the Ethernet-based CE devices shown.

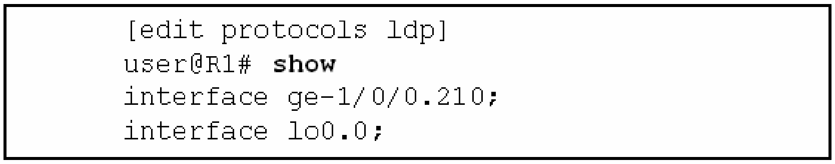
R1's Layer 2 Circuits Configuration

```
[edit protocols l2circuit]
user@R1# show
neighbor 192.168.1.3 {
    interface ge-1/0/4.512 {
        virtual-circuit-id 4;
    }
}
```

You create LDP Layer 2 circuits at the `[edit protocols l2circuit]` portion of the hierarchy. The **`neighbor`** statement specifies an IP address that is the LSP endpoint of the tunnel that should transport the Layer 2 connection to the remote PE router. In this example, it is configured to specify the R3 PE router's loopback address.

The PE-CE interface is also listed, along with its virtual circuit identifier value of 4. The VC ID must be unique within the context of a particular neighbor, as the combination of the *neighbor-IP-address* and *VC-ID* is the key for identifying a particular VC on a specific PE router. If this PE router had other connections to the R3 PE router, they would be listed under the existing **`neighbor`** statement. Connections to different PE routers require the declaration of a new neighbor address.

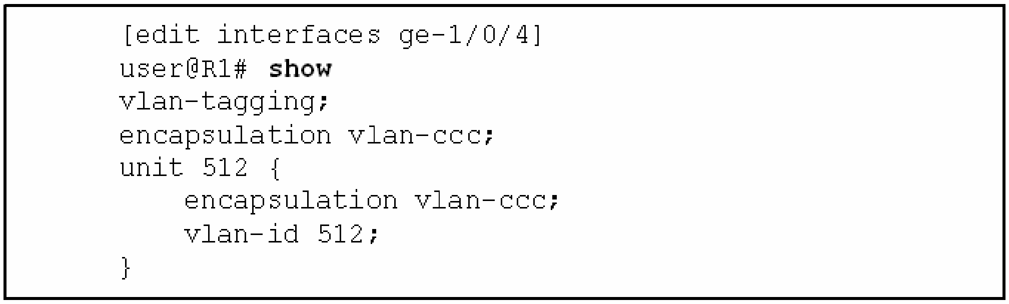
R1's LDP Configuration



```
[edit protocols ldp]
user@R1# show
interface ge-1/0/0.210;
interface lo0.0;
```

The graphic also shows the LDP-related configuration of the R1 PE router. You must configure LDP to run on the router's `lo0` interface when extended LDP neighbor relationships are required. Note that the MPLS configuration is not displayed.

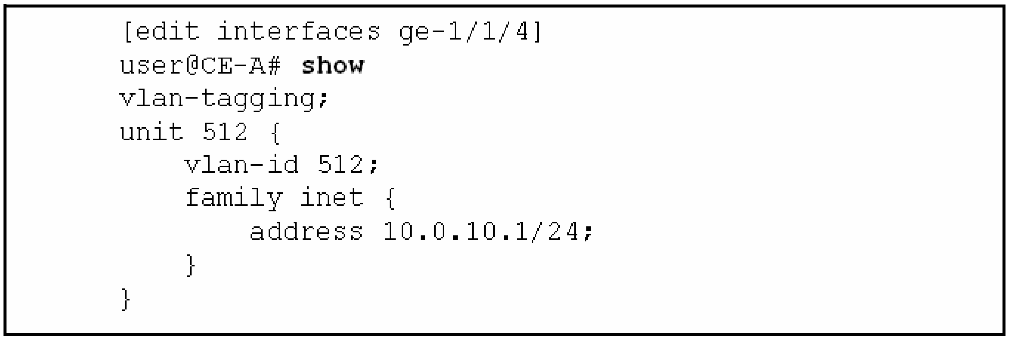
R1's PE-CE Interface Configuration



```
[edit interfaces ge-1/0/4]
user@R1# show
vlan-tagging;
encapsulation vlan-ccc;
unit 512 {
    encapsulation vlan-ccc;
    vlan-id 512;
}
```

The graphic shows the Layer 2 VPN interface-related configuration on the R1 PE router. It is identical to the configuration required for a CCC-based or BGP Layer 2 VPN application.

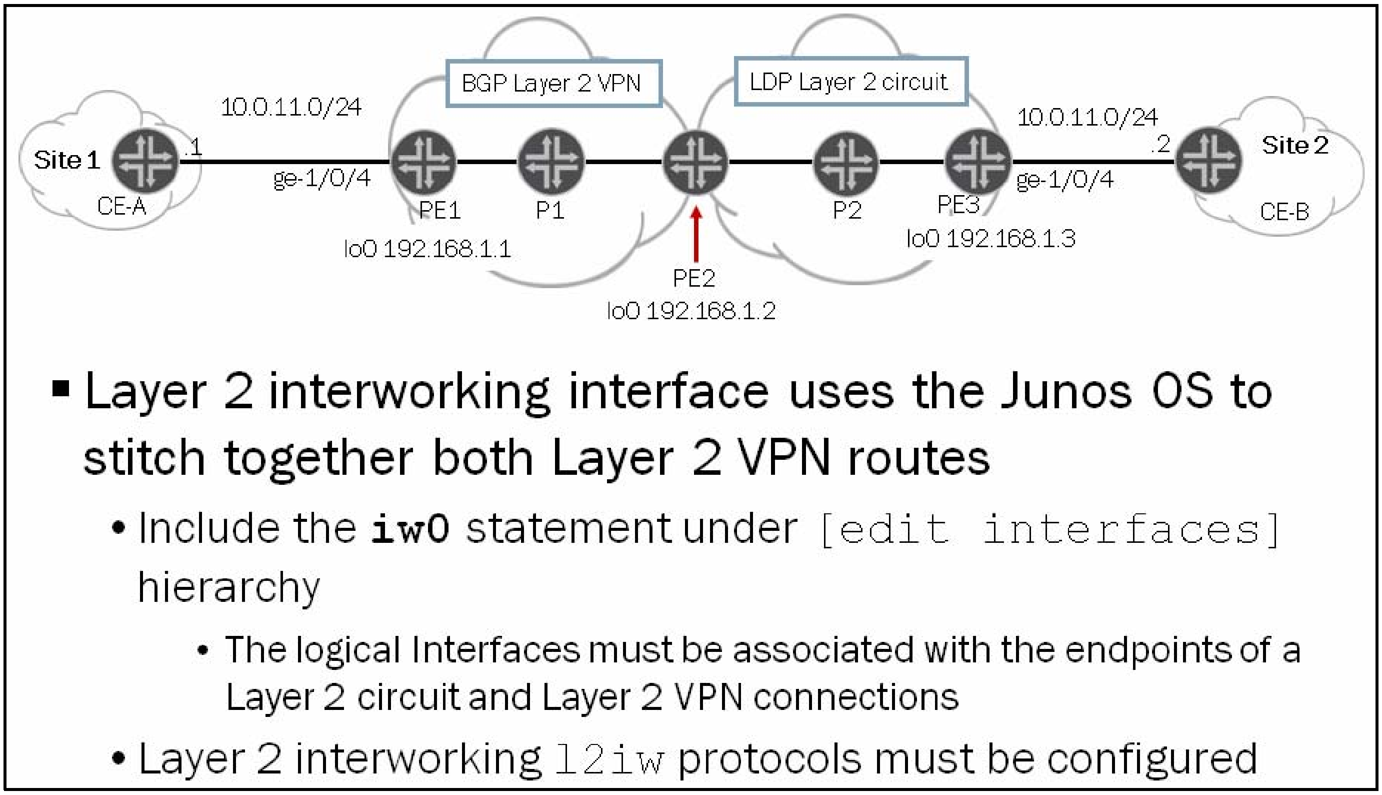
CE-A's CE-PE Interface Configuration



```
[edit interfaces ge-1/1/4]
user@CE-A# show
vlan-tagging;
unit 512 {
    vlan-id 512;
    family inet {
        address 10.0.10.1/24;
    }
}
```

For completeness, the graphic shows the interface configuration of the CE-1 router. As with the PE router, this same configuration could be used for either a CCC-based or BGP-based Layer 2 VPN. Note that the CE device's interface has a protocol family configured, and that these parameters must be compatible with the interface settings on the remote CE device.

Layer 2 Internetworking



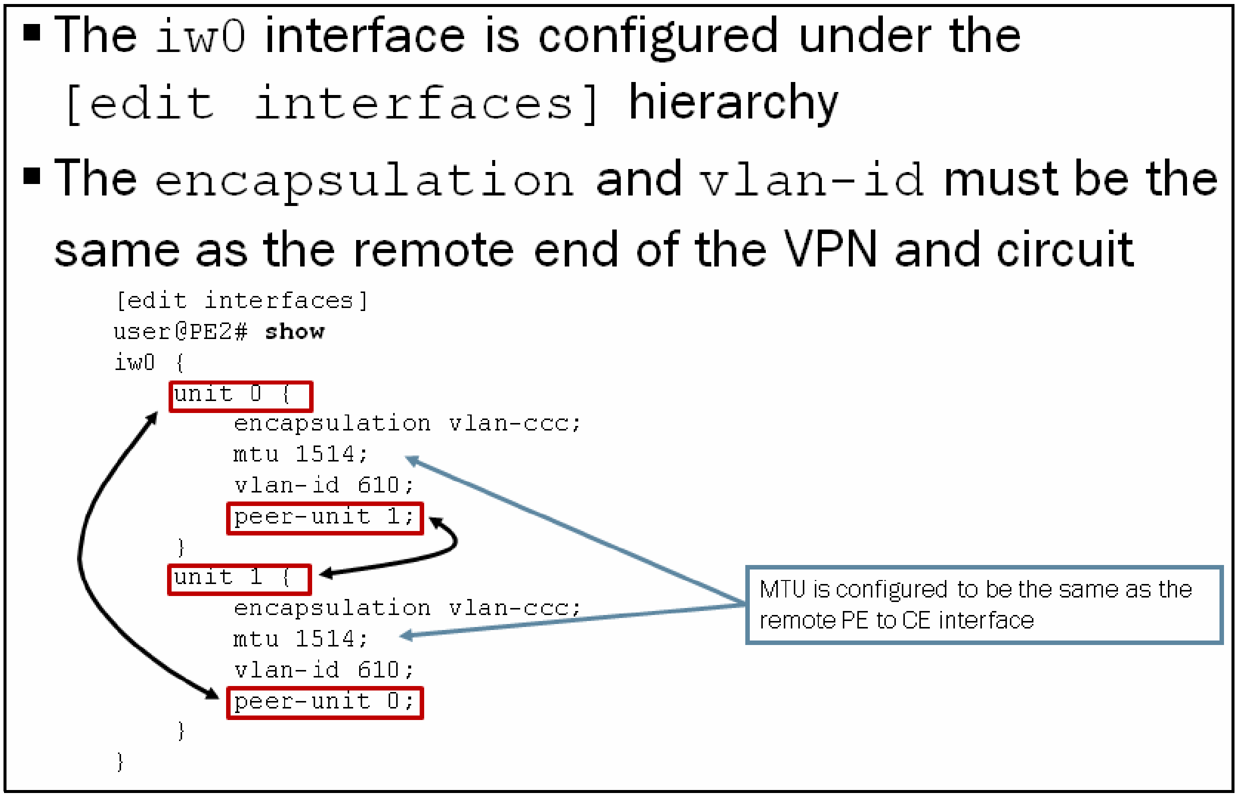
- Layer 2 interworking interface uses the Junos OS to stitch together both Layer 2 VPN routes
  - Include the **`iw0`** statement under `[edit interfaces]` hierarchy
    - The logical Interfaces must be associated with the endpoints of a Layer 2 circuit and Layer 2 VPN connections
  - Layer 2 interworking `l2iw` protocols must be configured

With the Junos OS it is possible to connect a Layer 2 VPN with a Layer 2 circuit by using an interworking interface. Instead of using a physical Tunnel PIC for looping the packet received from the Layer 2 circuit, the Layer 2 interworking interface uses Junos OS to stitch together both Layer 2 connections. The `iw0` statement is configured at the `[edit interfaces]` hierarchy level. This specifies the peering between two logical interfaces. This configuration is similar to the configuration for a logical tunnel interface. The logical Interfaces must be associated with the endpoints of a Layer 2 circuit and Layer 2 VPN connections.

In addition to configuring the interfaces and associating them with the Layer 2 protocols, the Layer 2 interworking `l2iw` protocol must be configured. Without the `l2iw` configuration, the `l2iw` routes will not be formed, regardless of whether any `iw` interfaces are present. Within the `l2iw` protocols, only trace options can be configured in the standard fashion.

This process can also be used to stitch together two Layer 2 circuits as well as stitch together two Layer 2 VPNs, as mentioned in an earlier chapter.

The **iw0** Interface Configuration



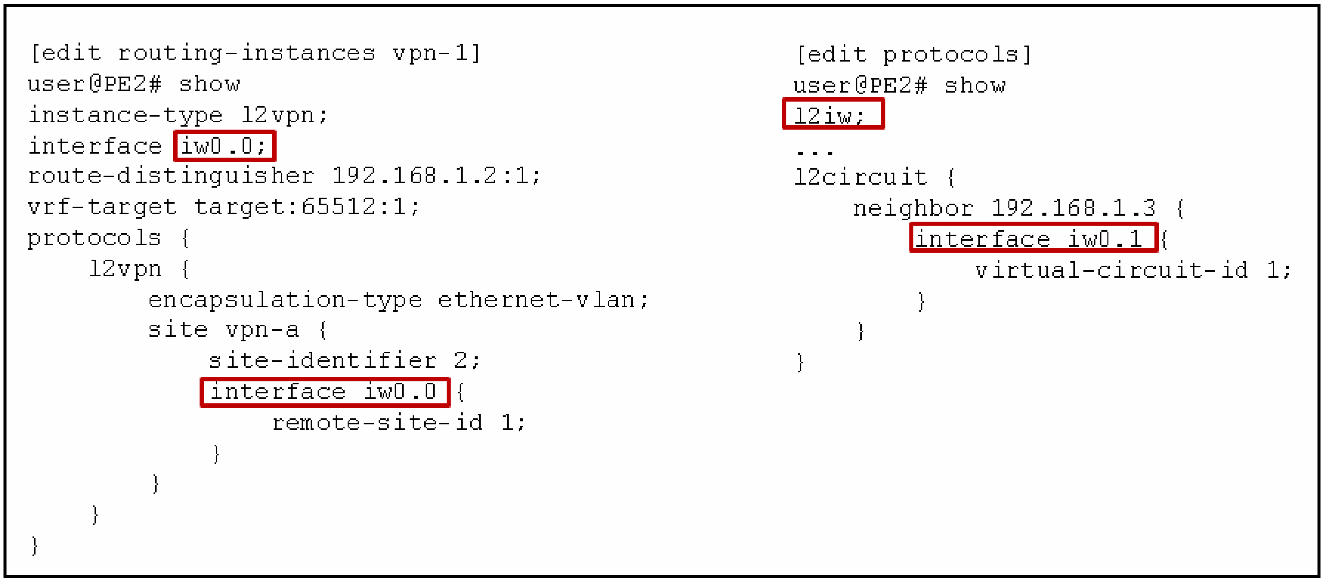
- The `iw0` interface is configured under the `[edit interfaces]` hierarchy
- The `encapsulation` and `vlan-id` must be the same as the remote end of the VPN and circuit

```
[edit interfaces]
user@PE2# show
iw0 {
    unit 0 {
        encapsulation vlan-ccc;
        mtu 1514;
        vlan-id 610;
        peer-unit 1;
    }
    unit 1 {
        encapsulation vlan-ccc;
        mtu 1514;
        vlan-id 610;
        peer-unit 0;
    }
}
```

MTU is configured to be the same as the remote PE to CE interface

The graphic demonstrates a basic `iw0` interface configuration. As indicated in the graphic you must configure two logical units. The same encapsulation and vlan-id must be configured on the `iw0` units as is configured on the PE to CE interfaces. Another requirement is that the MTU value for the interface be configured to be the same as the PE to CE interface. In our example, 1514 is used to match the ethernet MTU on the remote end. The MTU has to be specified because the Layer 2 circuit will not establish if the MTU does not match on both sides of the circuit. Traceoptions can be configured for the Layer 2 circuit to assist in determining the correct MTU value. The default MTU for the `iw0` interface is 65522 and must be set the same for both peer units configured. As displayed on the graphic, a `peer-unit` must be specified for each unit. This statement associates two units together so that traffic can be stitched between the two Layer 2 connections.

Layer 2 VPN and Layer 2 Circuit Configurations



```
[edit routing-instances vpn-1]
user@PE2# show
instance-type l2vpn;
interface iw0.0;
route-distinguisher 192.168.1.2:1;
vrf-target target:65512:1;
protocols {
    l2vpn {
        encapsulation-type ethernet-vlan;
        site vpn-a {
            site-identifier 2;
            interface iw0.0 {
                remote-site-id 1;
            }
        }
    }
}
```

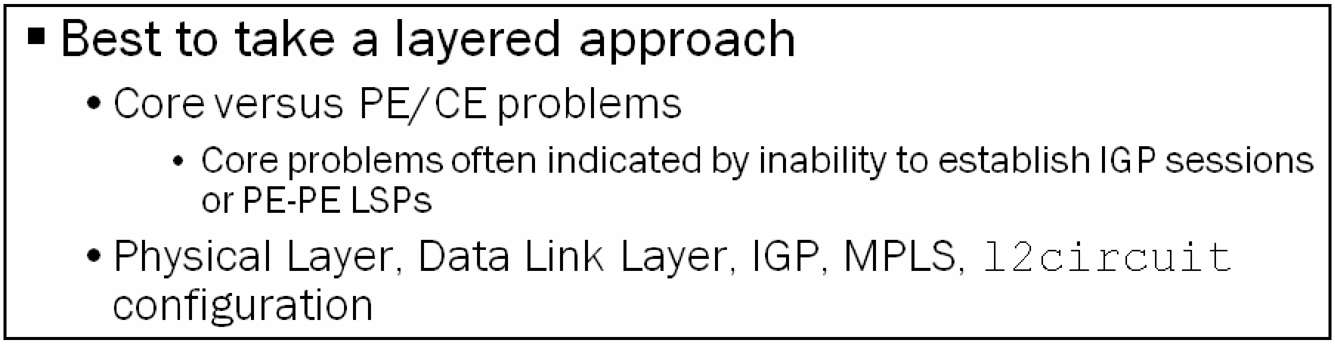
```
[edit protocols]
user@PE2# show
l2iw;
...
l2circuit {
    neighbor 192.168.1.3 {
        interface iw0.1 {
            virtual-circuit-id 1;
        }
    }
}
```

The iw0 interface is configured as the CE facing interface for each Layer 2 protocol. In the Layer 2 circuit you configure the IP address of the remote PE router by, include the **neighbor** statement and specify the IP address of the loopback interface on PE2. Configure the virtual circuit ID to be the same as the virtual circuit ID on the neighbor router. To allow a Layer 2 circuit to be established even though the MTU configured on the local PE router does not match the MTU configured on the remote PE router, you can include the **`ignore-mtu-mismatch`** statement. You can also disable the use of the control word for demultiplexing by including the **`no-control-word`** statement. If control-word is turned off, it must be turned off for both Layer 2 protocols throughout the circuit.

To configure the Layer 2 VPN protocol, including the **`l2vpn`** statement at the `[edit routing-instances routing-instances-name protocols]` hierarchy level. To configure the `iw0` interface, include the interfaces statement and specify `iw0` as the interface name. In the example provided, the `iw0.0` interface is configured under the Layer 2 VPN protocols to receive the looped packet from the `iw0.1`.

In addition to the `iw0` interface configuration, Layer 2 interworking `l2iw` protocols must be configured. Without the `l2iw` configuration, the `l2iw` routes are not formed, regardless of whether any `iw` interfaces are present.The minimum configuration necessary for the feature to work is shown on the graphic.

Take a Layered Approach

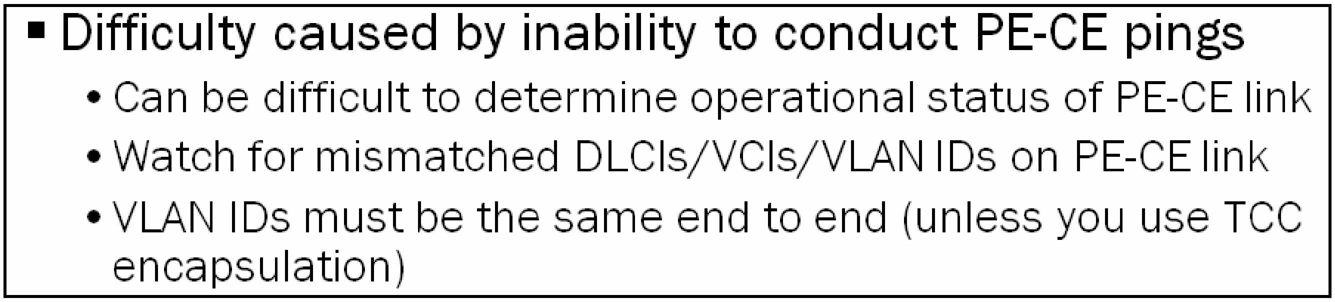


- Best to take a layered approach
  - Core versus PE/CE problems
    - Core problems often indicated by inability to establish IGP sessions or PE-PE LSPs
  - Physical Layer, Data Link Layer, IGP, MPLS, `l2circuit` configuration

Any number of configuration and operational problems can result in a dysfunctional VPN. With this much complexity, we encourage you to take a layered approach to the provisioning and troubleshooting of Layer 2 VPN services.

Is the problem core or PE-CE related? and *Are my pings failing because an interface is down, or because an LSP cannot be established?* are the types of questions that await you when troubleshooting. Fortunately, Layer 2 VPNs have several natural boundaries that allow for expedient problem isolation. As an example, consider a call reporting that three different LDP Layer 2 circuits on two different PE routers are down. Here, you look for core-related issues (the P routers are common to all VPNs) rather than looking for PE-CE-related problems at the sites reporting problems.

PE-CE Ping Testing No Longer Possible



- Difficulty caused by inability to conduct PE-CE pings
  - Can be difficult to determine operational status of PE-CE link
  - Watch for mismatched DLCIs/VCIs/VLAN IDs on PE-CE link
  - VLAN IDs must be the same end to end (unless you use TCC encapsulation)

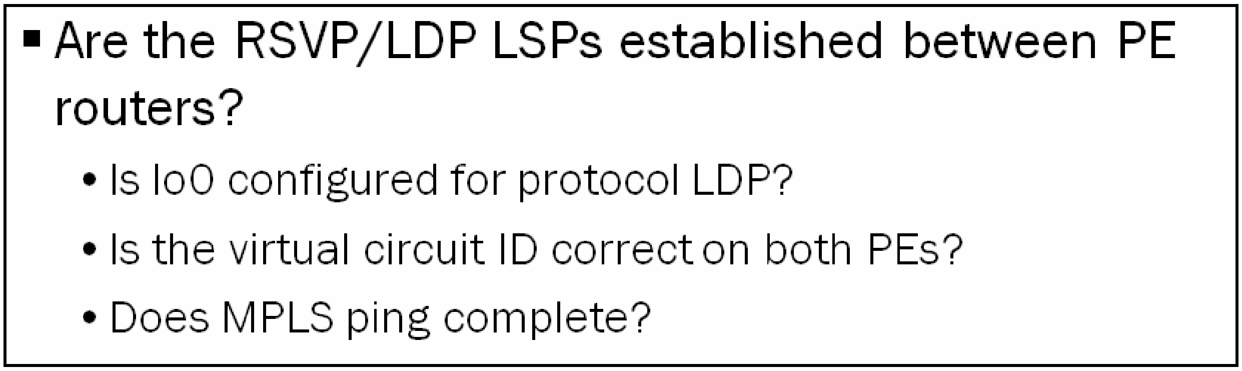
Layer 2 VPN troubleshooting differs from Layer 3 VPN troubleshooting in many ways. A significant difference is that the PE router and CE devices do not share IP connectivity, which makes the testing of the local PE-CE link difficult. In some cases you can determine the operational status of the PE-CE link by verifying the correct operation of the data link layer's keepalive function.

Mismatches between the connection identifiers configured on the PE-CE link are common sources of problems. VLAN ID must be the same end to end, unless you are using TCC encapsulation. Sometimes you can provision an out-of-band management interface that permits ping testing and Telnet access to the local CE device. This interface should be another logical unit on the interface also providing Layer 2 VPN connectivity.

Core IGP

A functional core IGP is critical to the operation of LSP signaling protocols. You always should check the IGP when LSP problems are evident. Generally, you verify IGP operation by enforcing such tasks as looking at routing tables and neighbor states (adjacencies) and conducting ping and traceroute testing.

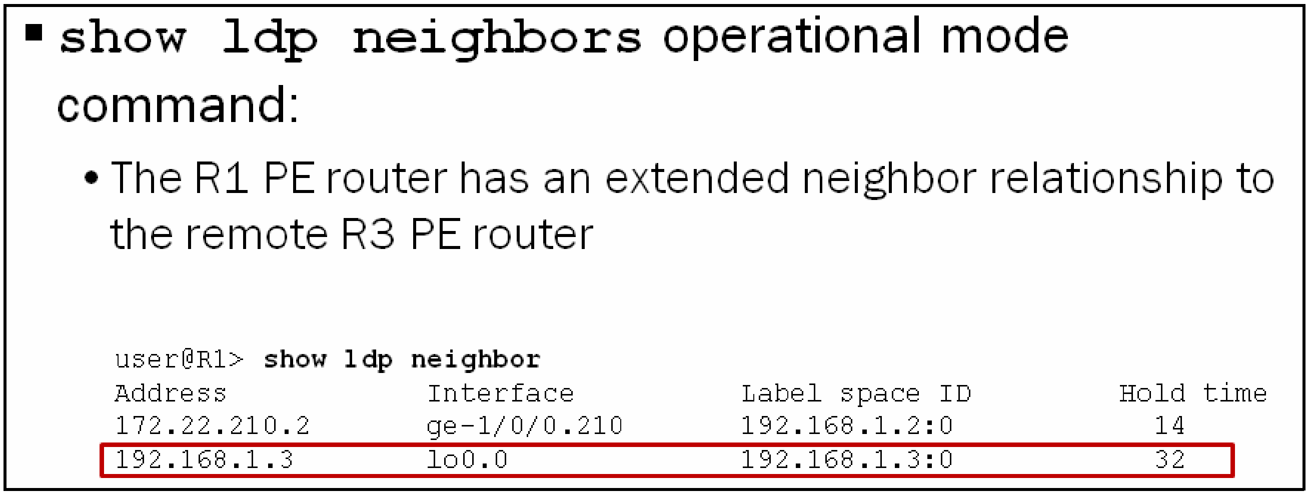
LSPs



- Are the RSVP/LDP LSPs established between PE routers?
  - Is lo0 configured for protocol LDP?
  - Is the virtual circuit ID correct on both PEs?
  - Does MPLS ping complete?

Each pair of PE routers sharing VPN membership must have LSPs established in both directions before traffic can be forwarded over the VPN. When dealing with extended LDP sessions you should verify that your LDP interfaces include the loopback. You should also look at the l2circuit configuration to ensure you have the proper neighbor configured as well as the proper virtual circuit ID. Another useful step is using MPLS ping, as mentioned in previous chapters, this is a valuable utility for checking MPLS connectivity to remote PE routers.

Confirming LDP Operation



- **show ldp neighbors** operational mode command:
  - The R1 PE router has an extended neighbor relationship to the remote R3 PE router

```
user@R1> show ldp neighbor
Address            Interface          Label space ID         Hold time
172.22.210.2       ge-1/0/0.210       192.168.1.2:0            14
192.168.1.3        lo0.0              192.168.1.3:0            32
```

An important step is to verify the LDP protocol that is used both to signal LSPs and to communicate Layer 2 VPN VC identifiers between PE routers. Also, because LDP relies on a functional IGP, you can often validate IGP operation by assessing how well things are going for LDP.

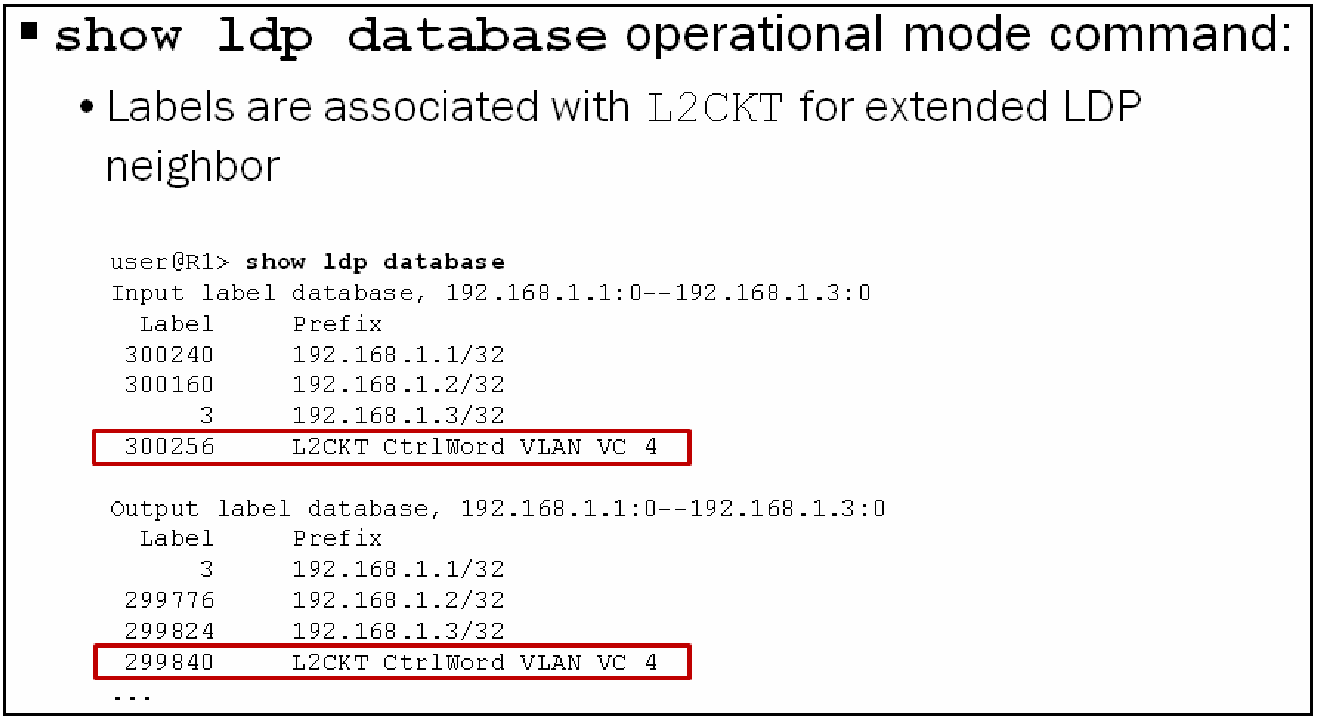
The **show ldp neighbors** command indicates if the PE router has successfully formed neighbor relationships with the directly connected and extended neighbors. The highlight in this graphic draws attention to the extended neighbor session that the R1 PE router has established to the remote R3 PE router. The other neighbor session is to the P1 router.

The effect of these neighbor relationships should be the establishment of LSPs to the router IDs of all routers running LDP. You can verify this establishment with the **show route table inet.3** command:

```
user@R1> show route table inet.3

inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
192.168.1.2/32      *[LDP/9] 1d 02:00:03, metric 1
                     > to 172.22.210.2 via ge-1/0/0.210
192.168.1.3/32      *[LDP/9] 1d 02:00:03, metric 1
                     > to 172.22.210.2 via ge-1/0/0.210, Push 300592
```

Showing the LDP Database



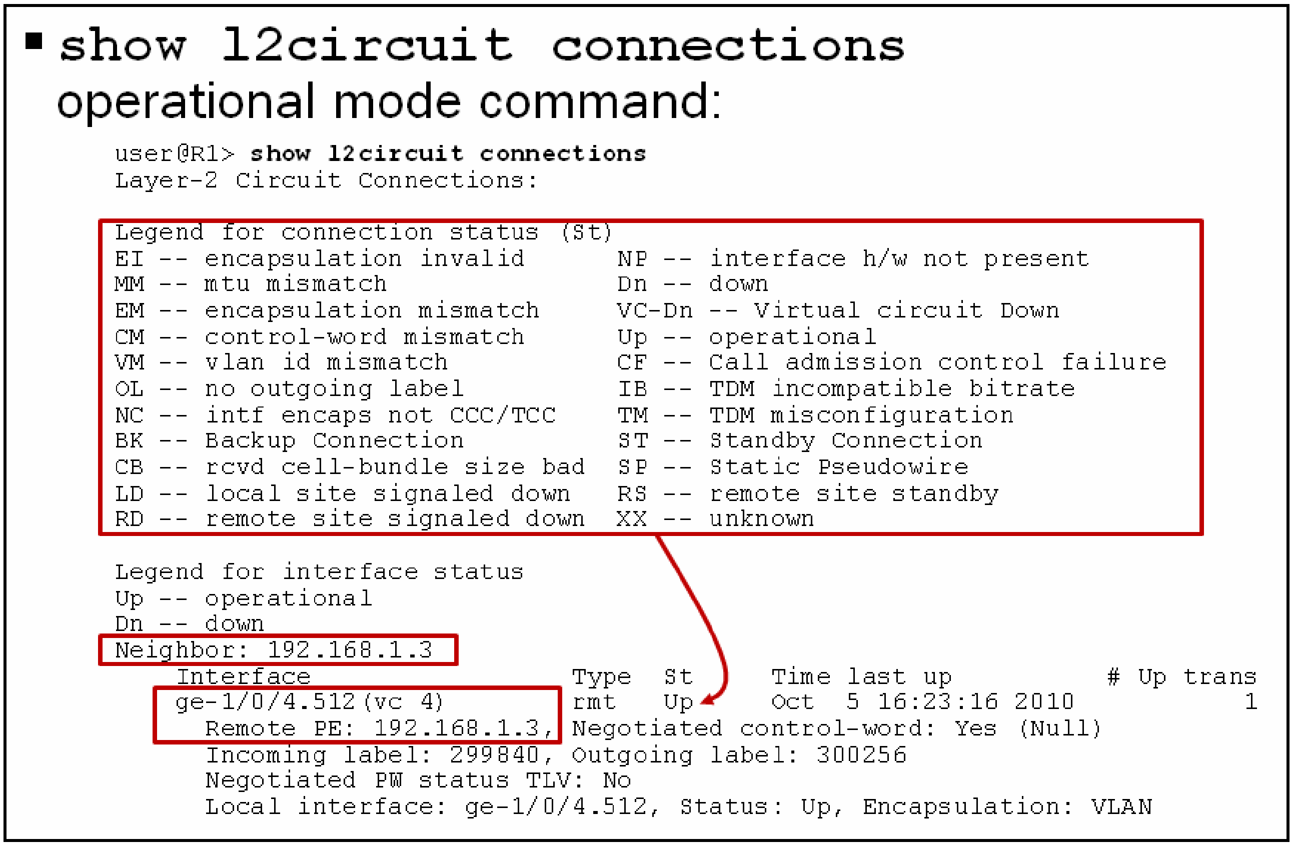
- `show ldp database` operational mode command:
  - Labels are associated with L2CKT for extended LDP neighbor

```
user@R1> show ldp database
Input label database, 192.168.1.1:0--192.168.1.3:0
  Label     Prefix
 300240     192.168.1.1/32
 300160     192.168.1.2/32
      3     192.168.1.3/32
 300256     L2CKT CtrlWord VLAN VC 4

Output label database, 192.168.1.1:0--192.168.1.3:0
  Label     Prefix
      3     192.168.1.1/32
 299776     192.168.1.2/32
 299824     192.168.1.3/32
 299840     L2CKT CtrlWord VLAN VC 4
...
```

To confirm that all is well with the operation of LDP, you should examine the LDP database using the **`show ldp database`** command. The highlight here is on the extended neighbor relationship to 192.168.1.3 and the presence of an LDP label (FEC) associated with an `L2CKT`.

The input label database for the 192.168.1.3 session shows the label that was advertised by the remote PE router (R3) for the connection identified as VC 4. This label (300256) is the label that the R1 PE router pushes onto packets received on the `ge-1/0/4.512` interface for transmission to the remote PE router. Similarly, you can see that the R1 PE router has advertised Label 299840 to R3 as the label it uses to associate received traffic with the `ge-1/0/4.512` interface connection.

Viewing Layer 2 Circuit Connections



- **show l2circuit connections** operational mode command:

```
user@R1> show l2circuit connections
Layer-2 Circuit Connections:

Legend for connection status (St)
EI -- encapsulation invalid      NP -- interface h/w not present
MM -- mtu mismatch               Dn -- down
EM -- encapsulation mismatch     VC-Dn -- Virtual circuit Down
CM -- control-word mismatch      Up -- operational
VM -- vlan id mismatch           CF -- Call admission control failure
OL -- no outgoing label          IB -- TDM incompatible bitrate
NC -- intf encaps not CCC/TCC    TM -- TDM misconfiguration
BK -- Backup Connection          ST -- Standby Connection
CB -- rcvd cell-bundle size bad  SP -- Static Pseudowire
LD -- local site signaled down   RS -- remote site standby
RD -- remote site signaled down  XX -- unknown

Legend for interface status
Up -- operational
Dn -- down
Neighbor: 192.168.1.3
    Interface                 Type  St     Time last up          # Up trans
    ge-1/0/4.512(vc 4)        rmt   Up     Oct  5 16:23:16 2010           1
      Remote PE: 192.168.1.3, Negotiated control-word: Yes (Null)
      Incoming label: 299840, Outgoing label: 300256
      Negotiated PW status TLV: No
      Local interface: ge-1/0/4.512, Status: Up, Encapsulation: VLAN
```

You can display the status of `l2circuits` with the **`show l2circuit connections`** operational-mode command.

The top of the display provides a legend for the connection and circuit status portion of each Layer 2 circuit. The circuit's incoming and outgoing labels are also displayed. Though not displayed here, you can include the `extensive` switch, which causes the output to list time-stamped entries, which indicate signaling and operational state changes for each Layer 2 connection.

Is the PE-CE Interface Up?

The lack of inherent IP connectivity between the PE and CE routers can make PE-CE interface troubleshooting problematic. Without the ability to conduct ping testing, you must rely on the absence or presence of physical layer and data-link layer alarms and status indications. For example, a loss of light (`LoL`) indication of a SONET link is a sure indication that physical layer problems are present on the PE-CE link. You can monitor ATM and Frame Relay links for proper permanent virtual connection (PVC) management protocol operation. In the case of ATM, you can issue **`ping atm`** to validate VC level connectivity to the attached CE device.

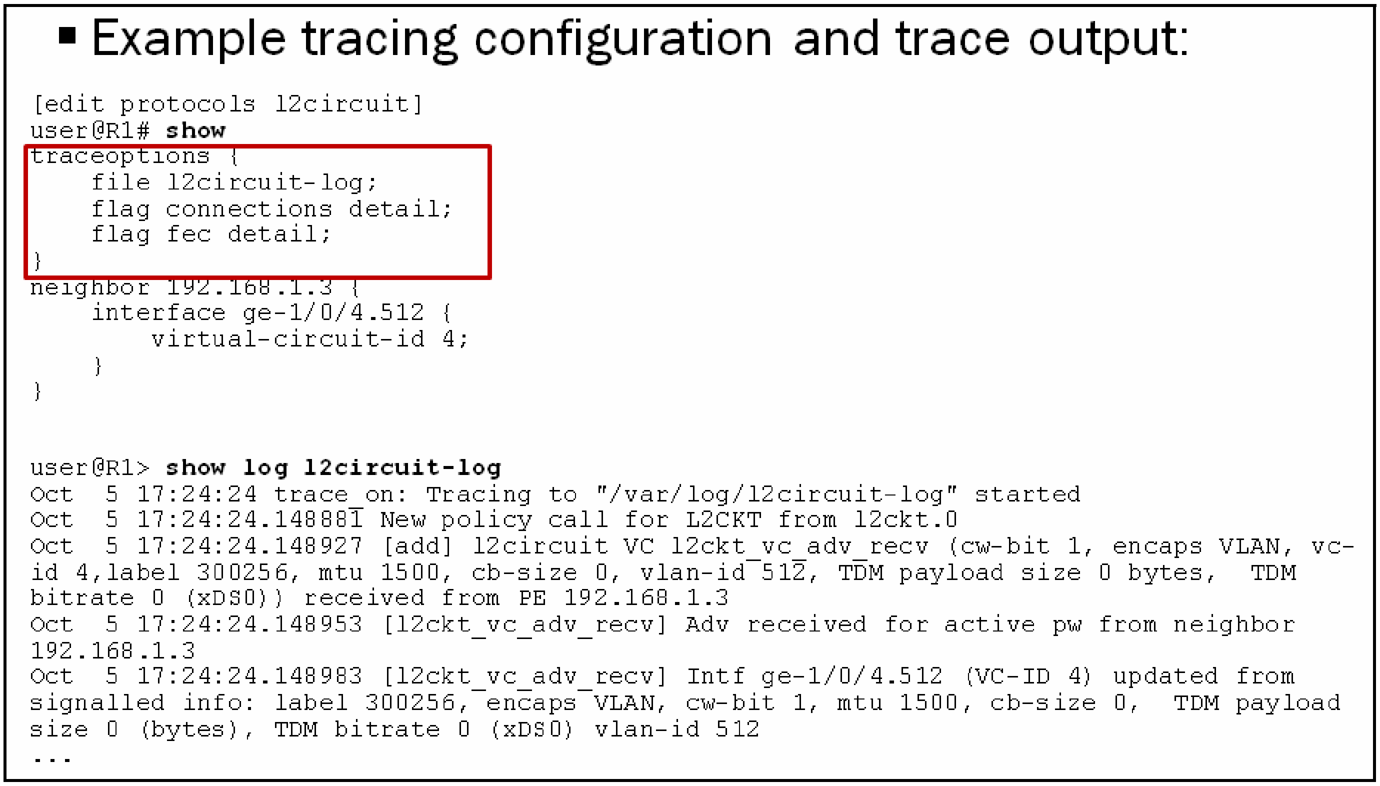
Compatible Circuit IDs

When the physical layer and data link layer operation of the PE-CE links appears normal, you should confirm that compatible connection identifiers are configured on the local PE-CE link. With VLAN tagging, you must ensure that the same VLAN ID values are configured on the remote PE-CE interface as well, unless you are using TCC encapsulation.

Out-of-Band Management

We recommend that the service provider provision a non-Layer 2 VPN connection between the PE and CE routers to simplify troubleshooting. This connection is normally just another logical unit on the existing PE-CE interface. However, it has the family `inet` and compatible IP addressing configured. You can use the resulting logical IP subnet to verify PE-CE interface operation and to enable Ping, Telnet, FTP, and other such services between the PE and CE routers.

LDP Layer 2 Circuit Traceoptions

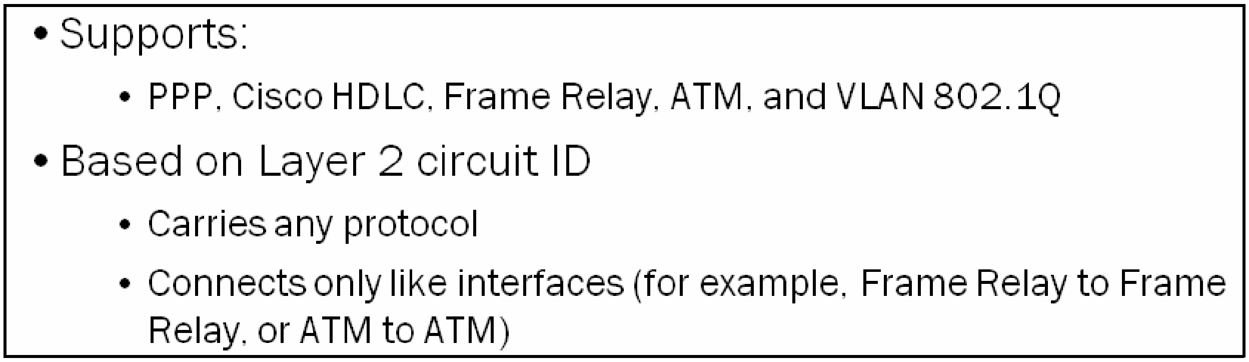


- Example tracing configuration and trace output:

```
[edit protocols l2circuit]
user@R1# show
traceoptions {
    file l2circuit-log;
    flag connections detail;
    flag fec detail;
}
neighbor 192.168.1.3 {
    interface ge-1/0/4.512 {
        virtual-circuit-id 4;
    }
}


user@R1> show log l2circuit-log
Oct  5 17:24:24 trace_on: Tracing to "/var/log/l2circuit-log" started
Oct  5 17:24:24.148881 New policy call for L2CKT from l2ckt.0
Oct  5 17:24:24.148927 [add] l2circuit VC l2ckt_vc_adv_recv (cw-bit 1, encaps VLAN, vc-
id 4,label 300256, mtu 1500, cb-size 0, vlan-id 512, TDM payload size 0 bytes,  TDM
bitrate 0 (xDS0)) received from PE 192.168.1.3
Oct  5 17:24:24.148953 [l2ckt_vc_adv_recv] Adv received for active pw from neighbor
192.168.1.3
Oct  5 17:24:24.148983 [l2ckt_vc_adv_recv] Intf ge-1/0/4.512 (VC-ID 4) updated from
signalled info: label 300256, encaps VLAN, cw-bit 1, mtu 1500, cb-size 0,  TDM payload
size 0 (bytes), TDM bitrate 0 (xDS0) vlan-id 512
...
```

Layer 2 circuit tracing can provide invaluable assistance when troubleshooting LDP-based Layer 2 circuit operational problems. This graphic shows an example of tracing parameters and a portion of the tracing output generated.

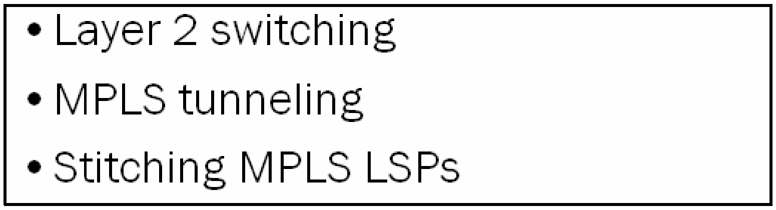
Connects Two Layer 2 Sites



- Supports:
  - PPP, Cisco HDLC, Frame Relay, ATM, and VLAN 802.1Q
- Based on Layer 2 circuit ID
  - Carries any protocol
  - Connects only like interfaces (for example, Frame Relay to Frame Relay, or ATM to ATM)

CCC allows you to configure transparent connections between two sites, where the circuit can be a Frame Relay data-link connection identifier (DLCI), an ATM VC, a Point-to-Point Protocol (PPP) interface, a Cisco High-Level Data Link Control (HDLC) interface, or an MPLS LSP. Using CCC, packets from the source circuit are delivered to the destination circuit with, at most, the Layer 2 address being changed. No other processing—such as header checksums, time-to-live (TTL) decrementing, or protocol processing—is done.

Cross-Connect Types

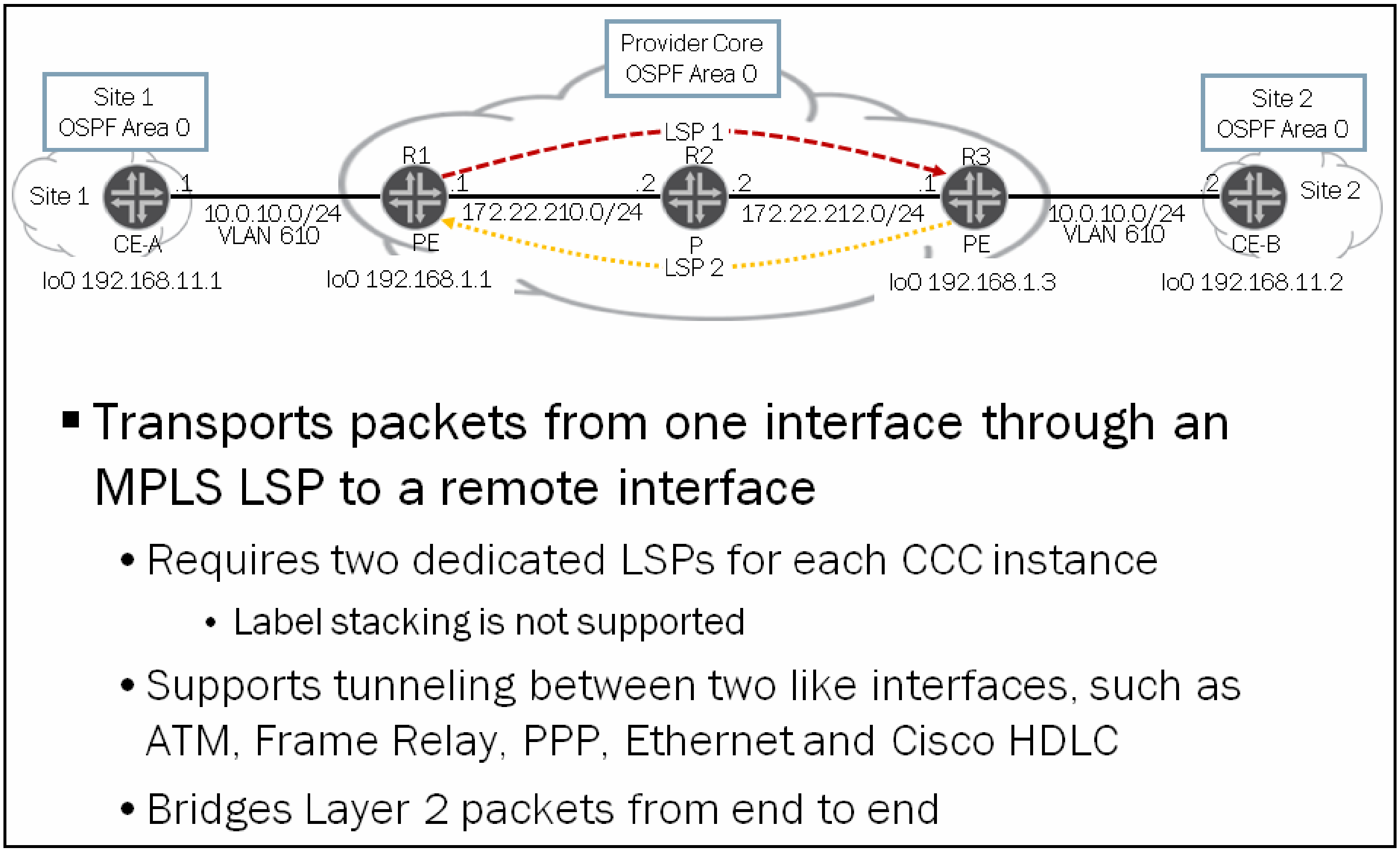


- Layer 2 switching
- MPLS tunneling
- Stitching MPLS LSPs

CCC circuits fall into two categories: logical interfaces, which include DLCIs, VCs, and PPP and Cisco HDLC interfaces; and LSPs. The two circuit categories provide the following three types of cross-connect:

- *Layer 2 switching*: Cross-connects between logical interfaces provide what is essentially Layer 2 switching. The interfaces that you connect must be of the same type.
- *MPLS tunneling*: Cross-connects between interfaces and LSPs allow you to connect two distant interface circuits of the same type by creating MPLS tunnels that use LSPs as the conduit.
- *LSP stitching*: Cross-connects between LSPs provide a way to *stitch* together two LSPs, including paths that fall in two different traffic engineering database (TED) areas.

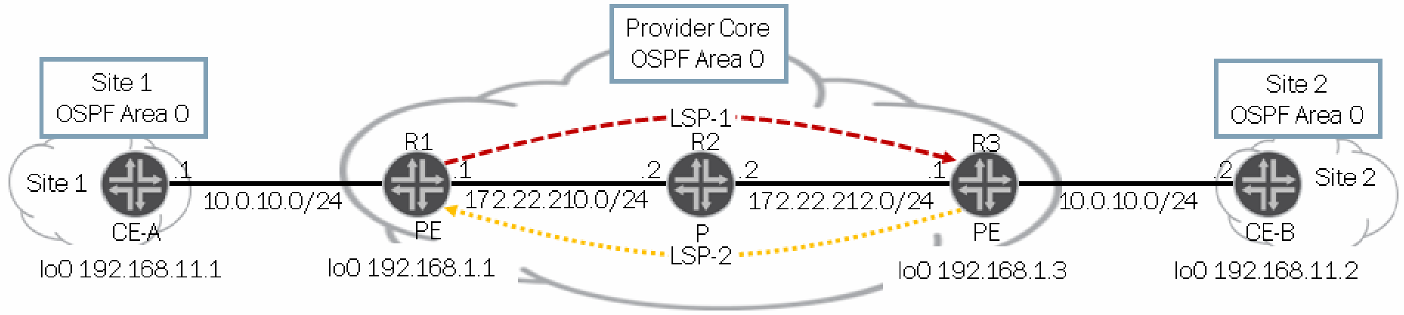
CCC MPLS Interface Tunneling



- Transports packets from one interface through an MPLS LSP to a remote interface
  - Requires two dedicated LSPs for each CCC instance
    - Label stacking is not supported
  - Supports tunneling between two like interfaces, such as ATM, Frame Relay, PPP, Ethernet and Cisco HDLC
  - Bridges Layer 2 packets from end to end

CCC requires that you have 2 dedicated LSPs to accommodate transmit and receive traffic for every CCC connection configuration. This is because CCC does not support label stacking like Layer 2 circuits, Layer 2 VPNs and VPLS. This is one of the primary reasons that CCC is not a scalable solution for larger networks. CCC allows you to connect two ATM, Frame Relay, PPP, Ethernet, or Cisco HDLC access links using an MPLS tunnel. Layer 2 packets are essentially bridged from end to end in this configuration. In the preceding figure, MPLS LSPs connect two Ethernet networks across an IP cloud. The Ethernet interface on the R1 expects a VLAN value of 610 (on whatever path is enabled on that interface). R3 will also transmit and receive using VLAN 610 (on whatever path is enabled on the output interface). The IP backbone between the two routers has two LSPs—one in each direction—that connect the two PE routers. When the packets come in to R1 destined to the network attached to R3, R1 places an MPLS header on the packets and transmit them down the LSP. Once the packets reaches R3, the MPLS headers are stripped off, and the packet are forwarded out the CE facing interface.

Configuration Example



```
[edit protocols]
user@R1# show
connections {
    remote-interface-switch vpn-a {
        interface ge-1/0/4.610;
        transmit-lsp LSP-1;
        receive-lsp LSP-2;
    }
}
```

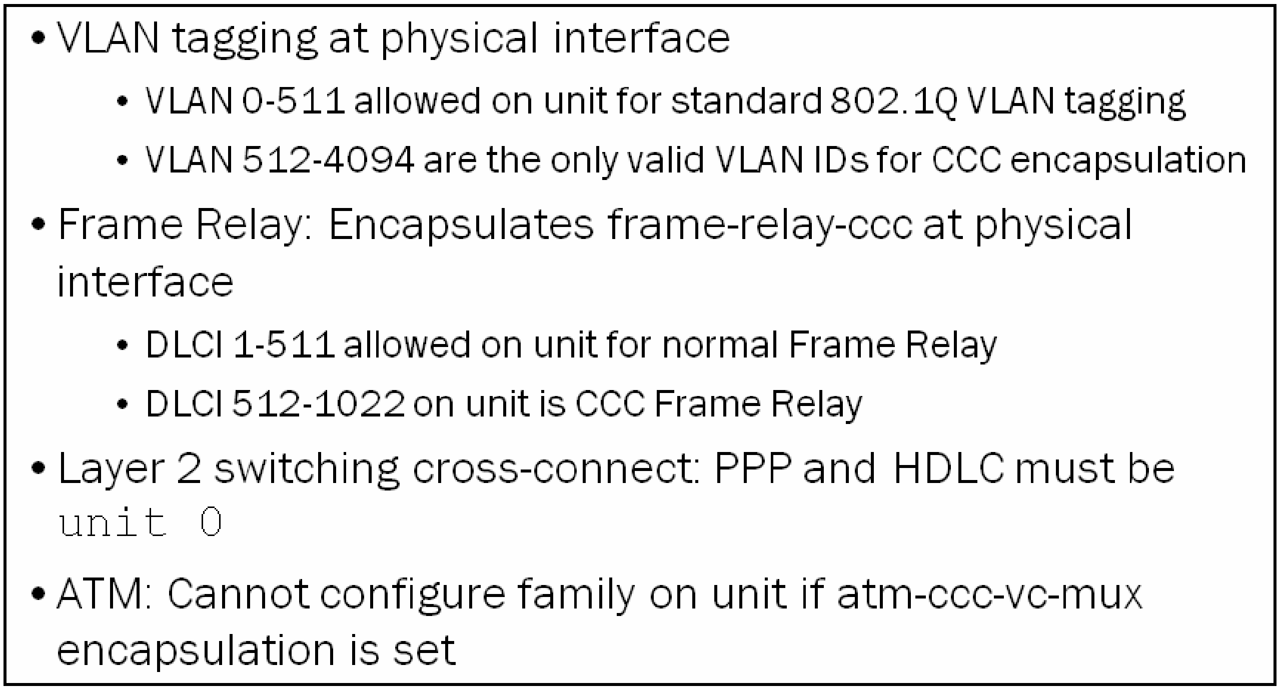
```
[edit protocols]
user@R3# show
connections {
    remote-interface-switch vpn-a {
        interface ge-1/0/4.610;
        transmit-lsp LSP-2;
        receive-lsp LSP-1;
    }
}
```

The configuration examples on the graphic show that the receive LSP on one router is the transmit LSP on the other router. The names referenced are the names of the transmit or receive LSPs displayed when you issue the **show mpls lsp** command.

To configure LSP tunnel cross-connects, you must also configure the CCC encapsulation on the ingress and egress router's CE facing interfaces. Below is an example from R1:

```
[edit]
user@R1# show interfaces ge-1/0/4
vlan-tagging;
encapsulation vlan-ccc;
unit 610 {
    encapsulation vlan-ccc;
    vlan-id 610;
}
```

CCC Caveats

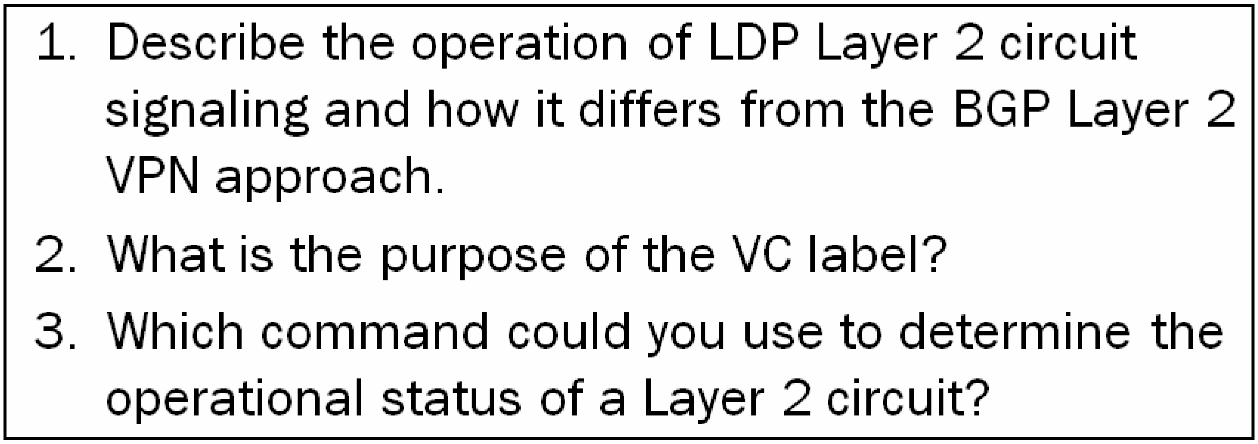


- VLAN tagging at physical interface
  - VLAN 0-511 allowed on unit for standard 802.1Q VLAN tagging
  - VLAN 512-4094 are the only valid VLAN IDs for CCC encapsulation
- Frame Relay: Encapsulates frame-relay-ccc at physical interface
  - DLCI 1-511 allowed on unit for normal Frame Relay
  - DLCI 512-1022 on unit is CCC Frame Relay
- Layer 2 switching cross-connect: PPP and HDLC must be `unit 0`
- ATM: Cannot configure family on unit if atm-ccc-vc-mux encapsulation is set

There are a variety of caveats for configuring CCC:

- *VLAN-ID number*: If the VLAN CCC encapsulation is not specified, GE/FE interfaces support VLAN-IDs from 0 to 4094. Regardless of the range of numbers supported, there is a limit of 1024 logical units. If the VLAN CCC encapsulation at the physical interface level is specified, then on logical units that do VLAN CCC, the VLAN CCC encapsulation is specified again AND the VLAN-ID must fall in the range of 512 to 4094. Logical units between 0 and 511 only support normal IEEE 802.1Q VLAN tagging.
- *Frame Relay*: The only issue with Frame Relay is the DLCI range. As stated earlier, when the physical interface is configured for Frame Relay CCC encapsulation, the logical units can be either normal Frame Relay interfaces or they can be CCC Frame Relay interfaces. Normal Frame Relay logical interfaces use a DLCI value between 1 and 511. CCC Frame Relay logical interfaces use a DLCI value between 512 and 1022. Additionally, the Frame Relay CCC encapsulation must also be configured on the logical interface.
- *PPP and Cisco HDLC*: Because both protocols are point-to-point serial protocols, the logical `unit` can be `0` only. This is not a requirement of the CCC capability, but a requirement of the physical-layer encapsulation.
- *ATM*: If an ATM interface is configured for `atm-ccc-vc-mux` encapsulation (which is another way of saying CCC), no families can be configured on the logical interface. CCC only works for ATM Adaptation Layer 5, unless Cell Relay is configured.

Review Questions



1. Describe the operation of LDP Layer 2 circuit signaling and how it differs from the BGP Layer 2 VPN approach.
2. What is the purpose of the VC label?
3. Which command could you use to determine the operational status of a Layer 2 circuit?

Answers to Review Questions

1.

LDP Layer 2 circuit signaling exchanges virtual circuit labels with targeted peers to indicate what parameters are needed to establish a session. Layer 2 circuits also uses these values to uniquely identify circuit connection to ensure traffic is delivered to the correct networks. It differs in that, Layer 2 circuits require the use of LDP to carry the virtual circuit information to remote peers. Some other differences are that Layer 2 circuits do not require a VRF instance configuration and do not require a route-distinguisher or VRF target policy, instead LDP Layer 2 circuits use the virtual circuit ID to identify which incoming circuit-connection requests are allowed. Also LDP Layer 2 circuits can only be used to connect remote sites and can not be used to connect local sites which are connected to the same PE router.

2.

The virtual circuit label is used to send circuit information to targeted remote PE routers. The remote router uses this label value as its output label when forwarding traffic associated with this FEC back to the originating router.

3.

The command to display the operational status of a LDP Layer 2 circuit is **`show l2circuit connections`**.

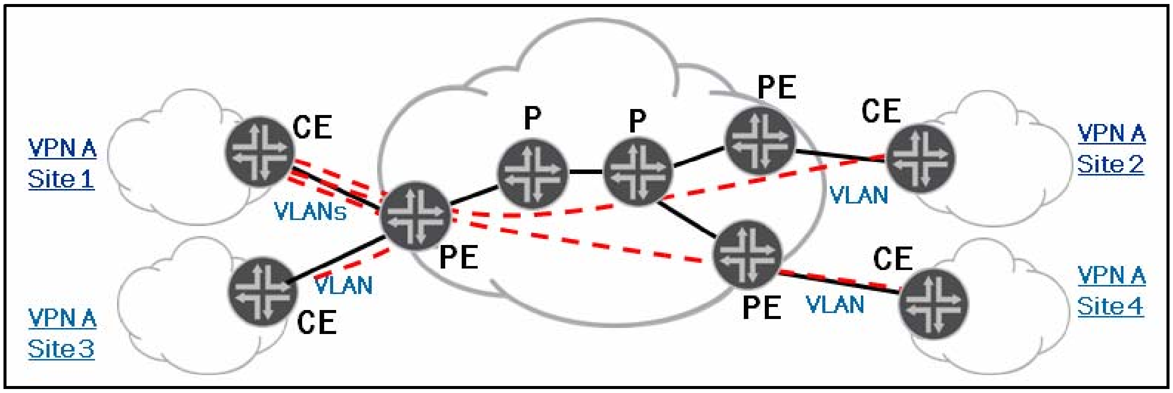
JNCIS-SP Study Guide—Part 3

Chapter 16: Virtual Private LAN Service

This Chapter Discusses:

- The difference between Layer 2 MPLS virtual private networks (VPNs) and virtual private LAN service (VPLS);
- The purpose of the provider edge (PE), customer edge (CE), and provider (P) devices;
- Provisioning of CE and PE routers;
- The signaling process of VPLS;
- The learning and forwarding process of VPLS; and
- The potential loops in a VPLS environment.

Layer 2 VPNs Are Point-to-Point

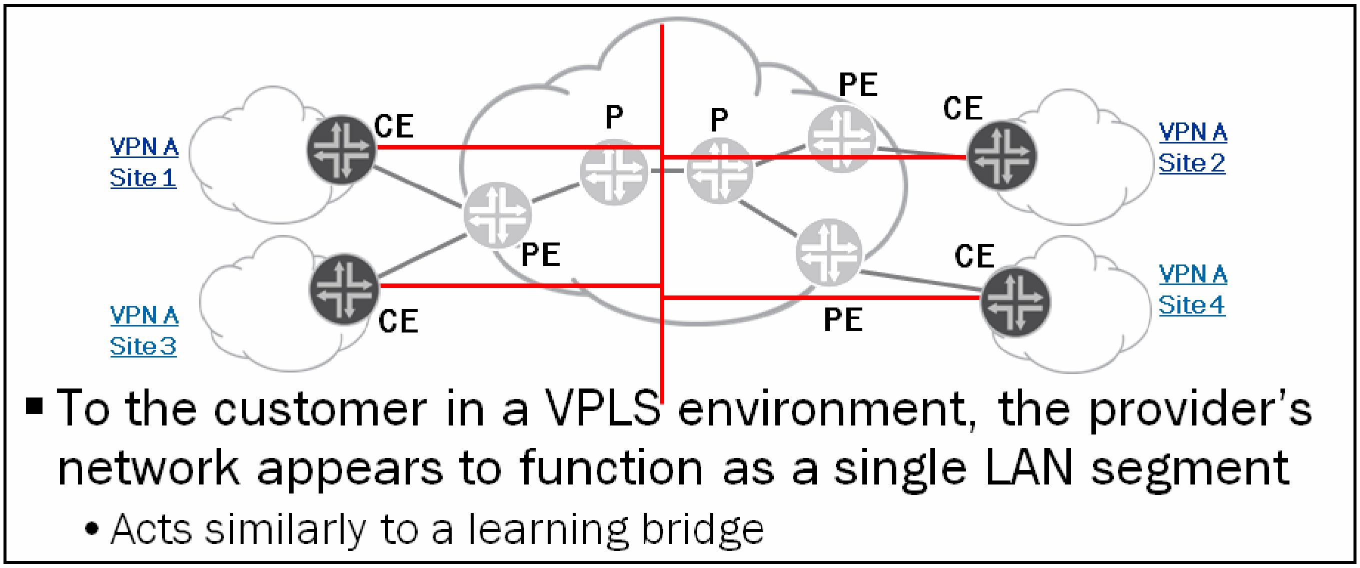


Border Gateway Protocol (BGP) Layer 2 VPNs and LDP Layer 2 circuits are point to point in nature and support Ethernet, Frame Relay, Point-to-Point Protocol (PPP), and Cisco's High-Level Data Link Control (Cisco HDLC). Even though Ethernet media can be used between PE and CE devices, only two CE devices can interact over a single emulated Layer 2 circuit or virtual LAN (VLAN). Although this behavior works well as it is, some customers prefer to have their Ethernet media behave like Ethernet so that more than two hosts or routers can interact over the Layer 2 circuit. This need on behalf of the customer, and other factors, is what led to the development of VPLS.

Mapping Local Circuits to Remote Sites

In both of the Layer 2 point to point VPNs scenarios, you must manually map local Layer 2 circuits on the PE device to the remote sites. This mapping might be labor intensive and sometimes confusing, especially when designing a full-mesh network between PE devices. BGP-based Layer 2 VPNs allows for over-provisioning, which eases the process of adding a new site, however.

Appearing to Be a Single LAN Segment



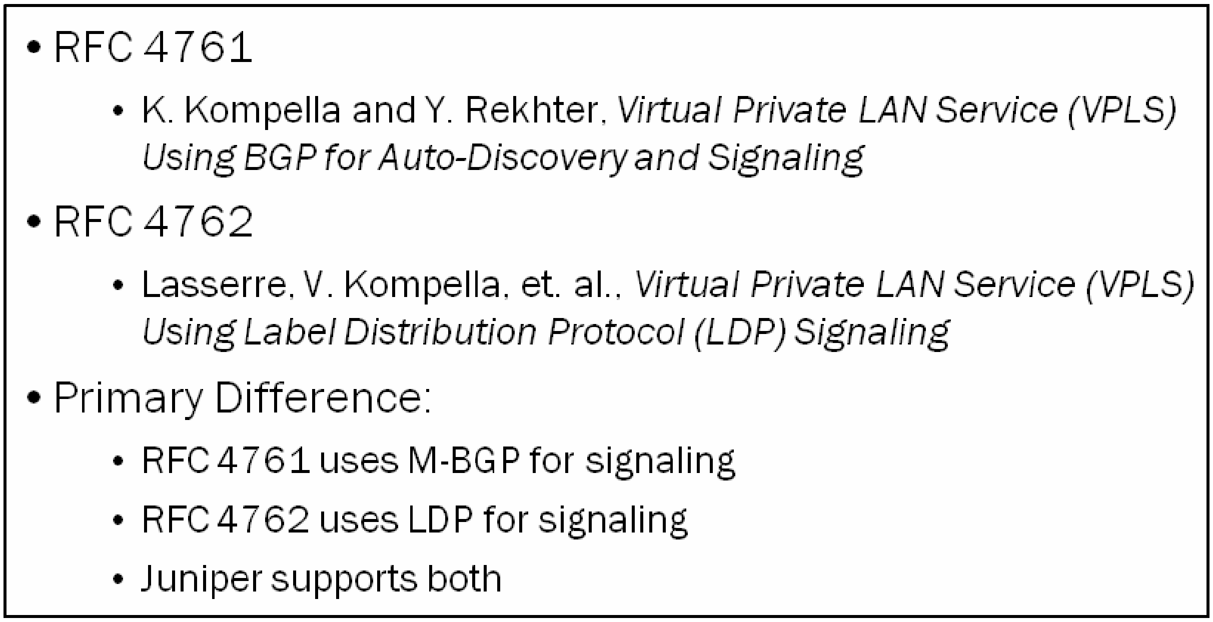
- To the customer in a VPLS environment, the provider's network appears to function as a single LAN segment
  - Acts similarly to a learning bridge

A new service that can be provided to the customer is VPLS. To the customer, a VPLS appears to be a single LAN segment. In fact, it appears to act similarly to a learning bridge. That is, when the destination media access control (MAC) address is not known, an Ethernet frame is sent to all remote sites. If the destination MAC address is known, it is sent directly to the site that owns it.

No Need to Map Local Circuit to Remote Sites

In a VPLS, PE devices learn MAC addresses from the frames that it receives. They will use the source and destination addresses to dynamically create a forwarding table (`vpn-name.vpls`) for Ethernet frames. Based on this table, frames are forwarded out directly connected interfaces or over MPLS label-switched paths (LSPs) across the provider core. This behavior allows you to not have to manually map Layer 2 circuits to remote sites.

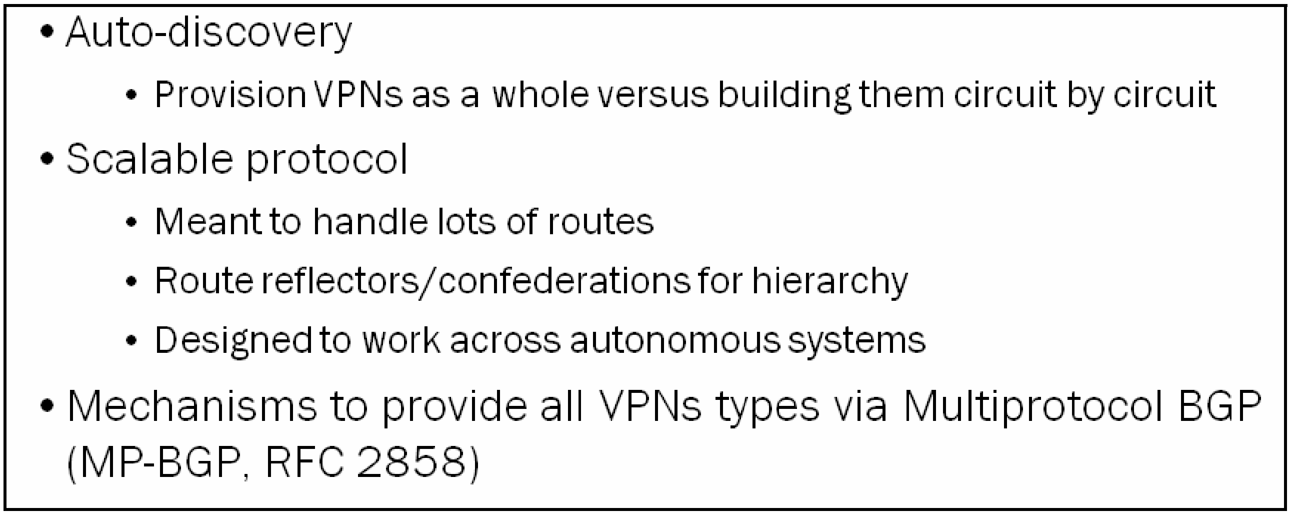
Standards for VPLS



- RFC 4761
  - K. Kompella and Y. Rekhter, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*
- RFC 4762
  - Lasserre, V. Kompella, et. al., *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*
- Primary Difference:
  - RFC 4761 uses M-BGP for signaling
  - RFC 4762 uses LDP for signaling
  - Juniper supports both

Two competing RFCs for VPLS exist. One of the remarkable things about these two competing RFCs is that their primary developers happen to be brothers, Kireeti Kompella (BGP) and Vach Kompella (LDP). The primary difference between the two RFCs is that one uses BGP and one uses LDP for signaling. Currently the Junos operating system supports both LDP and BGP for signaling, with BGP the preferred solution.

Benefits of BGP

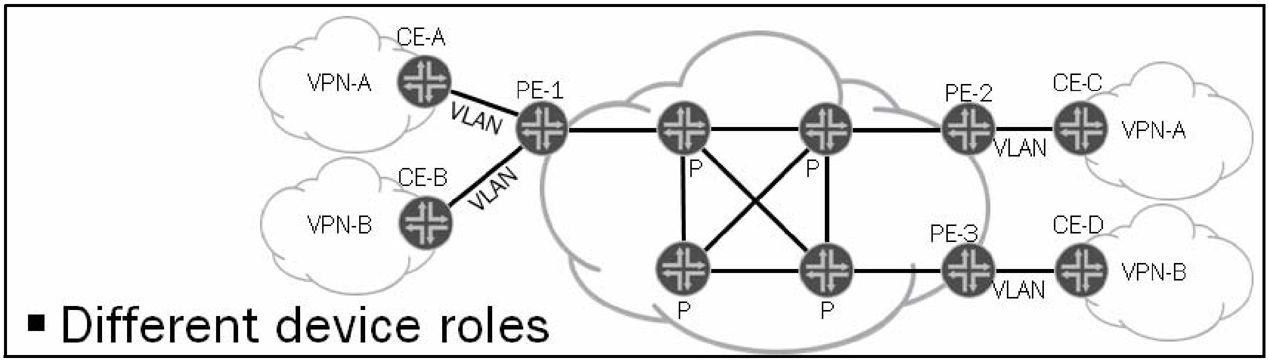


- Auto-discovery
  - Provision VPNs as a whole versus building them circuit by circuit
- Scalable protocol
  - Meant to handle lots of routes
  - Route reflectors/confederations for hierarchy
  - Designed to work across autonomous systems
- Mechanisms to provide all VPNs types via Multiprotocol BGP (MP-BGP, RFC 2858)

There are a few benefits to using BGP as the signaling protocol for VPLS. For instance, BGP allows for the auto-discovery of new sites as they are added to a VPLS. When a new site is added to a VPLS, you only need to configure the PE router connected to the new site. All other PE routers discover the new site with the use of the target extended community. Also, BGP is a very scalable protocol. BGP works well when dealing with large number of routes. Provisioning a VPLS network in a large provider network can be made easier with the use of route reflectors and/or confederations. Finally, BGP was designed to advertise routes between autonomous systems. Thus, it is inherently possible to build a VPLS across autonomous system (AS) boundaries.

Based on RFC 2858 (*Multiprotocol Extensions for BGP-4*), BGP can be extended to carry information for which it was not originally designed. The BGP draft for VPLS relies on Multiprotocol Border Gateway Protocol (MP-BGP) to carry its routing information between PE devices.

Customer Edge Devices



- Different device roles

The customer edge device is normally a router or Layer 2 switch that provides access to the provider's edge device. Because the Layer 2 frames generated by the customer are carried across the core using MPLS, there is inherent independence between the Layer 2 technology used at the provider's edge and the technologies used in the core. This independence extends to the upper protocol layers as well, because the provider does not interpret in any way the contents of the Layer 2 frames.

Both ends of a VPLS must use the Ethernet technology. Unlike point to point Layer 2 VPNs, each remote site does not need to be associated with a unique Layer 2 circuit identifier to map traffic to a given site. All mapping will be performed automatically through the MAC learning function of a PE.

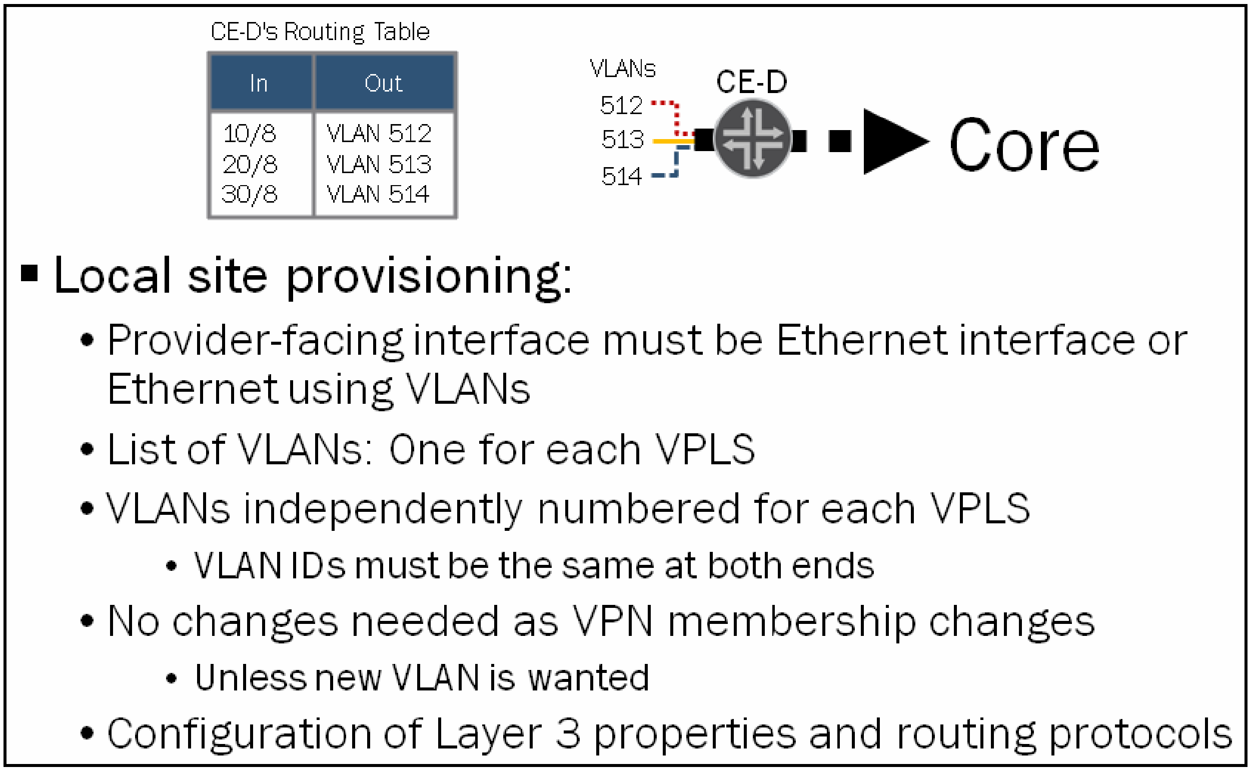
Provider Edge Routers

The provider edge routers connect to customer sites and maintain VPLS-specific information. This VPN information is obtained through local configuration and through signaling exchanges with either BGP or LDP. As with a Layer 3 VPN, the PE routers forward traffic across the provider's core using MPLS LSPs. PE routers perform MAC learning and store MACs in a VPLS-specific MAC table.

Provider Routers

The provider routers do not carry any VPLS state. They simply provide label-switching router (LSR) services to facilitate the transfer of labeled frames between PE routers.

Provisioning the Local CE Device



CE-D's Routing Table

In	Out
10/8	VLAN 512
20/8	VLAN 513
30/8	VLAN 514

VLANs
512
513
514
CE-D
Core

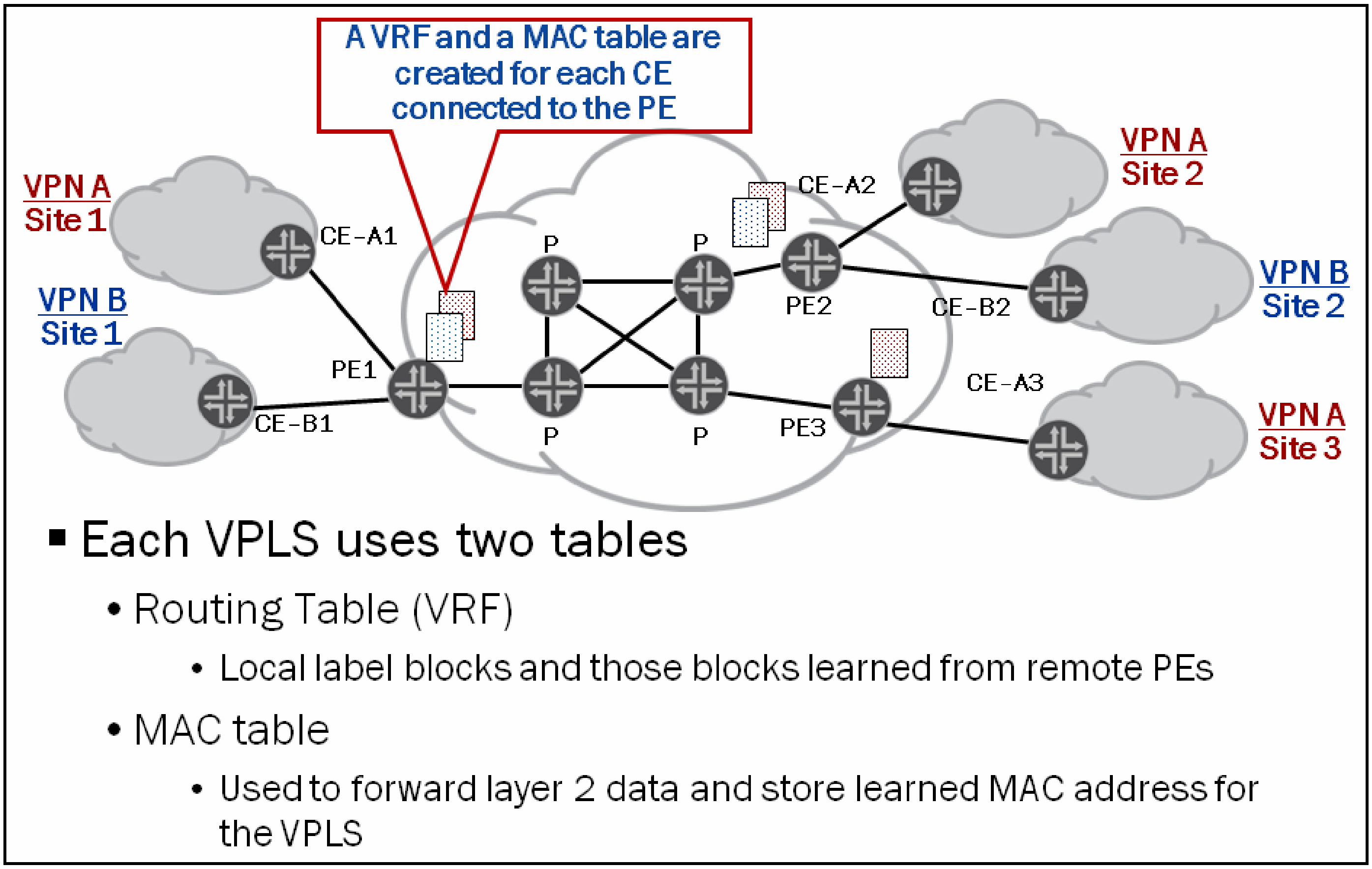
- Local site provisioning:
  - Provider-facing interface must be Ethernet interface or Ethernet using VLANs
  - List of VLANs: One for each VPLS
  - VLANs independently numbered for each VPLS
    - VLAN IDs must be the same at both ends
  - No changes needed as VPN membership changes
    - Unless new VLAN is wanted
  - Configuration of Layer 3 properties and routing protocols

The first step in building a VPLS is the configuration of the local CE device. This configuration normally entails assigning a range of Layer 2 circuit identifiers to logical interfaces on the CE device and having the correct encapsulation settings.

For Ethernet with VLAN tagging, it is required that VLAN IDs be the same at both ends of a VPLS. The VPLS standard allows for the expansion of VPN membership without reconfiguring existing sites.

The CE device also requires the configuration of upper-layer protocols to be compatible with the remote CE router. Unlike a Layer 3 VPN solution, the PE router has no IP or routing protocol configuration because these functions are configured on the CE routers with end-to-end significance. With VPLS, the CE routers form adjacencies with each other as if they were connected to the same Ethernet segment, as opposed to becoming adjacent to the local PE router.

VPLS Route and Forwarding Tables



A VRF and a MAC table are created for each CE connected to the PE

- Each VPLS uses two tables
  - Routing Table (VRF)
    - Local label blocks and those blocks learned from remote PEs
  - MAC table
    - Used to forward layer 2 data and store learned MAC address for the VPLS

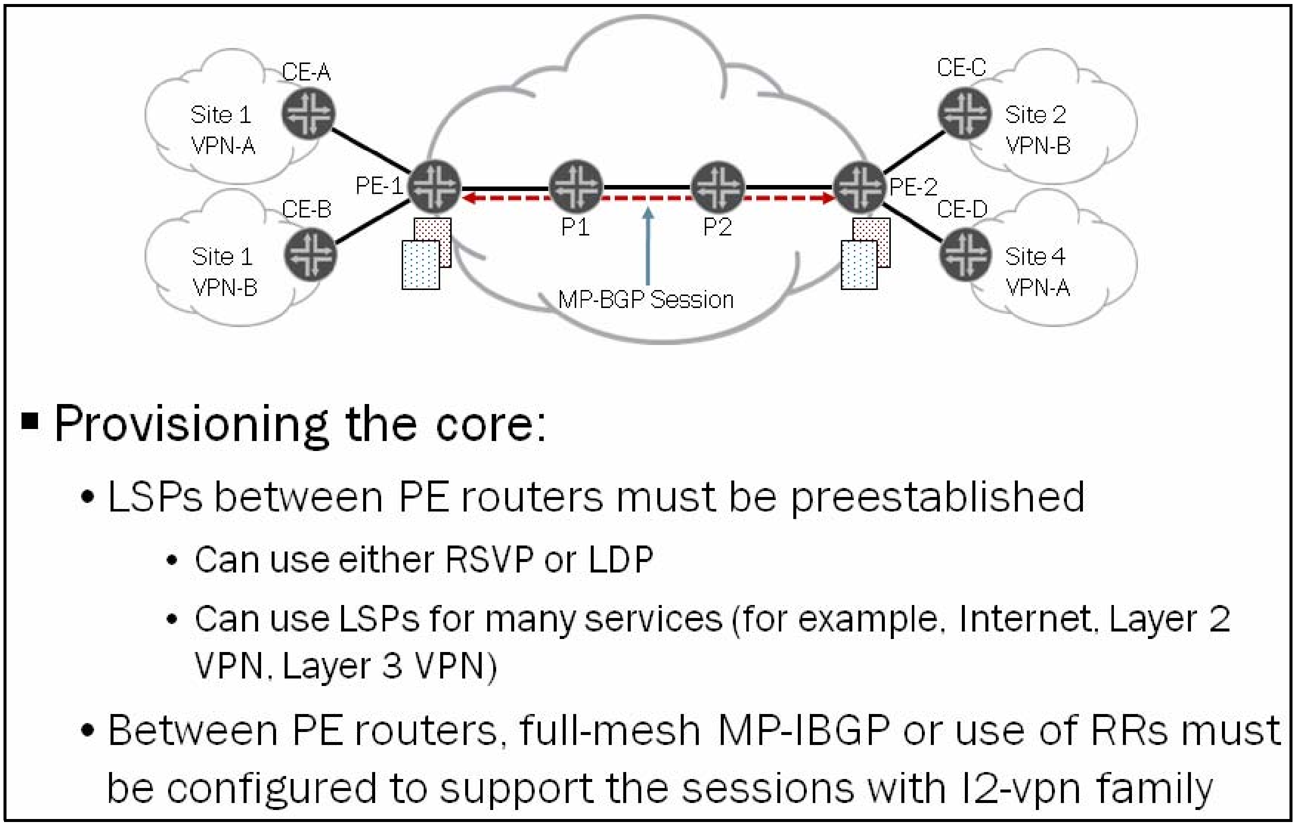
A VPN routing and forwarding table (VRF) and a VPLS-specific MAC-table are created in the PE router for each VPLS. The VRF table is populated with information provisioned for the local CE device and contains:

- The local site ID;
- The site's Layer 2 encapsulation;
- The logical interfaces provisioned to the local CE device; and
- A label base used to associated received traffic with one of the logical interfaces.

The VRF is also populated with information received from other PE routers in MP-IBGP updates. These updates contain the remote site's ID, label base, label, offset, and Layer 2 encapsulation.

The combination of locally provisioned information and Layer 2 VPN network layer reachability information (NLRI) received from remote PE routers results in a Layer 2 VPN VRF table and an associate MAC table which are used to map traffic to and from the LSPs connecting the PE routers.

Provisioning the Core

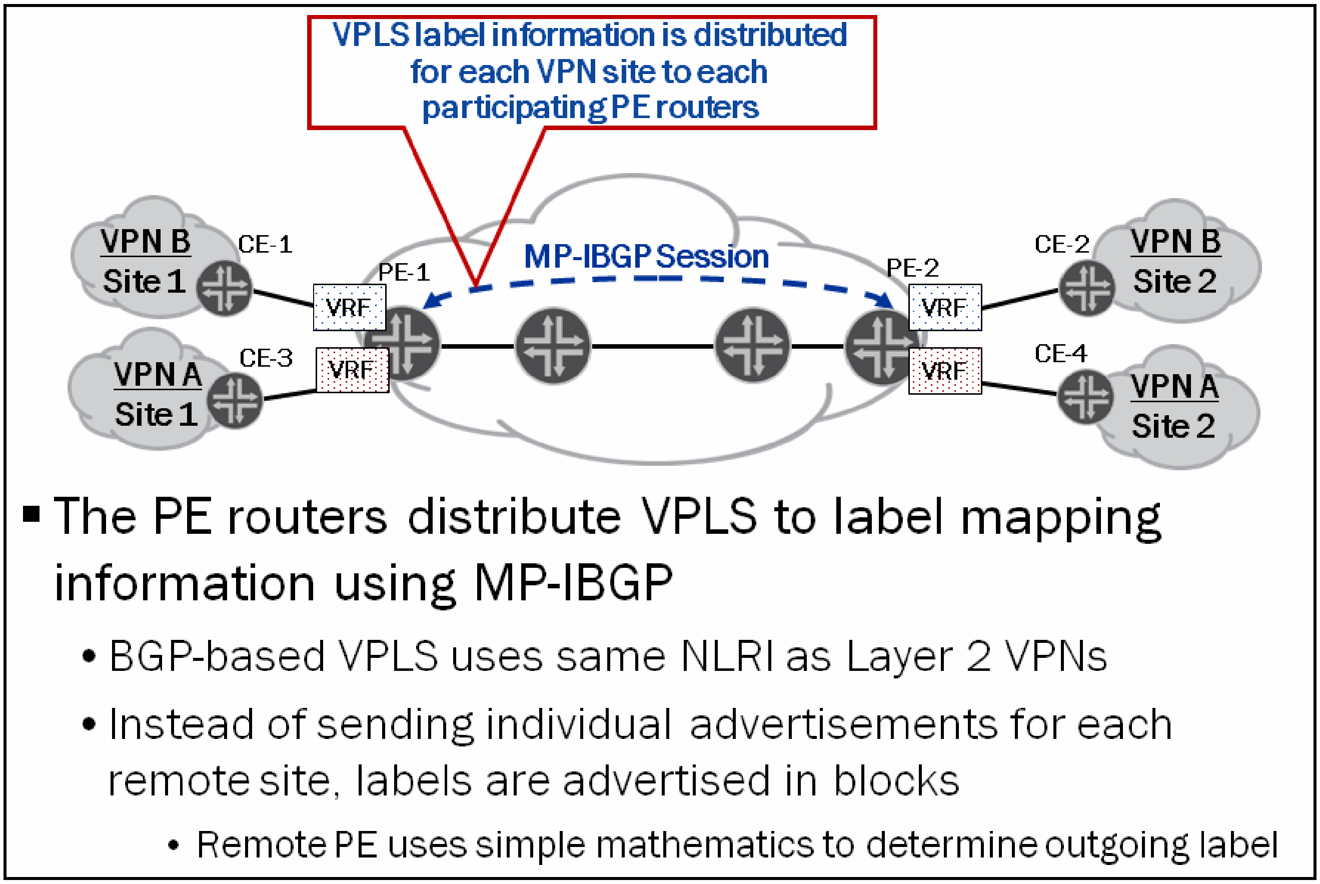


- Provisioning the core:
  - LSPs between PE routers must be preestablished
    - Can use either RSVP or LDP
    - Can use LSPs for many services (for example, Internet, Layer 2 VPN, Layer 3 VPN)
  - Between PE routers, full-mesh MP-IBGP or use of RRs must be configured to support the sessions with l2-vpn family

As with a Layer 3 VPN, the provider's core must be provisioned to support the Layer 2 VPN service. Besides a functional interior gateway protocol (IGP), this support normally involves the establishment of MPLS LSPs between PE routers to be used for data forwarding. The PE-PE LSPs are not dedicated to any particular service. With label stacking, the same LSP can be used to support multiple VPLS customers while also supporting Layer 3 VPNs and non-VPN traffic.

Each PE router must also be configured with MP-BGP to peer to other PE routers having local sites belonging to the same VPN. These MP-BGP sessions must be configured to support the **`l2-vpn signaling`** address family so that they can send and receive VPLS NLRI updates.

VPLS Label Distribution

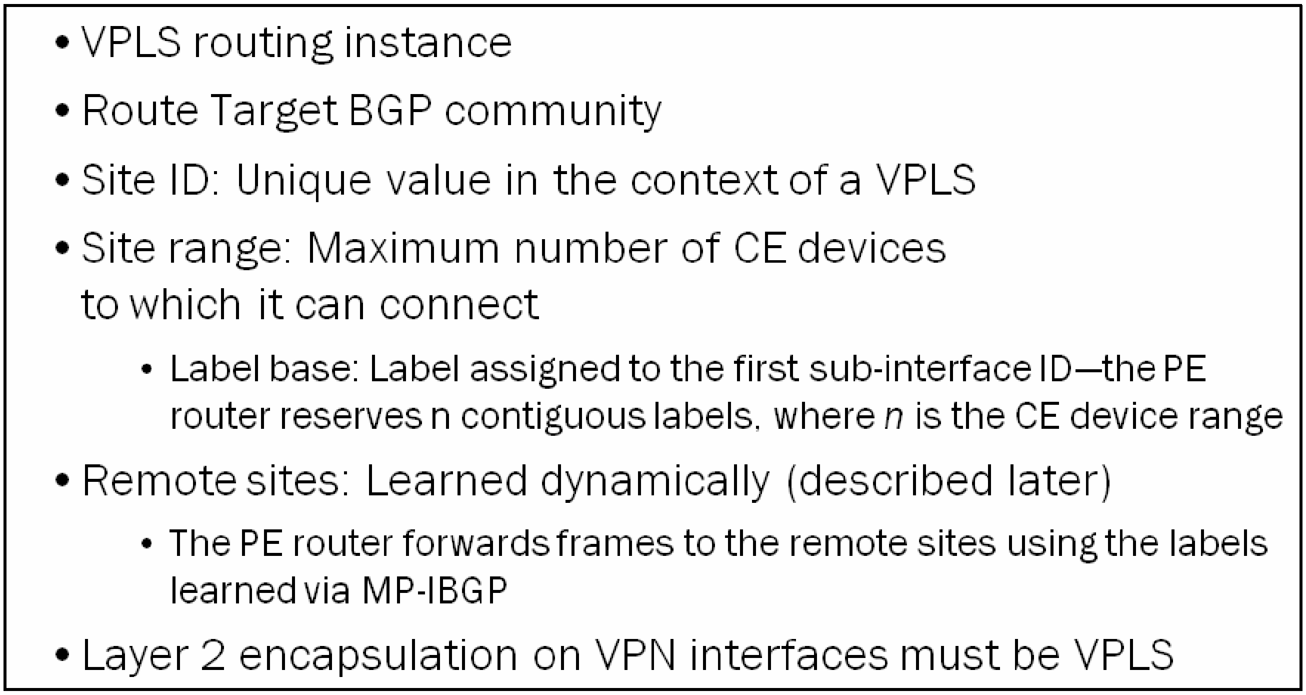


VPLS label information is distributed for each VPN site to each participating PE routers

- The PE routers distribute VPLS to label mapping information using MP-IBGP
  - BGP-based VPLS uses same NLRI as Layer 2 VPNs
  - Instead of sending individual advertisements for each remote site, labels are advertised in blocks
    - Remote PE uses simple mathematics to determine outgoing label

PE routers exchange MPLS label information using the same MP-BGP NLRI as Layer 2 VPNs. For a given site, a PE will advertise a block of labels that can be used by remote PEs to forward traffic to the sending PE. Using simple mathematics receiving PEs, can determine which label to use to reach the sending PE.

Provisioning the PE Router



- VPLS routing instance
- Route Target BGP community
- Site ID: Unique value in the context of a VPLS
- Site range: Maximum number of CE devices to which it can connect
  - Label base: Label assigned to the first sub-interface ID—the PE router reserves n contiguous labels, where *n* is the CE device range
- Remote sites: Learned dynamically (described later)
  - The PE router forwards frames to the remote sites using the labels learned via MP-IBGP
- Layer 2 encapsulation on VPN interfaces must be VPLS

After configuring the local CE device properties, you must provision the site's VRF on the PE router. The following list shows what is typically involved:

- Specification of route targets or VRF policy.
- CE device identifier (site ID), which must be unique in the context of a specific VPN.
- CE device range, which helps determine the size of the site's label block and therefore how many remote sites to which it can connect.
- Logical interfaces associated with this VRF.

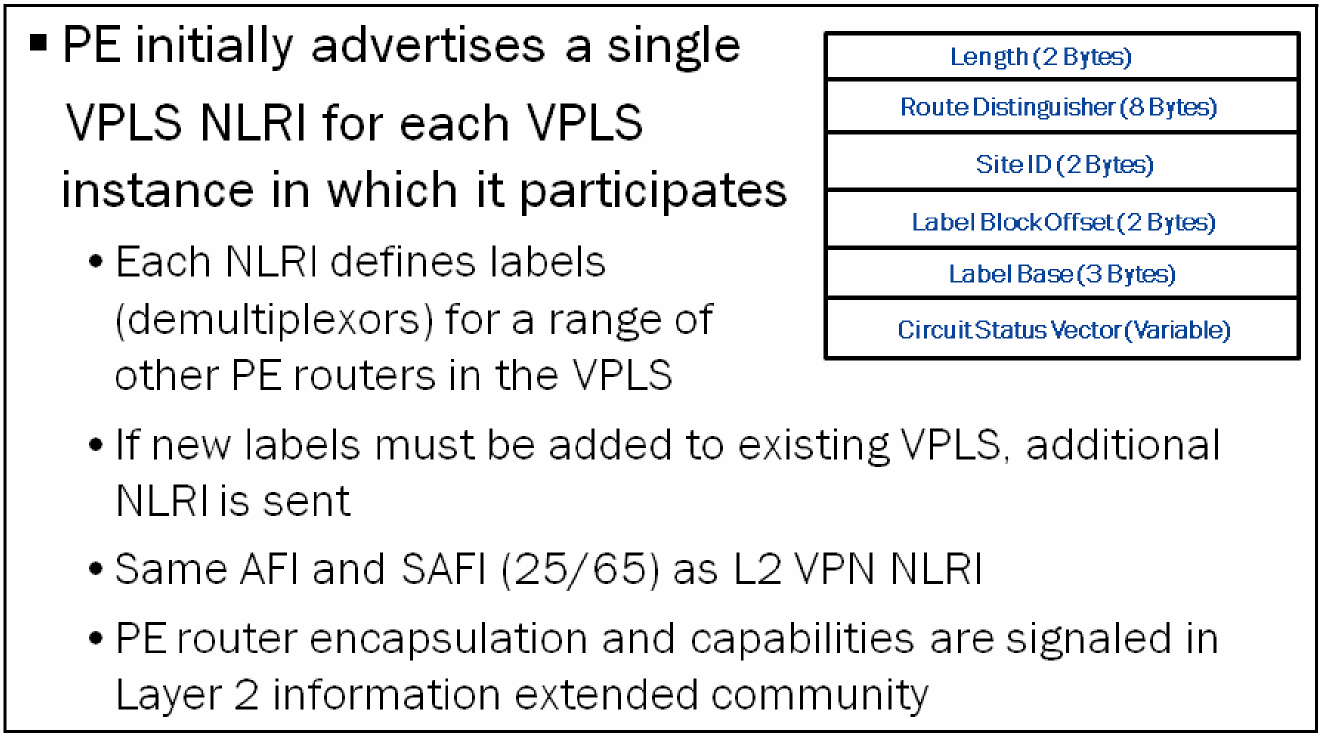
Some of the steps outlined in the graphic can occur automatically and therefore do not require explicit configuration.

PE Router Layer 2 Configuration

Each VPLS interface must be configured to support `family vpls` and the appropriate encapsulation. Support encapsulations are:

- `ethernet-vpls`:
  - Standard Ethernet encapsulation; and
  - Accepts packets with Tag Protocol Identifier (TPID) values.
- `vlan-vpls`:
  - For VLAN 802.1q tagging; and
  - Accepts standard TPID values onl.y
- `extended-vlan-vpls`:
  - 802.1q tagging; and
  - Accepts special TPID values: 0x8100, 0x9100, and 0x9901.
- `ether-vpls-over-atm-llc`:
  - Bridges Ethernet and Asynchronous Transfer Mode (ATM) interfaces for Ethernet over ATM (AAL-5);
  - RFC 2684; and
  - Supported on ATM IQ interfaces only.

VPLS AFI/SAFI



- PE initially advertises a single VPLS NLRI for each VPLS instance in which it participates
  - Each NLRI defines labels (demultiplexors) for a range of other PE routers in the VPLS
  - If new labels must be added to existing VPLS, additional NLRI is sent
  - Same AFI and SAFI (25/65) as L2 VPN NLRI
  - PE router encapsulation and capabilities are signaled in Layer 2 information extended community

Length (2 Bytes)

Route Distinguisher (8 Bytes)

Site ID (2 Bytes)

Label Block Offset (2 Bytes)

Label Base (3 Bytes)

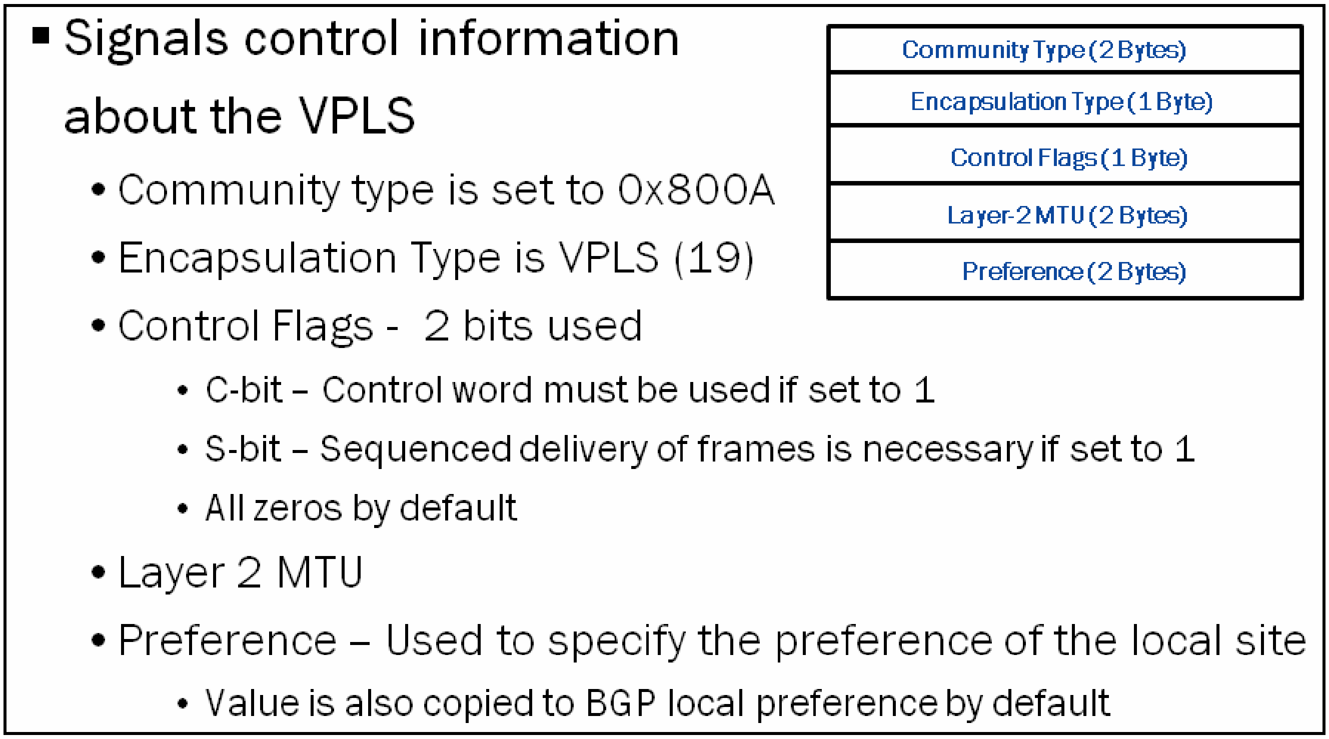
Circuit Status Vector (Variable)

The graphic displays the structure of VPLS NLRI. The address family indicator (AFI) and subsequent address family identifier (SAFI) values of 25 and 65 are shared with Kompella Layer 2 VPN NLRIs.

The NLRI consists of the site ID, the label base, and the label block offset, which are used when multiple label blocks are generated for a particular site. Each label block is carried as a separate update when multiple blocks exist.

The circuit status vector (CSV) is a bit vector used to indicate the site’s label range (that is, block size) and to report failures of a PE router’s local circuits.

Layer 2 Information Extended Community



- Signals control information about the VPLS
  - Community type is set to 0x800A
  - Encapsulation Type is VPLS (19)
  - Control Flags - 2 bits used
    - C-bit – Control word must be used if set to 1
    - S-bit – Sequenced delivery of frames is necessary if set to 1
    - All zeros by default
  - Layer 2 MTU
  - Preference – Used to specify the preference of the local site
    - Value is also copied to BGP local preference by default

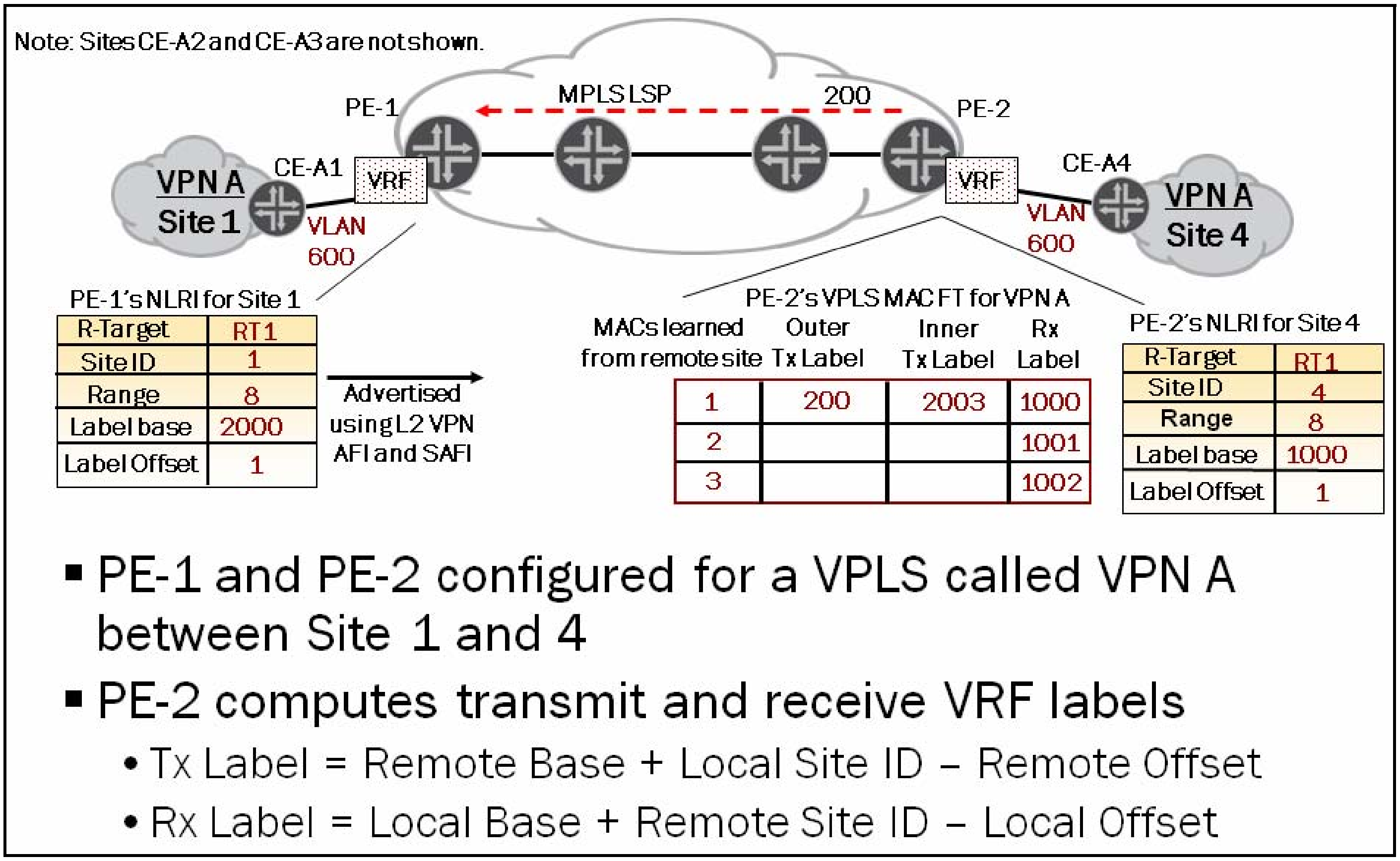
Community Type (2 Bytes)
Encapsulation Type (1 Byte)
Control Flags (1 Byte)
Layer-2 MTU (2 Bytes)
Preference (2 Bytes)

The Layer 2 information extended communities (carried as part of the Layer 2 NLRI) communicate the following information between PE routers:

- The Layer 2 encapsulation type (VPLS is the encapsulation type in a VPLS environment).
- The Layer 2 maximum transmission unit (MTU) field, which reports the MTU configured on the sending PE router's PE-CE link (because fragmentation is not supported in a Layer 2 VPN environment, the receiving PE router ignores Layer 2 NLRI with MTU values that differ from the PE router's local VRF interface).

The graphic shows the control flags field and the meaning of each of the specific flags. The reserved field is currently undefined and is set to all zeros.

PE-1 and PE-2 are Configured for a VPLS Called VPN A


Note: Sites CE-A2 and CE-A3 are not shown.
PE-1
MPLS LSP
200
PE-2
VPN A Site 1
CE-A1
VRF
VLAN 600
VRF
CE-A4
VLAN 600
VPN A Site 4
Advertised using L2 VPN AFI and SAFI

PE-1's NLRI for Site 1

R-Target	RT1
Site ID	1
Range	8
Label base	2000
Label Offset	1

PE-2's VPLS MAC FT for VPN A

MACs learned from remote site	Outer Tx Label	Inner Tx Label	Rx Label
1	200	2003	1000
2			1001
3			1002

PE-2's NLRI for Site 4

R-Target	RT1
Site ID	4
Range	8
Label base	1000
Label Offset	1

- PE-1 and PE-2 configured for a VPLS called VPN A between Site 1 and 4
- PE-2 computes transmit and receive VRF labels
  - Tx Label = Remote Base + Local Site ID – Remote Offset
  - Rx Label = Local Base + Remote Site ID – Local Offset

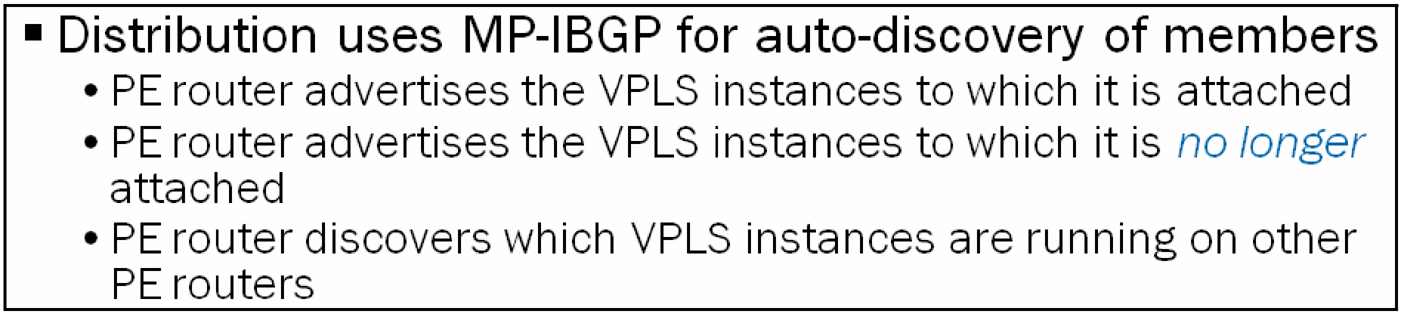
In the example, PE-2 is configured with a VRF for its local connection to CE-A4. This configuration assigns CE-A4 the site ID of 4 and associates this VPLS with a route target of *RT1*. Also, the local site is configured using VLAN tagging with a single VLAN ID of 600.

Computes Labels Automatically

Based on the MP-BGP advertisement that results from the information in PE-2's VRF, PE-2 automatically computes the label received when traffic is sent to PE-2 from remote PE routers. A single label from the labels in PE-2's label block is associated with each of the remote sites. The result is that PE-2 expects to receive traffic from CE-A1 with a label value of 1000.

The preceding graphic also shows how transmit labels are calculated based on the local site ID and the received MP-BGP advertisements from a remote site. Based on the received advertisement from PE-1, PE-2 sends frames destined for Site 1 using an inner label of 2003 (label-base-remote + local-site-id – label-block-offset). The outer MPLS label used for transmission is 200, based on the existing MPLS LSP from PE-2 to PE-1.

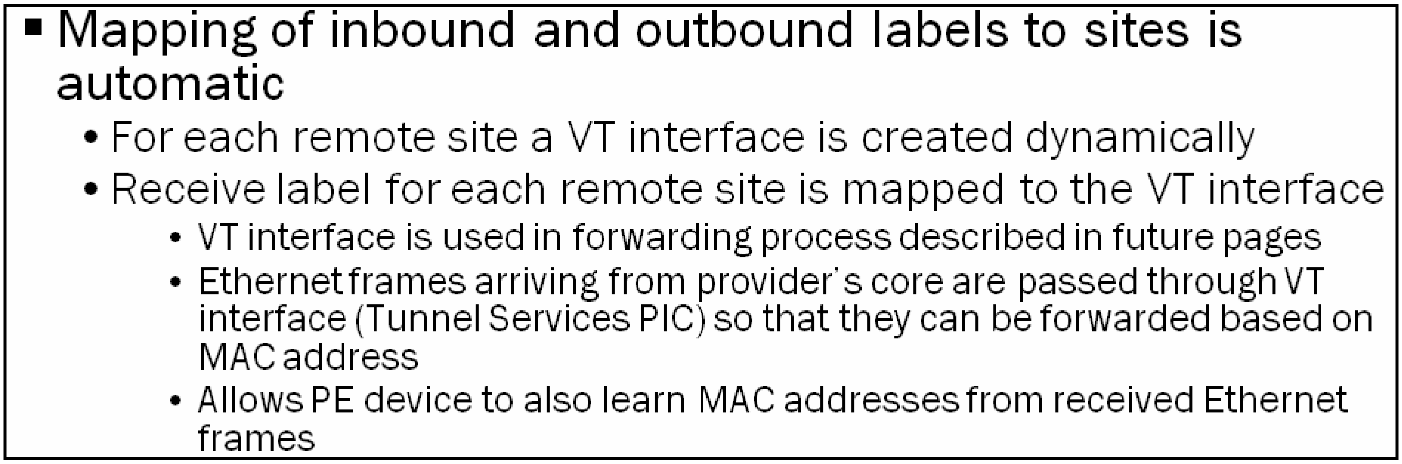
MP-IBGP Used for Signaling



- Distribution uses MP-IBGP for auto-discovery of members
  - PE router advertises the VPLS instances to which it is attached
  - PE router advertises the VPLS instances to which it is *no longer* attached
  - PE router discovers which VPLS instances are running on other PE routers

The distribution of label blocks between PE routers is facilitated with MP-IBGP using a Layer 2 VPN address family. Because all PE routers advertise the VPLS instances to which they are attached or no longer attached, every PE router can discover automatically which VPLS instances are running on other PE router by using the target extended community.

Automatic Label Mapping



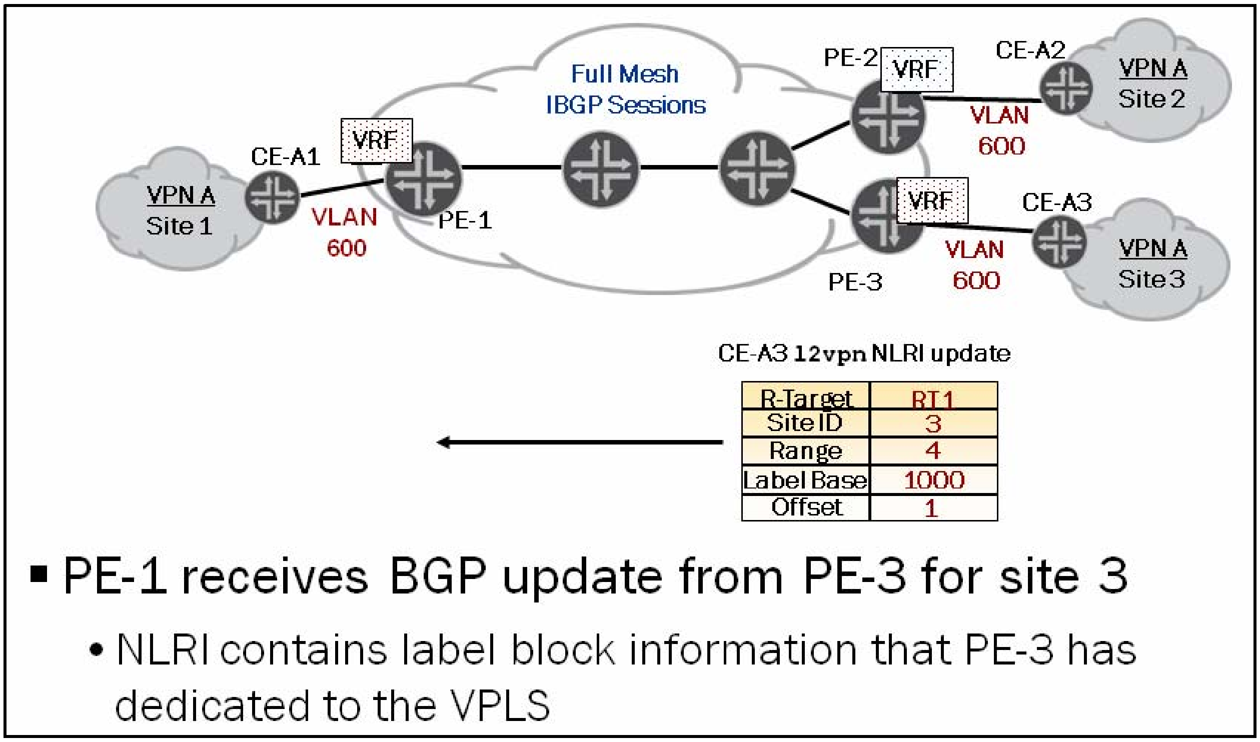
- Mapping of inbound and outbound labels to sites is automatic
  - For each remote site a VT interface is created dynamically
  - Receive label for each remote site is mapped to the VT interface
    - VT interface is used in forwarding process described in future pages
    - Ethernet frames arriving from provider's core are passed through VT interface (Tunnel Services PIC) so that they can be forwarded based on MAC address
    - Allows PE device to also learn MAC addresses from received Ethernet frames

The algorithm defined in the VPLS draft allows each PE router to compute automatically the mapping between remote site IDs and the label values used to send and receive traffic from them. The labels advertised by a site also are mapped automatically to VPN tunnel interfaces within a services PIC (Tunnel Services, Adaptive Services, Link Services). Thus, the connections between sites are created automatically. The PE device learns the MAC addresses during the forwarding process with the help of the VPN tunnel interfaces.

VPN Policy

VPN policy using route target communities to filter and accept label blocks from remote PE routers results in a VPLS topology.

PE-1 Receives Label Block from PE-3



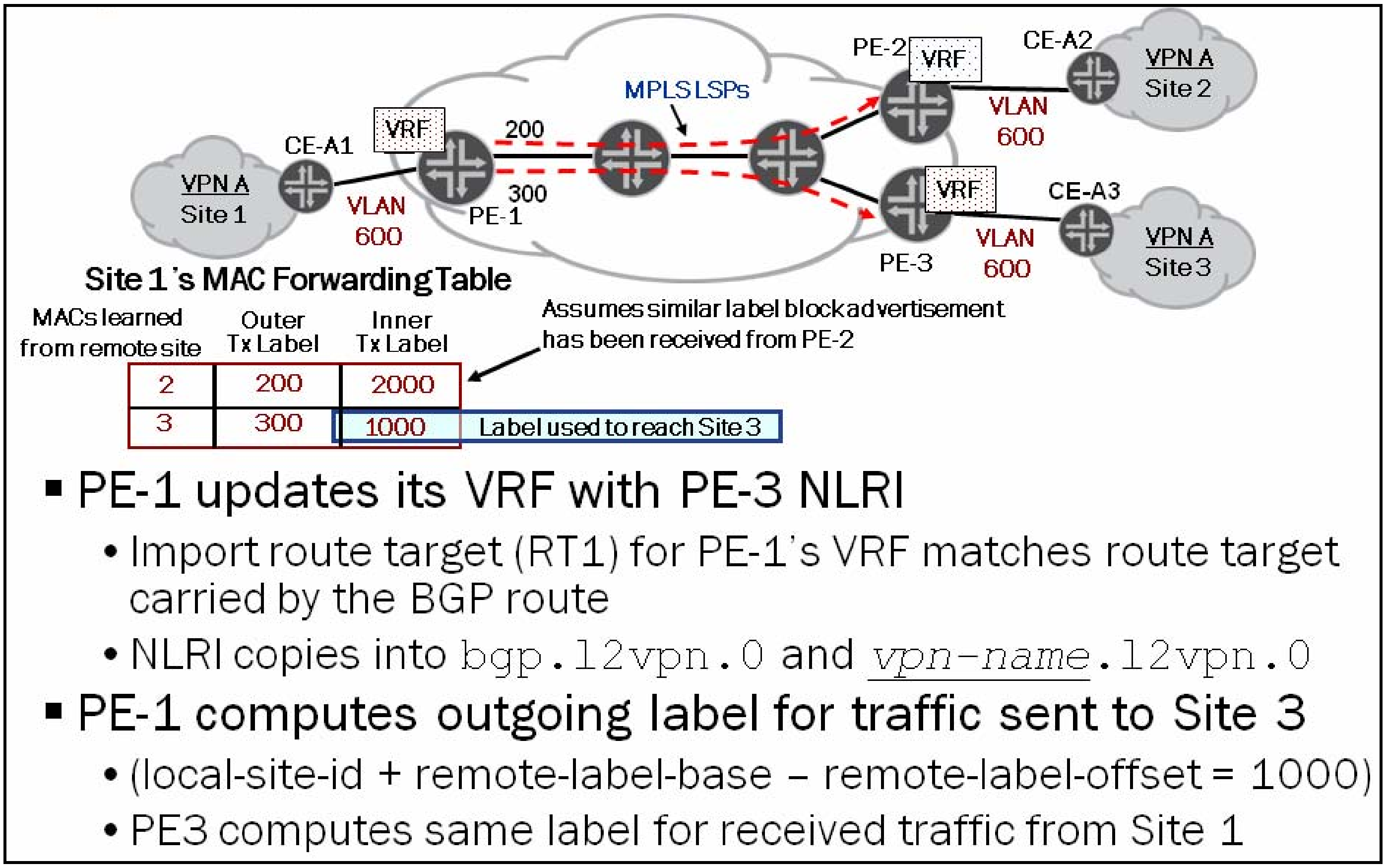
CE-A3 12vpn NLRI update

R-Target	RT1
Site ID	3
Range	4
Label Base	1000
Offset	1

- PE-1 receives BGP update from PE-3 for site 3
  - NLRI contains label block information that PE-3 has dedicated to the VPLS

The label block for CE-A3 contains the site's ID, label block size, label offset, and label base. This update also is associated with the route target extended BGP community.

PE-1 Updates Its VRF



Site 1's MAC Forwarding Table

MACs learned from remote site	Outer Tx Label	Inner Tx Label
2	200	2000
3	300	1000

Assumes similar label block advertisement has been received from PE-2

Label used to reach Site 3

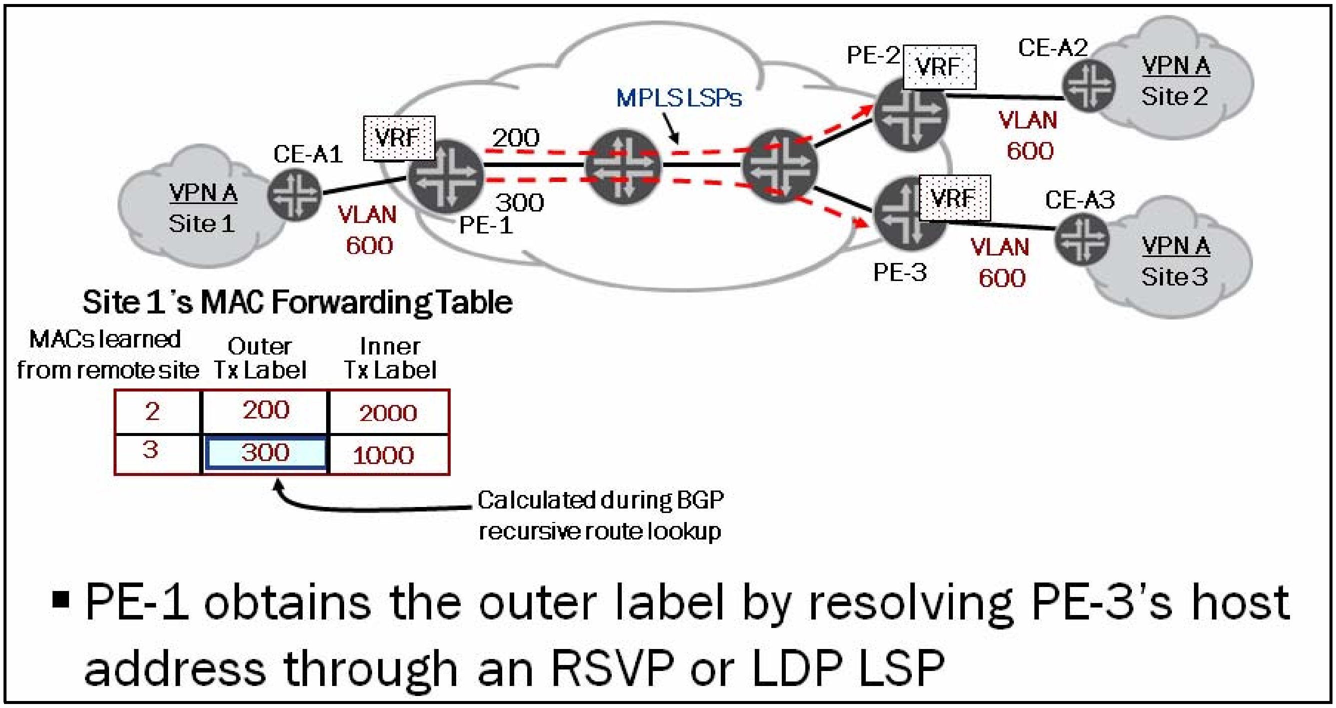
- PE-1 updates its VRF with PE-3 NLRI
  - Import route target (RT1) for PE-1's VRF matches route target carried by the BGP route
  - NLRI copies into `bgp.l2vpn.0` and *`vpn-name`*`.l2vpn.0`
- PE-1 computes outgoing label for traffic sent to Site 3
  - (local-site-id + remote-label-base – remote-label-offset = 1000)
  - PE3 computes same label for received traffic from Site 1

PE-1 receives the update from PE-3 and checks the route target for a match. Because the route target matches, the update is installed in the VRF associated with CE-A1. The L2 VPN NLRI is copied into both the `bgp.l2vpn.0` and *`vpn-name`*`.l2vpn.0` table as in Layer 2 VPNs.

PE-1 Computes Outgoing Label

PE-1 uses the update from PE-3 to compute automatically the labels to be used when sending traffic from CE-A1 to CE-A3. PE-1 uses the algorithm that subtracts the remote PE router's label offset from its local site ID and adds the resulting value to the received label base. In this example, PE-1 computes Label 1000 for traffic destined to CE-A3 (1–1 = 0 + 1000 = 1000). PE-3 computes the same label value (1000) as the label it expects to receive on traffic sent by CE-A1.

PE-1 Computes the Outer Label



Site 1's MAC Forwarding Table

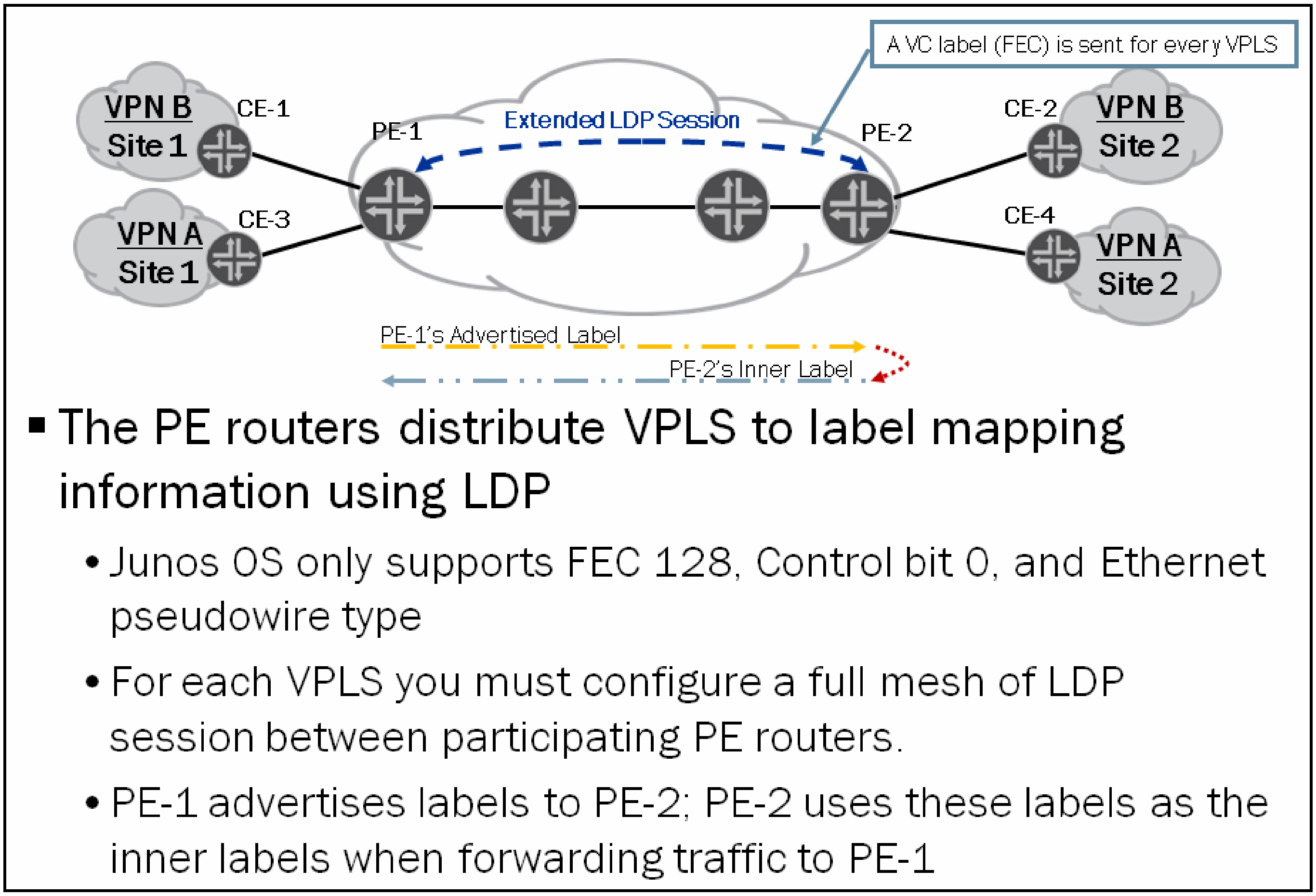
MACs learned from remote site	Outer Tx Label	Inner Tx Label
2	200	2000
3	300	1000

Calculated during BGP recursive route lookup

- PE-1 obtains the outer label by resolving PE-3's host address through an RSVP or LDP LSP

PE-1 computes the outer MPLS label by resolving PE-3's router ID to an LSP in the inet.3 routing table. In this example, the LSP from PE-1 to PE-3 is associated with label value 300.

PE Routers Advertise Labels Using LDP

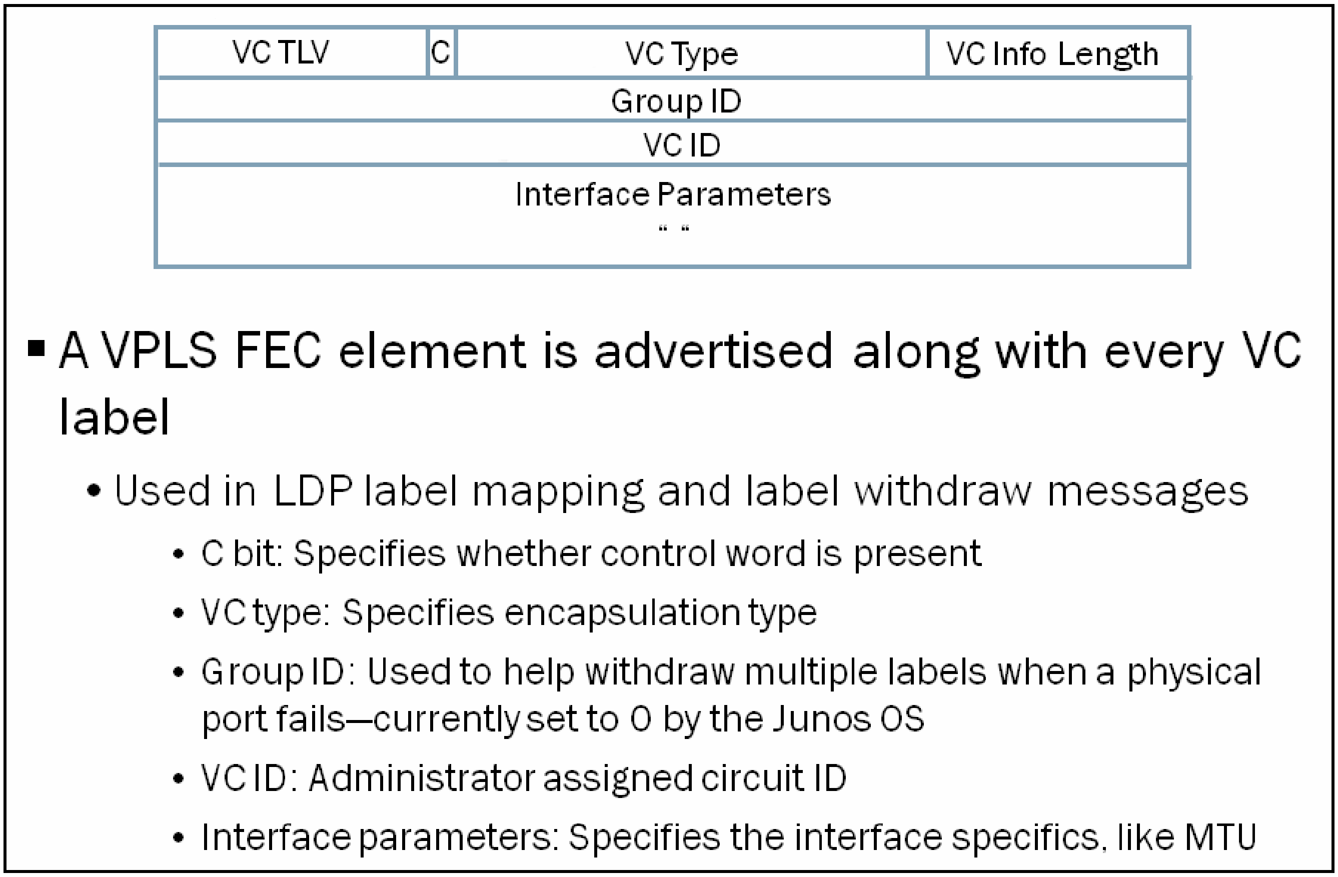


- The PE routers distribute VPLS to label mapping information using LDP
  - Junos OS only supports FEC 128, Control bit 0, and Ethernet pseudowire type
  - For each VPLS you must configure a full mesh of LDP session between participating PE routers.
  - PE-1 advertises labels to PE-2; PE-2 uses these labels as the inner labels when forwarding traffic to PE-1

In operation, a PE router advertises a label for each remote PE configured. To LDP, this label advertisement is just another forwarding equivalence class (FEC). These labels are advertised to targeted peers using extended LDP sessions.

As shown in the preceding diagram, the remote PE router (PE-2) uses the input label value advertised by PE-1 as its output label when forwarding traffic associated with this FEC to PE-1. Although not shown on the graphic, you can assume that PE-2 has also advertised an input label to PE-1, and that PE-1 pushes this label when sending traffic (for this connection) to PE-2.

VPLS FEC Element



VC TLV	C	VC Type	VC Info Length
Group ID			
VC ID			
Interface Parameters “ “			

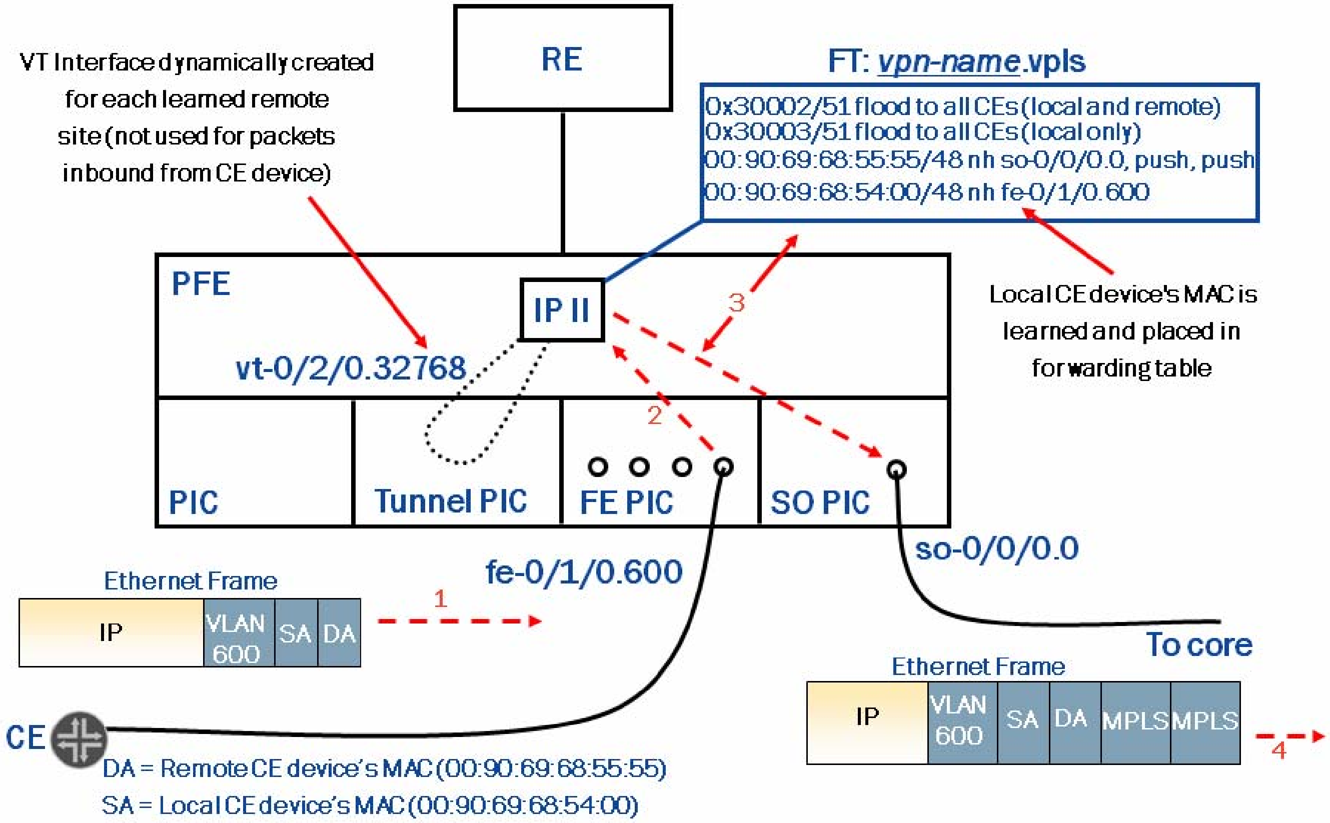
- A VPLS FEC element is advertised along with every VC label
  - Used in LDP label mapping and label withdraw messages
    - C bit: Specifies whether control word is present
    - VC type: Specifies encapsulation type
    - Group ID: Used to help withdraw multiple labels when a physical port fails—currently set to 0 by the Junos OS
    - VC ID: Administrator assigned circuit ID
    - Interface parameters: Specifies the interface specifics, like MTU

Using the LDP extended neighbor relationship, PE routers can exchange the virtual circuit labels associated with the VPLS. Along with each label, an associated FEC element is also advertised. This FEC element is used to describe the parameters of a PE router to the remote LDP neighbor.

The fields in this FEC element are described as follows:

- *C bit*: Specifies whether the Martini control word is present. This bit is set by default (control word present) in the Junos OS.
- *VC type*: Layer 2 encapsulation on VPN interface.
- *Group ID (optional)*: Used to group a set of labels together that relate to a particular port or tunnel. Makes withdrawal of labels easier when there is a failure of a port and there are many VPN labels associated with that same port.
- *VC ID*: An administrator-configurable value that represents the Layer 2 circuit.
- *Interface parameters*: Used to validate interoperability between ingress and egress ports. Possible parameter can be MTU, maximum number of concatenated ATM cells, interface description string, and other circuit emulation parameters.

PE Forwarding: Inbound from CE



VT Interface dynamically created for each learned remote site (not used for packets inbound from CE device)

0x30002/51 flood to all CEs (local and remote)
0x30003/51 flood to all CEs (local only)
00:90:69:68:55:55/48 nh so-0/0/0.0, push, push
00:90:69:68:54:00/48 nh fe-0/1/0.600

Local CE device's MAC is learned and placed in forwarding table

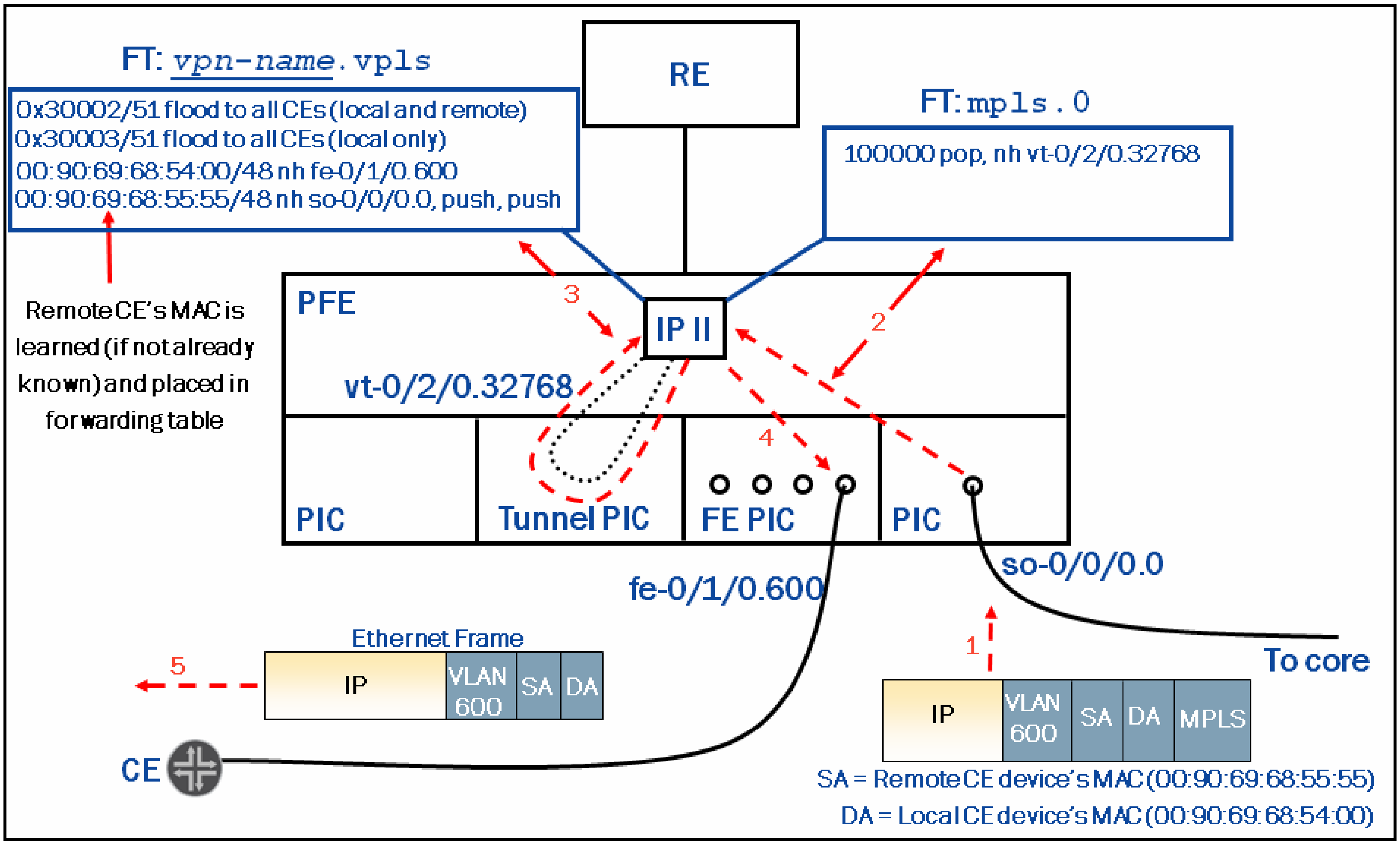
DA = Remote CE device's MAC (00:90:69:68:55:55)
SA = Local CE device's MAC (00:90:69:68:54:00)

Inbound Frame from Local CE Device

The graphic shows how Ethernet frames from the local CE devise of a VPLS logically flow through the Packet Forwarding Engine (PFE) of a router running the Junos OS. Notice that for each learned remote site (by means of L2VPN advertisements), a VPN tunnel interface is created within the Tunnel Services PIC. The VPN tunnel interface is not used for forwarding traffic inbound from the local CE device. The VPN tunnel interface's use is described in the next section.

1. An Ethernet frame arrives on interface `fe-0/1/0.600`. Because the interface is configured as part of a VPLS, the Ethernet framing is not stripped from the Layer 3 packet inside.
2. The Internet Processor II (or equivalent route lookup ASIC) can learn (if not already known) the local CE device's MAC address from the Ethernet header's source address. Because of this learning process, an entry is stored in the *`vpn-name`*`.vpls` forwarding table (MAC table) with its associated next hop. This entry is used to forward traffic to the local CE device as MPLS-encapsulated frames arrive from the core (shown in next section).
3. The Internet Processor II performs a forwarding lookup for this Ethernet frame using the *`vpn-name`*`.vpls` table. If the destination MAC address is not known, the Internet Processor II uses a flood route in the forwarding table, which causes the frame to be flooded to all sites except for the site where the frame originally came from. In the case described in the graphic, there is a specific entry for the destination MAC address so the frame is encapsulated in two MPLS headers and passed to the outgoing interface.
4. The MPLS-encapsulated Ethernet frame is forwarded to the remote site across the core network by means of the MPLS LSP built between PE devices.

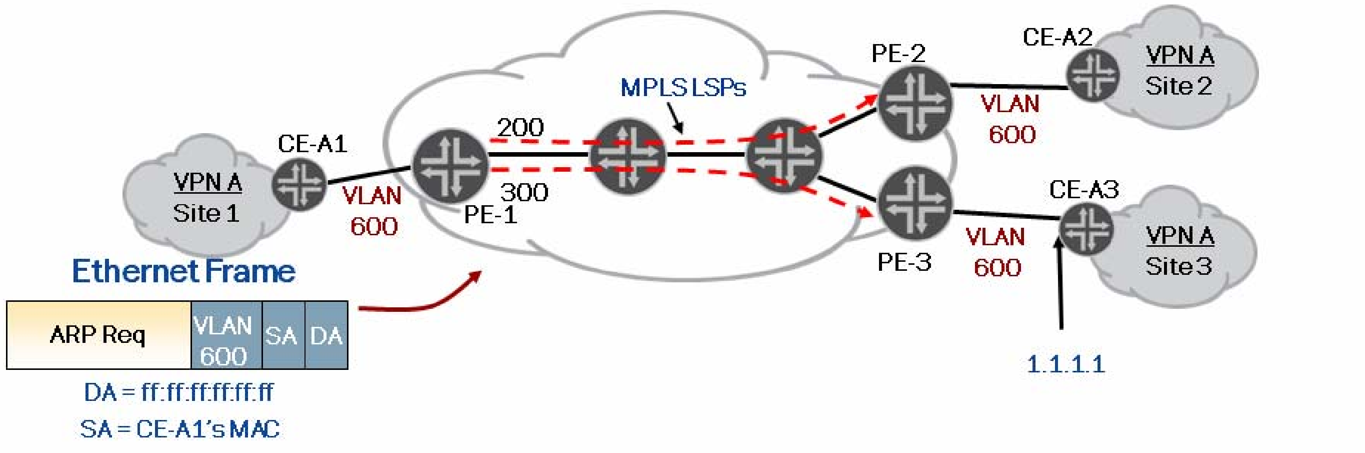
Inbound Frame from Core (Remote Site)


FT: *vpn-name*.vpls
0x30002/51 flood to all CEs (local and remote)
0x30003/51 flood to all CEs (local only)
00:90:69:68:54:00/48 nh fe-0/1/0.600
00:90:69:68:55:55/48 nh so-0/0/0.0, push, push
RE
FT: mpls.0
100000 pop, nh vt-0/2/0.32768
Remote CE's MAC is learned (if not already known) and placed in forwarding table
PFE
IP II
vt-0/2/0.32768
3
2
4
PIC
Tunnel PIC
FE PIC
PIC
so-0/0/0.0
fe-0/1/0.600
Ethernet Frame
5
IP
VLAN 600
SA
DA
1
IP
VLAN 600
SA
DA
MPLS
To core
CE
SA = Remote CE device's MAC (00:90:69:68:55:55)
DA = Local CE device's MAC (00:90:69:68:54:00)

The graphic shows how MPLS-encapsulated Ethernet frames arriving from the core network of a VPLS logically flow through the PFE of a router running the Junos OS.

1. Because of penultimate-hop popping (PHP), an Ethernet frame encapsulated by a single MPLS header arrives on the SONET interface.
2. As with all MPLS-encapsulated data, the Internet Processor II performs a forwarding lookup using the `mpls.0` table. Unlike point to point Layer 2 VPNs, instead of having a mapping of inbound labels to an outbound interface, the inbound labels are mapped to the dynamically created VPN tunnel interface. The reason this interface mapping is needed on a router running the Junos OS is because after the Internet Processor II processor does a pop operation based on the `mpls.0` table, no other Internet Processor II functions can be performed. By passing the resulting Ethernet frame through the VPN tunnel interface, the Internet Processor II is given a second chance to perform another function on the frame (that is, MAC address learning).
3. The Internet Processor II can learn (if not already known) the remote CE device's MAC address from the Ethernet headers source address. Because of this learning process, an entry is stored in the *vpn-name*`.vpls` forwarding table with its associated next hop and MPLS encapsulation. The Internet Processor II knows in which VPLS forwarding table to store the new entry based upon which VPN tunnel interface the packet arrives from. This entry is used to forward traffic to the remote CE device as Ethernet frames arrive from the local CE device (shown on previous graphic).
4. The Internet Processor II performs a forwarding lookup for this Ethernet frame using the *vpn-name*`.vpls` table. If the destination MAC address is not known, the Internet Processor II uses the default route in the forwarding table, which causes the Internet Processor II to flood the frame to all local sites but not to any remote sites. In the case described in the graphic, there is a specific entry for the destination MAC address so the frame is passed to the appropriate outgoing interface.
5. The Ethernet frame is forwarded to the local site.

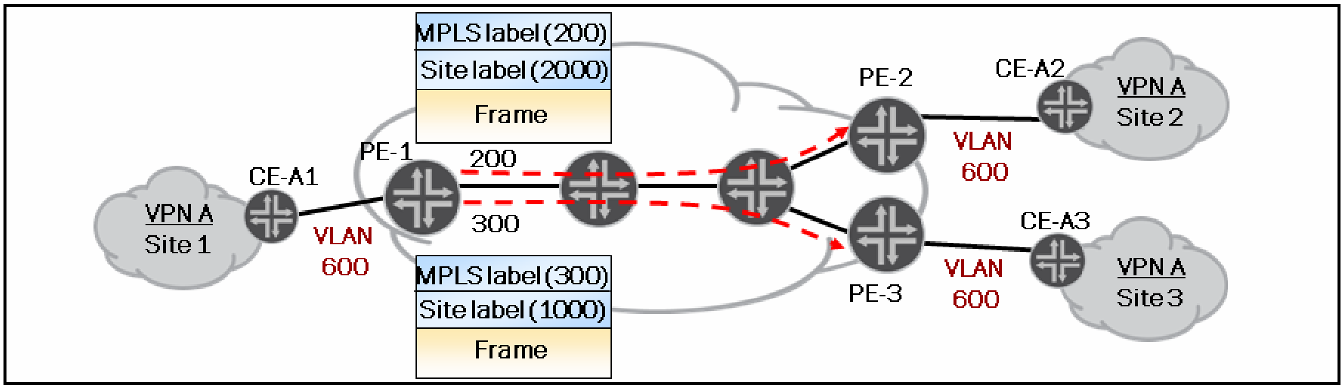
CE-A1 Sends Broadcast Traffic to CE-A3



- CE-A1 attempts to ping CE-A3's interface
  - CE-A1 does not know the MAC address of 1.1.1.1, so CE-A1 must send ARP request

On the graphic, the administrator of CE-A1 attempts to ping the core-facing interface of CE-A3. Because CE-A1 does not know the MAC address to use to send traffic to CE-A3, it must send an Address Resolution Protocol (ARP) request onto the Ethernet segment. This graphic shows CE-A1 sending an ARP request frame on VLAN 600. The frame arrives on PE-1 VPLS interface with a broadcast destination MAC address.

PE-1 Learns MAC Address from Frame



Before PE-1 forwards the Ethernet frame, it analyzes the addresses in the Ethernet header. PE-1 learns and stores the MAC address of CE-A1 and related interface in its `vpn-name.vpls` forwarding table.

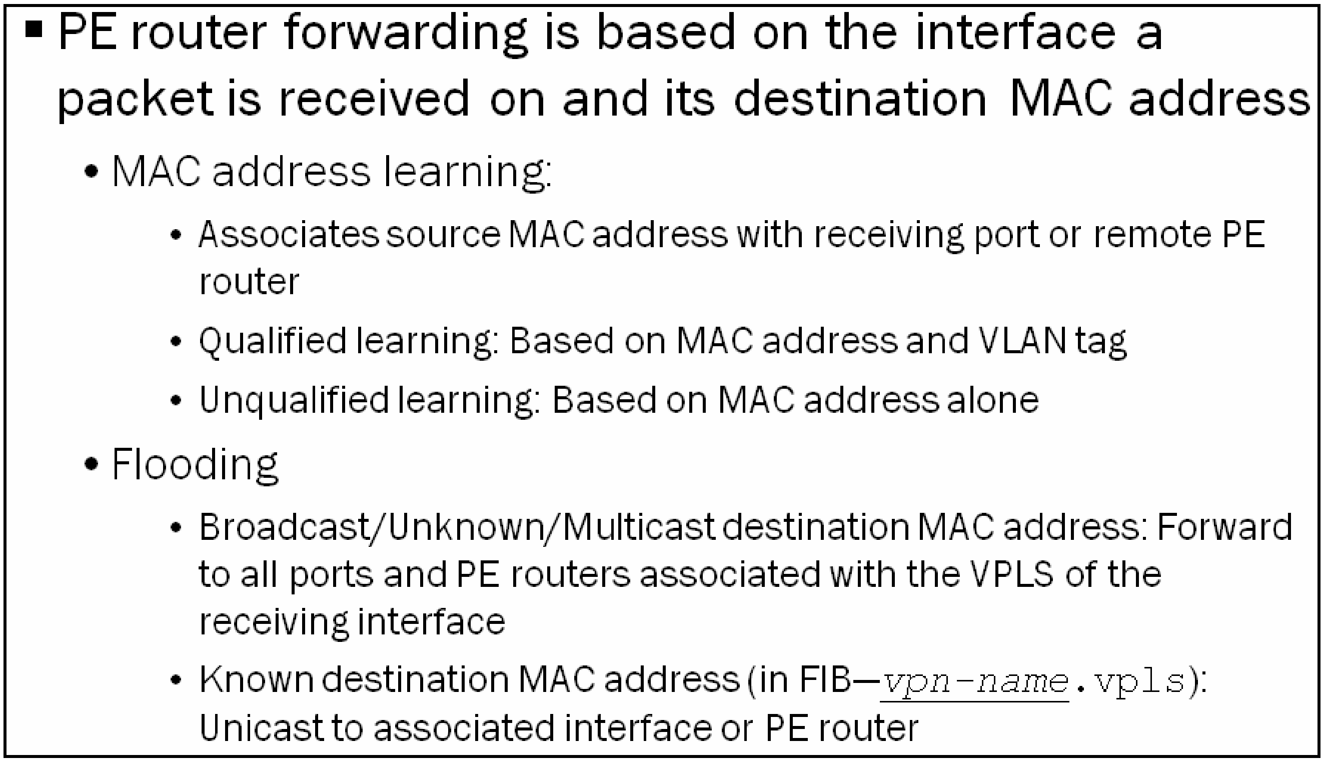
Broadcast Frame Is Flooded

When the destination MAC address of a received Ethernet frame is unknown or is broadcast from a CE device, the Ethernet frame is duplicated and sent to all remote PE routers for the VPLS. The flooding behavior is based on a default route in the VPLS forwarding table, `vpn-name.vpls`.

Lookup Derives Two Labels

Based on a forwarding table lookup, the Ethernet frames are encapsulated into the appropriate inner and outer header, as shown on the graphic.

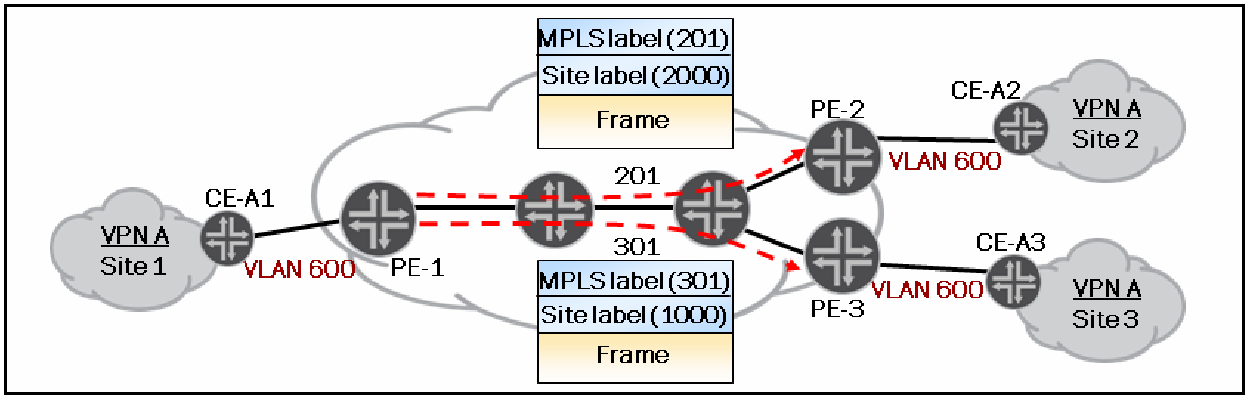
PE Router Forwarding



- PE router forwarding is based on the interface a packet is received on and its destination MAC address
  - MAC address learning:
    - Associates source MAC address with receiving port or remote PE router
    - Qualified learning: Based on MAC address and VLAN tag
    - Unqualified learning: Based on MAC address alone
  - Flooding
    - Broadcast/Unknown/Multicast destination MAC address: Forward to all ports and PE routers associated with the VPLS of the receiving interface
    - Known destination MAC address (in FIB—*vpn-name*.vpls): Unicast to associated interface or PE router

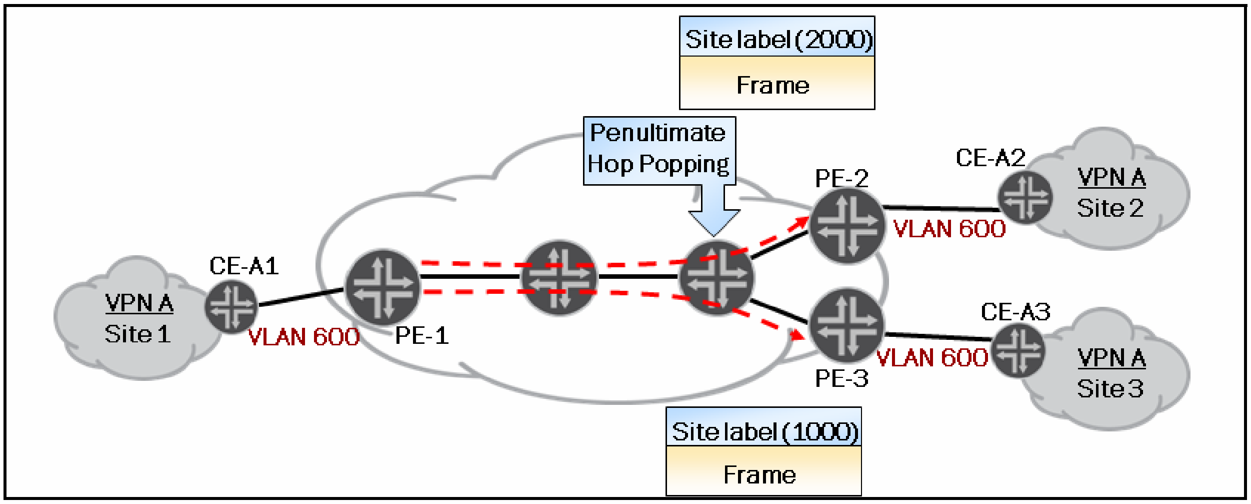
The graphic summarizes the forwarding and learning behavior of a PE router.

MPLS Switching in Core



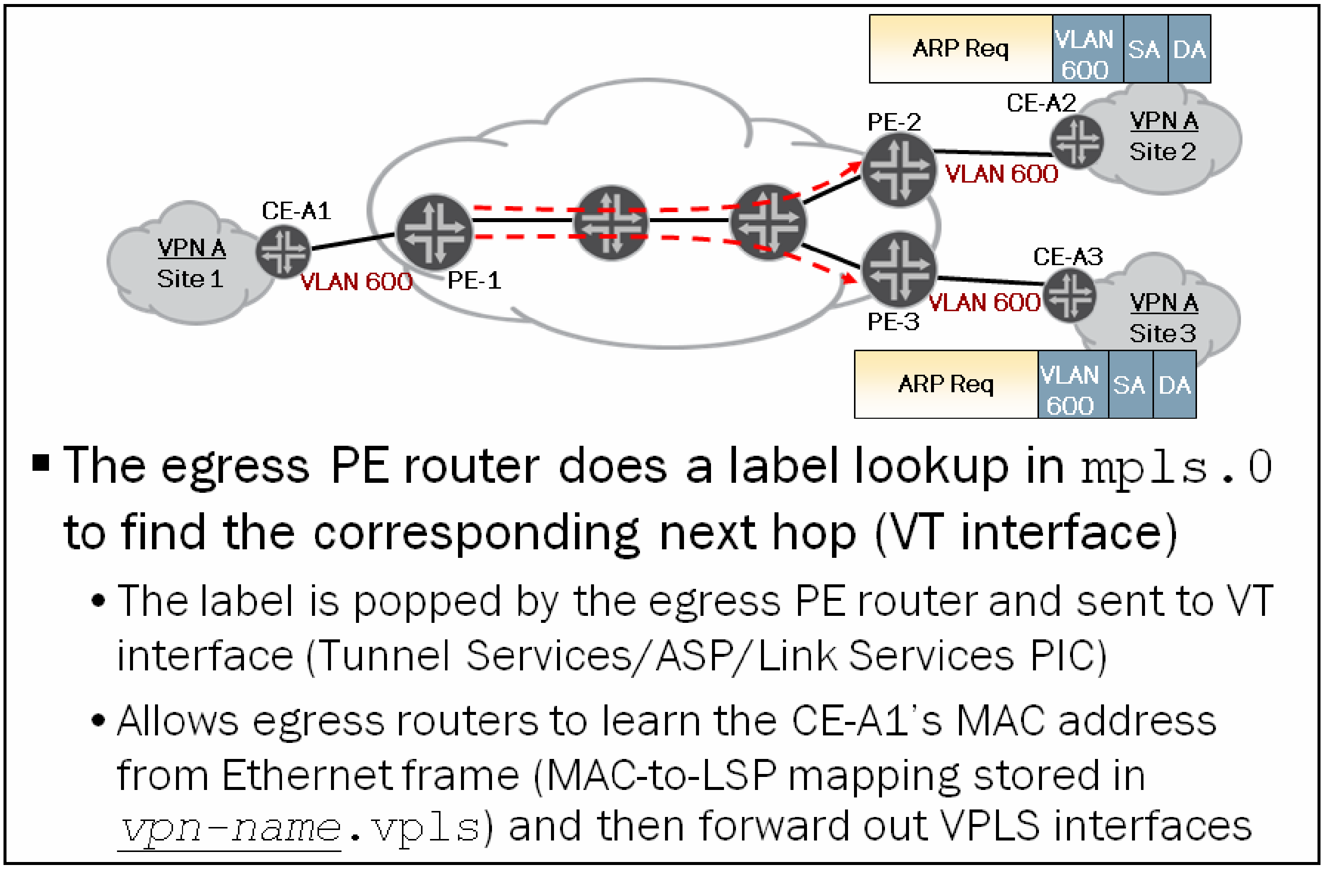
The labeled Ethernet frames are forwarded over the LSPs connecting the ingress PE router to the remote PE routers. The P routers in the core perform swap operations on the outer label. The P routers are not aware of the inner label, which remains unchanged throughout this process.

Outer Label Removed



The penultimate router pops the label stack, resulting in PE-2 and PE-3 receiving an Ethernet Frame with a single label.

Egress PE Router Looks Up VPLS Label



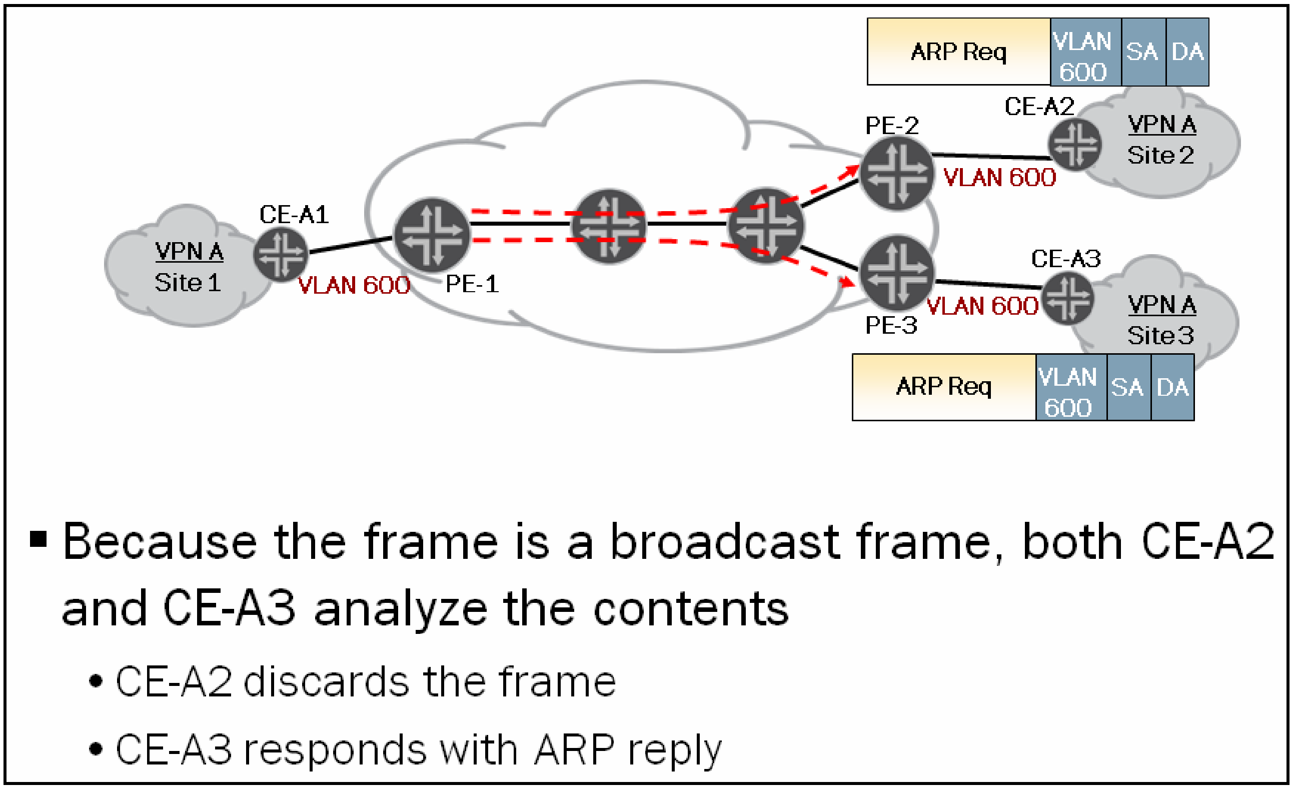
- The egress PE router does a label lookup in `mpls.0` to find the corresponding next hop (VT interface)
  - The label is popped by the egress PE router and sent to VT interface (Tunnel Services/ASP/Link Services PIC)
  - Allows egress routers to learn the CE-A1's MAC address from Ethernet frame (MAC-to-LSP mapping stored in *`vpn-name`*`.vpls`) and then forward out VPLS interfaces

The egress PE router performs a lookup on the VPLS label in the `mpls.0` table. The entry in the `mpls.0` table tells the router to pop the MPLS label and forward the Ethernet frame through the VPN tunnel interface that was created in response to learning PE-1's label block. The packet is essentially passed through the VPN tunnel interface so that a second lookup can occur.

Egress PE Router Learns and Performs Second Lookup

When an unlabeled Ethernet frame returns from VPN tunnel interface (Tunnel Services PIC), the egress PE router does two things. First, in the example on the graphic, PE-2 and PE-3 learn the MAC address for CE-A1 from source address of Ethernet Frame. Based on the newly learned MAC address, a dynamically generated route/MAC entry is placed into the *`vpn-name`*`.vpls`. The new route table entry is a route to the MAC address of CE-A1 with an LSP next hop (push-push operation based on VCT learned from PE-1). Second, the PE routers perform a lookup in the *vpn-name*`.vpls` table. Because the frame is a broadcast frame that arrived from the provider core, the frame is flooded to all attached CE devices.

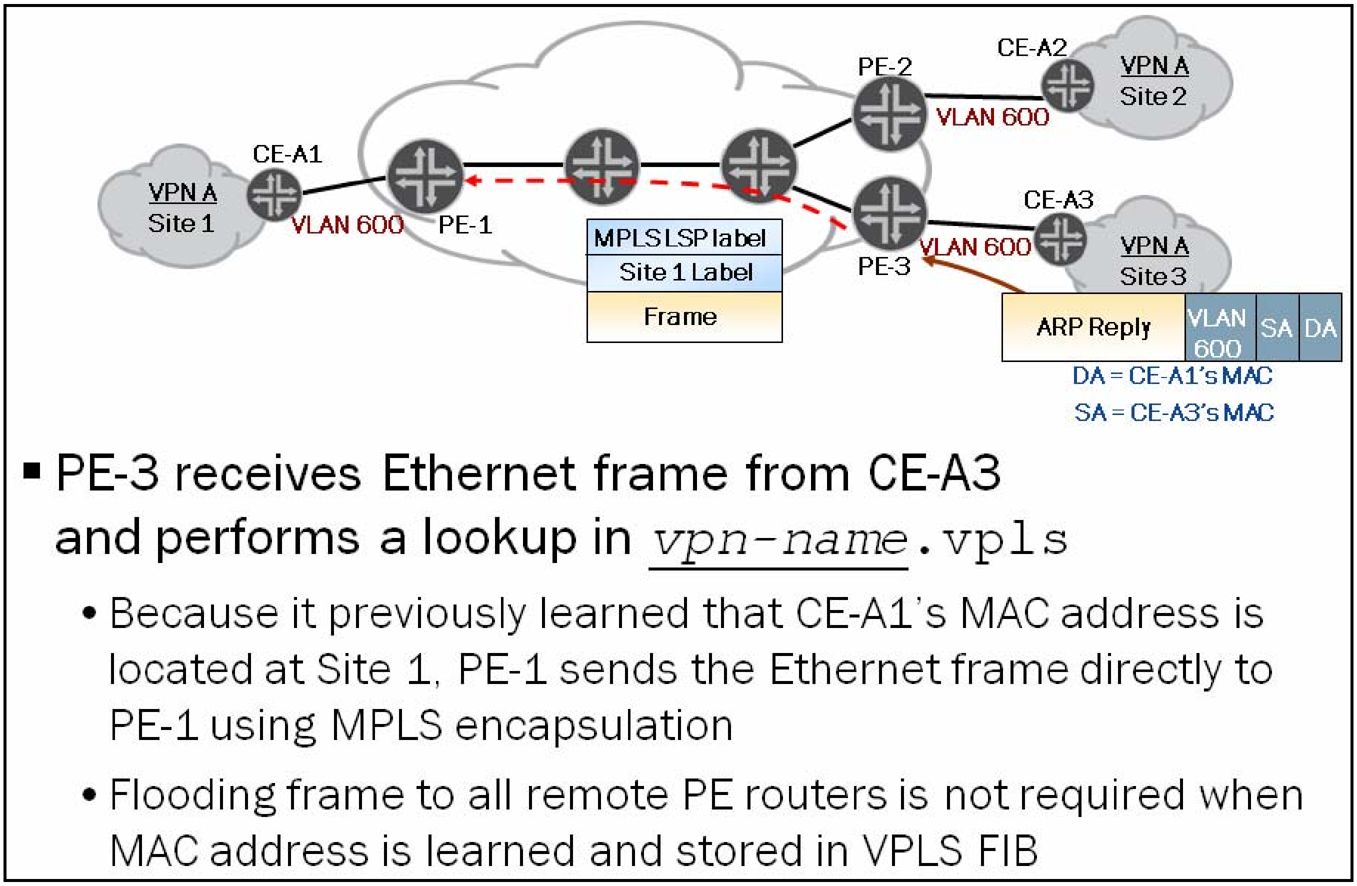
Broadcast Frame Analyzed by Remote CE Devices



- Because the frame is a broadcast frame, both CE-A2 and CE-A3 analyze the contents
  - CE-A2 discards the frame
  - CE-A3 responds with ARP reply

CE-A2 discards ARP frame because 1.1.1.1 does not belong to it. Because 1.1.1.1 belongs to CE-A3, it responds with and ARP reply.

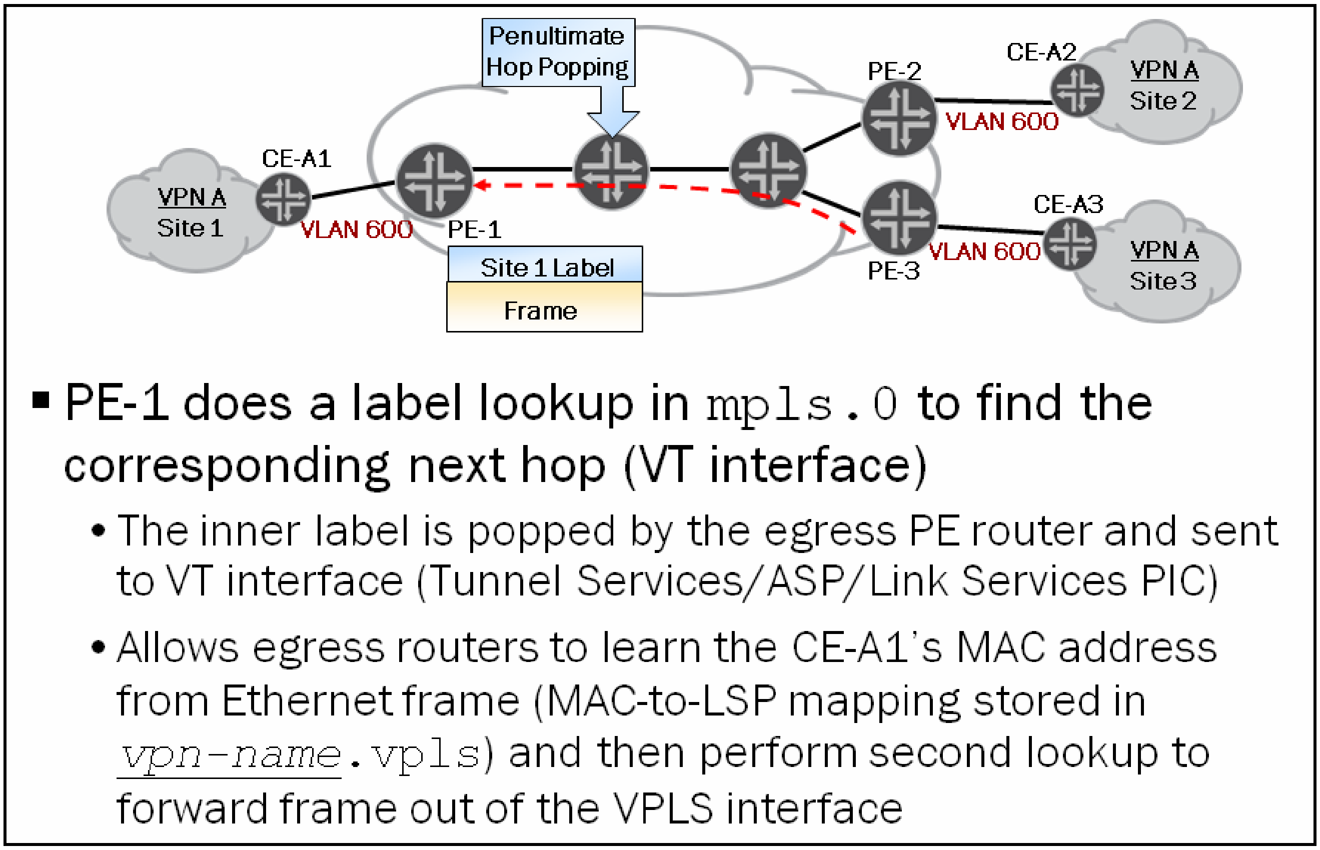
PE Router Receives ARP Reply



- PE-3 receives Ethernet frame from CE-A3 and performs a lookup in *vpn-name*.vpls
  - Because it previously learned that CE-A1's MAC address is located at Site 1, PE-1 sends the Ethernet frame directly to PE-1 using MPLS encapsulation
  - Flooding frame to all remote PE routers is not required when MAC address is learned and stored in VPLS FIB

PE-3 receives the ARP reply from the attached CE device. PE-3 learns the MAC address of CE-A3 and installs a route in the ***vpn-name*.vpls** table. Also, because PE-3 previously installed a route in the ***vpn-name*.vpls** table for CE-A1's MAC address, it can encapsulate and send the Ethernet frame to PE-1 directly without the need for flooding to all PE routers, as in the initial flow.

PE-1 Looks Up VPLS Label



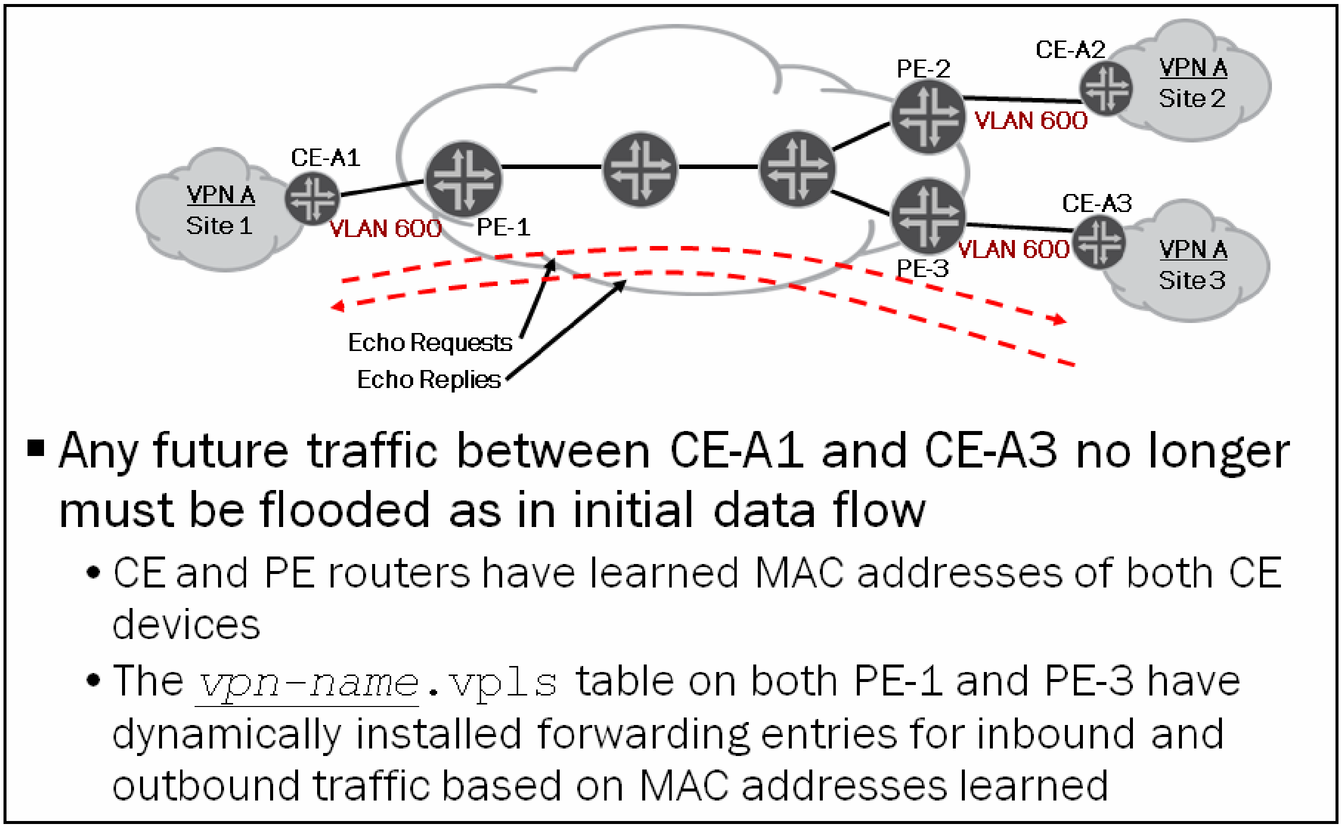
- PE-1 does a label lookup in `mpls.0` to find the corresponding next hop (VT interface)
  - The inner label is popped by the egress PE router and sent to VT interface (Tunnel Services/ASP/Link Services PIC)
  - Allows egress routers to learn the CE-A1's MAC address from Ethernet frame (MAC-to-LSP mapping stored in *vpn-name*`.vpls`) and then perform second lookup to forward frame out of the VPLS interface

The egress PE router performs a lookup on the VPLS label in the `mpls.0` table. The entry in the `mpls.0` table tells the router to pop the MPLS label and forward the Ethernet frame through the VPN tunnel interface that was created in response to learning PE-3's VCT.

Egress PE Router Learns and Performs Second Lookup

When the unlabeled Ethernet frame returns from VPN tunnel interface (Tunnel Services PIC), PE-1 learns the MAC address for CE-A3 from source address of Ethernet Frame. Based on the newly learned MAC address a dynamically generated route entry is placed into the ***vpn-name*.vpls**. The new route table entry is a route to the MAC address of CE-A3 with an LSP next-hop (push-push operation based on label block learned from PE-1). Second, PE-1 performs a lookup in the ***vpn-name*.vpls** table. Because the MAC address of CE-A1 was learned earlier, which caused a route to CE-A1 to be dynamically installed, the frame is sent directly to CE-A1.

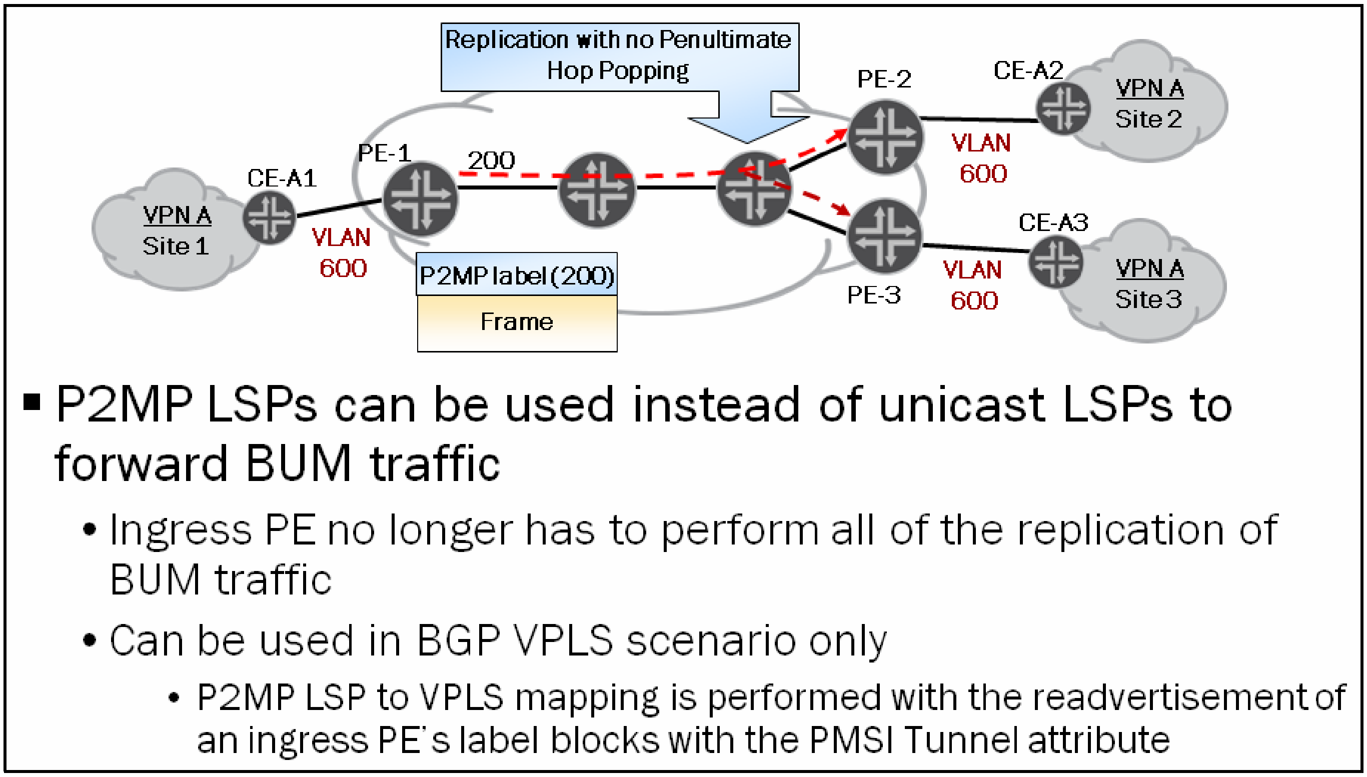
Future Traffic Is Not Flooded



- Any future traffic between CE-A1 and CE-A3 no longer must be flooded as in initial data flow
  - CE and PE routers have learned MAC addresses of both CE devices
  - The `vpn-name.vpls` table on both PE-1 and PE-3 have dynamically installed forwarding entries for inbound and outbound traffic based on MAC addresses learned

As MAC addresses are learned over time, packets no longer need to be flooded to all remote PE devices across provider core. Ethernet frames that are now passed between CE-A1 and CE-A3 will be forwarded only between the related PE devices.

BUM Replication



- P2MP LSPs can be used instead of unicast LSPs to forward BUM traffic
  - Ingress PE no longer has to perform all of the replication of BUM traffic
  - Can be used in BGP VPLS scenario only
    - P2MP LSP to VPLS mapping is performed with the readvertisement of an ingress PE's label blocks with the PMSI Tunnel attribute

Broadcast, unicast with unknown destination, and multicast (BUM) traffic is replicated and flooded solely by the ingress PE by default. This behavior can put a tremendous burden on the PE if it happens to be services several hundred VPLS instances. point to multipoint LSPs can be used specifically for the purpose of carrying BUM traffic. When using a point to multipoint LSP for this purpose, the ingress PE only needs to send 1 copy of the BUM traffic into the core. The downstream routers along the LSP will perform replication of the traffic. To notify remote PEs that a point to multipoint LSP will be used for BUM forwarding, the ingress PE re-advertises all label blocks along with the Provider Multicast Service Interface (PMSI) Tunnel attribute which carries the RSVP session identification information.

Fully Meshed PE Routers

Based on the flooding behavior of VPLS, PE routers must be fully meshed in terms of MPLS LSPs as well as extended LDP or MP-BGP sessions (route reflectors and confederations can be used).

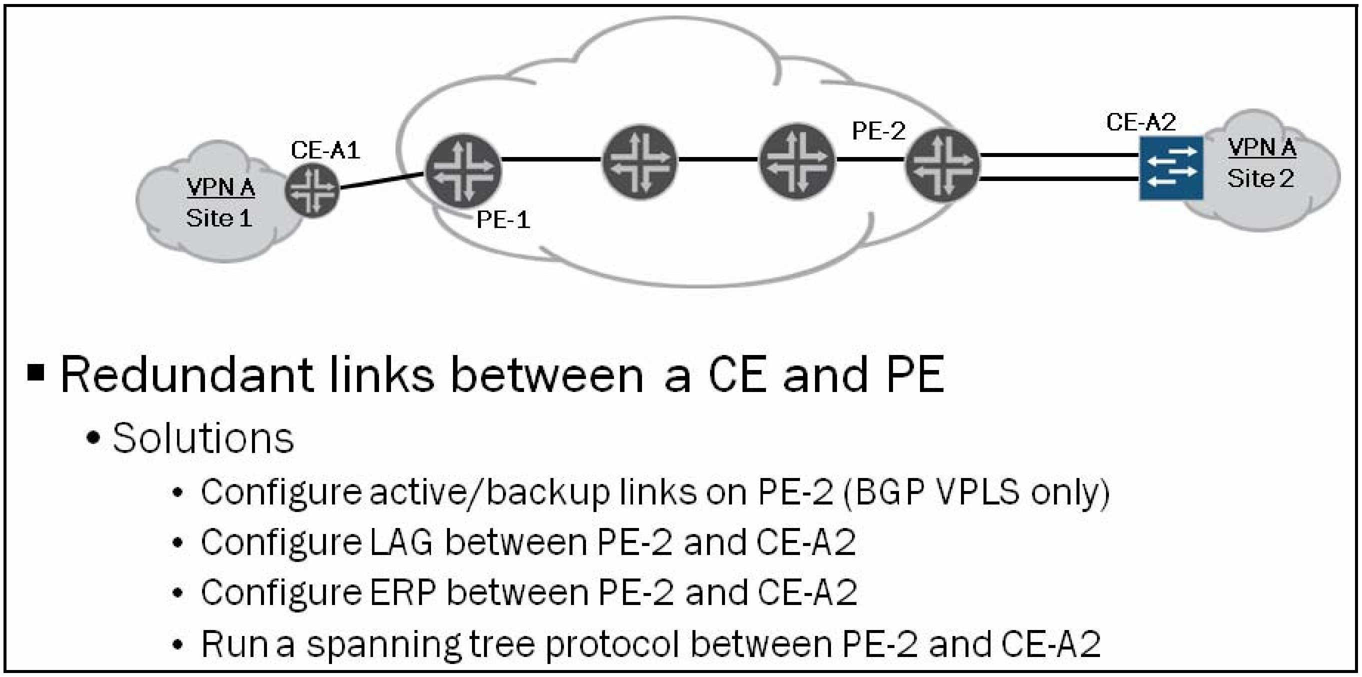
Split Horizon

One of the flooding rules of VPLS is that a router cannot flood a packet from a remote PE router to another remote PE router. Although this behavior causes a need for the full mesh, it helps eliminate the need for a spanning tree protocol in the provider core.

PE Routers Perform MAC Learning and Flooding

As described on the previous graphics, a PE router learns MAC addresses based on received Ethernet frames. A PE device will not request another PE device to flood or learn on its behalf.

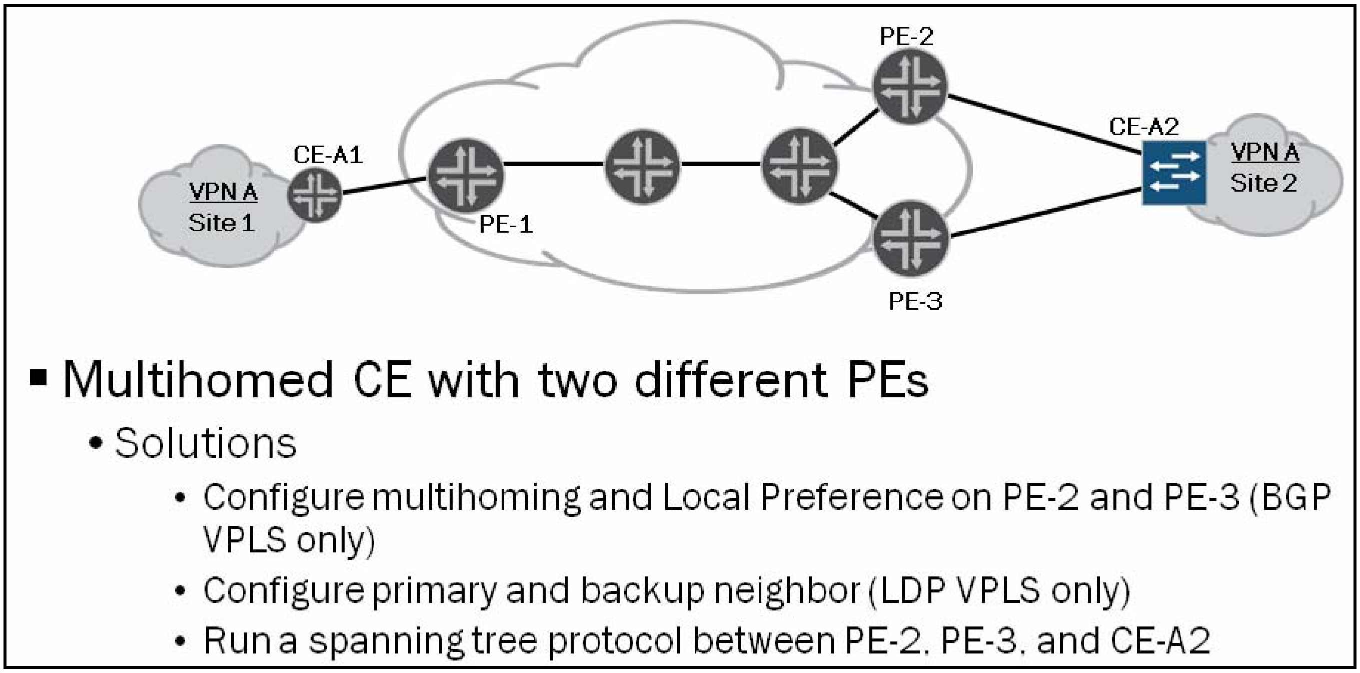
Redundant Links Between CE and PE



- Redundant links between a CE and PE
  - Solutions
    - Configure active/backup links on PE-2 (BGP VPLS only)
    - Configure LAG between PE-2 and CE-A2
    - Configure ERP between PE-2 and CE-A2
    - Run a spanning tree protocol between PE-2 and CE-A2

The graphic shows a potential loop situation that can occur when there are multiple links between a CE and the local PE. If CE-A2 is a router operating at Layer 3, then there should be no Layer 2 loop possible. However, if CE-A2 is a Layer 2 switch then a Layer 2 loop is possible. To prevent Layer 2 data from looping between the CE and PE with redundant links you must configure either a spanning tree protocol between PE and CE, active and backup links on the PE, Ethernet Ring Protection (ERP), or a link aggregation group (LAG). Each solution will be shown in the next chapter.

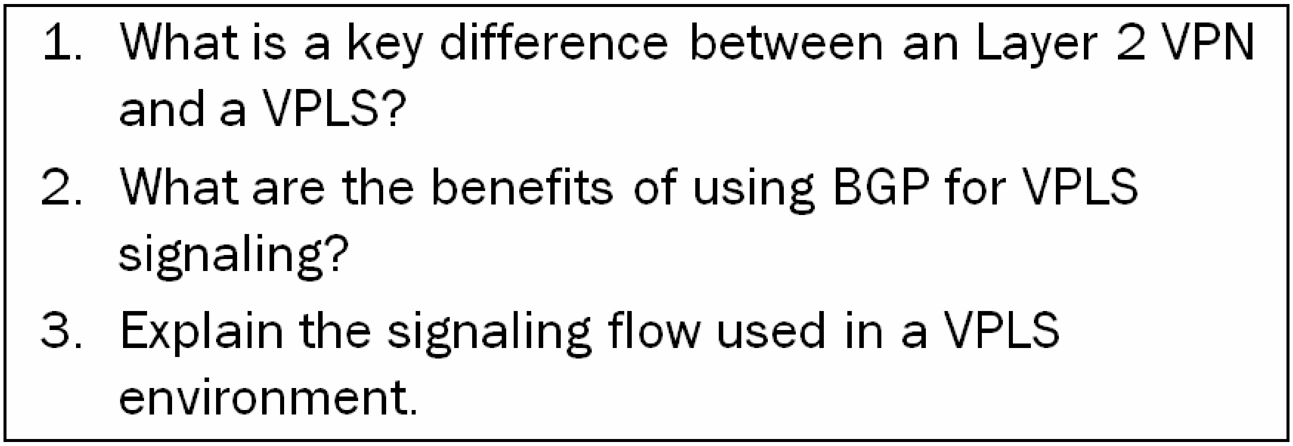
Multihomed CE



- Multihomed CE with two different PEs
  - Solutions
    - Configure multihoming and Local Preference on PE-2 and PE-3 (BGP VPLS only)
    - Configure primary and backup neighbor (LDP VPLS only)
    - Run a spanning tree protocol between PE-2. PE-3. and CE-A2

The graphic shows a potential loop situation that can occur when there are links between a single CE multiple PEs. If CE-A2 is a router operating at Layer 3, then there should be no Layer 2 loop possible. However, if CE-A2 is a Layer 2 switch then a Layer 2 loop is possible. To prevent Layer 2 data from looping between the CE and two PEs you must configure either a spanning tree protocol between PEs and CE, BGP multihoming, or a primary and backup neighbor. Each solution will be shown in the next chapter.

Review Questions



1. What is a key difference between an Layer 2 VPN and a VPLS?
2. What are the benefits of using BGP for VPLS signaling?
3. Explain the signaling flow used in a VPLS environment.

Answers to Review Questions

1.

A Layer 2 VPN is point to point in nature while a VPLS is point to multipoint.

2.

Adding and removing sites from a BGP VPLS requires the configuration of only one PE. All other PE's will automatically discover the added or removed site.

3.

In a BGP VPLS, PEs advertise label blocks to remote PEs. The label blocks have enough labels to reach and be reached by all currently configured sites in the VPLS. In an LDP VPLS, individual labels are advertised using an LDP extended neighbor relationship to all remote PEs participating in the VPLS.

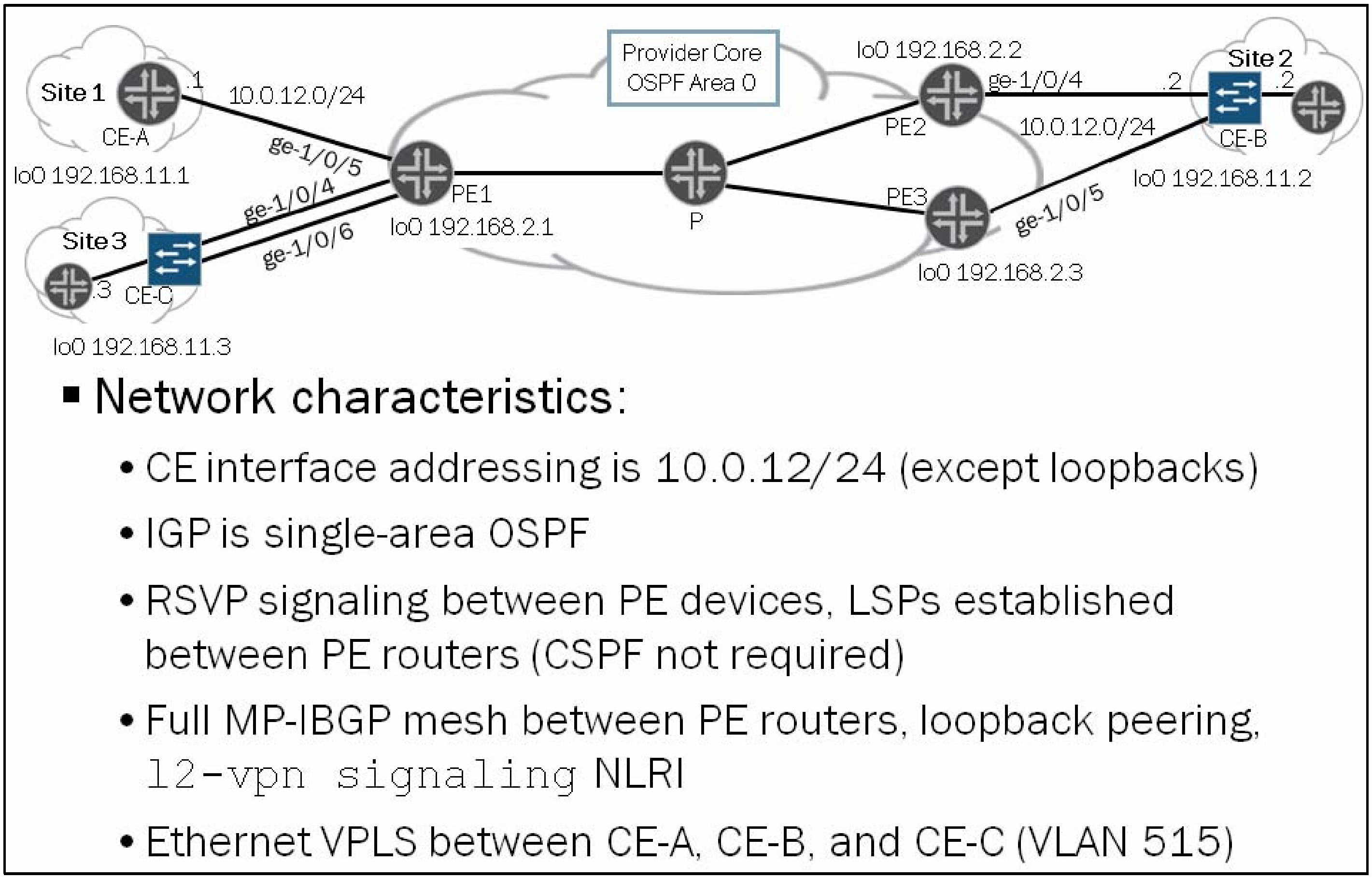
JNCIS-SP Study Guide—Part 3

Chapter 17: VPLS Configuration

This Chapter Discusses:

- Virtual private LAN service (VPLS) configuration, and
- VPLS troubleshooting.

Sample VPLS Topology



- Network characteristics:
  - CE interface addressing is 10.0.12/24 (except loopbacks)
  - IGP is single-area OSPF
  - RSVP signaling between PE devices, LSPs established between PE routers (CSPF not required)
  - Full MP-IBGP mesh between PE routers, loopback peering, `l2-vpn signaling` NLRI
  - Ethernet VPLS between CE-A, CE-B, and CE-C (VLAN 515)

The diagram serves as the basis for the various configuration-mode and operational-mode examples that follow.

All customer edge (CE) and provider edge (PE) router interfaces use 10.0.12.0/24 addresses. The drawings show only the interfaces' subnet and host IDs. Loopback addresses are assigned from the 192.168/16 address block.

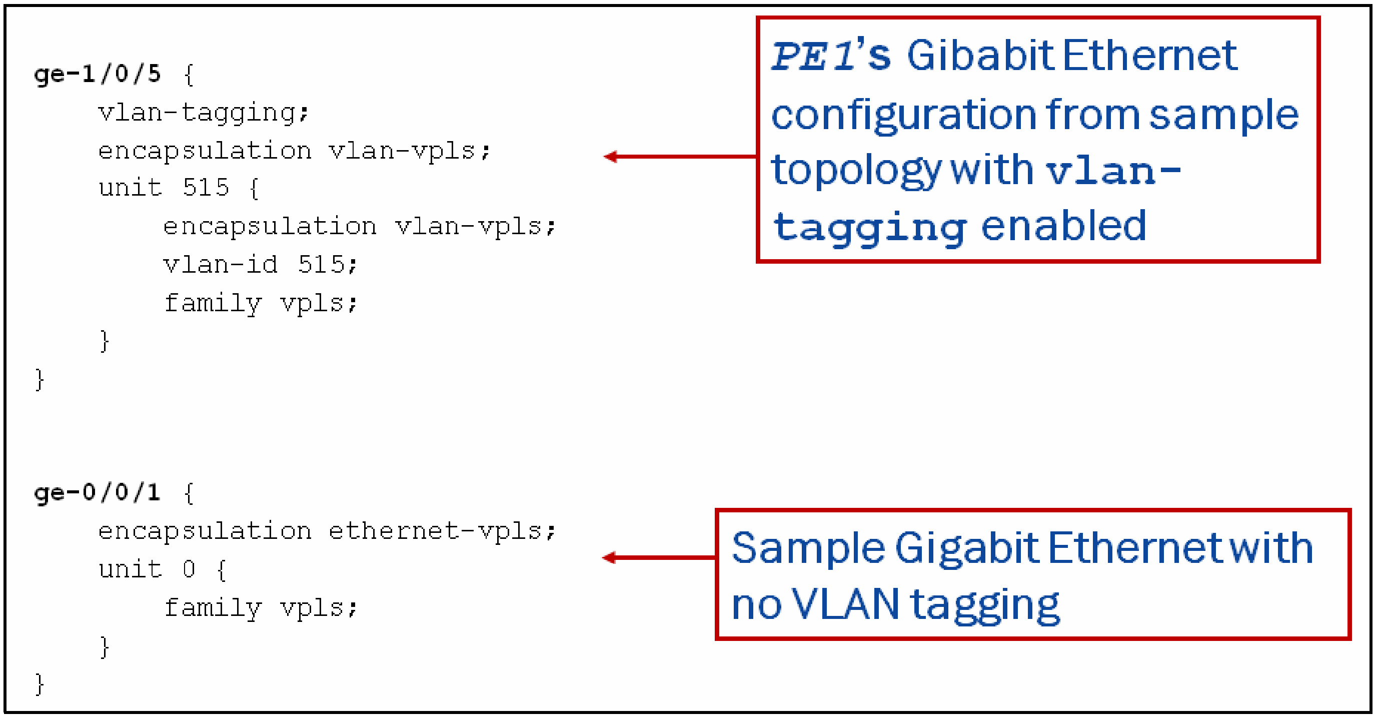
The core interior gateway protocol (IGP) is Open Shortest Path First (OSPF), and a single area (Area 0) is configured. Because the examples do not rely on the functionality of the Constrained Shortest Path First (CSPF) algorithm, traffic engineering extensions need not be enabled.

RSVP is deployed as the MPLS signaling protocol, and label-switched paths (LSPs) are configured between all three PE routers.

A multiprotocol IBGP(MP-IBGP) peering session is configured between the loopback addresses of the PE routers. The `l2-vpn signaling` and `inet unicast` address families are configured.

The goal of this network is to provide point-to-multipoint connectivity between the three CE routers shown. This network is considered a full-mesh application because the resulting configuration readily accommodates additional sites with any-to-any connectivity.

PE Interface Configuration



```
ge-1/0/5 {
    vlan-tagging;
    encapsulation vlan-vpls;
    unit 515 {
        encapsulation vlan-vpls;
        vlan-id 515;
        family vpls;
    }
}
```

***PE1*'s Gibabit Ethernet configuration from sample topology with `vlan-tagging` enabled**

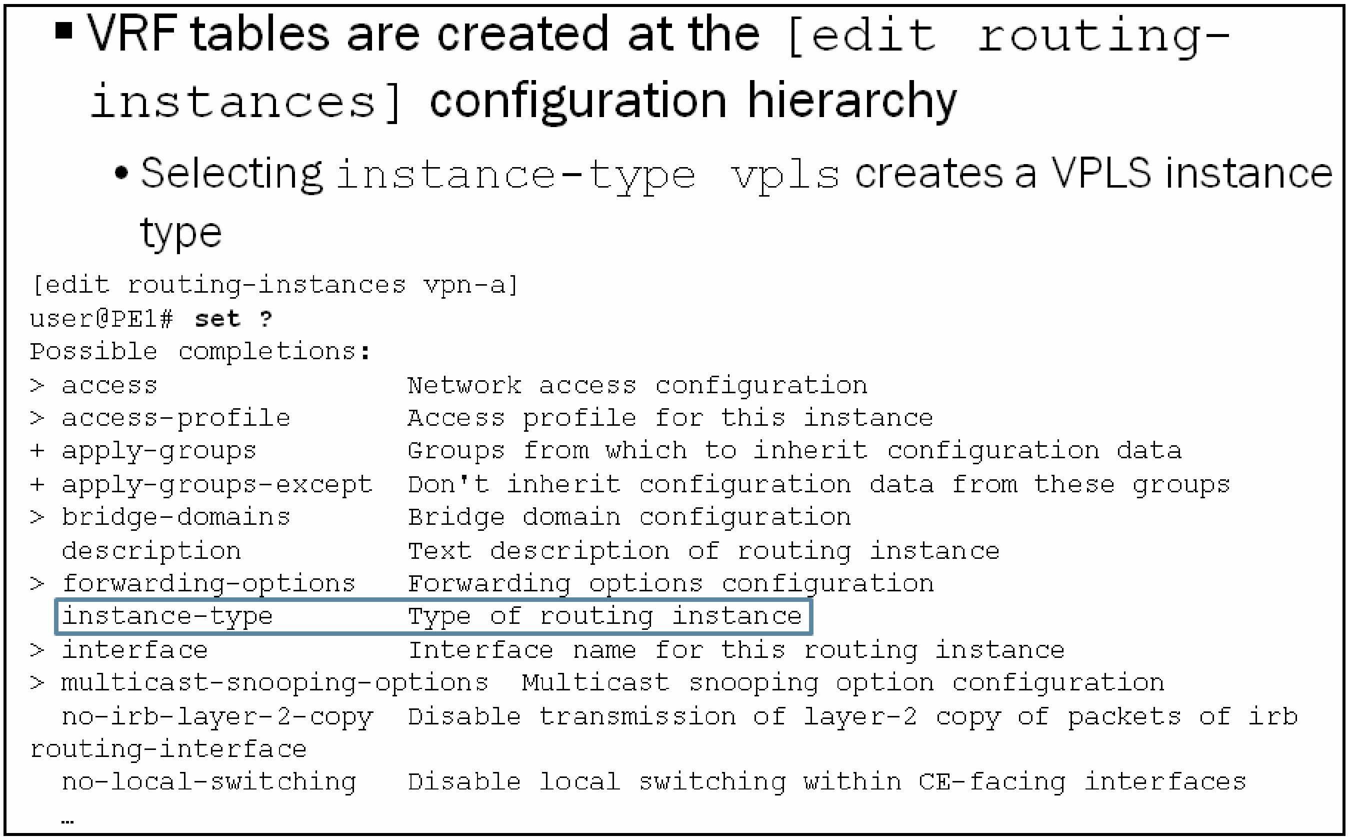
```
ge-0/0/1 {
    encapsulation ethernet-vpls;
    unit 0 {
        family vpls;
    }
}
```

Sample Gigabit Ethernet with no VLAN tagging

This graphic provides an example of Gigabit Ethernet interface configurations for use with VPLS.

Virtual LAN (VLAN) tagging is possible but not mandatory, and you must specify the use of VPLS encapsulation at both the device and logical unit levels. When you enable VPLS encapsulation, VLAN IDs from 512 to 4094 are reserved for circuit cross-connect (CCC) and VPLS encapsulation. You can configure VLAN IDs 0 to 511 as normal VLAN tagged interfaces, if wanted. All logical unit levels might also be configured for `family vpls`, but in later versions of the Junos operating system, this configuration is optional.

VPLS VRF Table Creation

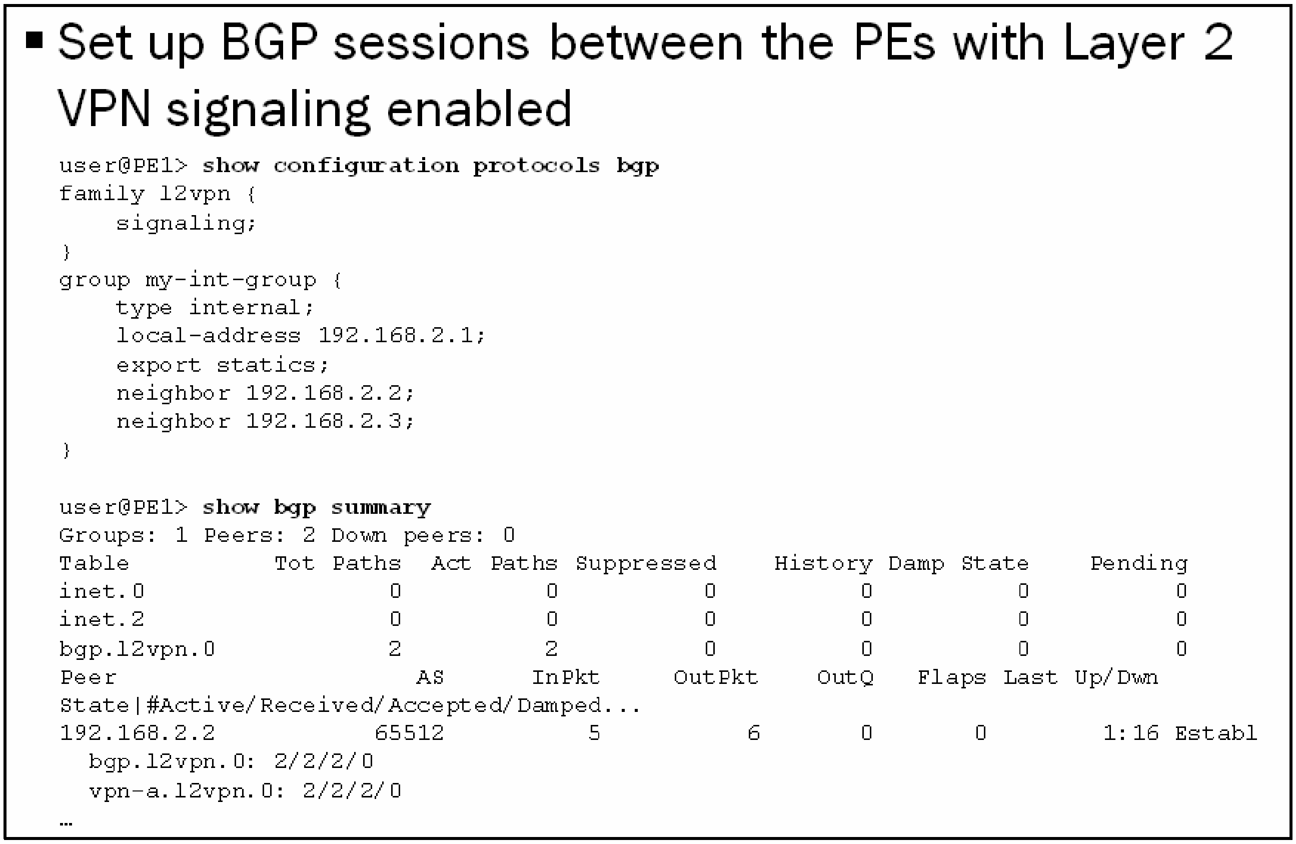


- VRF tables are created at the `[edit routing-instances]` configuration hierarchy
  - Selecting `instance-type vpls` creates a VPLS instance type

```
[edit routing-instances vpn-a]
user@PE1# set ?
Possible completions:
> access                 Network access configuration
> access-profile         Access profile for this instance
+ apply-groups           Groups from which to inherit configuration data
+ apply-groups-except    Don't inherit configuration data from these groups
> bridge-domains         Bridge domain configuration
  description            Text description of routing instance
> forwarding-options     Forwarding options configuration
  instance-type          Type of routing instance
> interface              Interface name for this routing instance
> multicast-snooping-options  Multicast snooping option configuration
  no-irb-layer-2-copy    Disable transmission of layer-2 copy of packets of irb
routing-interface
  no-local-switching     Disable local switching within CE-facing interfaces
  …
```

You create VPLS virtual private network (VPN) routing and forwarding (VRF) tables at the **`[edit routing-instances]`** portion of the hierarchy. You specify a VPLS instance with arguments applied to the **`instance-type`** statement. As with a Layer 3 VPN VRF instance, you must assign a route distinguisher, list the VRF interfaces, and link the instance with a **`vrf-target`** community or VRF import and export policies. You also must configure local site properties under the **`[edit routing-instances`** ***`instance-name`*** **`protocols vpls]`** hierarchy.

BGP VPLS Signaling

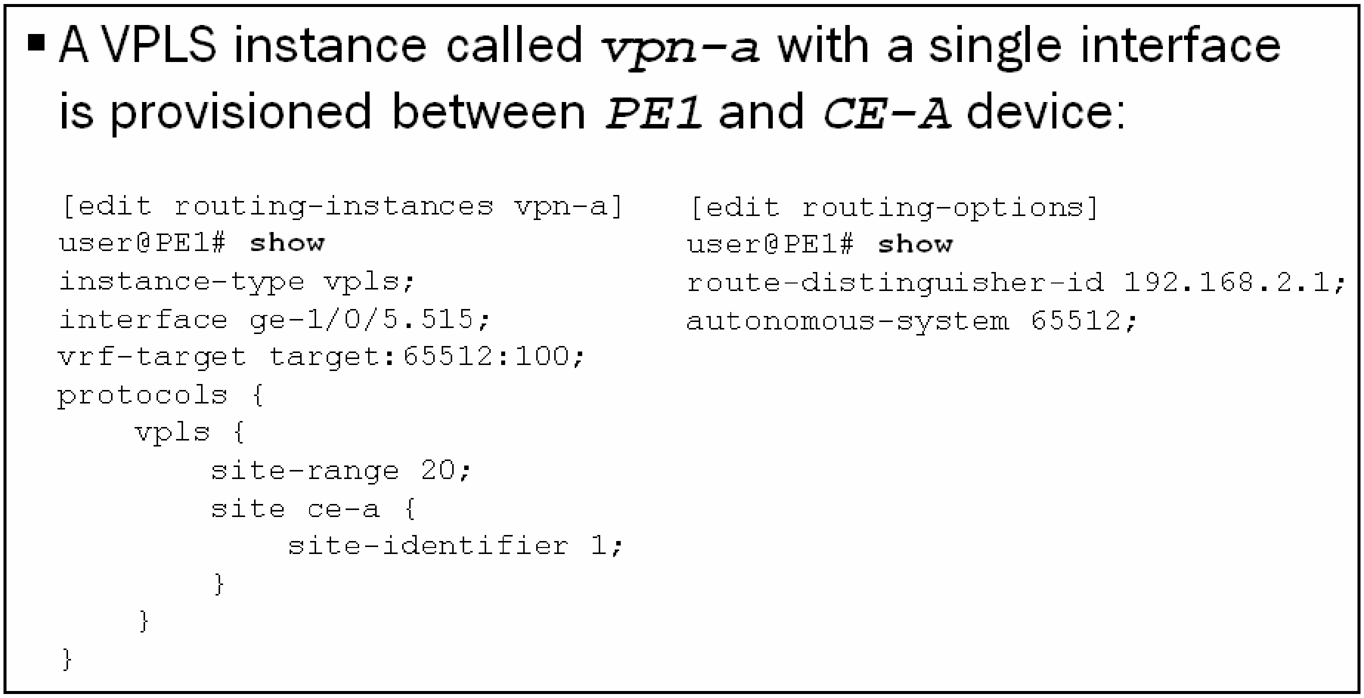


- Set up BGP sessions between the PEs with Layer 2 VPN signaling enabled

```
user@PE1> show configuration protocols bgp
family l2vpn {
    signaling;
}
group my-int-group {
    type internal;
    local-address 192.168.2.1;
    export statics;
    neighbor 192.168.2.2;
    neighbor 192.168.2.3;
}

user@PE1> show bgp summary
Groups: 1 Peers: 2 Down peers: 0
Table          Tot Paths  Act Paths Suppressed    History Damp State    Pending
inet.0                 0          0          0          0          0          0
inet.2                 0          0          0          0          0          0
bgp.l2vpn.0            2          2          0          0          0          0
Peer                     AS      InPkt     OutPkt    OutQ   Flaps Last Up/Dwn
State|#Active/Received/Accepted/Damped...
192.168.2.2           65512          5          6       0       0        1:16 Establ
  bgp.l2vpn.0: 2/2/2/0
  vpn-a.l2vpn.0: 2/2/2/0
…
```

In this example, BGP sessions are configured between the PE with `family l2vpn` signaling configured. This configuration is the same family that is used for point-to-point Layer 2 VPNs.

Sample BGP VPLS Instance



- A VPLS instance called *vpn-a* with a single interface is provisioned between *PE1* and *CE-A* device:

```
[edit routing-instances vpn-a]
user@PE1# show
instance-type vpls;
interface ge-1/0/5.515;
vrf-target target:65512:100;
protocols {
    vpls {
        site-range 20;
        site ce-a {
            site-identifier 1;
        }
    }
}
```

```
[edit routing-options]
user@PE1# show
route-distinguisher-id 192.168.2.1;
autonomous-system 65512;
```

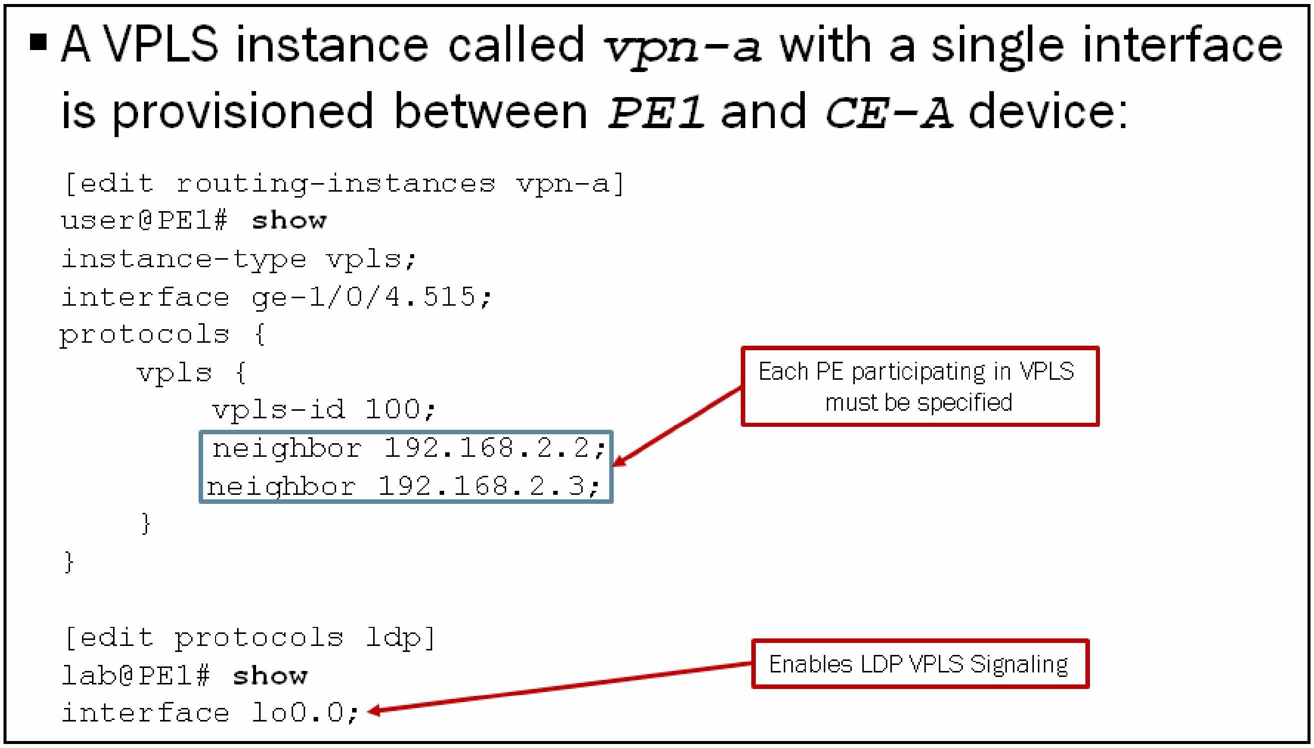
This graphic shows a sample VPLS routing instance based on the sample topology. This instance is called ***vpn-a***. The instance is assigned a route distinguisher based on the PE router's loopback address (Type 1 format). The `instance-type vpls` setting creates a VPLS VRF.

This ***vpn-a*** instance is associated with a single logical interface (`ge-1/0/5.515`). Additional interfaces can be listed if the customer wants to be multihomed.

You can link the VPLS VRF table to either VRF import and export policies or a `vrf-target` statement, which is used to match and add route target communities.

The local site properties are configured under the protocols portion of the VPLS instance. These parameters were discussed in the previous pages.

LDP VPLS Instance Example



- A VPLS instance called *vpn-a* with a single interface is provisioned between *PE1* and *CE-A* device:

```
[edit routing-instances vpn-a]
user@PE1# show
instance-type vpls;
interface ge-1/0/4.515;
protocols {
    vpls {
        vpls-id 100;
        neighbor 192.168.2.2;
        neighbor 192.168.2.3;
    }
}

[edit protocols ldp]
lab@PE1# show
interface lo0.0;
```

Each PE participating in VPLS must be specified

Enables LDP VPLS Signaling

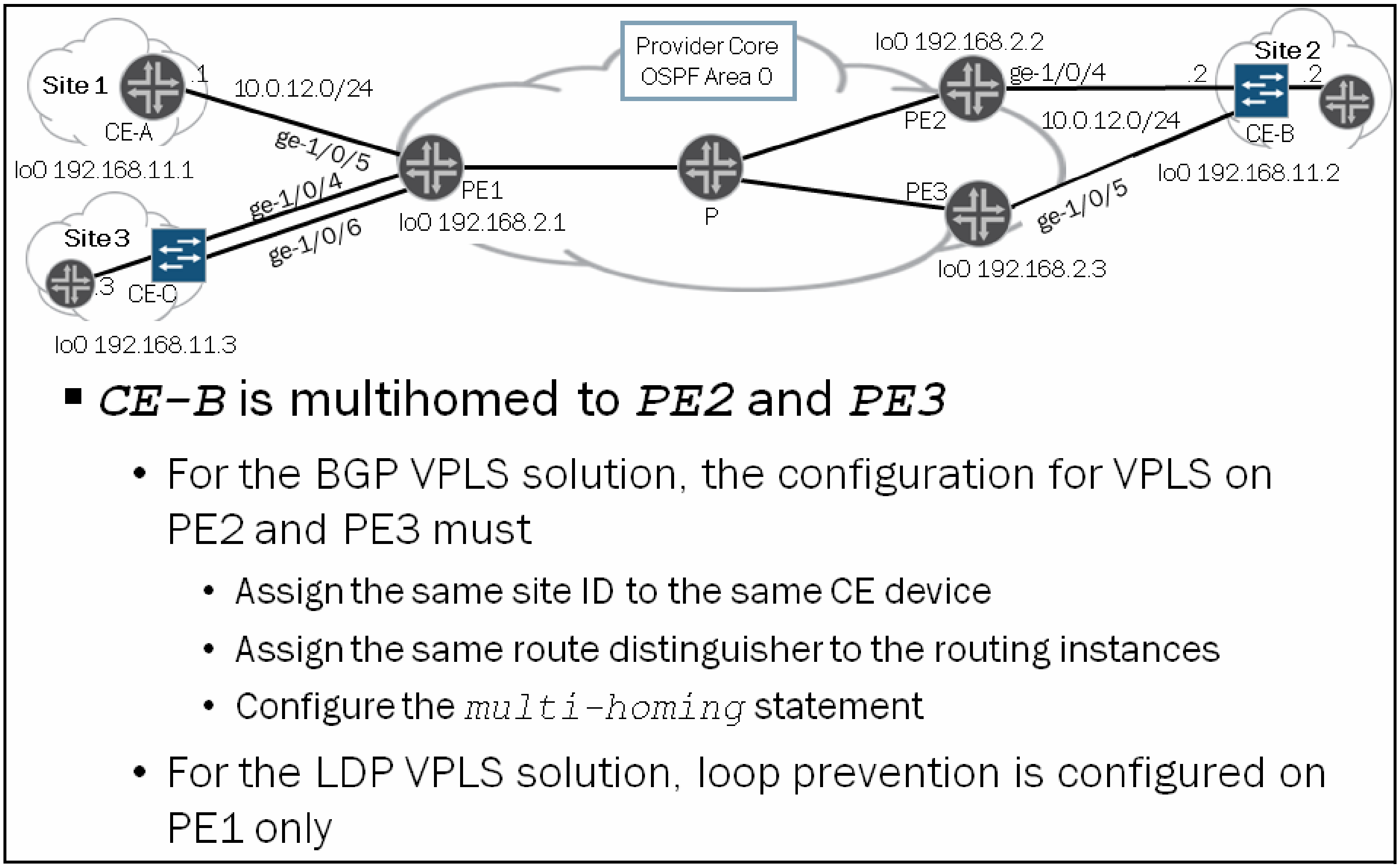
You can configure LDP as the signaling protocol for a VPLS routing instance instead of BGP. The functionality is described in RFC 4762, “VPLS Using LDP Signaling”.

The Junos OS does not support all of RFC 4762. When enabling LDP signaling for a VPLS routing instance, network engineers should be aware that only the following values are supported:

- Forwarding equivalence class (FEC)—FEC 128;
- Control bit—0; and
- Ethernet pseudowire type—hexadecimal 0x0005.

To enable LDP signaling for the set of PE routers participating in the same VPLS routing instance, you need to use the `vpls-id` statement configured at the `[edit routing-instances routing-instance-name protocols vpls]` hierarchy level to configure the same VPLS identifier on each of the PE routers. The VPLS identifier must be globally unique. When each VPLS routing instance (domain) has a unique VPLS identifier, it is possible to configure multiple VPLS routing instances between a given pair of PE routers.

LDP signaling requires that you configure a full mesh LDP session between the PE routers in the same VPLS routing instance. Neighboring PE routers are statically configured. Tunnels are created between the neighboring PE routers to aggregate traffic from one PE router to another. Pseudowires are then signaled to demultiplex traffic between VPLS routing instances. These PE routers exchange the pseudowire label, the MPLS label that acts as the VPLS pseudowire demultiplexer field, by using LDP FECs. Tunnels based on both MPLS and generic routing encapsulation (GRE) are supported.

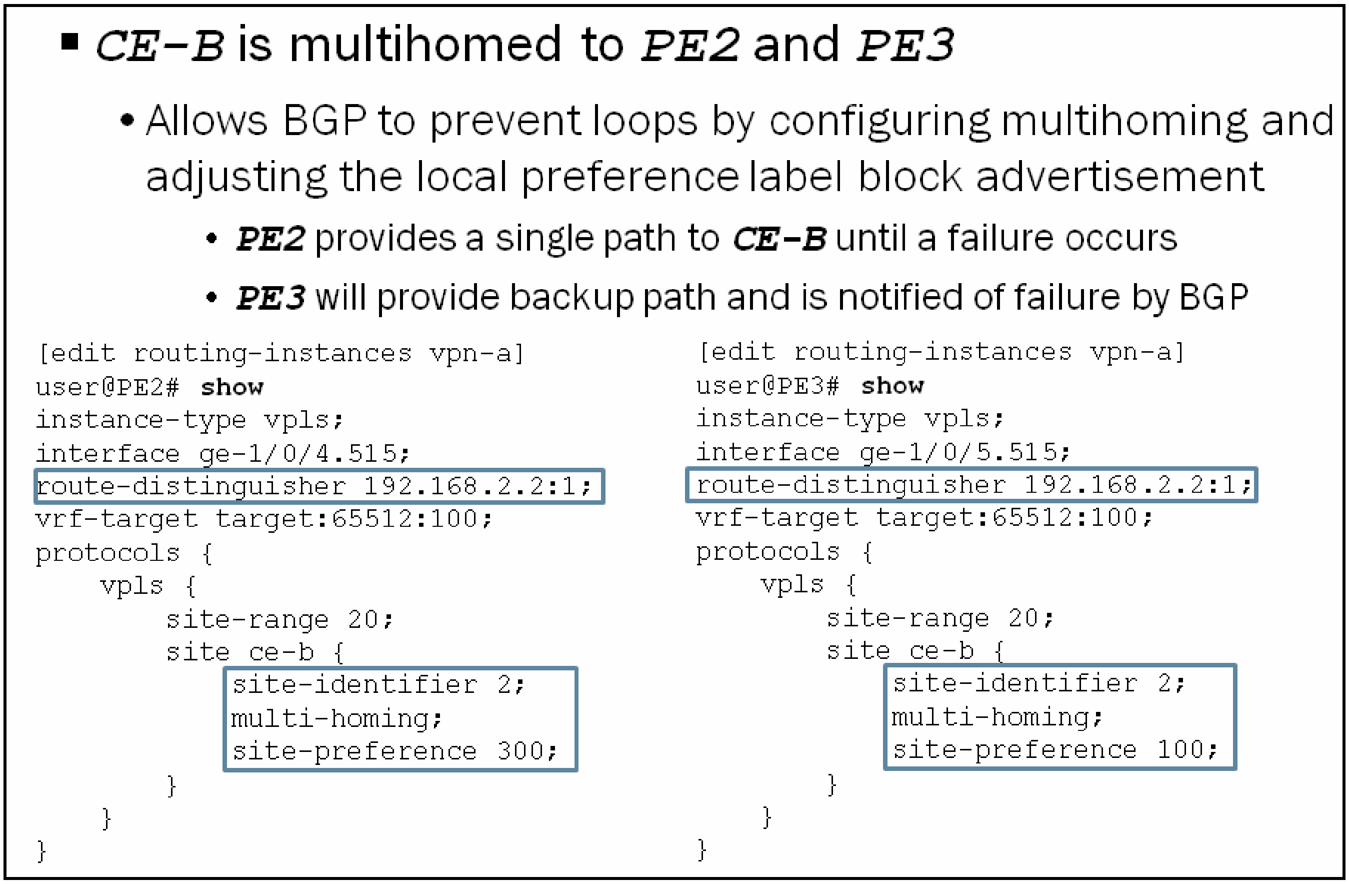


- ***CE-B*** is multihomed to ***PE2*** and ***PE3***
  - For the BGP VPLS solution, the configuration for VPLS on PE2 and PE3 must
    - Assign the same site ID to the same CE device
    - Assign the same route distinguisher to the routing instances
    - Configure the *multi-homing* statement
  - For the LDP VPLS solution, loop prevention is configured on PE1 only

CE with Multiple Interfaces to Multiple PEs

The graphic discusses the options to prevent a Layer 2 loop in the case that CE-B is an Ethernet switch.

BGP Solution



- ***CE-B*** is multihomed to ***PE2*** and ***PE3***
  - Allows BGP to prevent loops by configuring multihoming and adjusting the local preference label block advertisement
    - ***PE2*** provides a single path to ***CE-B*** until a failure occurs
    - ***PE3*** will provide backup path and is notified of failure by BGP

```
[edit routing-instances vpn-a]
user@PE2# show
instance-type vpls;
interface ge-1/0/4.515;
route-distinguisher 192.168.2.2:1;
vrf-target target:65512:100;
protocols {
    vpls {
        site-range 20;
        site ce-b {
            site-identifier 2;
            multi-homing;
            site-preference 300;
        }
    }
}
```

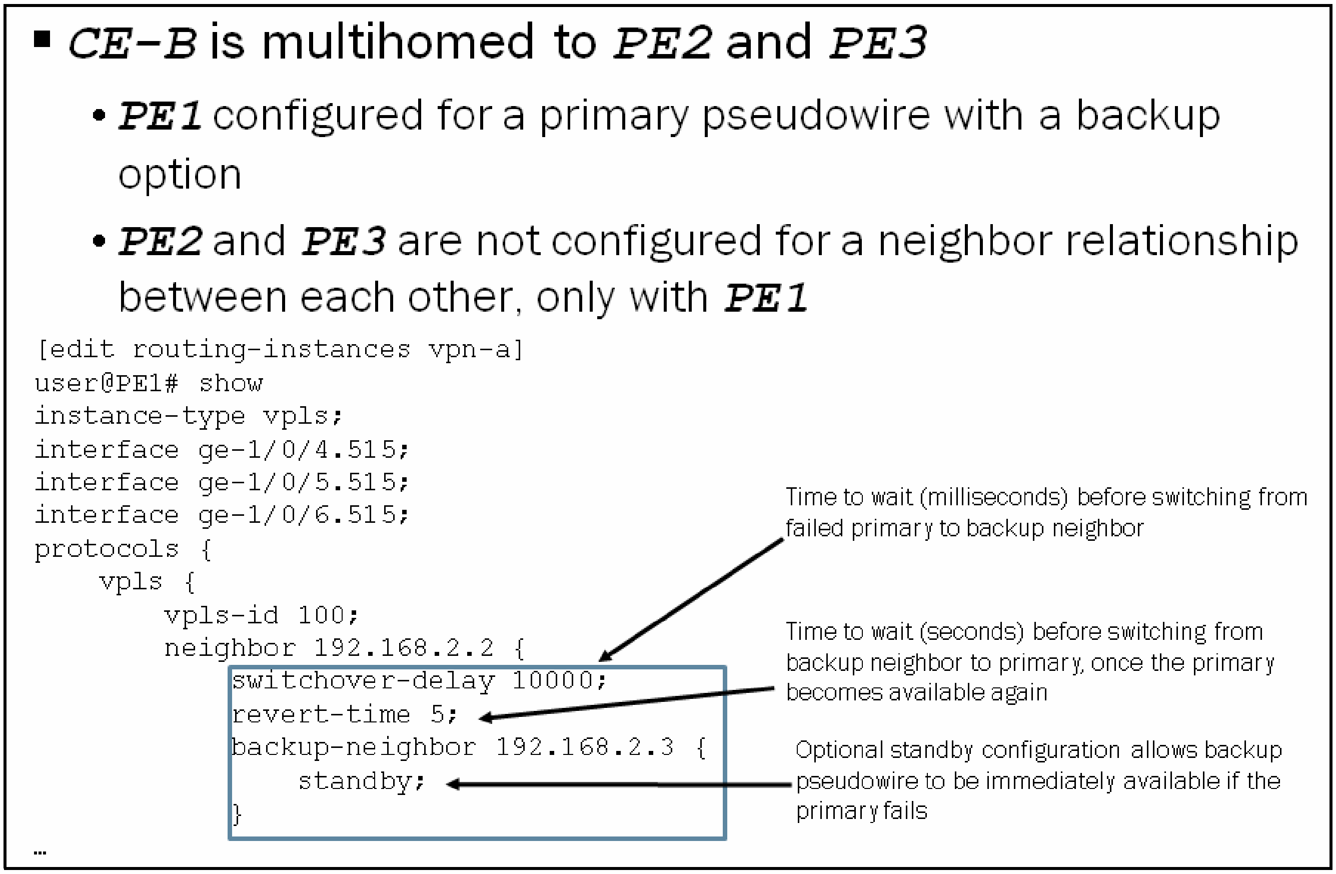
```
[edit routing-instances vpn-a]
user@PE3# show
instance-type vpls;
interface ge-1/0/5.515;
route-distinguisher 192.168.2.2:1;
vrf-target target:65512:100;
protocols {
    vpls {
        site-range 20;
        site ce-b {
            site-identifier 2;
            multi-homing;
            site-preference 100;
        }
    }
}
```

In the example topology, CE-B is a Layer 2 switch that is multihomed to both PE2 and PE3. This redundant topology causes a potential Layer 2 loop. Luckily, because BGP is used to signal the VPLS, BGP's normal route selection process will almost completely prevent a loop from occurring. To allow BGP to prevent the potential loop, you must configure the following on PE2 and PE3:

1. Both routing-instances should be configured for the same route distinguisher.
2. Both sites should be configured with the same site ID.
3. Both sites should be configured with the same target extended community.

The configuration settings in the graphic will make PE2 and PE3 send label block advertisements that appear to be identical except for the BGP next-hop. When the remote PE, PE1, receives the 2 sets of label blocks from PE2 and PE3, PE1 will go through its normal route selection process to determine one set of routes (label blocks) to use for forwarding. The example in the graphic shows that you can affect which label blocks will be chosen by modifying the BGP local preference and set the site-preference (default is 100). In this case, because the label blocks from PE2 are more preferred (Local Preference is 300), PE2 will be chosen as the designated forwarder for the site. PE3 will not forward or learn on its ge-1/0/5.515 interface until PE2 withdraws its label block advertisements (PE2's ge-1/0/4 interface fails). The `multi-homing` command is used to prevent a corner case loop that can occur when BGP connectivity to the core is lost by PE3, it could assume that PE2 is no longer advertising its label blocks and then assume the role of designated forwarder.

LDP Solution



- *CE-B* is multihomed to *PE2* and *PE3*
  - *PE1* configured for a primary pseudowire with a backup option
  - *PE2* and *PE3* are not configured for a neighbor relationship between each other, only with *PE1*

```
[edit routing-instances vpn-a]
user@PE1# show
instance-type vpls;
interface ge-1/0/4.515;
interface ge-1/0/5.515;
interface ge-1/0/6.515;
protocols {
    vpls {
        vpls-id 100;
        neighbor 192.168.2.2 {
            switchover-delay 10000;
            revert-time 5;
            backup-neighbor 192.168.2.3 {
                standby;
            }
…
```

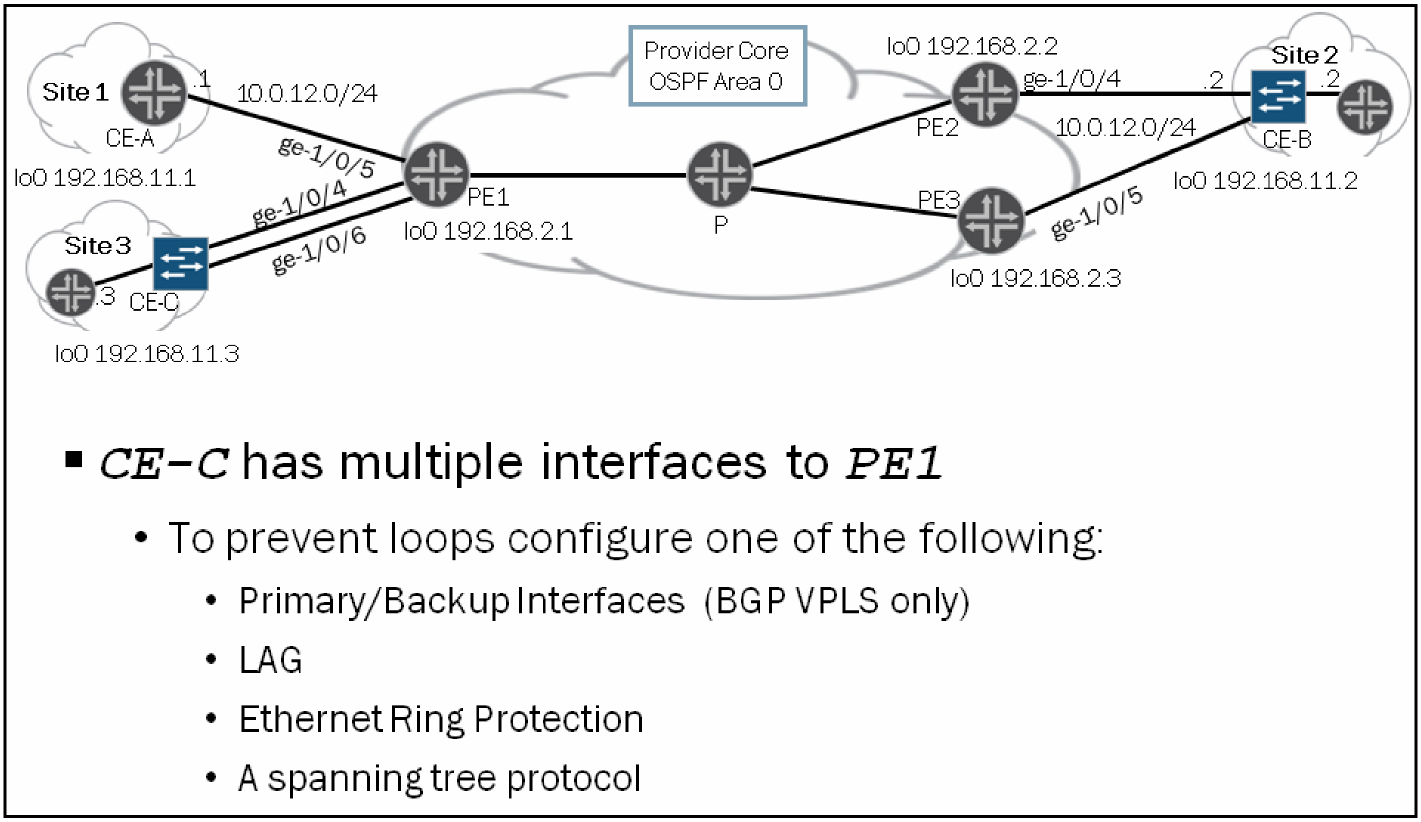
Time to wait (milliseconds) before switching from failed primary to backup neighbor

Time to wait (seconds) before switching from backup neighbor to primary, once the primary becomes available again

Optional standby configuration allows backup pseudowire to be immediately available if the primary fails

To prevent a Layer 2 forwarding loop in this scenario when using an LDP VPLS, special configuration is made on PE1. Assuming that the desired primary forwarding path is between PE1 and PE2 with PE3 acting as a backup. In the VPLS configuration for PE1, PE2 would be listed as a neighbor and PE3 would be listed as a backup neighbor in the event of PE2 failure. PE2 and PE3 would be configured as normal. In PE1's configuration, it is also possible to configure PE3 in standby mode. In that case, PE1 and PE3 would establish a pseudowire between one another even when PE2 is available. Although, PE3 would send broadcast, unicast unknown, multicast (BUM) and so forth to PE1, PE1 will not learn or forward any traffic to or from PE3 while PE2 is available.

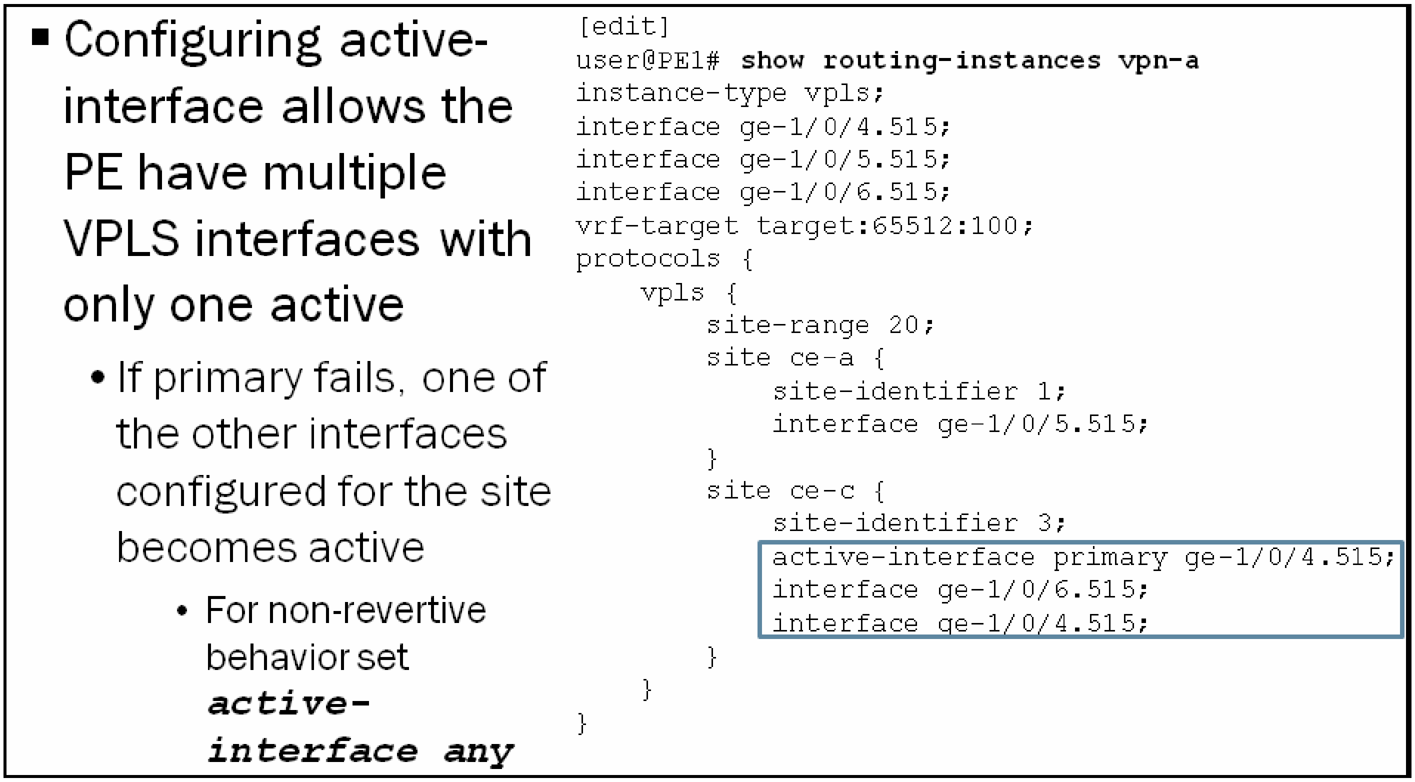
CE with Multiple Interfaces to One PE



- *CE-C* has multiple interfaces to *PE1*
  - To prevent loops configure one of the following:
    - Primary/Backup Interfaces (BGP VPLS only)
    - LAG
    - Ethernet Ring Protection
    - A spanning tree protocol

The graphic discusses the options to prevent a Layer 2 loop in the case that CE-C is an Ethernet switch.

Primary Interface

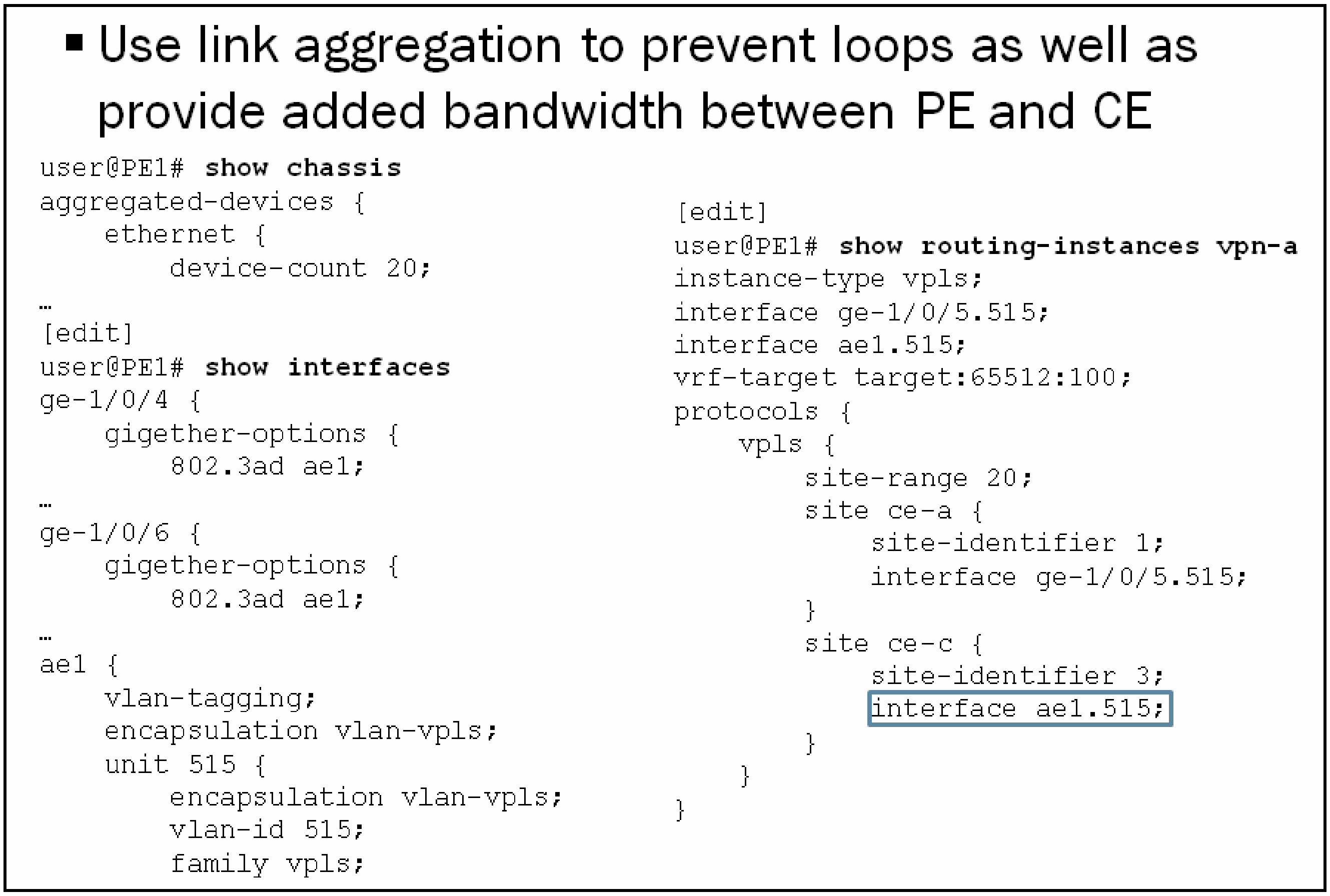


- Configuring active-interface allows the PE have multiple VPLS interfaces with only one active
  - If primary fails, one of the other interfaces configured for the site becomes active
    - For non-revertive behavior set ***active-interface any***

```
[edit]
user@PE1# show routing-instances vpn-a
instance-type vpls;
interface ge-1/0/4.515;
interface ge-1/0/5.515;
interface ge-1/0/6.515;
vrf-target target:65512:100;
protocols {
    vpls {
        site-range 20;
        site ce-a {
            site-identifier 1;
            interface ge-1/0/5.515;
        }
        site ce-c {
            site-identifier 3;
            active-interface primary ge-1/0/4.515;
            interface ge-1/0/6.515;
            interface ge-1/0/4.515;
        }
    }
}
```

It is possible to configure the PE to monitor its own VPLS interfaces, allowing only one interface to be primary and active at any one time. A benefit of this feature is that there is no requirement to run a spanning tree protocol and yet the PE behaves similarly. To use this feature, you must list each interface connected to the site under the site-level configuration. Finally, you must specify an active interface. If you specify a particular interface as the active interface then that interface will be used by the PE for learning and forwarding for the site. All other interfaces will not forward or learn during this period. If the active interface goes down then one of the other configured interfaces will take over as active. Once the primary comes back up, it will again become the active forwarder. For non-revertive behavior, set the active interface to any. There might be packet loss during the failover from one interface to another, however the main concern is Layer 2 loop prevention.

Link Aggregation



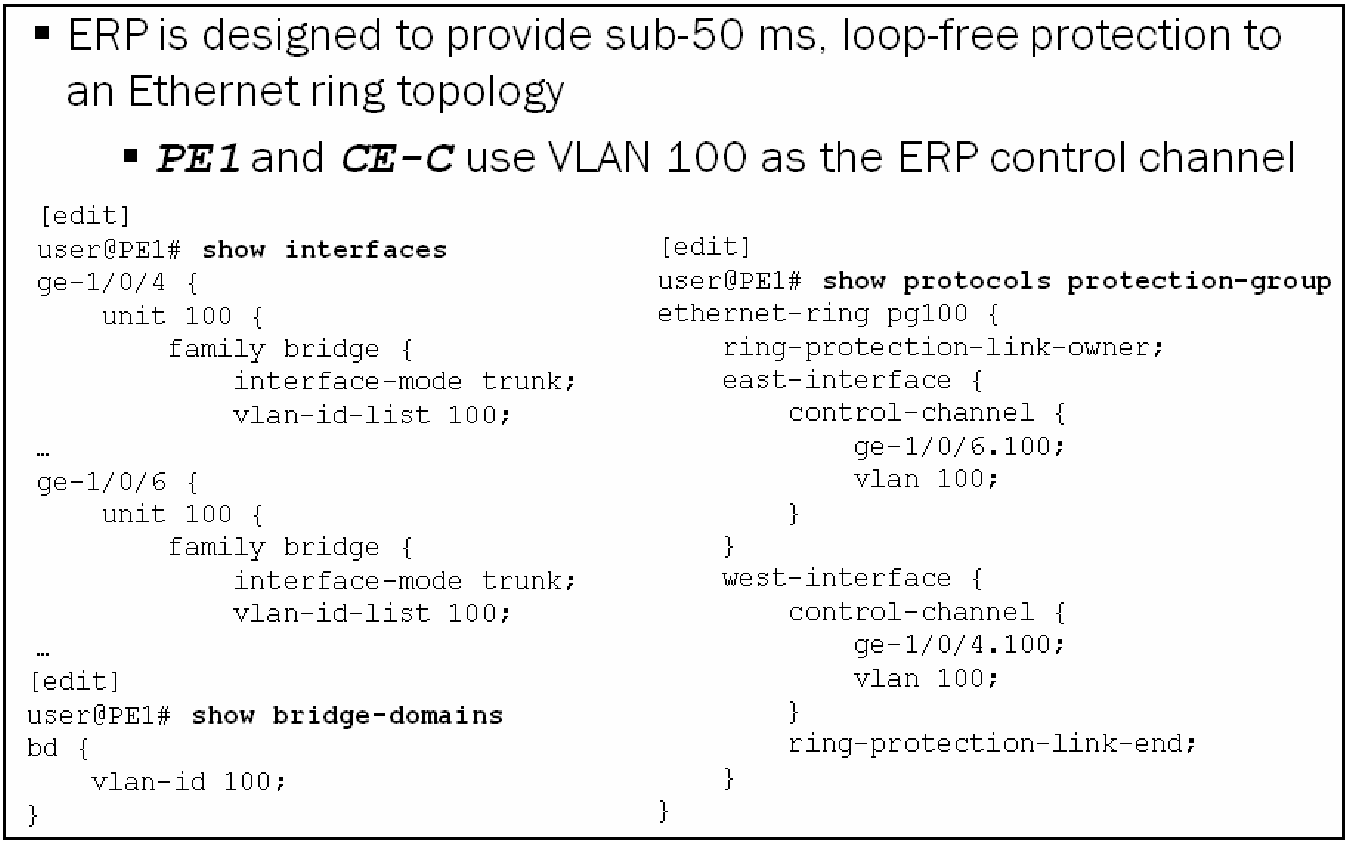
- Use link aggregation to prevent loops as well as provide added bandwidth between PE and CE

```
user@PE1# show chassis
aggregated-devices {
    ethernet {
        device-count 20;
…
[edit]
user@PE1# show interfaces
ge-1/0/4 {
    gigether-options {
        802.3ad ae1;
…
ge-1/0/6 {
    gigether-options {
        802.3ad ae1;
…
ae1 {
    vlan-tagging;
    encapsulation vlan-vpls;
    unit 515 {
        encapsulation vlan-vpls;
        vlan-id 515;
        family vpls;
```

```
[edit]
user@PE1# show routing-instances vpn-a
instance-type vpls;
interface ge-1/0/5.515;
interface ae1.515;
vrf-target target:65512:100;
protocols {
    vpls {
        site-range 20;
        site ce-a {
            site-identifier 1;
            interface ge-1/0/5.515;
        }
        site ce-c {
            site-identifier 3;
            interface ae1.515;
        }
    }
}
```

The configuration example shows the use of a link aggregation group (LAG) to prevent a Layer 2 loop. Instead of two separate 1 Gbps interfaces, it is possible bind them together to make the 2 interfaces logically appear as a single 2 Gbps interfaces to the PE and CE involved. Not only will this configuration allow for Layer 2 loop prevention, it will also double the customer’s access speed to the network.

Ethernet Ring Protection

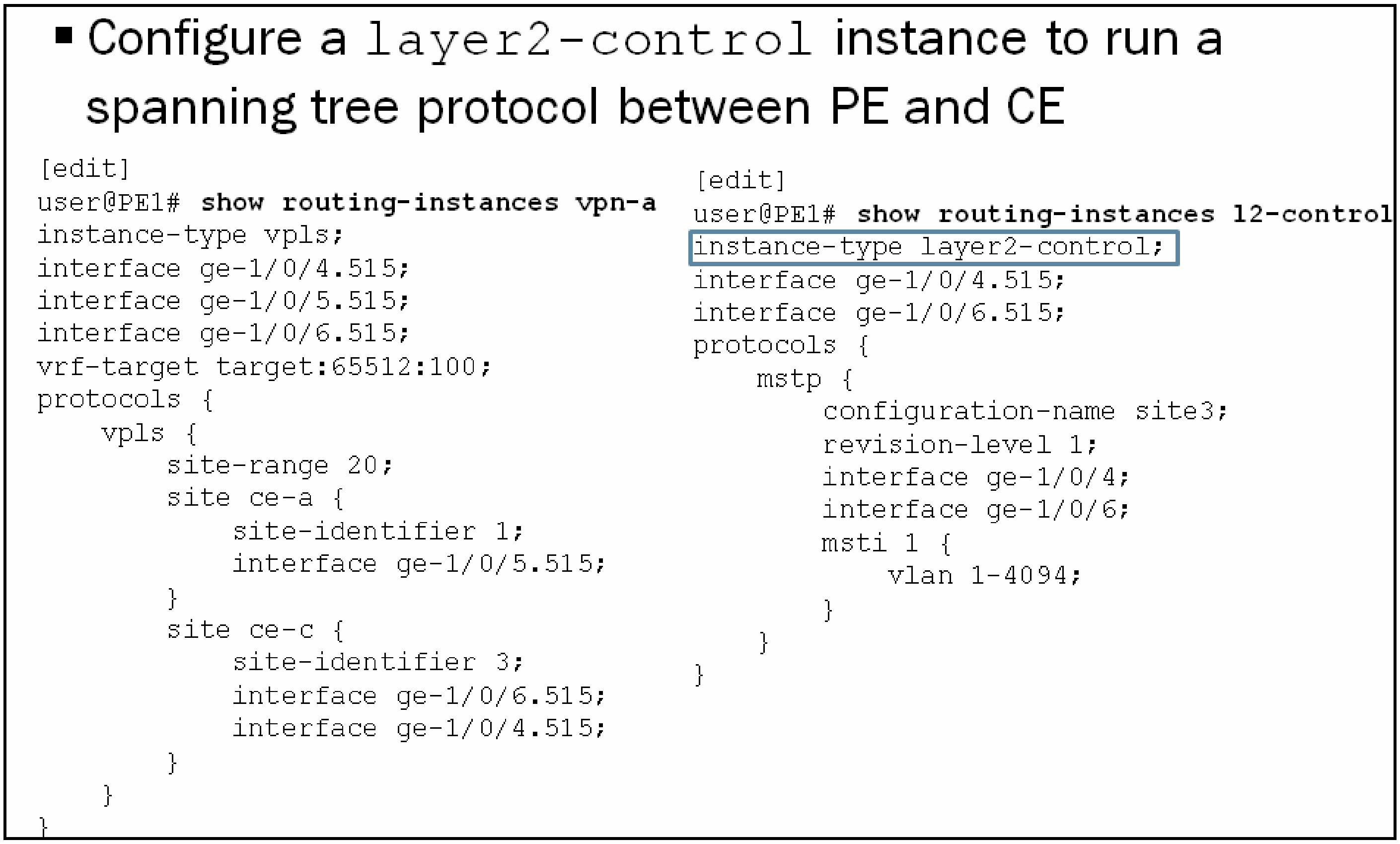


- ERP is designed to provide sub-50 ms, loop-free protection to an Ethernet ring topology
  - **PE1** and **CE-C** use VLAN 100 as the ERP control channel

```
[edit]
user@PE1# show interfaces
ge-1/0/4 {
    unit 100 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 100;
…
ge-1/0/6 {
    unit 100 {
        family bridge {
            interface-mode trunk;
            vlan-id-list 100;
…
[edit]
user@PE1# show bridge-domains
bd {
    vlan-id 100;
}
```

```
[edit]
user@PE1# show protocols protection-group
ethernet-ring pg100 {
    ring-protection-link-owner;
    east-interface {
        control-channel {
            ge-1/0/6.100;
            vlan 100;
        }
    }
    west-interface {
        control-channel {
            ge-1/0/4.100;
            vlan 100;
        }
        ring-protection-link-end;
    }
}
```

The graphic shows the configuration to enable the Ethernet Ring Protection (ERP) control channel on VLAN 100. To protect the Ethernet ring, a single link between ***PE1*** and ***CE-C*** acts as the ring protection link (RPL) on the ring (ge-1/0/4 from the perspective of ***PE1***). ***PE1*** acts as the RPL owner and controls the state of the RPL. During normal operation with no failures (idle state), the RPL owner places the RPL in the blocking state, which results in a loop-free topology. If a link failure occurs somewhere on the ring, the RPL owner places the RPL in a forwarding state until the failed link is repaired. Once the failed link is repaired, the Junos OS acts in a revertive manner, returning the RPL to the blocking state.



- Configure a `layer2-control` instance to run a spanning tree protocol between PE and CE

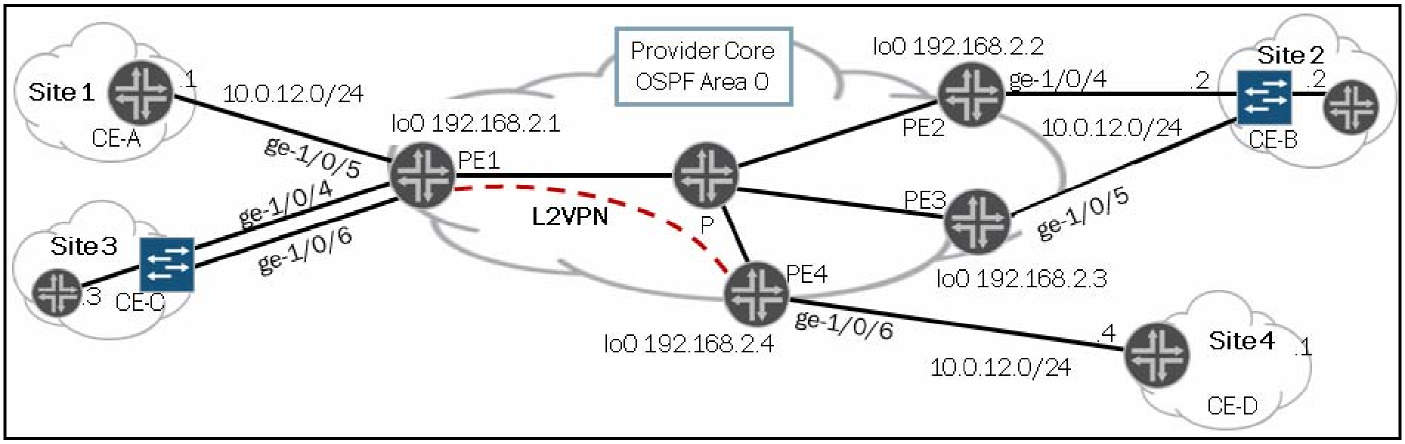
```
[edit]
user@PE1# show routing-instances vpn-a
instance-type vpls;
interface ge-1/0/4.515;
interface ge-1/0/5.515;
interface ge-1/0/6.515;
vrf-target target:65512:100;
protocols {
    vpls {
        site-range 20;
        site ce-a {
            site-identifier 1;
            interface ge-1/0/5.515;
        }
        site ce-c {
            site-identifier 3;
            interface ge-1/0/6.515;
            interface ge-1/0/4.515;
        }
    }
}
```

```
[edit]
user@PE1# show routing-instances l2-control
instance-type layer2-control;
interface ge-1/0/4.515;
interface ge-1/0/6.515;
protocols {
    mstp {
        configuration-name site3;
        revision-level 1;
        interface ge-1/0/4;
        interface ge-1/0/6;
        msti 1 {
            vlan 1-4094;
        }
    }
}
```

Spanning tree protocols cannot be configured directly in a VPLS routing-instance. However, you can use a Layer 2 control routing instance instead. A benefit of using this type of routing instance is that you can run a spanning tree protocol using interfaces that belong to several different VPLS routing instances, not just one. The graphic shows the use of a Layer 2 control routing instance to ensure that no loop exists between the PE and CE. For the topology to work properly, the CE (Layer 2 switch) should also be configured for a spanning tree protocol. Use the **`show spanning-tree interface routing-instance`** ***`instance-name`*** command to view the forwarding and blocking state of the interfaces.

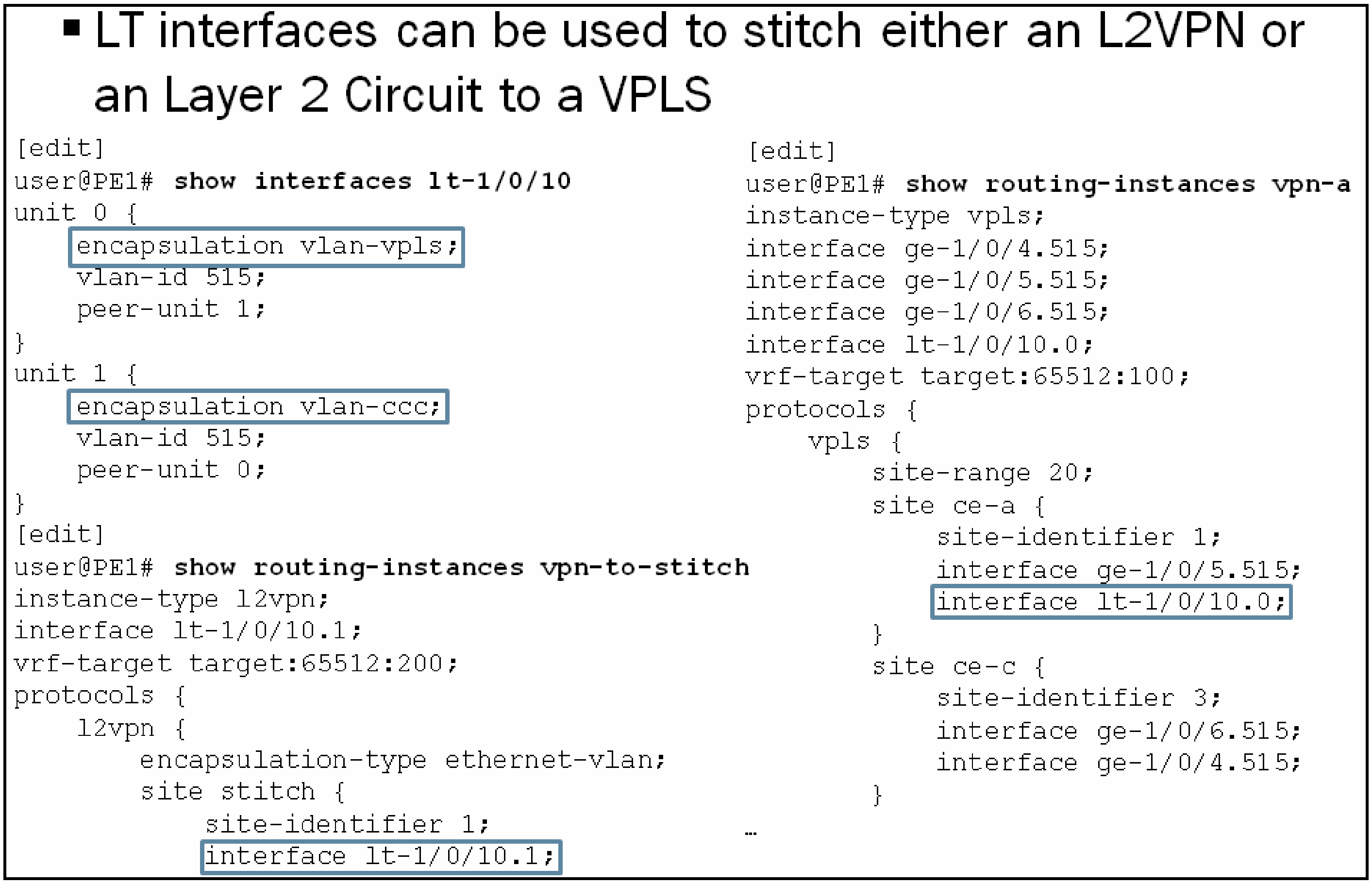
```
user@PE1> show spanning-tree interface routing-instance l2-control
...
Spanning tree interface parameters for instance 1

Interface    Port ID    Designated      Designated         Port    State  Role
                         port ID        bridge ID          Cost
ge-1/0/4        128:45       128:55  32769.80711fc307d1     20000  FWD    ROOT
ge-1/0/6        128:47       128:57  32769.80711fc307d1     20000  BLK    ALT
```

Stitching a Point-to-Point VPN to a VPLS



It is possible to add or stitch a point to point Layer 2 VPN (Layer 2 VPN or Layer 2 Circuit) into a VPLS. To do so, one end of the point-to-point Layer 2 VPN must terminate on a PE that is also a VPLS edge device. Logical tunnel interfaces can be used as the stitching mechanism on the terminating PE.

Stitch Configuration



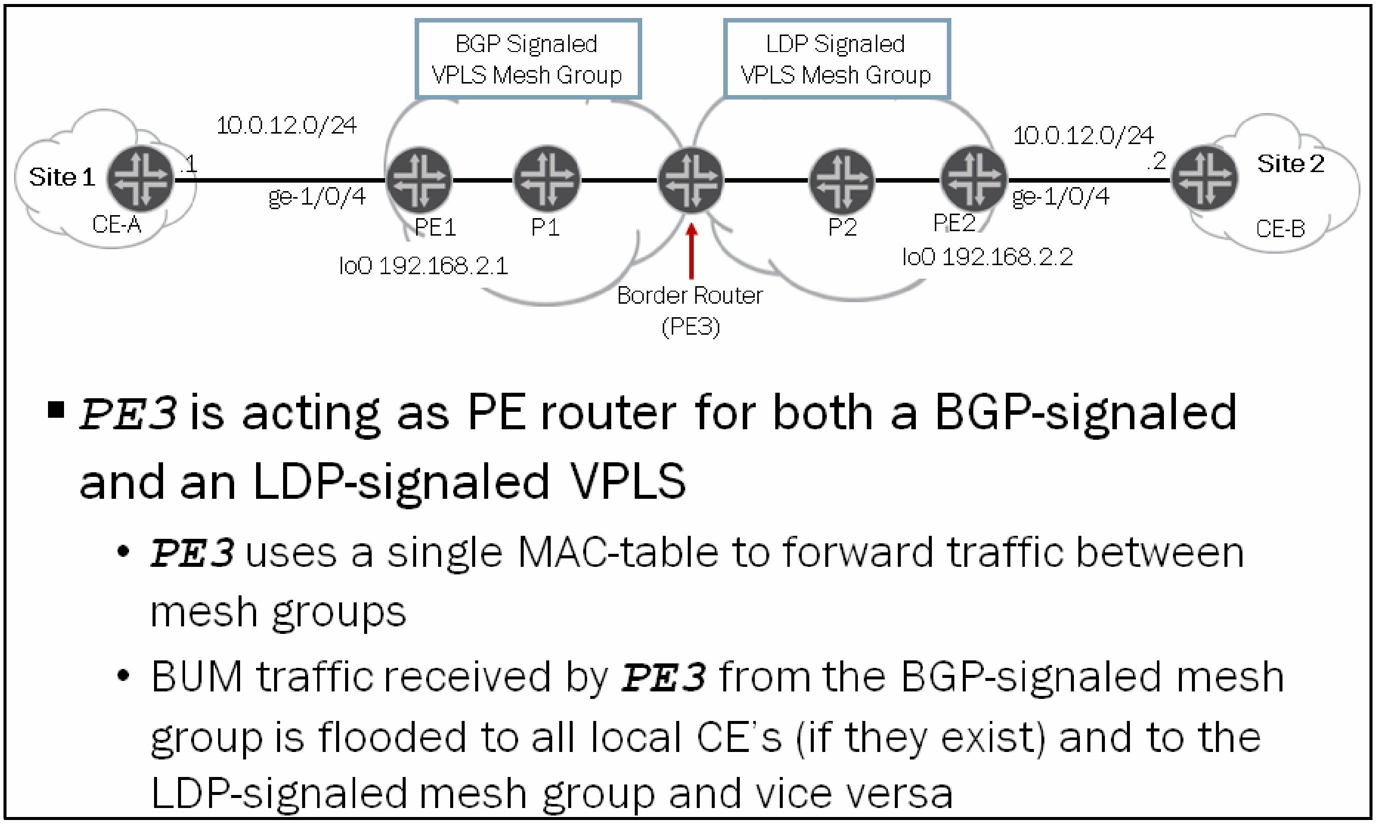
- LT interfaces can be used to stitch either an L2VPN or an Layer 2 Circuit to a VPLS

```
[edit]
user@PE1# show interfaces lt-1/0/10
unit 0 {
    encapsulation vlan-vpls;
    vlan-id 515;
    peer-unit 1;
}
unit 1 {
    encapsulation vlan-ccc;
    vlan-id 515;
    peer-unit 0;
}
[edit]
user@PE1# show routing-instances vpn-to-stitch
instance-type l2vpn;
interface lt-1/0/10.1;
vrf-target target:65512:200;
protocols {
    l2vpn {
        encapsulation-type ethernet-vlan;
        site stitch {
            site-identifier 1;
            interface lt-1/0/10.1;
```

```
[edit]
user@PE1# show routing-instances vpn-a
instance-type vpls;
interface ge-1/0/4.515;
interface ge-1/0/5.515;
interface ge-1/0/6.515;
interface lt-1/0/10.0;
vrf-target target:65512:100;
protocols {
    vpls {
        site-range 20;
        site ce-a {
            site-identifier 1;
            interface ge-1/0/5.515;
            interface lt-1/0/10.0;
        }
        site ce-c {
            site-identifier 3;
            interface ge-1/0/6.515;
            interface ge-1/0/4.515;
        }
…
```

The example shows the stitching of a BGP Layer 2 VPN (L2VPN) to a VPLS. On the terminating PE, there should be a separate routing instance for both the L2VPN and the VPLS. Instead of specifying a physical interface in the L2VPN routing instance, notice that lt-1/0/10.1 is used. The interface configuration for lt-1/0/10.1 uses vlan-ccc encapsulation as expected. Also, the peer interface, lt-1/0/10.0 uses vlan-vpls encapsulation. The last step to the stitching process is to add the lt-1/0/10.0 interface to the VPLS.

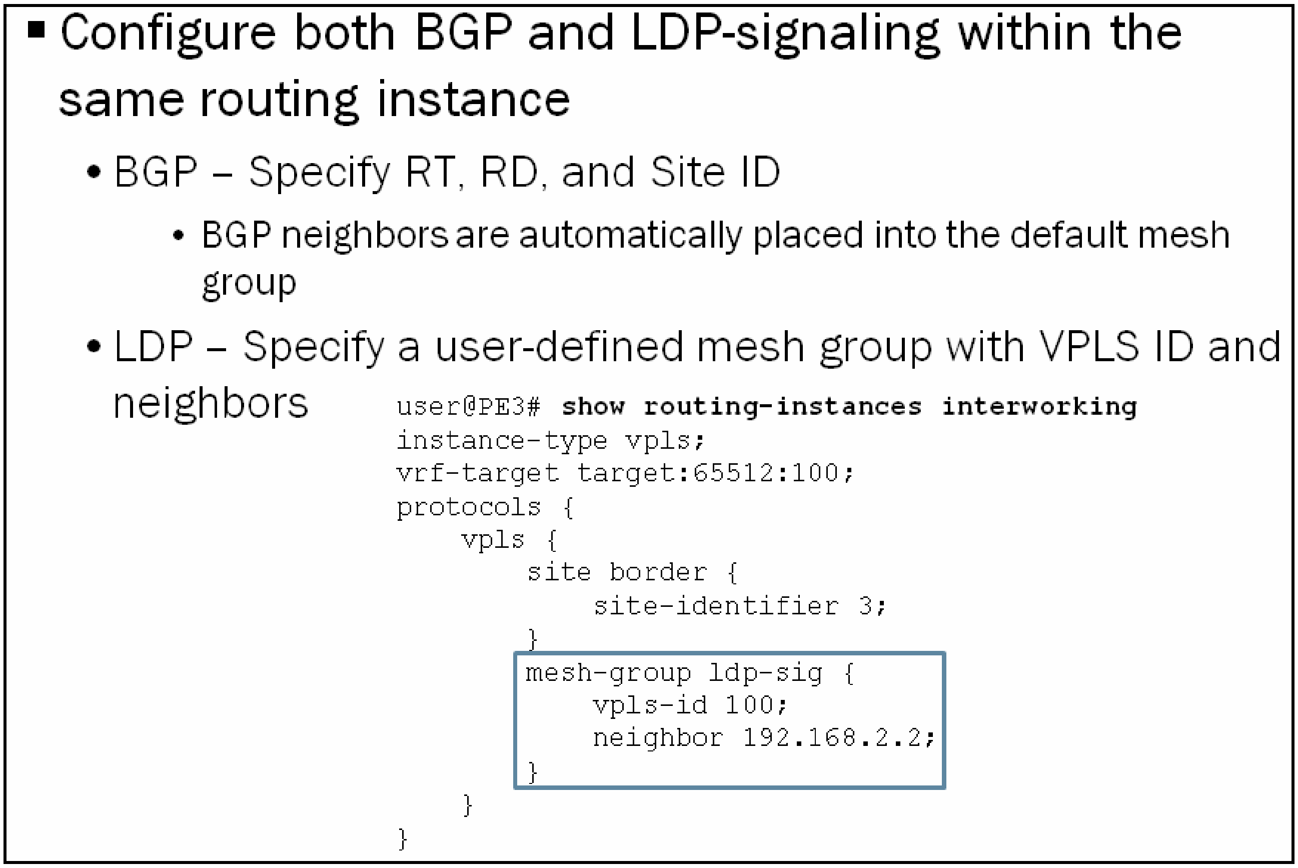
BGP and LDP VPLS Interworking



- ***PE3*** is acting as PE router for both a BGP-signaled and an LDP-signaled VPLS
  - ***PE3*** uses a single MAC-table to forward traffic between mesh groups
  - BUM traffic received by ***PE3*** from the BGP-signaled mesh group is flooded to all local CE's (if they exist) and to the LDP-signaled mesh group and vice versa

There are vendors that make routers that only support LDP-signaled VPLS. BGP and LDP VPLS interworking allows for routers of this type to coexist in a network that uses the benefits of BGP-signaled VPLS. To interconnect these two different VPLS types there must be a single border router that has a full mesh of BGP sessions to the PE's in the BGP-based network (unless route reflection or confederations are used) and a full mesh of LDP sessions to the PEs participating in the LDP VPLS. The border router, PE3, has a single media access control (MAC) table that it uses to learn and forward for both VPLS types. With interworking, the concept of mesh groups has been introduced. In the example in the graphic, the BGP session mesh will fall into one mesh group (the default mesh group) and the LDP session mesh will fall into another mesh group. When BUM traffic arrives from one mesh group, it will be flooded to all CE interfaces as well as all mesh groups for the VPLS except for the one from which the frame arrived. Essentially, the flooding behavior considers a mesh group to be just another CE interface.

Same Routing Instance

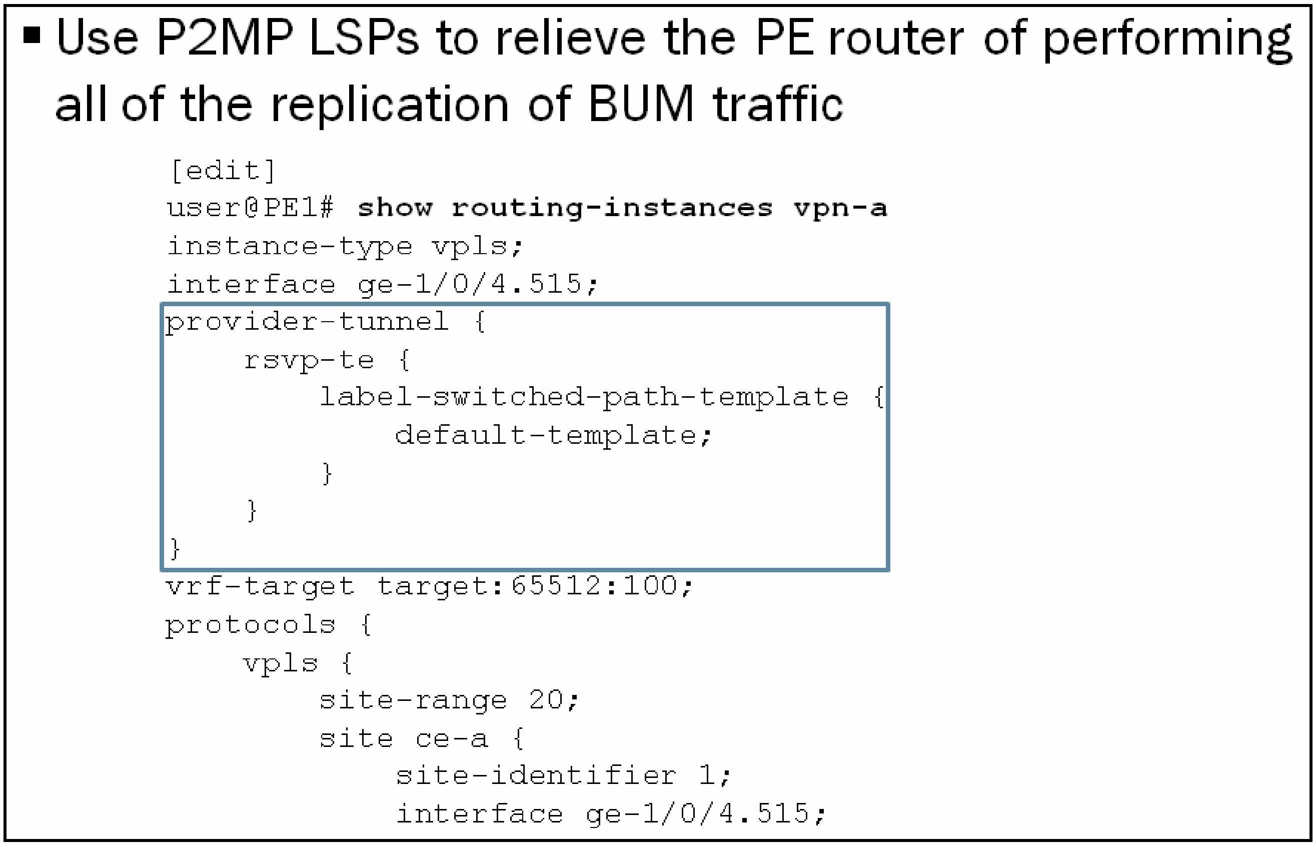


- Configure both BGP and LDP-signaling within the same routing instance
  - BGP – Specify RT, RD, and Site ID
    - BGP neighbors are automatically placed into the default mesh group
  - LDP – Specify a user-defined mesh group with VPLS ID and neighbors

```
user@PE3# show routing-instances interworking
instance-type vpls;
vrf-target target:65512:100;
protocols {
    vpls {
        site border {
            site-identifier 3;
        }
        mesh-group ldp-sig {
            vpls-id 100;
            neighbor 192.168.2.2;
        }
    }
}
```

Unlike the stitching example in the previous graphics, interworking uses a single routing instance. The configuration for both the BGP and LDP VPLS is performed in the single VPLS instance on the border router.

VPLS with Point-to-Multipoint LSPs

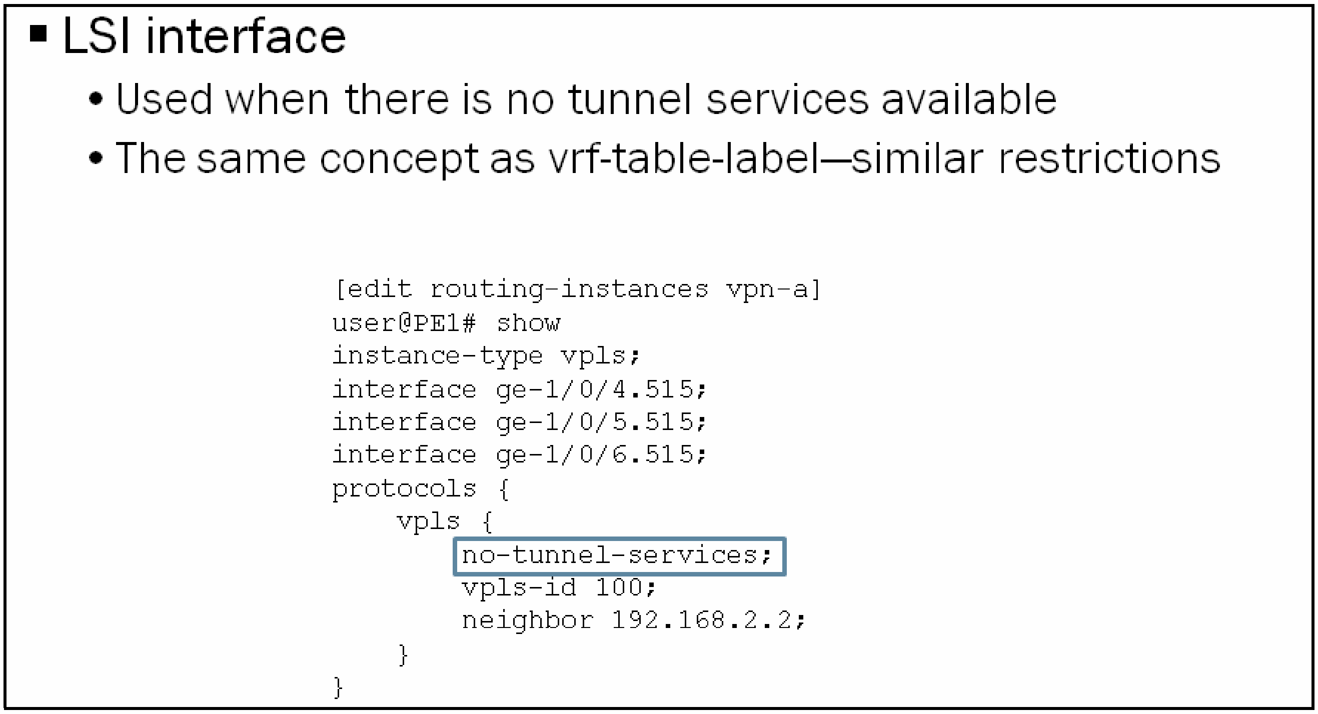


- Use P2MP LSPs to relieve the PE router of performing all of the replication of BUM traffic

```
[edit]
user@PE1# show routing-instances vpn-a
instance-type vpls;
interface ge-1/0/4.515;
provider-tunnel {
    rsvp-te {
        label-switched-path-template {
            default-template;
        }
    }
}
vrf-target target:65512:100;
protocols {
    vpls {
        site-range 20;
        site ce-a {
            site-identifier 1;
            interface ge-1/0/4.515;
```

To allow for a PE to flood BUM traffic using point-to-multipoint LSP, simply configure an RSVP provider tunnel. You can use the default template or you can use a user defined label switched path template.

No Tunnel PIC Required

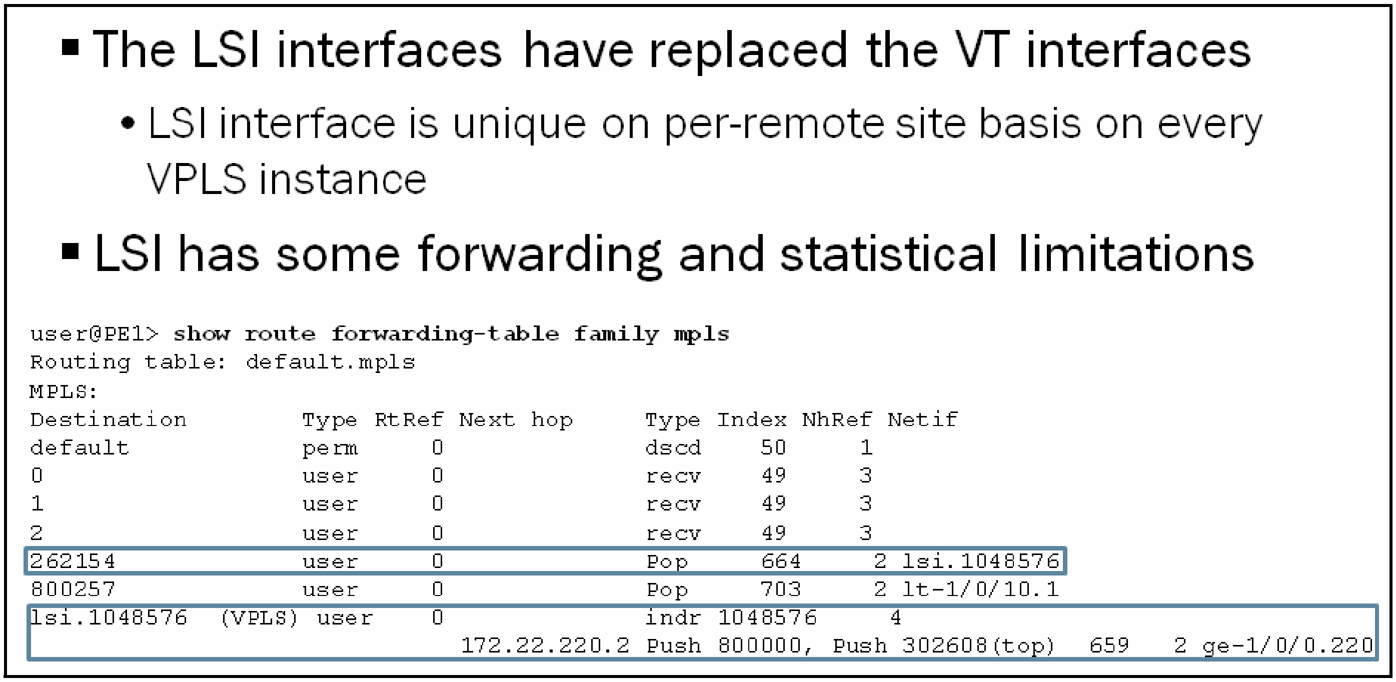


- LSI interface
  - Used when there is no tunnel services available
  - The same concept as vrf-table-label—similar restrictions

```
[edit routing-instances vpn-a]
user@PE1# show
instance-type vpls;
interface ge-1/0/4.515;
interface ge-1/0/5.515;
interface ge-1/0/6.515;
protocols {
    vpls {
        no-tunnel-services;
        vpls-id 100;
        neighbor 192.168.2.2;
    }
}
```

A tunnel pic is no longer required to run VPLS on the Junos OS because the command `no-tunnel-services` can be configured under the routing instance. When this command is configured, instead of seeing VPN tunnel interfaces, label-switched interfaces (LSIs) are used instead. This command has very similar restrictions to the `vrf-table-label` command.

LSI Interface



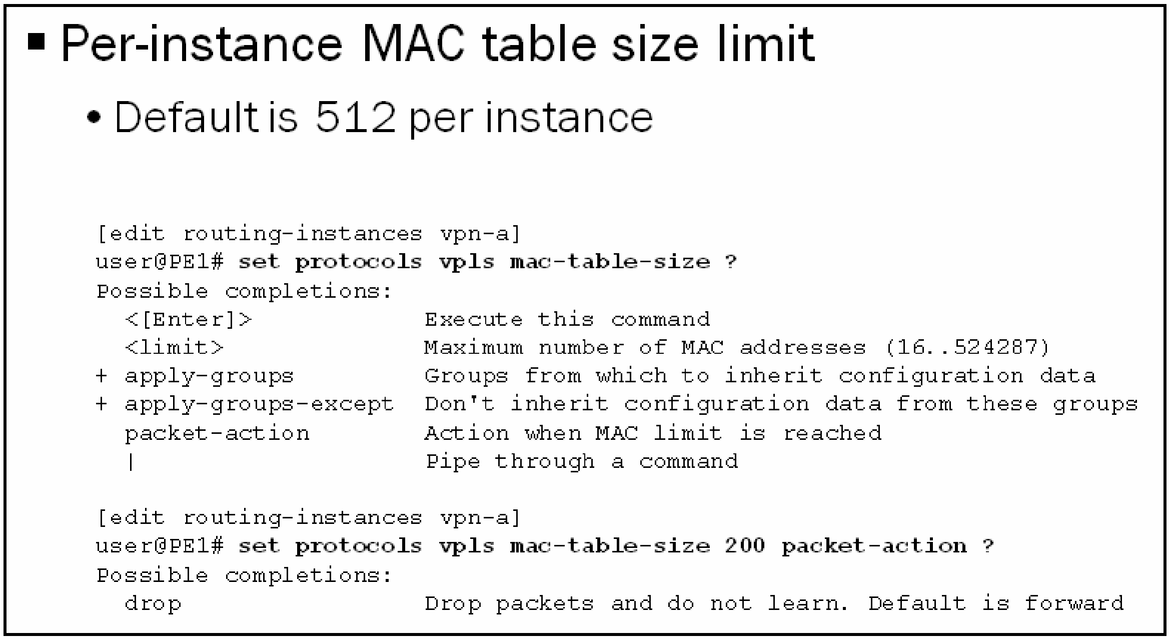
- The LSI interfaces have replaced the VT interfaces
  - LSI interface is unique on per-remote site basis on every VPLS instance
- LSI has some forwarding and statistical limitations

```
user@PE1> show route forwarding-table family mpls
Routing table: default.mpls
MPLS:
Destination        Type RtRef Next hop     Type Index NhRef Netif
default            perm     0              dscd    50     1
0                  user     0              recv    49     3
1                  user     0              recv    49     3
2                  user     0              recv    49     3
262154             user     0              Pop    664     2 lsi.1048576
800257             user     0              Pop    703     2 lt-1/0/10.1
lsi.1048576  (VPLS) user    0              indr 1048576     4
                               172.22.220.2 Push 800000, Push 302608(top)   659   2 ge-1/0/0.220
```

The LSIs replace the use of the VPN tunnel interfaces inside the forwarding table of the router, but all the forwarding concepts stay the same. In other words, a unique LSI interface is still created for every remote site and used for packets received from remote PEs.

The use of the LSI interface does have a few limitations. The forwarding rate is limited on a per LSI basis and his variable per router type. When using tunnel services, the forwarding rate can be increased by adding more Tunnel PICs to the router (or enabling them on an MX Series Ethernet Services router).

MAC Table Size



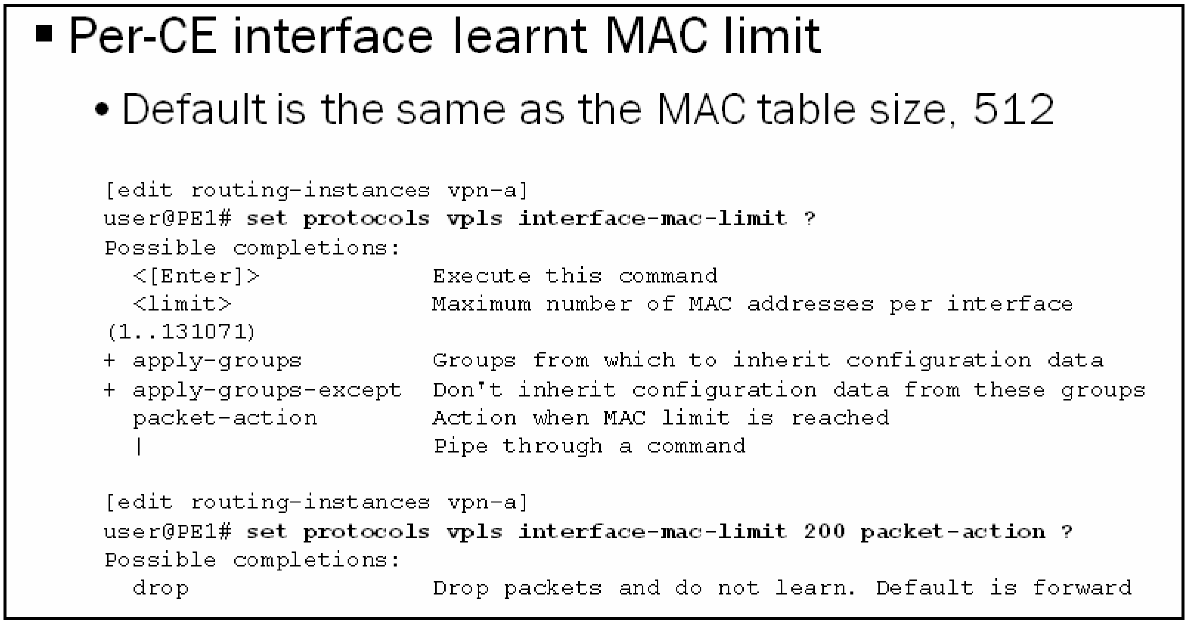
- Per-instance MAC table size limit
  - Default is 512 per instance

```
[edit routing-instances vpn-a]
user@PE1# set protocols vpls mac-table-size ?
Possible completions:
  <[Enter]>            Execute this command
  <limit>              Maximum number of MAC addresses (16..524287)
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  packet-action        Action when MAC limit is reached
  |                    Pipe through a command

[edit routing-instances vpn-a]
user@PE1# set protocols vpls mac-table-size 200 packet-action ?
Possible completions:
  drop                 Drop packets and do not learn. Default is forward
```

You can modify the size of the VPLS MAC address table. The default table size is 512 MAC addresses, the minimum is 16 addresses, and the maximum is 527,287 addresses.

If the MAC table limit is reached, new MAC addresses can no longer be added to the table. Eventually the oldest MAC addresses are removed from the MAC address table automatically. The removal of MAC addresses frees space in the table, allowing new entries to be added. However, as long as the table is full, new MAC addresses are not learned however traffic will continue to be forwarded using the process of flooding by default. You can also specify to have the router drop traffic to unknown destinations when the MAC table is full.

MAC Table Size Limit



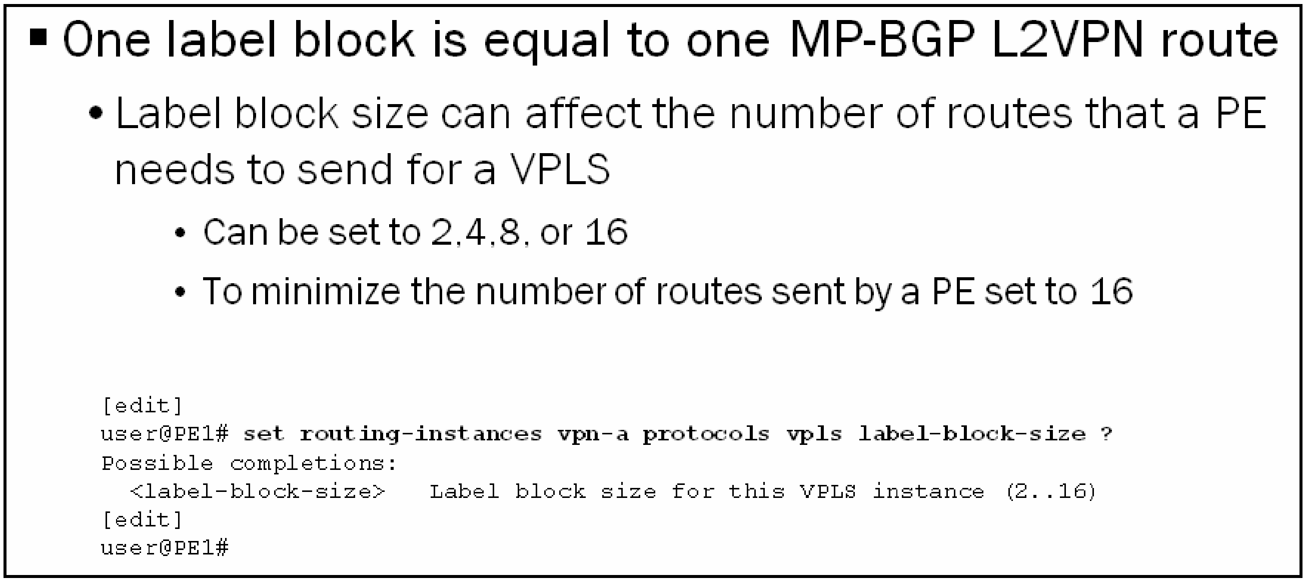
- Per-CE interface learnt MAC limit
  - Default is the same as the MAC table size, 512

```
[edit routing-instances vpn-a]
user@PE1# set protocols vpls interface-mac-limit ?
Possible completions:
  <[Enter]>            Execute this command
  <limit>              Maximum number of MAC addresses per interface
(1..131071)
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
  packet-action        Action when MAC limit is reached
  |                    Pipe through a command

[edit routing-instances vpn-a]
user@PE1# set protocols vpls interface-mac-limit 200 packet-action ?
Possible completions:
  drop                 Drop packets and do not learn. Default is forward
```

You can limit the total system MAC table size as shown on the previous graphic. Because this limit applies to each VPLS routing instance, the MAC addresses of a single interface can consume all the available space in the table, preventing the routing instance from acquiring addresses from other interfaces.

You can limit the number of MAC addresses learned from each interface configured for a VPLS routing instance.

Label Block Size

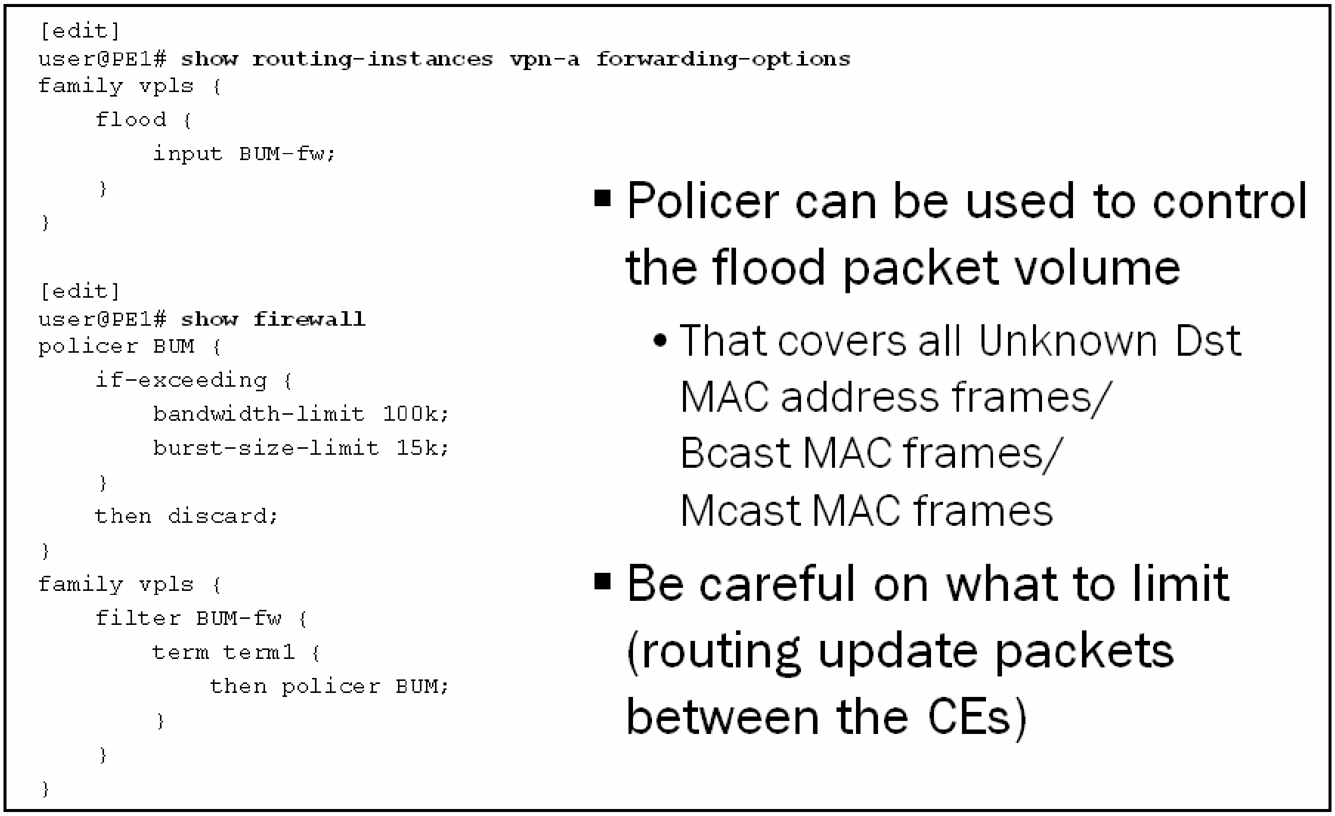


- One label block is equal to one MP-BGP L2VPN route
  - Label block size can affect the number of routes that a PE needs to send for a VPLS
    - Can be set to 2,4,8, or 16
    - To minimize the number of routes sent by a PE set to 16

```
[edit]
user@PE1# set routing-instances vpn-a protocols vpls label-block-size ?
Possible completions:
  <label-block-size>   Label block size for this VPLS instance (2..16)
[edit]
user@PE1#
```

A PE will advertise enough labels to ensure that each remote site that it learns can send traffic downstream and upstream from itself. A PE can advertise blocks of labels in sets of 2, 4, 8, or 16. If there are giant gaps in site IDs, then it is possible that many of the advertised labels will go unused. To minimize the wasted label allocations you can configure a lower label-block size. However, a lower label block size will force the PE to advertise many routes to represent the full set of sites. If your concern is to keep the number of route advertisement low, then set the label block size higher. The default is 8.

Rate Limits



```
[edit]
user@PE1# show routing-instances vpn-a forwarding-options
family vpls {
    flood {
        input BUM-fw;
    }
}

[edit]
user@PE1# show firewall
policer BUM {
    if-exceeding {
        bandwidth-limit 100k;
        burst-size-limit 15k;
    }
    then discard;
}
family vpls {
    filter BUM-fw {
        term term1 {
            then policer BUM;
        }
    }
}
```

- Policer can be used to control the flood packet volume
  - That covers all Unknown Dst MAC address frames/ Bcast MAC frames/ Mcast MAC frames
- Be careful on what to limit (routing update packets between the CEs)

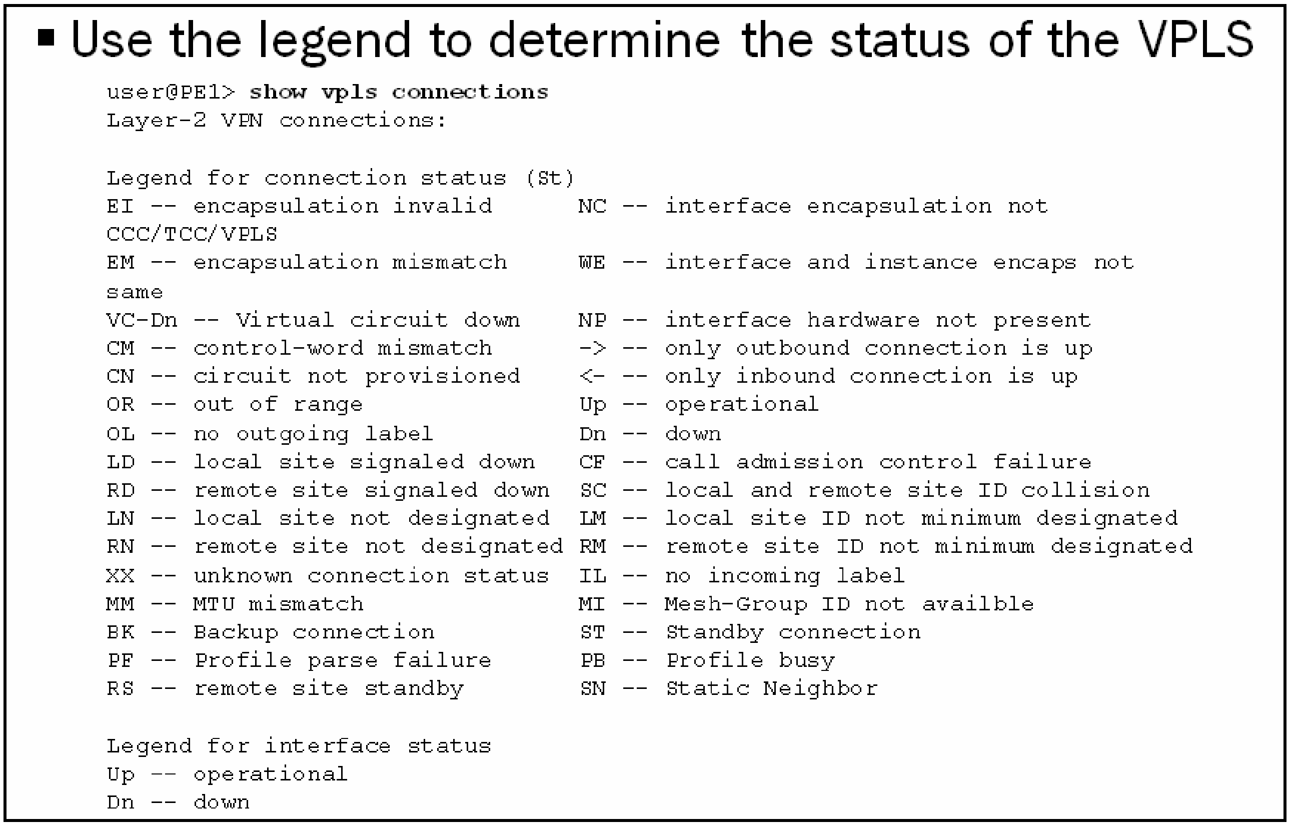
A policer can be used to rate limit traffic that is being flooded. Rate limiting can be configured for all traffic or from certain MAC addresses if matched in the from statement:

```
[edit]
user@PE# set firewall family vpls filter foo term 1 from ?
Possible completions:
+ apply-groups         Groups from which to inherit configuration data
+ apply-groups-except  Don't inherit configuration data from these groups
> destination-mac-address  Destination MAC address
+ ether-type           Match Ethernet type
+ ether-type-except    Do not match Ethernet type
+ forwarding-class     Match forwarding class
+ forwarding-class-except  Do not match forwarding class
+ interface-group      Match interface group
+ interface-group-except  Do not match interface group
> source-mac-address   Source MAC address
+ vlan-ether-type      Match VLAN Ethernet type
+ vlan-ether-type-except  Do not match VLAN Ethernet type
```

Be Careful What You Wish For

Take proper care when applying a VPLS policer and remember the variety of packets that are being flooded. For instance, routing protocol packets might be sent to other CEs and could be rate limited by the policer!

VPLS Connections Legend

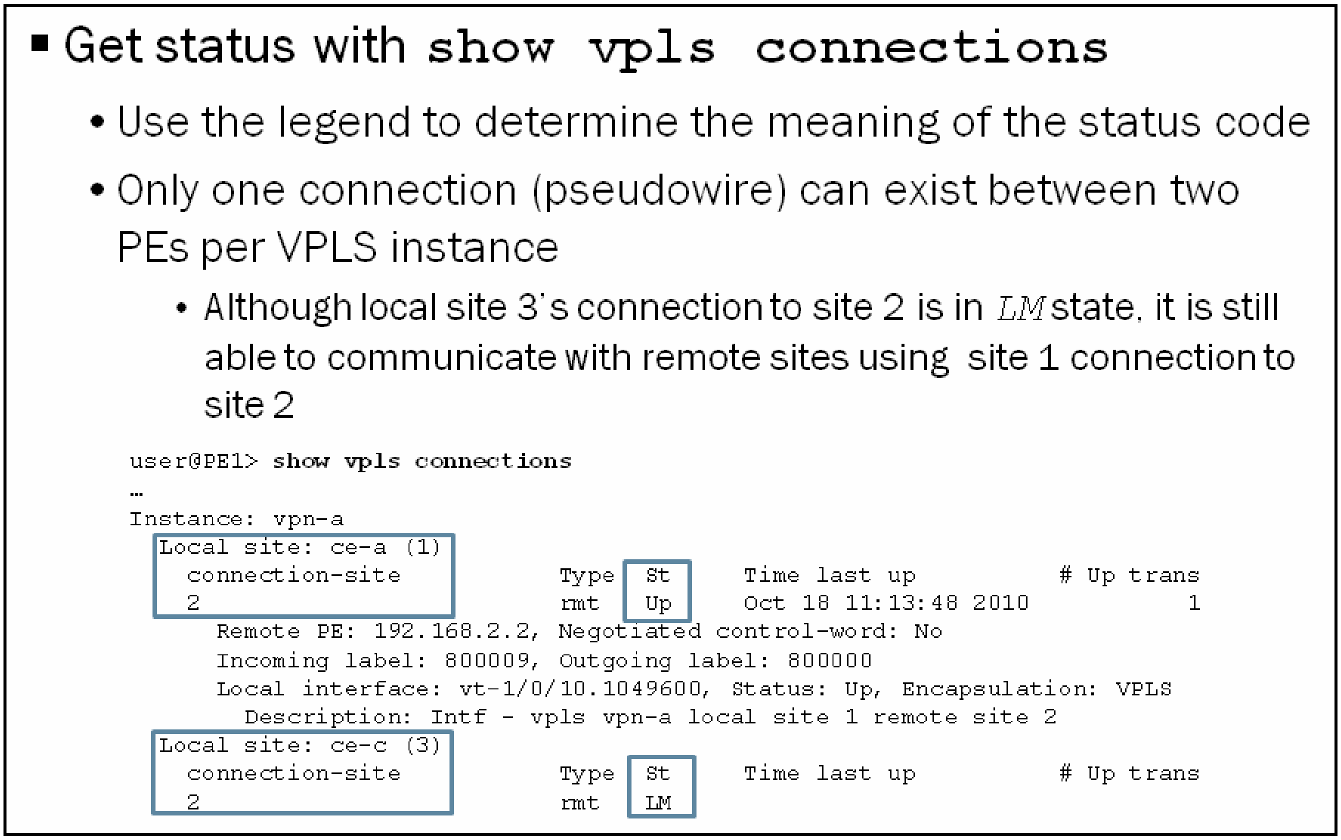


- Use the legend to determine the status of the VPLS

```
user@PE1> show vpls connections
Layer-2 VPN connections:

Legend for connection status (St)
EI -- encapsulation invalid      NC -- interface encapsulation not
CCC/TCC/VPLS
EM -- encapsulation mismatch     WE -- interface and instance encaps not
same
VC-Dn -- Virtual circuit down    NP -- interface hardware not present
CM -- control-word mismatch      -> -- only outbound connection is up
CN -- circuit not provisioned    <- -- only inbound connection is up
OR -- out of range               Up -- operational
OL -- no outgoing label          Dn -- down
LD -- local site signaled down   CF -- call admission control failure
RD -- remote site signaled down  SC -- local and remote site ID collision
LN -- local site not designated  LM -- local site ID not minimum designated
RN -- remote site not designated RM -- remote site ID not minimum designated
XX -- unknown connection status  IL -- no incoming label
MM -- MTU mismatch               MI -- Mesh-Group ID not availble
BK -- Backup connection          ST -- Standby connection
PF -- Profile parse failure      PB -- Profile busy
RS -- remote site standby        SN -- Static Neighbor

Legend for interface status
Up -- operational
Dn -- down
```

The **`show vpls connection`** command is an excellent command to help you determine the status of a VPLS. Every time that you issue the command the legend shown on the graphic will be displayed followed by a listing of each VPLS and its status (see next section). The legend will help you interpret the status code for each VPLS.

show vpls connections

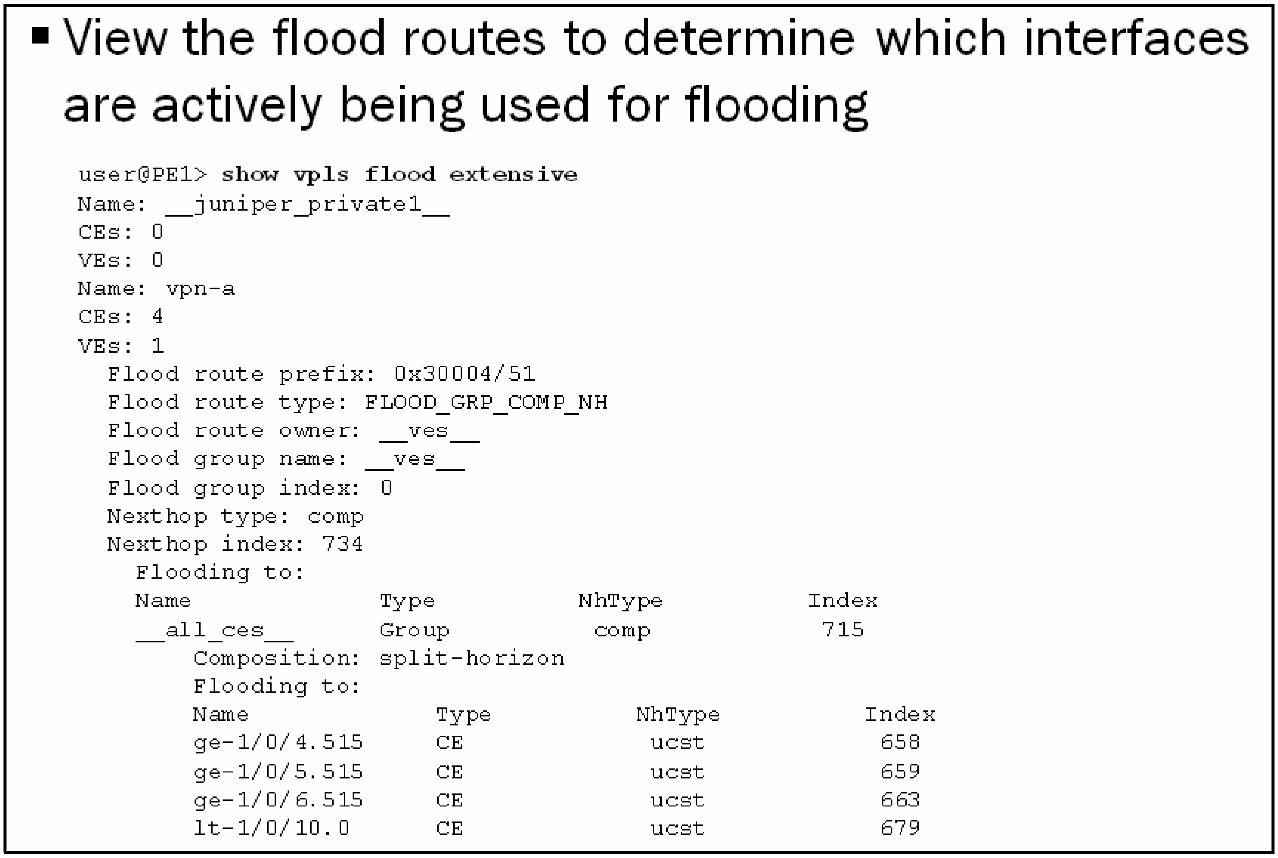


- Get status with `show vpls connections`
  - Use the legend to determine the meaning of the status code
  - Only one connection (pseudowire) can exist between two PEs per VPLS instance
    - Although local site 3's connection to site 2 is in *LM* state, it is still able to communicate with remote sites using site 1 connection to site 2

```
user@PE1> show vpls connections
…
Instance: vpn-a
  Local site: ce-a (1)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   Up     Oct 18 11:13:48 2010           1
      Remote PE: 192.168.2.2, Negotiated control-word: No
      Incoming label: 800009, Outgoing label: 800000
      Local interface: vt-1/0/10.1049600, Status: Up, Encapsulation: VPLS
        Description: Intf - vpls vpn-a local site 1 remote site 2
  Local site: ce-c (3)
    connection-site           Type  St     Time last up          # Up trans
    2                         rmt   LM
```

This output shows the status of the VPLS from site 1 and site 3 to site 2 (refer to the example network). Remember that site 1 and site 3 were configured under the same VPLS routing instance. Based on the output on the graphic, the 3 sites should be able to communicate with each other with no problems. You should not be alarmed when you see a status of LM for the site 3 to site 2 connection. When two or more sites are configured under one VPLS instance, the site with the lowest site ID will form the connection to the remote site. All other sites in the local VPLS instance, will use that same connection (pseudowire) to forward and learn. Remember, all of the sites configured under the same VPLS routing instance are also using the same, single MAC table.

Flood Routes



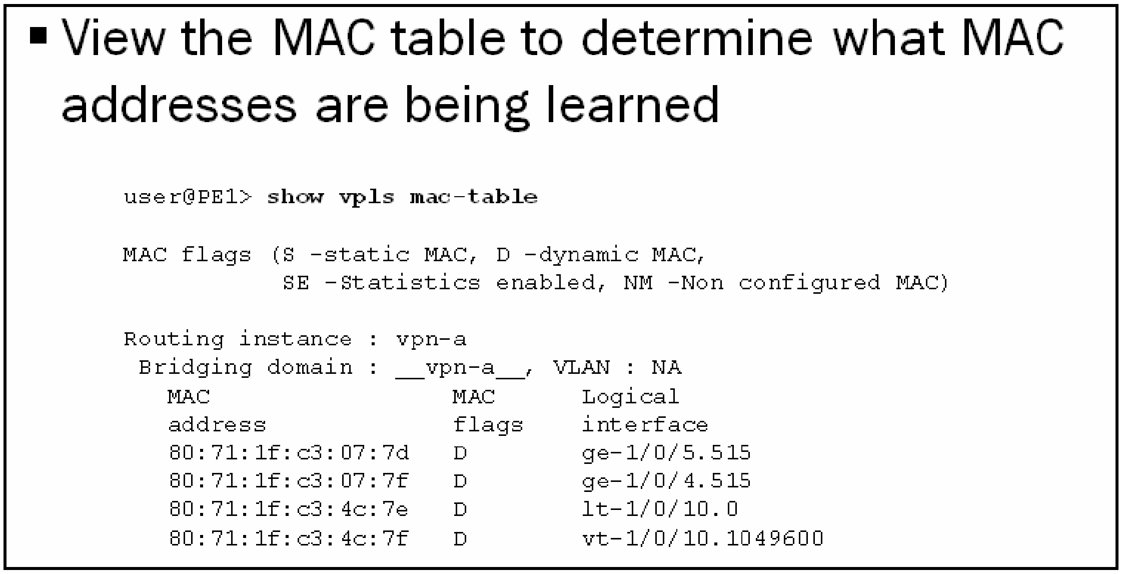
- View the flood routes to determine which interfaces are actively being used for flooding

```
user@PE1> show vpls flood extensive
Name: __juniper_private1__
CEs: 0
VEs: 0
Name: vpn-a
CEs: 4
VEs: 1
  Flood route prefix: 0x30004/51
  Flood route type: FLOOD_GRP_COMP_NH
  Flood route owner: __ves__
  Flood group name: __ves__
  Flood group index: 0
  Nexthop type: comp
  Nexthop index: 734
    Flooding to:
    Name             Type          NhType          Index
    __all_ces__      Group          comp            715
        Composition: split-horizon
        Flooding to:
        Name             Type          NhType          Index
        ge-1/0/4.515     CE              ucst            658
        ge-1/0/5.515     CE              ucst            659
        ge-1/0/6.515     CE              ucst            663
        lt-1/0/10.0      CE              ucst            679
```

To determine the current flooding behavior of the VPLS, use the **show vpls flood** command. This command will help you determine which interfaces are being used to learn and forward. If you have configured an active interface for a multihomed PE, this command is great to help determine which interface is currently active.

Why are there so many flood routes? You should normally expect to see 3 flood routes in the VPLS forwarding table. The flooding behavior on a PE is based upon the interface that BUM traffic arrives on. If BUM traffic arrives from a locally connect CE, then the traffic needs to be flooded to all local CEs (except the one the traffic came from) and to all remote PEs. If BUM traffic arrives from a remote PE, then the traffic needs to be flooded to only local CEs, not to any remote PE (because of PE full-mesh). If BUM traffic arrives from the Routing Engine (RE), then the traffic needs to be flooded to all local CEs and to all remote PEs. There should be a flood route for each of the three scenarios.

View the MAC Table

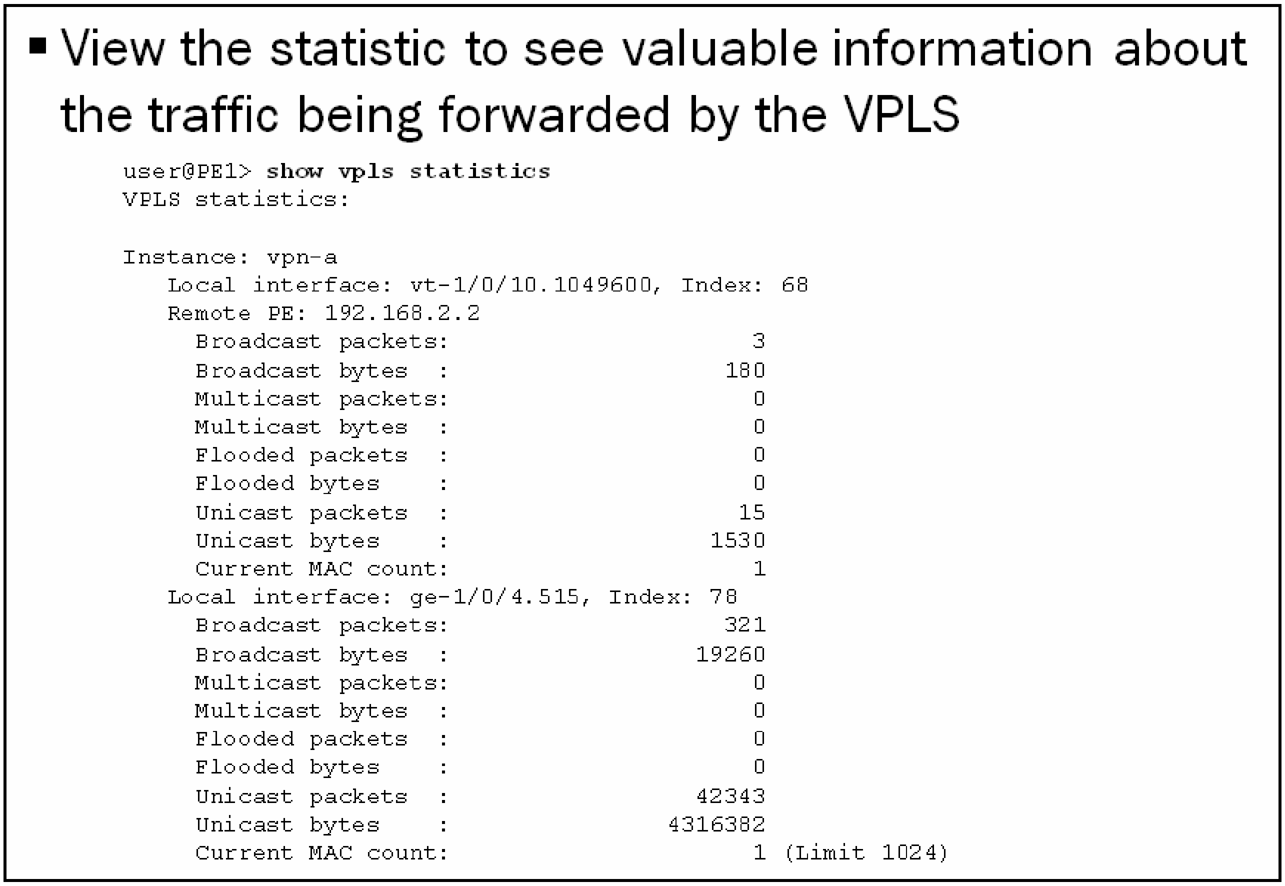


- View the MAC table to determine what MAC addresses are being learned

```
user@PE1> show vpls mac-table

MAC flags (S -static MAC, D -dynamic MAC,
           SE -Statistics enabled, NM -Non configured MAC)

Routing instance : vpn-a
 Bridging domain : __vpn-a__, VLAN : NA
   MAC                 MAC      Logical
   address             flags    interface
   80:71:1f:c3:07:7d   D        ge-1/0/5.515
   80:71:1f:c3:07:7f   D        ge-1/0/4.515
   80:71:1f:c3:4c:7e   D        lt-1/0/10.0
   80:71:1f:c3:4c:7f   D        vt-1/0/10.1049600
```

Use the **`show vpls mac-table`** command to see the routing engines copy of the MAC table. To clear the table, use the **`clear vpls mac-table`** command.

VPLS Statistics



- View the statistic to see valuable information about the traffic being forwarded by the VPLS

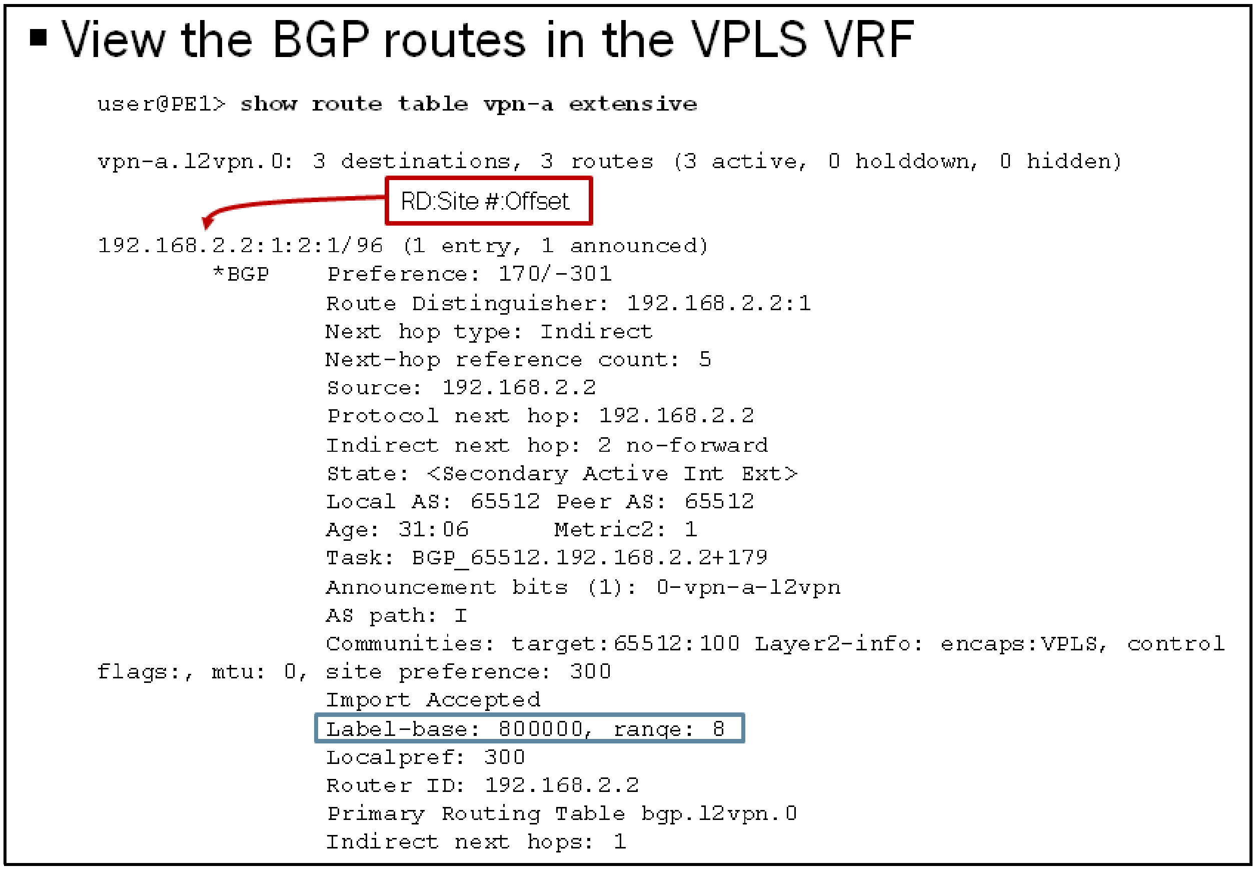
```
user@PE1> show vpls statistics
VPLS statistics:

Instance: vpn-a
   Local interface: vt-1/0/10.1049600, Index: 68
   Remote PE: 192.168.2.2
     Broadcast packets:                    3
     Broadcast bytes  :                  180
     Multicast packets:                    0
     Multicast bytes  :                    0
     Flooded packets  :                    0
     Flooded bytes    :                    0
     Unicast packets  :                   15
     Unicast bytes    :                 1530
     Current MAC count:                    1
   Local interface: ge-1/0/4.515, Index: 78
     Broadcast packets:                  321
     Broadcast bytes  :                19260
     Multicast packets:                    0
     Multicast bytes  :                    0
     Flooded packets  :                    0
     Flooded bytes    :                    0
     Unicast packets  :                42343
     Unicast bytes    :              4316382
     Current MAC count:                    1 (Limit 1024)
```

The following fields are present in the output of the statistics command:

- *Instance*: Name of the VPLS instance.
- *Local interface*: Name of the local VPLS virtual loopback tunnel interface, `vt-fpc/pic/port.`*nnnnn*, where *nnnnn* is a dynamically generated virtual port used to transport and receive packets from other PE routers in the VPLS domain.

- *Index*: Number associated with the next hop.
- *Remote provider edge router*: Address of the remote PE router.
- *Multicast packets*: Number of multicast packets received.
- *Multicast bytes*: Number of multicast bytes received.
- *Flood packets*: Number of VPLS flood packets received.
- *Flood bytes*: Number of VPLS flood bytes received.
- *Current MAC count*: Number of MAC addresses learned by the interface.

VPLS NLRI

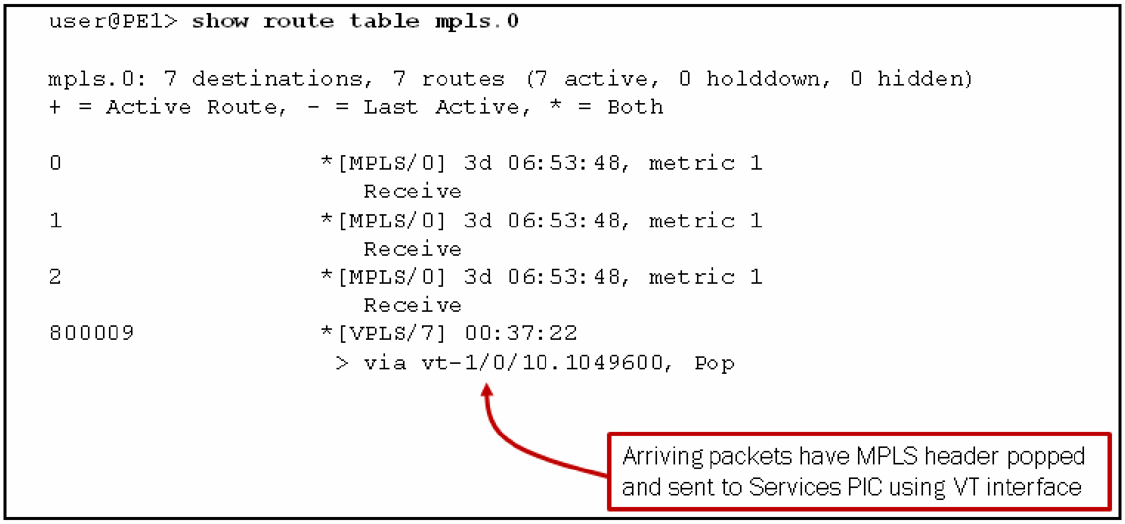


- View the BGP routes in the VPLS VRF

```
user@PE1> show route table vpn-a extensive

vpn-a.l2vpn.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)

192.168.2.2:1:2:1/96 (1 entry, 1 announced)
        *BGP    Preference: 170/-301
                Route Distinguisher: 192.168.2.2:1
                Next hop type: Indirect
                Next-hop reference count: 5
                Source: 192.168.2.2
                Protocol next hop: 192.168.2.2
                Indirect next hop: 2 no-forward
                State: <Secondary Active Int Ext>
                Local AS: 65512 Peer AS: 65512
                Age: 31:06      Metric2: 1
                Task: BGP_65512.192.168.2.2+179
                Announcement bits (1): 0-vpn-a-l2vpn
                AS path: I
                Communities: target:65512:100 Layer2-info: encaps:VPLS, control
flags:, mtu: 0, site preference: 300
                Import Accepted
                Label-base: 800000, range: 8
                Localpref: 300
                Router ID: 192.168.2.2
                Primary Routing Table bgp.l2vpn.0
                Indirect next hops: 1
```

This capture shows the contents the *vpn-name*`.l2vpn.0` table. This table displays all received label blocks from remote sites that have the correct target community attached.

MPLS Forwarding Table

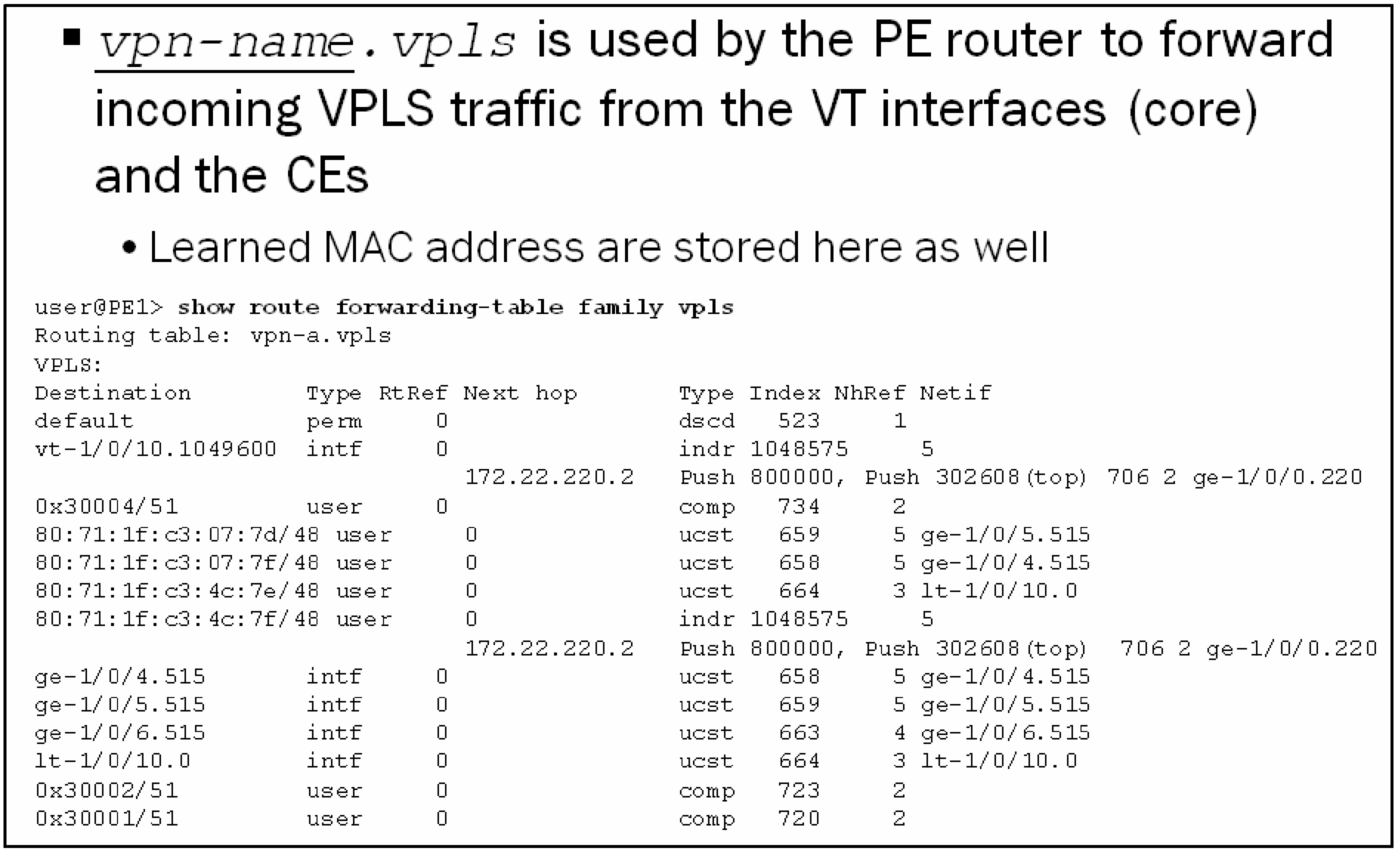


```
user@PE1> show route table mpls.0

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0                  *[MPLS/0] 3d 06:53:48, metric 1
                      Receive
1                  *[MPLS/0] 3d 06:53:48, metric 1
                      Receive
2                  *[MPLS/0] 3d 06:53:48, metric 1
                      Receive
800009             *[VPLS/7] 00:37:22
                    > via vt-1/0/10.1049600, Pop
```

Arriving packets have MPLS header popped and sent to Services PIC using VT interface

The capture displays the `mpls.0` table, which is the table used to forward packets that arrive from the provider's core. This graphic shows that packets that arrive from the provider encapsulated with an MPLS label of 800009 have the MPLS header popped and the resulting Ethernet frame forwarded to the VPN tunnel interface, `vt-1/3/10.1049600`.

VPLS Forwarding Table

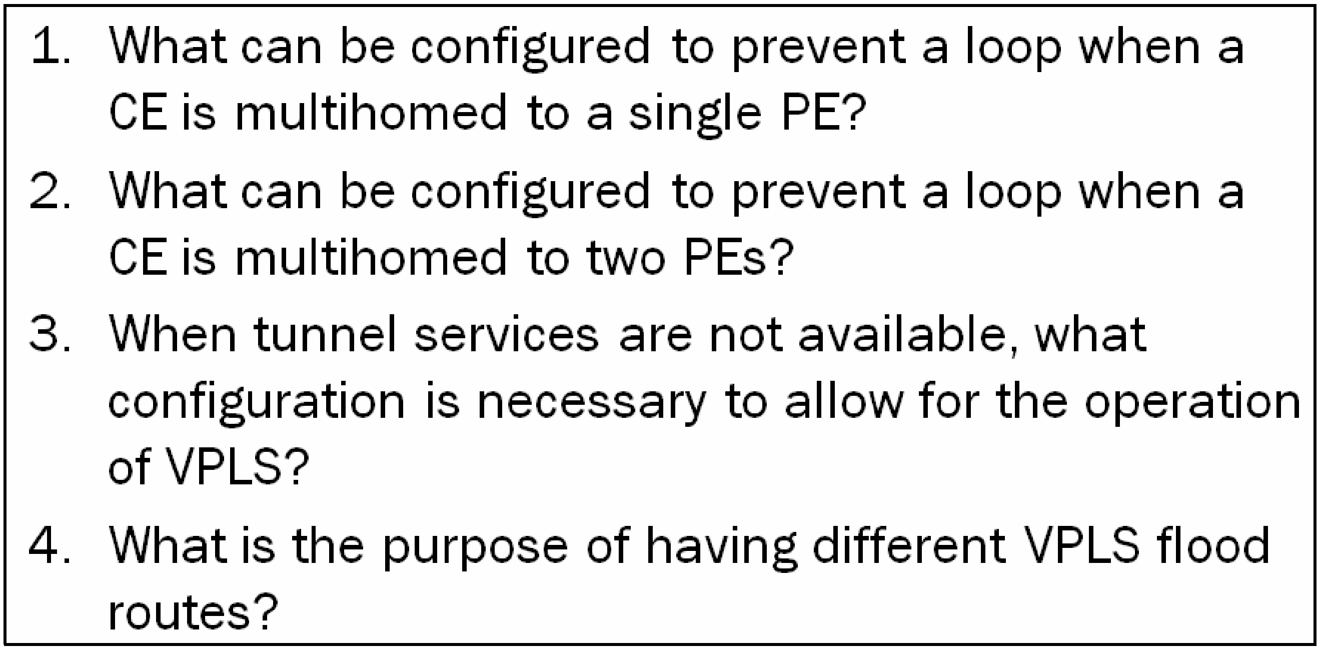


- *vpn-name*.vpls is used by the PE router to forward incoming VPLS traffic from the VT interfaces (core) and the CEs
    - Learned MAC address are stored here as well

```
user@PE1> show route forwarding-table family vpls
Routing table: vpn-a.vpls
VPLS:
Destination        Type RtRef Next hop        Type Index NhRef Netif
default            perm     0                 dscd   523     1
vt-1/0/10.1049600  intf     0                 indr 1048575     5
                              172.22.220.2    Push 800000, Push 302608(top)  706 2 ge-1/0/0.220
0x30004/51         user     0                 comp   734     2
80:71:1f:c3:07:7d/48 user     0               ucst   659     5 ge-1/0/5.515
80:71:1f:c3:07:7f/48 user     0               ucst   658     5 ge-1/0/4.515
80:71:1f:c3:4c:7e/48 user     0               ucst   664     3 lt-1/0/10.0
80:71:1f:c3:4c:7f/48 user     0               indr 1048575     5
                              172.22.220.2    Push 800000, Push 302608(top)   706 2 ge-1/0/0.220
ge-1/0/4.515       intf     0                 ucst   658     5 ge-1/0/4.515
ge-1/0/5.515       intf     0                 ucst   659     5 ge-1/0/5.515
ge-1/0/6.515       intf     0                 ucst   663     4 ge-1/0/6.515
lt-1/0/10.0        intf     0                 ucst   664     3 lt-1/0/10.0
0x30002/51         user     0                 comp   723     2
0x30001/51         user     0                 comp   720     2
```

The *vpn-name*`.vpls` table is essentially the packet forwarding engine's copy of the MAC table. All learned MAC addresses are placed in this table along with its next hop. A benefit of looking at this table is that it is possible to see the MPLS label stack that will be used when forwarded traffic to a particular MAC address.

Review Questions



1. What can be configured to prevent a loop when a CE is multihomed to a single PE?
2. What can be configured to prevent a loop when a CE is multihomed to two PEs?
3. When tunnel services are not available, what configuration is necessary to allow for the operation of VPLS?
4. What is the purpose of having different VPLS flood routes?

Answers to Review Questions

1.

To prevent a layer 2 loop when a CE is multihomed to a single PE, you must configure either primary/backup link, LAG, ERP, or a spanning tree protocol.

2.

To prevent a layer 2 loop when a CE is multihomed to multiple PEs, you must use BGP for signaling which automatically prevents a loop or when using LDP for VPLS signaling specify a neighbor and a backup neighbor.

3.

When tunnel services are not available (no Tunnel PIC) the command no-tunnel-services can be enabled to use LSI interfaces instead of vt interfaces.

4.

The flooding behavior on a PE is based upon the interface that BUM traffic arrives on. If BUM traffic arrives from a locally connect CE, then the traffic needs to be flooded to all local CEs (except the one the traffic came from) and to all remote PEs. If BUM traffic arrives from a remote PE, then the traffic needs to be flooded to only local CEs, not to any remote PE (because of PE full-mesh). If BUM traffic arrives from the RE, then the traffic needs to be flooded to all local CEs and to all remote PEs. There should be a flood route for each of the 3 scenarios.

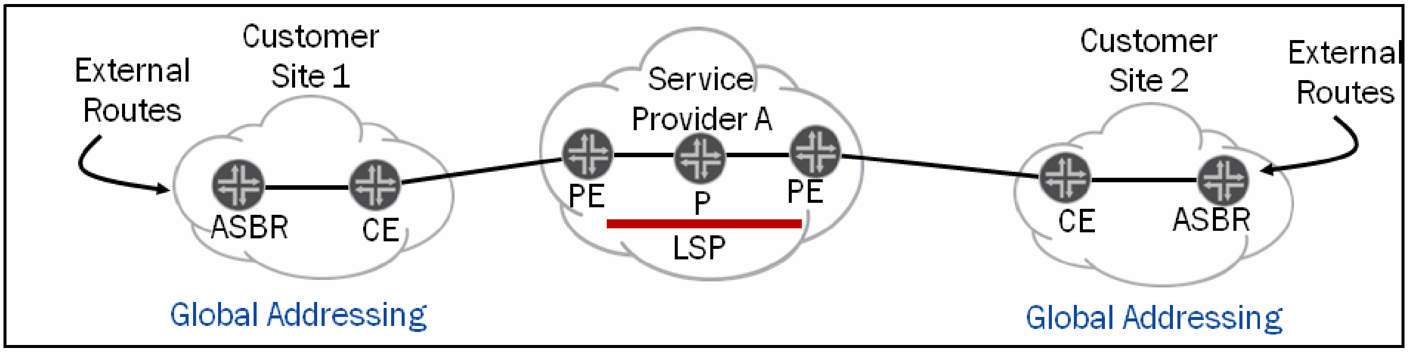
JNCIS-SP Study Guide—Part 3

Chapter 18: Interprovider VPNs

This Chapter Discusses:

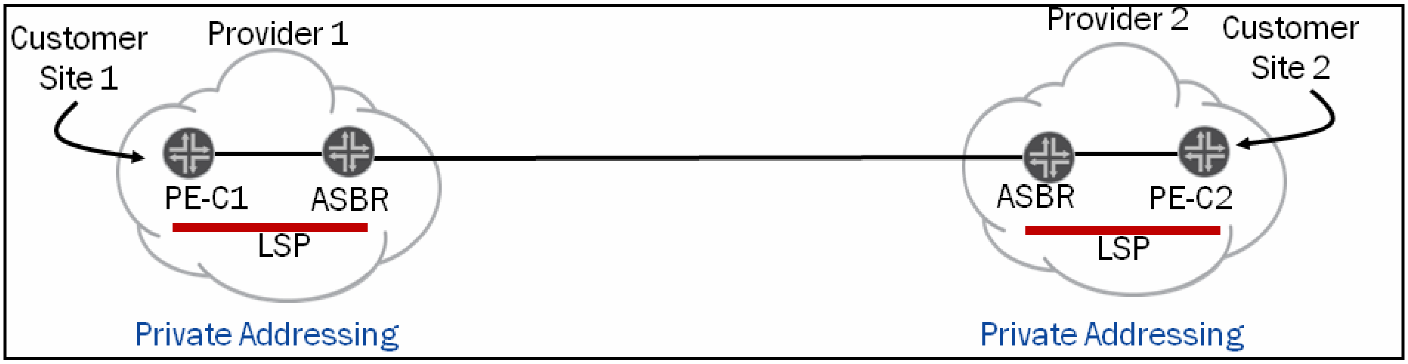
- Junos operating system support for carrier of carriers; and
- Junos support for interprovider virtual private networks (VPNs).

Carrier-of-Carriers Model



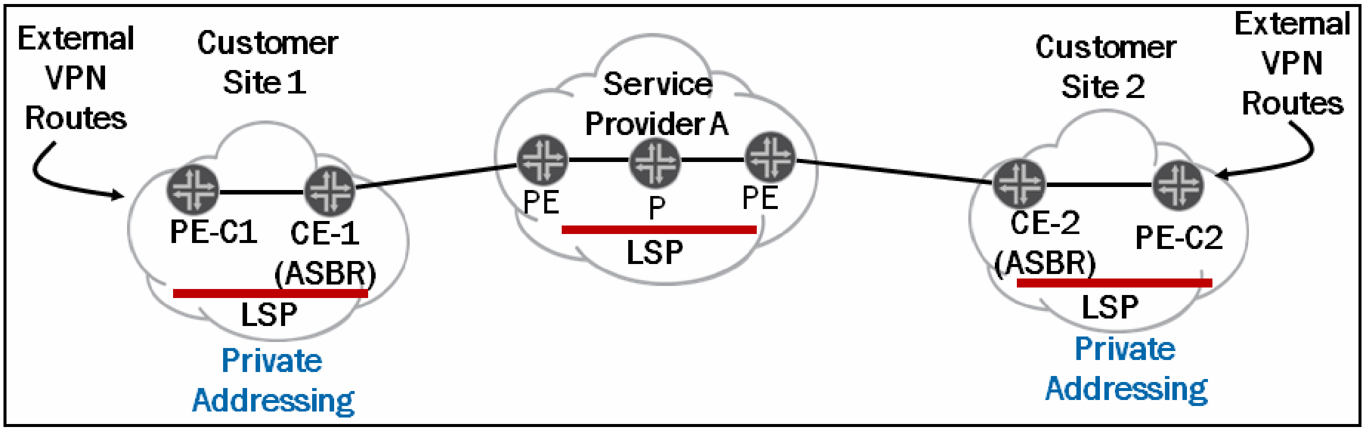
This model allows service provider A to offer a backbone service to the customer, another service provider. Assume the customer is a new service provider that has a point of presence (POP) in a few sparse locations with no backbone network to interconnect those POPs. The customer (the new service provider) can purchase the carrier of carrier service from the service provider A to interconnect its sites making the customer network appear as a single autonomous system (AS) without service provider A having to carry the external routes learned by the customer. The details of this model are discussed in the subsequent sections.

Interprovider VPN Model



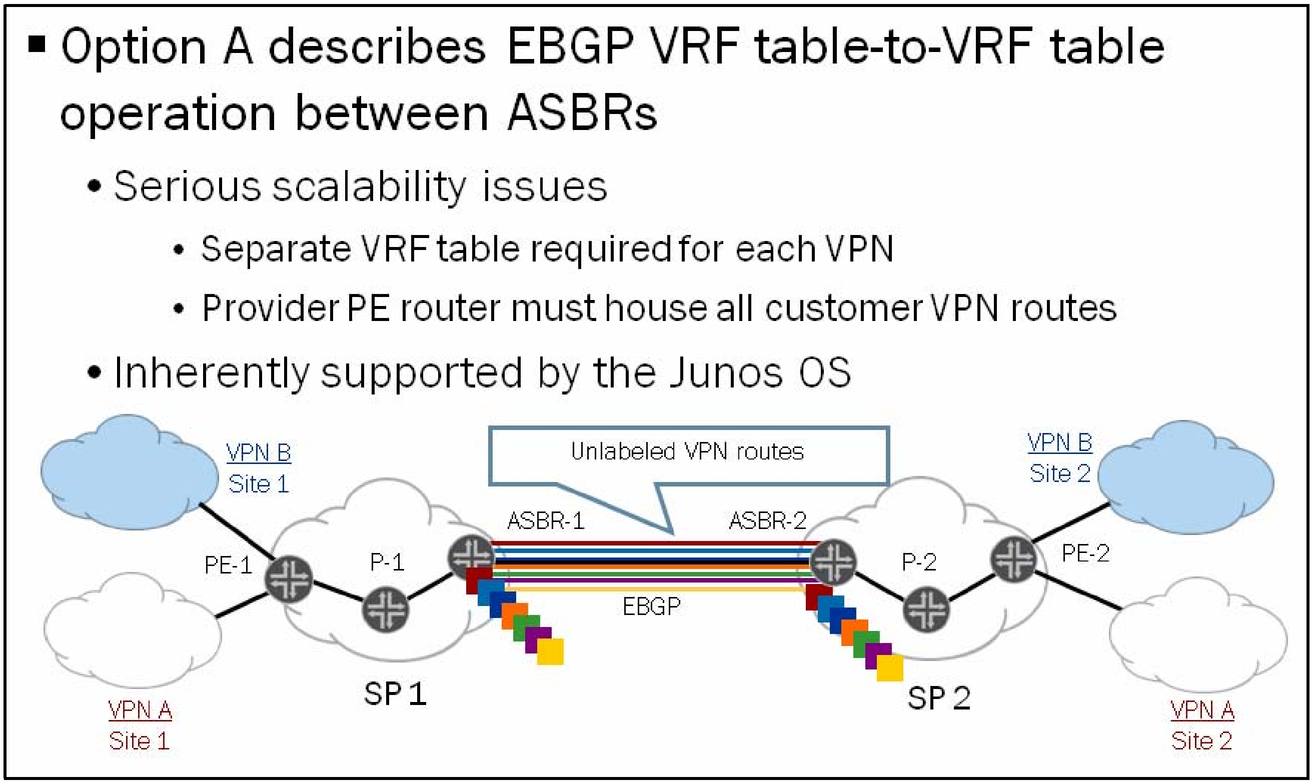
This model allows for a Layer 3 VPN, BGP Layer 2 VPN, or a BGP virtual private LAN service (VPLS) to extend between autonomous system or service providers.

Carrier-of-Carriers VPN Support



This model is a combination of the two models discussed on the previous section. In this model, the customers of service provider A will be providing VPN service to its own customers. The details of this model are described in subsequent sections.

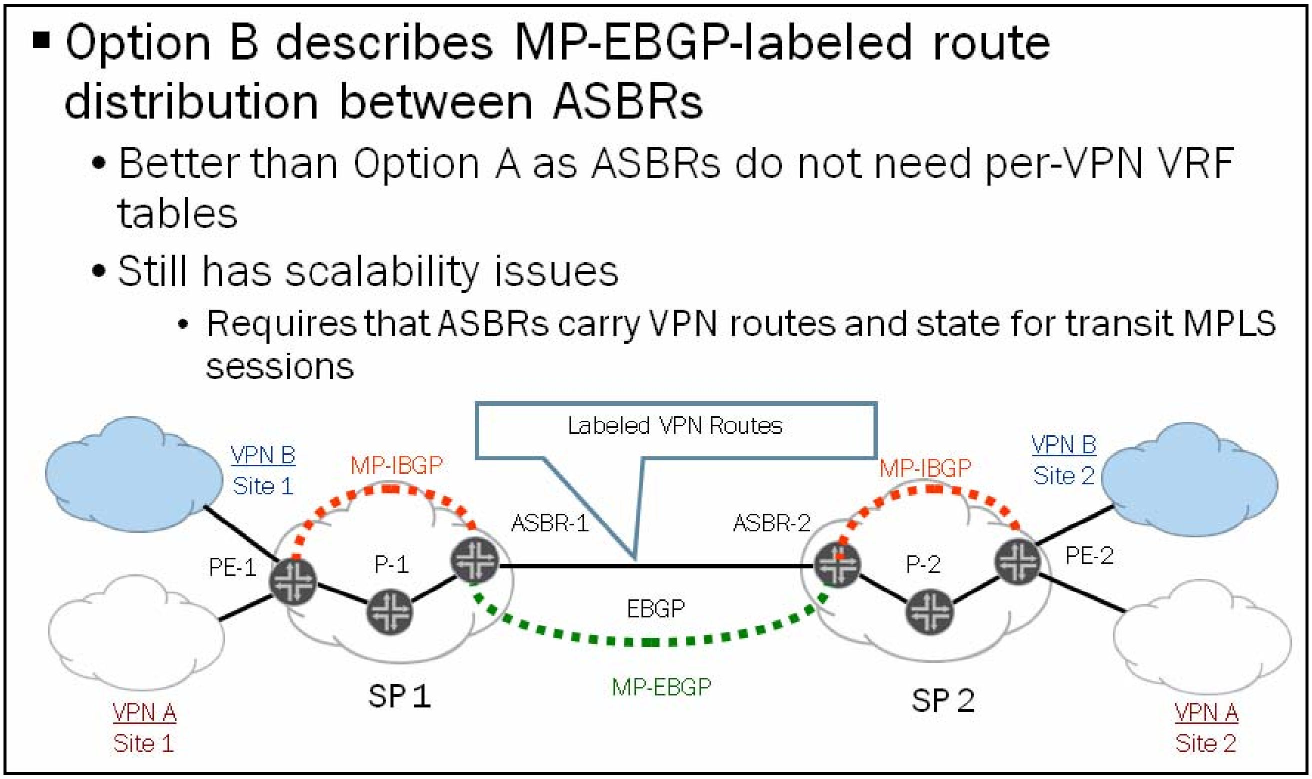
Option A



- Option A describes EBGP VRF table-to-VRF table operation between ASBRs
  - Serious scalability issues
    - Separate VRF table required for each VPN
    - Provider PE router must house all customer VPN routes
  - Inherently supported by the Junos OS

RFC 4364 describes three methods of providing multiple AS backbones. Option A is the least scalable of the options. This option requires that the autonomous system boundary routers (ASBRs) maintain separate VPN routing and forwarding tables (VRFs) and store all of the associated routes for every one of its customers. Although this option is supported by the Junos OS, it is not a recommended solution.

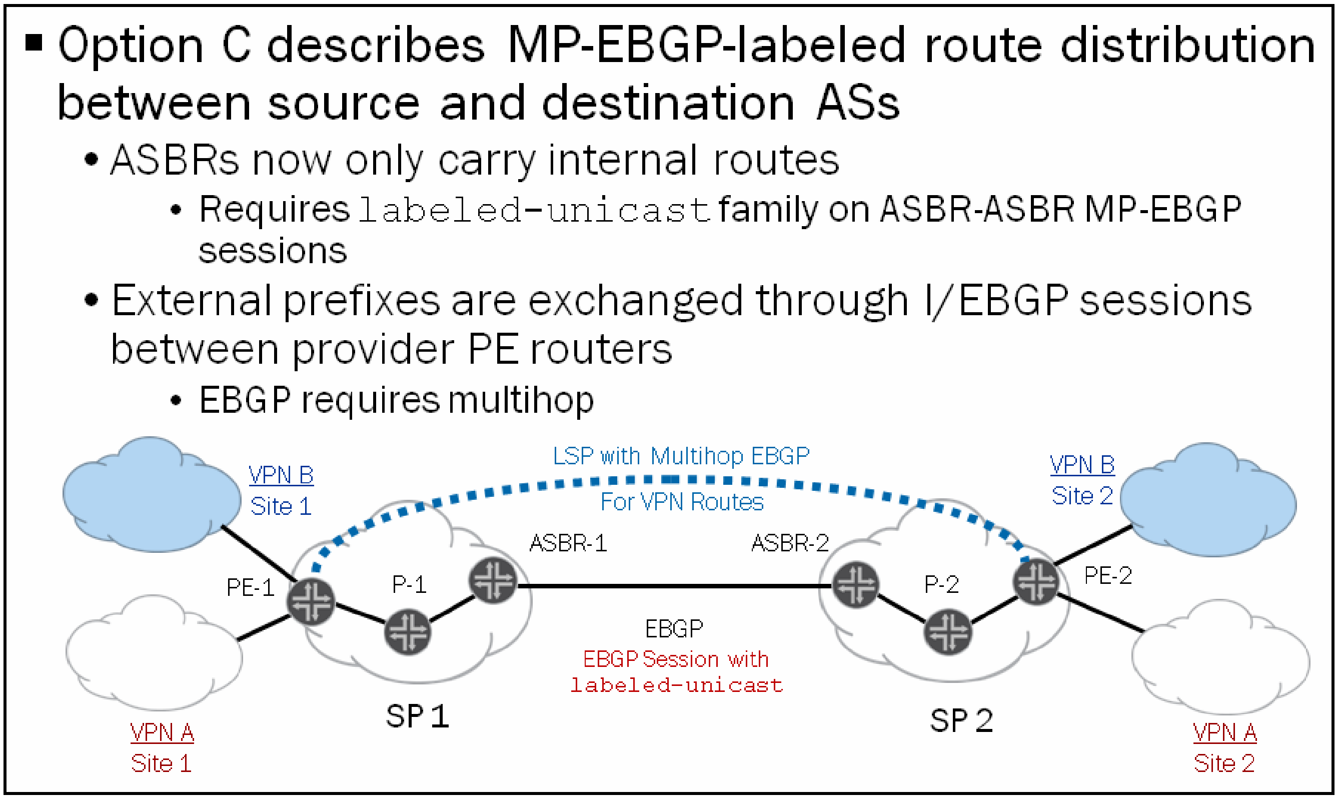
Option B



- Option B describes MP-EBGP-labeled route distribution between ASBRs
  - Better than Option A as ASBRs do not need per-VPN VRF tables
  - Still has scalability issues
    - Requires that ASBRs carry VPN routes and state for transit MPLS sessions

With option B, the ASBRs does not need to maintain separate VRF instances for each VPN. However, the ASBR will still have to keep VPN routes in a single routing table, bgp.l3vpn.0 for L3VPN routes. Through an EBGP session between one another, the ASBRs will then exchange VPN routes as label routes. The EBGP advertised labels are used stitch together the label-switched paths (LSPs) that terminate between provider edge (PE) and ASBR.

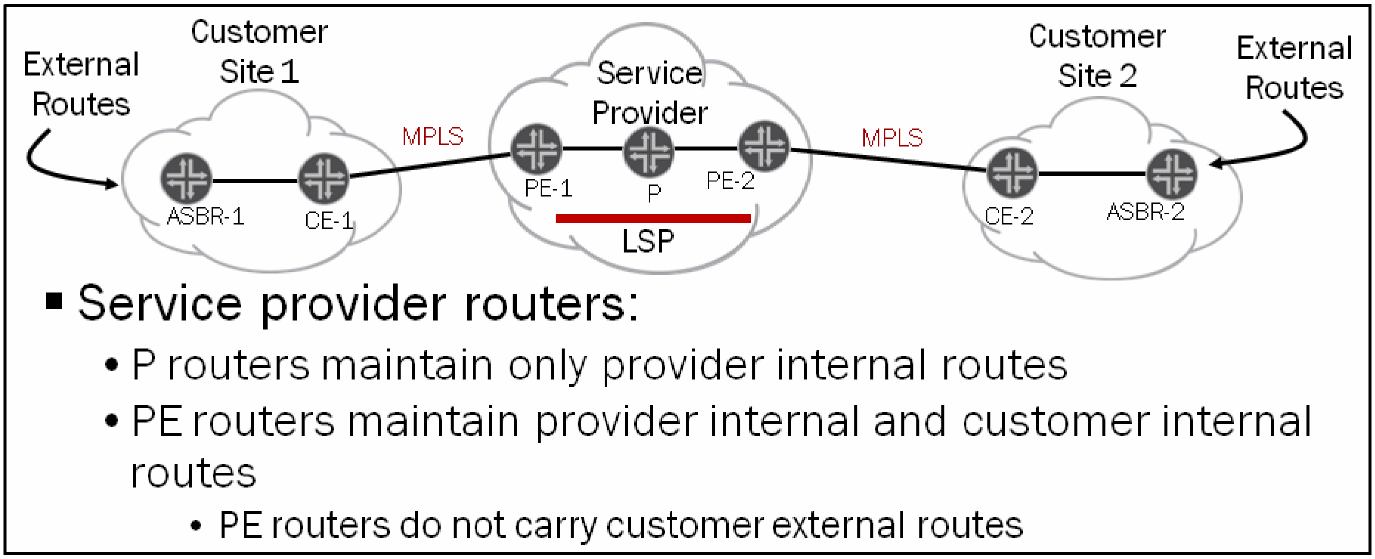
Option C



- Option C describes MP-EBGP-labeled route distribution between source and destination ASs
  - ASBRs now only carry internal routes
    - Requires `labeled-unicast` family on ASBR-ASBR MP-EBGP sessions
  - External prefixes are exchanged through I/EBGP sessions between provider PE routers
    - EBGP requires multihop

This option is generally accepted as the most scalable solution for interprovider VPNs. This option allows the PE routers in different autonomous systems to exchange VPN routes (Layer 3 VPN, BGP Layer 2 VPN, or BGP VPLS) using a multihop BGP session. The ASBRs do not need to store any VPN routes in this case. Instead, the ASBRs will exchange the internal networks of each service provider (most importantly the loopback addresses of the PEs) using labeled IP version 4 (IPv4) routes. The labels associated with the internal networks will be used to stitch together the MPLS LSPs that exist between PE and ASBR in the service provider networks.

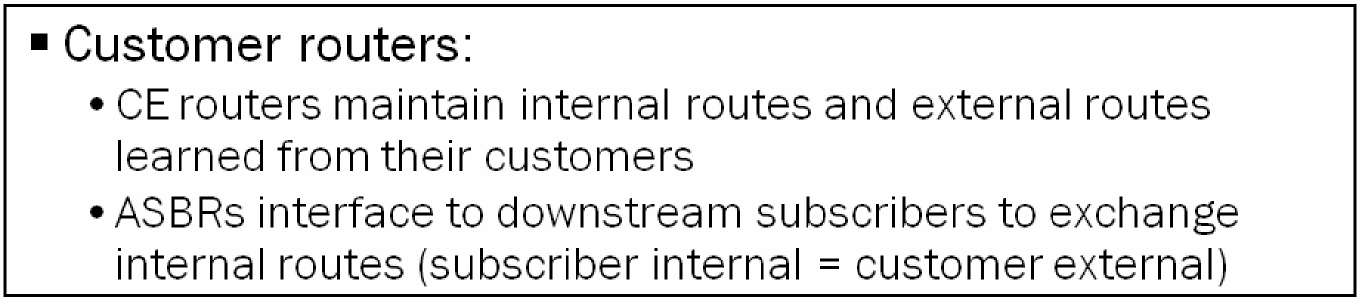
Service Provider Routers



- Service provider routers:
  - P routers maintain only provider internal routes
  - PE routers maintain provider internal and customer internal routes
    - PE routers do not carry customer external routes

The service provider's P routers only maintain routes internal to the provider's network. The PE routers maintain both provider internal routes and customer internal routes. Customer-specific VRF tables on the PE routers house the customer's internal routes. These routes normally consist of at least the customer's /32 loopback addresses. The provider's PE routers do not carry the customer's external routes, which is critical to the overall scalability of this model.

Customer Routers



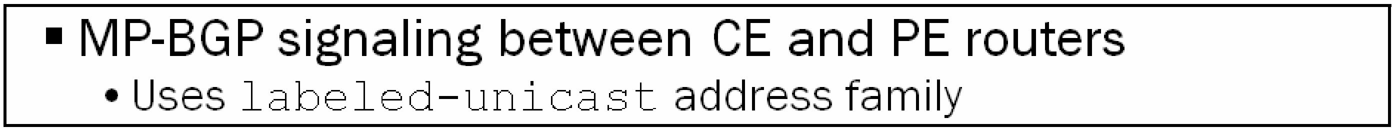
- Customer routers:
  - CE routers maintain internal routes and external routes learned from their customers
  - ASBRs interface to downstream subscribers to exchange internal routes (subscriber internal = customer external)

The customer's routers must maintain both customer internal and external routes. The customer's external routes are those learned from the customer's downstream subscribers.

LSP Signaling Needed in Service Provider Network

Because the provider's network uses MPLS forwarding, an LSP must be established between provider PE routers. This LSP can be established with RSVP or LDP signaling. In this example, the LSP is established using RSVP; PE-1 is assigned MPLS label 30 by the P router.

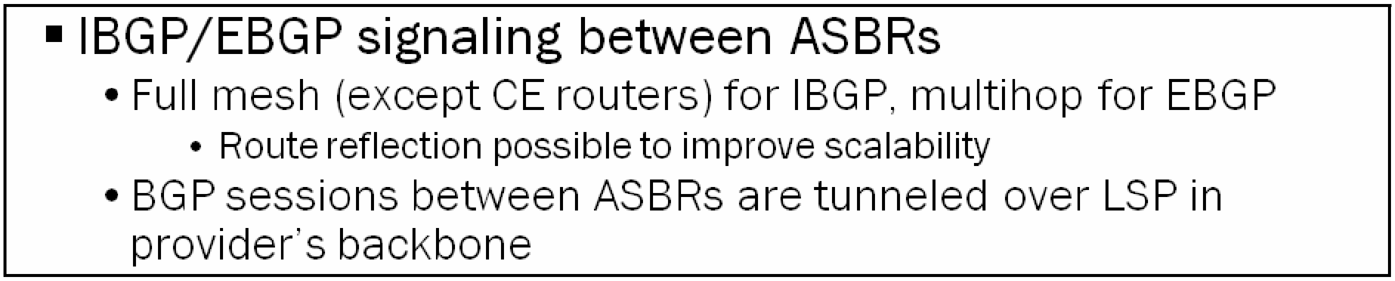
MP-BGP Signaling Between PE and CE Routers



- MP-BGP signaling between CE and PE routers
  - Uses `labeled-unicast` address family

The customer edge (CE) routers use EBGP with labeled-unicast network layer reachability information (NLRI) to exchange labeled routes with the provider's PE routers. The use of labeled routes allows the provider to extend its LSPs to the customer CE router, which thereby eliminates the need to carry customer internal routes in its P routers. While the customer's network does not require MPLS signaling, the CE router must support the family MPLS on its PE-facing interface, because it must send labeled packets.

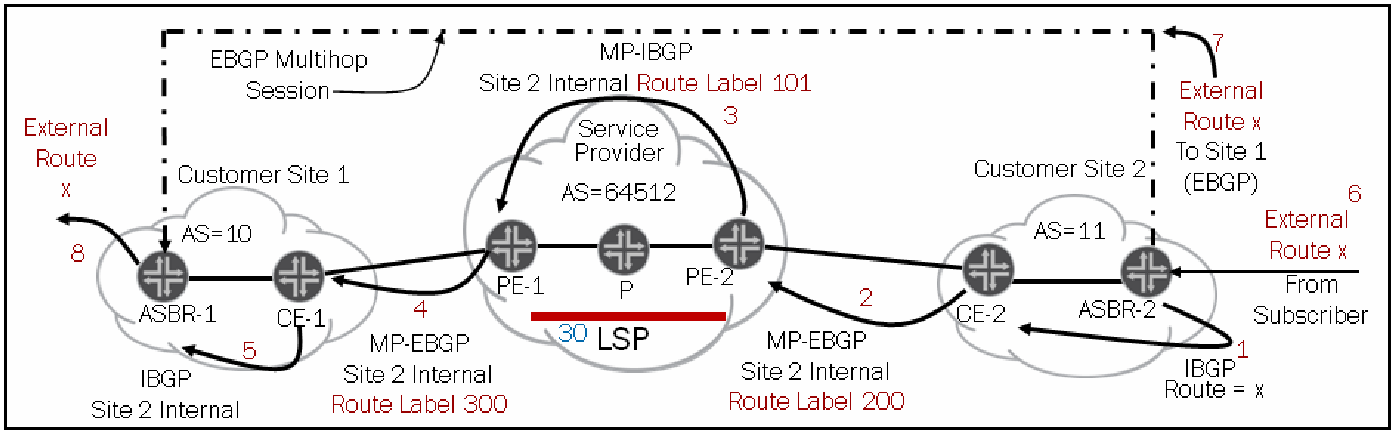
IBGP/EBGP Signaling Between Customer ASBRs



- IBGP/EBGP signaling between ASBRs
  - Full mesh (except CE routers) for IBGP, multihop for EBGP
    - Route reflection possible to improve scalability
  - BGP sessions between ASBRs are tunneled over LSP in provider's backbone

Once the customer's internal routes are exchanged across the provider's backbone, the ASBRs can establish internal BGP (IBGP) (same AS numbers) sessions or multihop EBGP (different AS numbers) sessions through the provider's backbone for the purposes of exchanging external routes. A full IBGP mesh is needed between routers at the customer sites when using IBGP, except for the CE routers, which peer indirectly using the provider's backbone. Because this example demonstrates the use of EBGP, only the peering session between ASBR-2 and CE-1 is needed. The second BGP session between the two ASBRs (shown as a dotted line) is only required for IBGP peering when the customer sites share the same AS number.

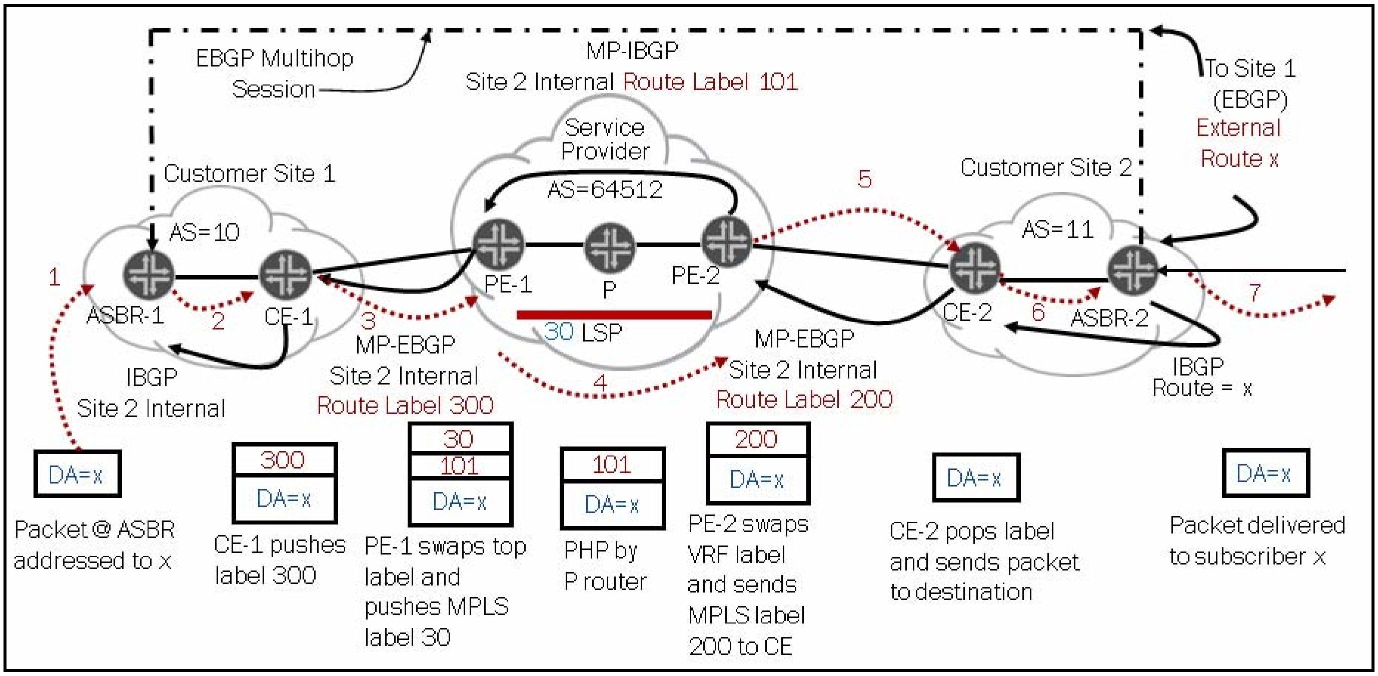
Signaling: Step by Step



The details of the signaling exchanges shown on the graphic are:

1. The IGP at customer Site 2 exchanges internal reachability with CE-2. ASBR-2 establishes an IBGP neighbor relationship with CE-2.
2. CE-2 selectively advertises Site 2's internal routes to the provider's PE-2 router using multiprotocol EBGP (MP-EBGP) with support of `labeled-unicast` routes. These routes are advertised with a valid label, which is 200 in this example.
3. PE-2 houses Site 2's internal routes in a VRF table and uses MP-IBGP to send labeled VPN-IPv4 routes to PE-1. The route to ASBR-2 is assigned Label 101 in this example.
4. PE-1 uses MP-EBGP to send Site 2's internal routes to CE-1. PE-1 changes the BGP next hop. Therefore, it must assign a new label to the prefix advertised (Label 300 in this example).
5. After receiving the labeled route, CE-1 distributes Site 2's internal routes to ASBR-1 using IBGP. No labels are needed, because conventional IP forwarding is used within the customer sites. At this point, the ASBRs can establish an EBGP multihop session through the provider's backbone. This session is tunneled through the LSP in the provider's network.
6. ASBR-2 learns an external route x from one of its subscribers. IBGP conveys external routes from ASBR-2 to CE-2. PE-1, PE-2, and P routers never become aware of the external route advertisement x.
7. The external route x is now advertised to ASBR-1 using the EBGP session established at Step 5. No labels are associated with this route due to the lack of MPLS forwarding in the customer networks.
8. External route x is advertised by ASBR-1 to its downstream subscribers as well as to CE-1.

Carrier-of-Carriers Data Forwarding



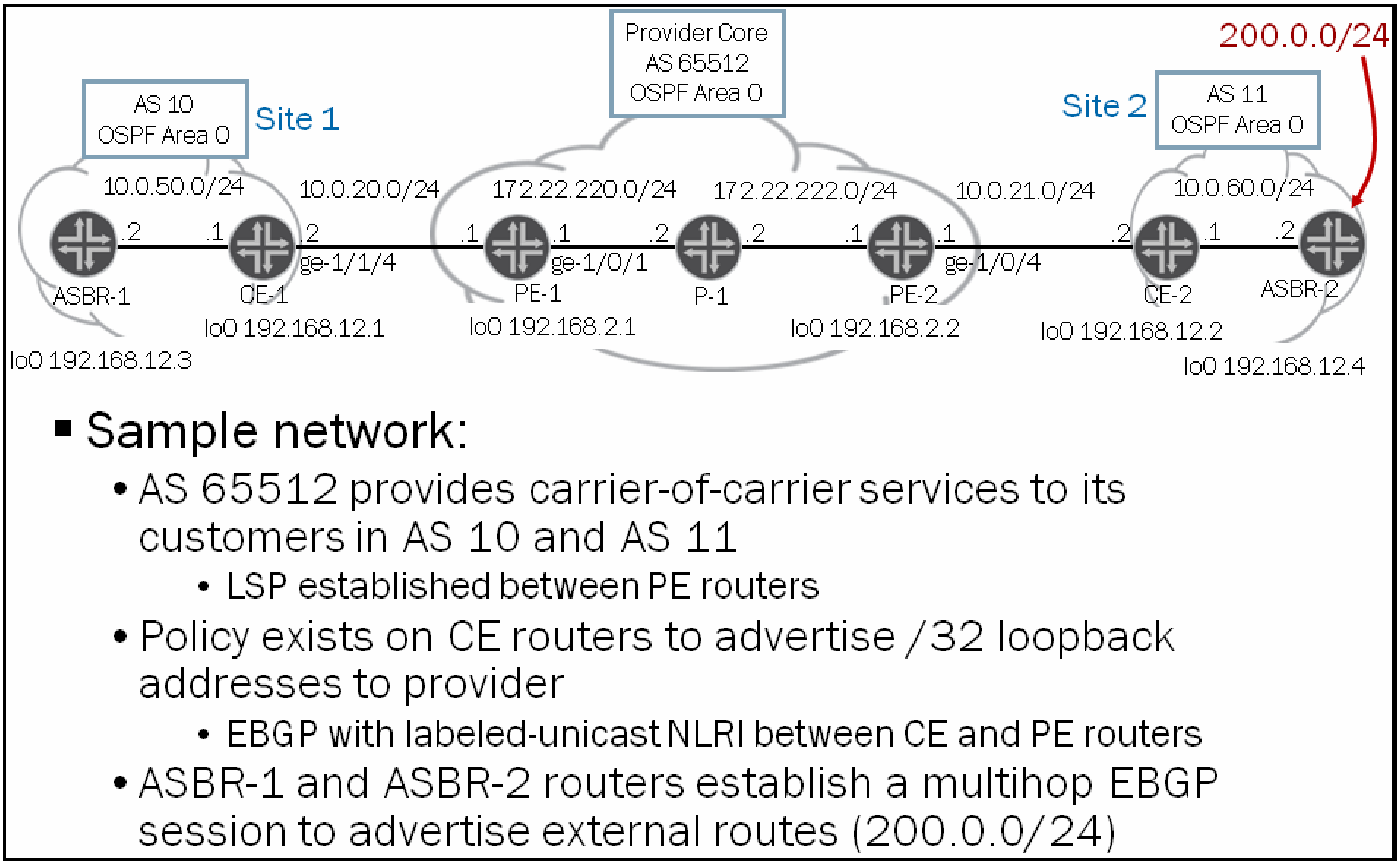
This graphic uses step numbers to describe the forwarding operations between ASBR-1 and ASBR-2. The result is the need for a two-level label stack in the provider's network.

Forwarding: Step by Step

The details of the forwarding operation shown on the preceding graphic are:

1. A packet addressed to external route x arrives at ASBR-1.
2. ASBR-1 forwards this unlabeled packet towards CE-1 using the IGP's shortest path.
3. CE-1 pushes Label 300 onto the packet and forwards it to PE-1.
4. PE-1 swaps the top label with the value received from PE-2, and pushes an MPLS label (30 in this example) onto the stack. The P router pops this top label (PHP) such that PE-2 receives a packet with a single label.
5. PE-2 swaps the VRF label with the label advertised by CE-2 and forwards the packet out the VRF interface to CE-2.
6. CE-2 pops the MPLS label and routes the native packet using Site 2's interior gateway protocol (IGP).
7. ASBR-2 performs a longest-match lookup and routes the packet towards destination *x*.

Carrier-of-Carriers Sample Network

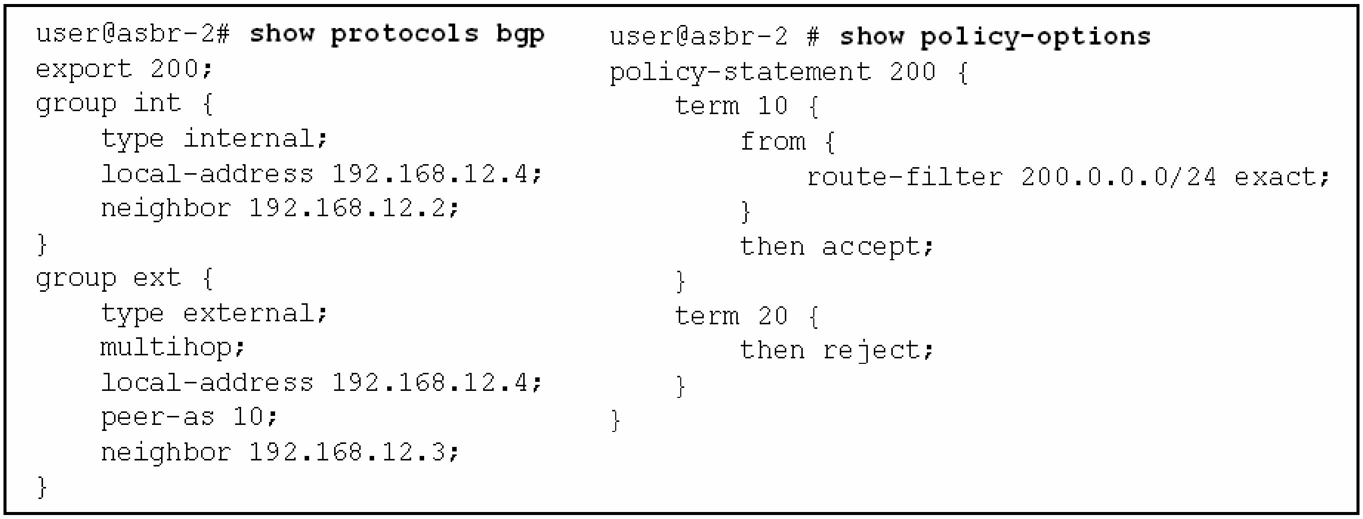


- Sample network:
  - AS 65512 provides carrier-of-carrier services to its customers in AS 10 and AS 11
    - LSP established between PE routers
  - Policy exists on CE routers to advertise /32 loopback addresses to provider
    - EBGP with labeled-unicast NLRI between CE and PE routers
  - ASBR-1 and ASBR-2 routers establish a multihop EBGP session to advertise external routes (200.0.0/24)

This graphic provides a sample network; the following list provides the details of this network. The following sections show various configuration-mode and operational-mode screen captures relating to this network.

- *Provider network*: The provider’s network is assigned AS 65512 and has already established an LSP between PE-1 and PE-2 using RSVP. The PE routers have a VRF table configured, along with the necessary VRF target community and route distinguishers.
- *Policy on CE routers*: The CE routers are configured to run MP-EBGP with the PE routers and have a policy in place to ensure that only internal prefixes are advertised to the PE routers.
- *ASBR-1 and ASBR-2 routers exchange external routes*: A multihop EBGP session is configured between the ASBRs because the customer networks are assigned differing AS numbers. ASBR-2 advertises the external route 200.0.0/24 to ASBR-1 using this EBGP session.

ASBR-2 Configuration



```
user@asbr-2# show protocols bgp
export 200;
group int {
    type internal;
    local-address 192.168.12.4;
    neighbor 192.168.12.2;
}
group ext {
    type external;
    multihop;
    local-address 192.168.12.4;
    peer-as 10;
    neighbor 192.168.12.3;
}
```

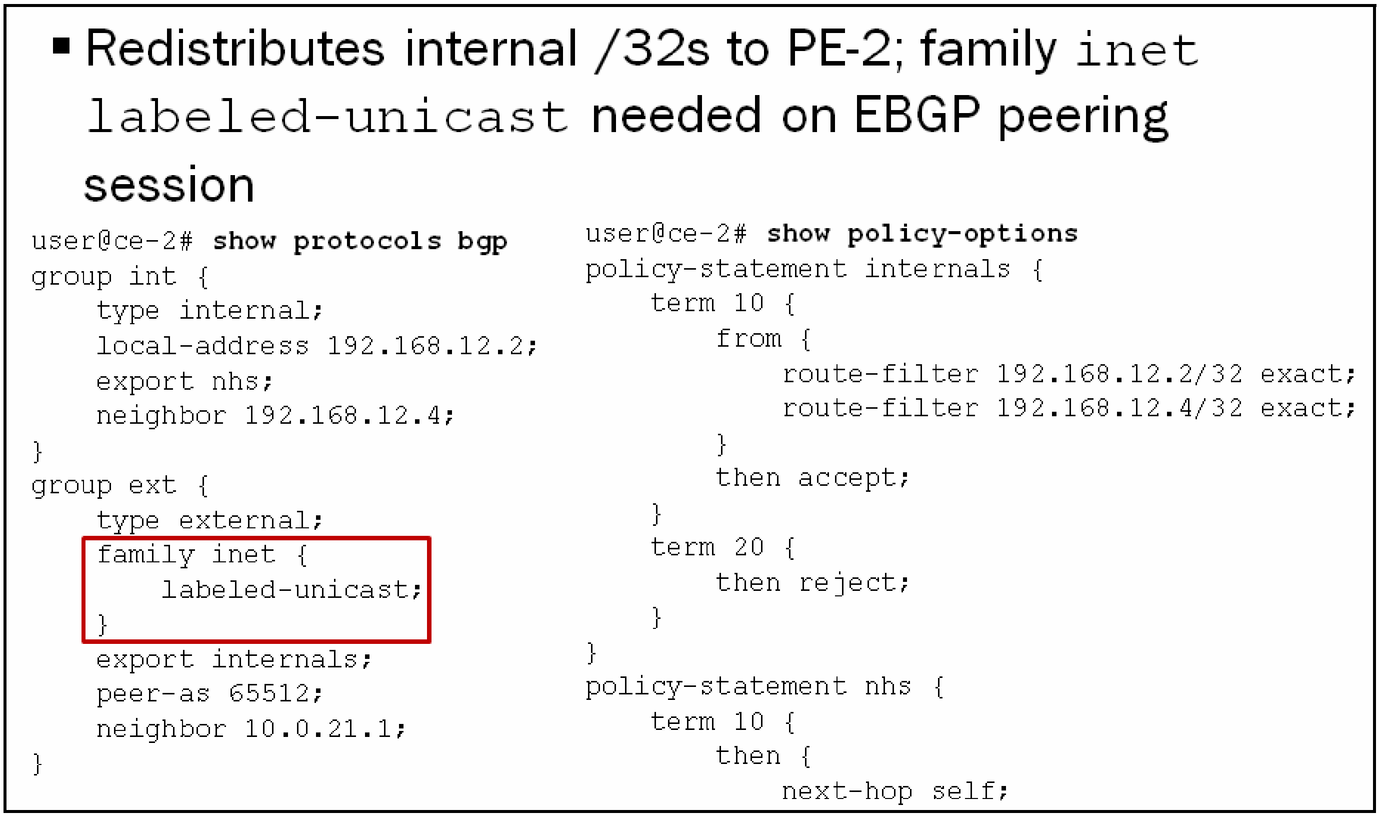
```
user@asbr-2 # show policy-options
policy-statement 200 {
    term 10 {
        from {
            route-filter 200.0.0.0/24 exact;
        }
        then accept;
    }
    term 20 {
        then reject;
    }
}
```

This graphic lists the key aspects of ASBR-2's configuration. An IBGP session is configured to CE-2, and a multihop EBGP session is configured for ASBR-1 at Site 1.

The `200` policy in ASBR-2 ensures that only external routes (200.0.0/24 in this example) are sent to ASBR-1. The default IBGP policy causes all external routes ASBR-2 learns through EBGP to be sent to CE-2.

This policy is rather simple and requires changes for each new external route. A more scalable solution involves an AS path regex that blocks all internal routes and only accepts routes whose AS-path attribute does not begin with 11.

CE-2 Configuration



- Redistributes internal /32s to PE-2; family `inet labeled-unicast` needed on EBGP peering session

```
user@ce-2# show protocols bgp
group int {
    type internal;
    local-address 192.168.12.2;
    export nhs;
    neighbor 192.168.12.4;
}
group ext {
    type external;
    family inet {
        labeled-unicast;
    }
    export internals;
    peer-as 65512;
    neighbor 10.0.21.1;
}
```

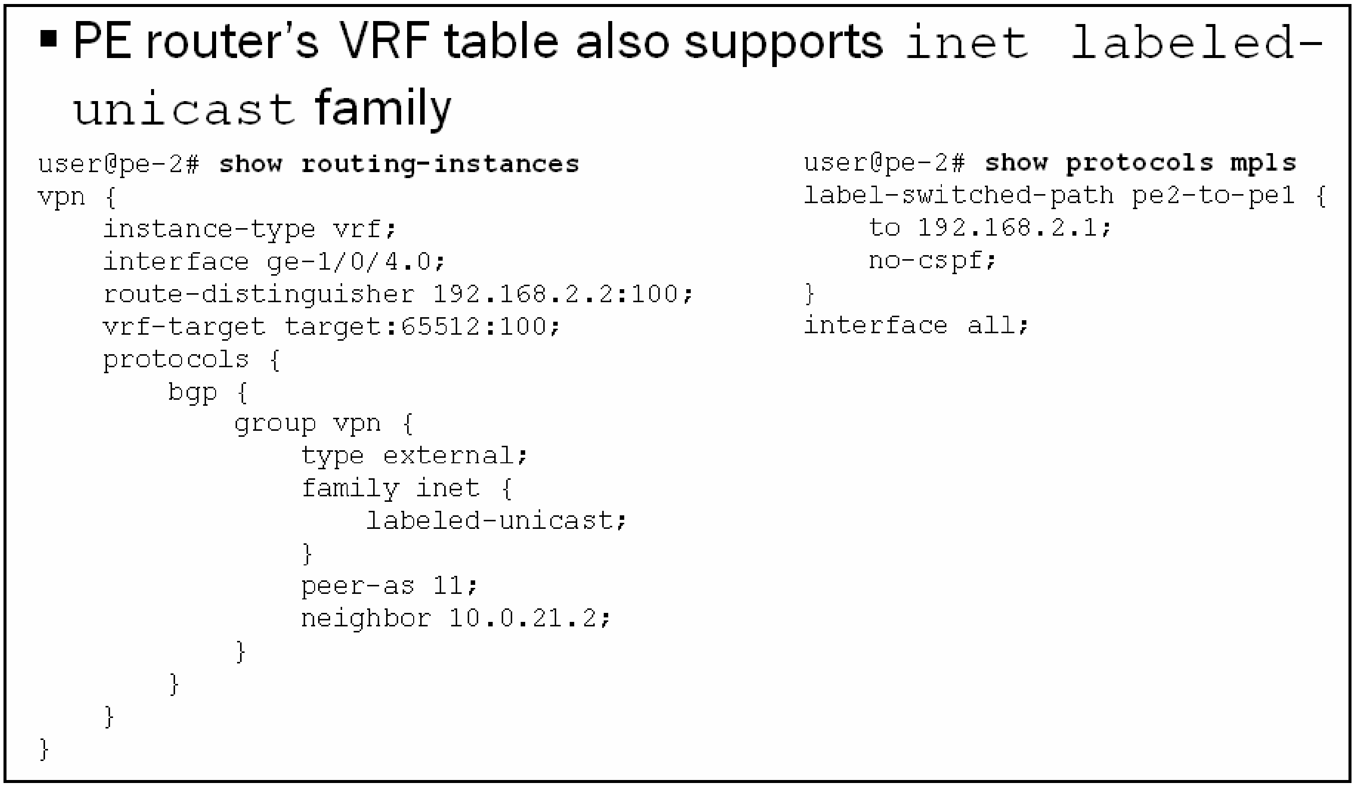
```
user@ce-2# show policy-options
policy-statement internals {
    term 10 {
        from {
            route-filter 192.168.12.2/32 exact;
            route-filter 192.168.12.4/32 exact;
        }
        then accept;
    }
    term 20 {
        then reject;
    }
}
policy-statement nhs {
    term 10 {
        then {
            next-hop self;
```

This graphic lists the key aspects of CE-2's configuration. An IBGP session is configured to ASBR-2, and an MP-EBGP session is configured for communications with PE-2.

The MP-EBGP session has the `labeled-unicast` family configured, which is required for the exchange of labeled routes between CE and PE routers.

CE-2 has an EBGP export policy in place that causes it to only advertise the /32 routes associated with Site 2's loopback addresses. The leaking of other internal routes (that is, OSPF and direct connect) are not strictly required but can aid in troubleshooting. With this configuration, we must take care to source pings and traceroutes for the loopback addresses of customer site routers.

PE-2 Configuration



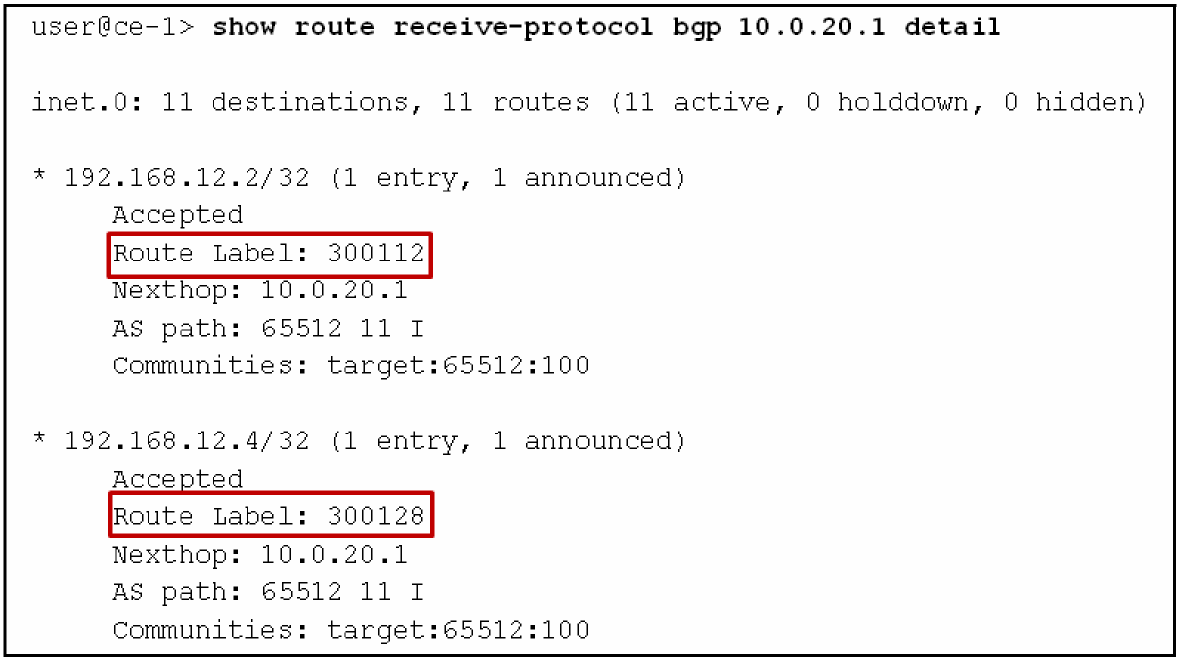
- PE router's VRF table also supports `inet labeled-unicast` family

```
user@pe-2# show routing-instances
vpn {
    instance-type vrf;
    interface ge-1/0/4.0;
    route-distinguisher 192.168.2.2:100;
    vrf-target target:65512:100;
    protocols {
        bgp {
            group vpn {
                type external;
                family inet {
                    labeled-unicast;
                }
                peer-as 11;
                neighbor 10.0.21.2;
            }
        }
    }
}
```

```
user@pe-2# show protocols mpls
label-switched-path pe2-to-pe1 {
    to 192.168.2.1;
    no-cspf;
}
interface all;
```

This graphic lists the key aspects of PE-2's configuration. An MP-EBGP VRF routing instance is configured for communications with CE-2. Also shown is an LSP that terminates on PE-1.

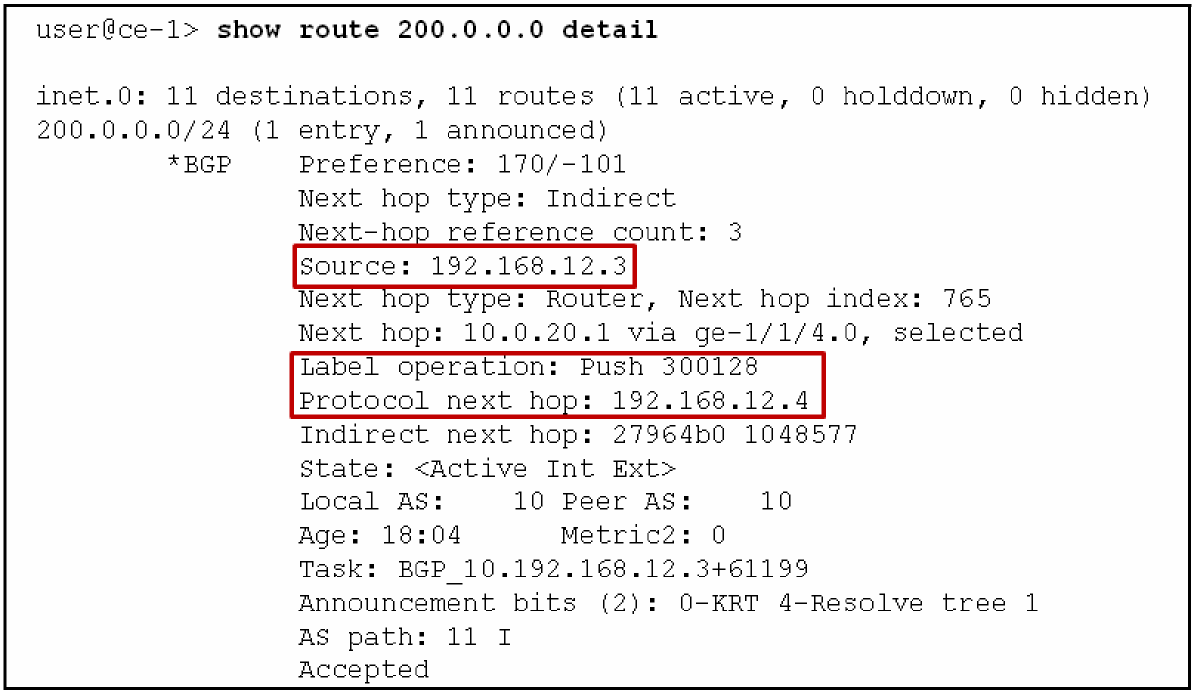
Carrier-of-Carriers Operation: CE-1



```
user@ce-1> show route receive-protocol bgp 10.0.20.1 detail

inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)

* 192.168.12.2/32 (1 entry, 1 announced)
     Accepted
     Route Label: 300112
     Nexthop: 10.0.20.1
     AS path: 65512 11 I
     Communities: target:65512:100

* 192.168.12.4/32 (1 entry, 1 announced)
     Accepted
     Route Label: 300128
     Nexthop: 10.0.20.1
     AS path: 65512 11 I
     Communities: target:65512:100
```

This graphic shows that CE-1 is receiving the internal routes from Site 2 through its Multiprotocol Border Gateway Protocol (MP-BGP) session to PE-1. These routes are labeled due to the provisioning of family `labeled-unicast` on the MP-EBGP session.

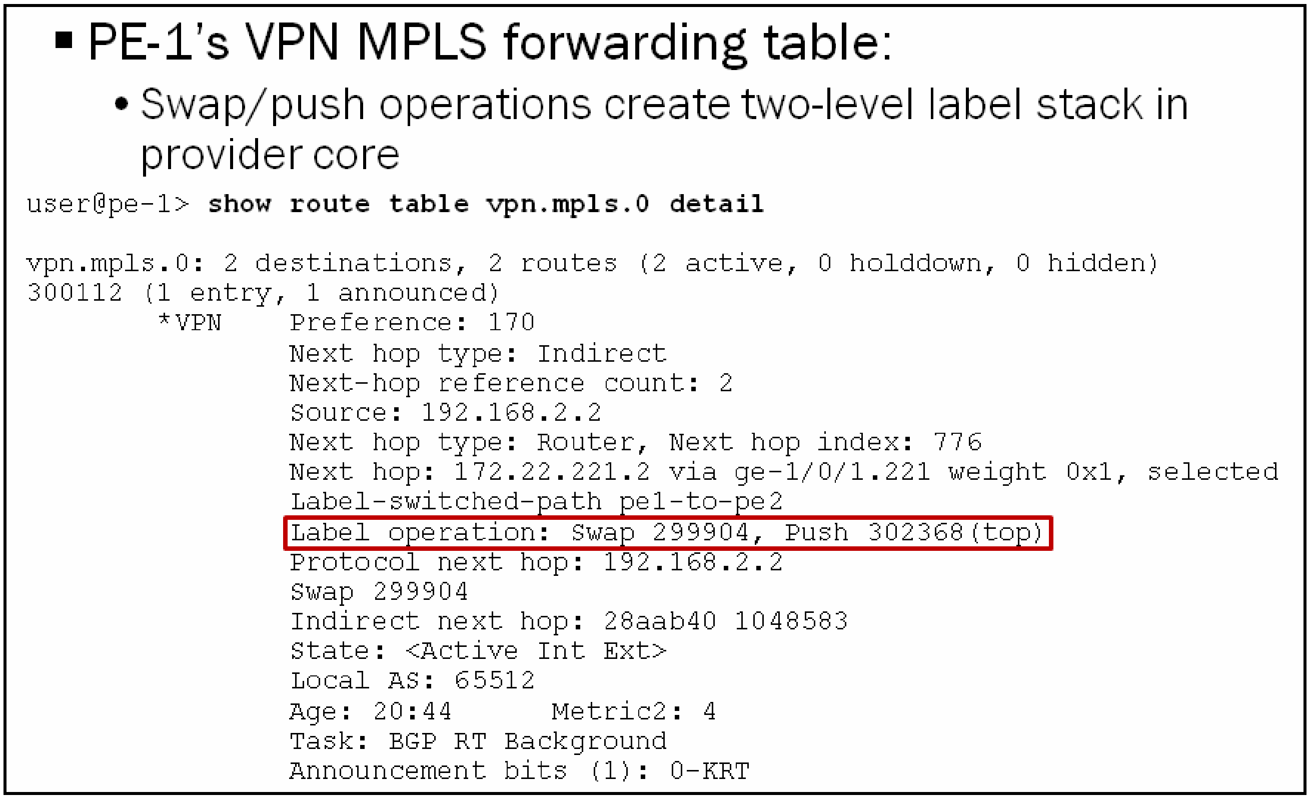
Carrier-of-Carriers Operation: CE-1



```
user@ce-1> show route 200.0.0.0 detail

inet.0: 11 destinations, 11 routes (11 active, 0 holddown, 0 hidden)
200.0.0.0/24 (1 entry, 1 announced)
        *BGP    Preference: 170/-101
                Next hop type: Indirect
                Next-hop reference count: 3
                Source: 192.168.12.3
                Next hop type: Router, Next hop index: 765
                Next hop: 10.0.20.1 via ge-1/1/4.0, selected
                Label operation: Push 300128
                Protocol next hop: 192.168.12.4
                Indirect next hop: 27964b0 1048577
                State: <Active Int Ext>
                Local AS:    10 Peer AS:    10
                Age: 18:04      Metric2: 0
                Task: BGP_10.192.168.12.3+61199
                Announcement bits (2): 0-KRT 4-Resolve tree 1
                AS path: 11 I
                Accepted
```

This graphic shows that CE-1 learns about the external prefix 200.0.0/24 from ASBR-1 through its IBGP peering session. Even though the route is learned from ASBR-1, the next hop is ASBR-2 (192.168.12.4). The BGP next hop is associated with a label and push operation. Thus, CE-1 routes packets addressed to 200.0.0/24 by pushing label 300128 and forwarding the labeled packet to PE-1 (10.0.20.1) for ultimate delivery to ASBR-2.

Carrier-of-Carriers Operation: PE-1



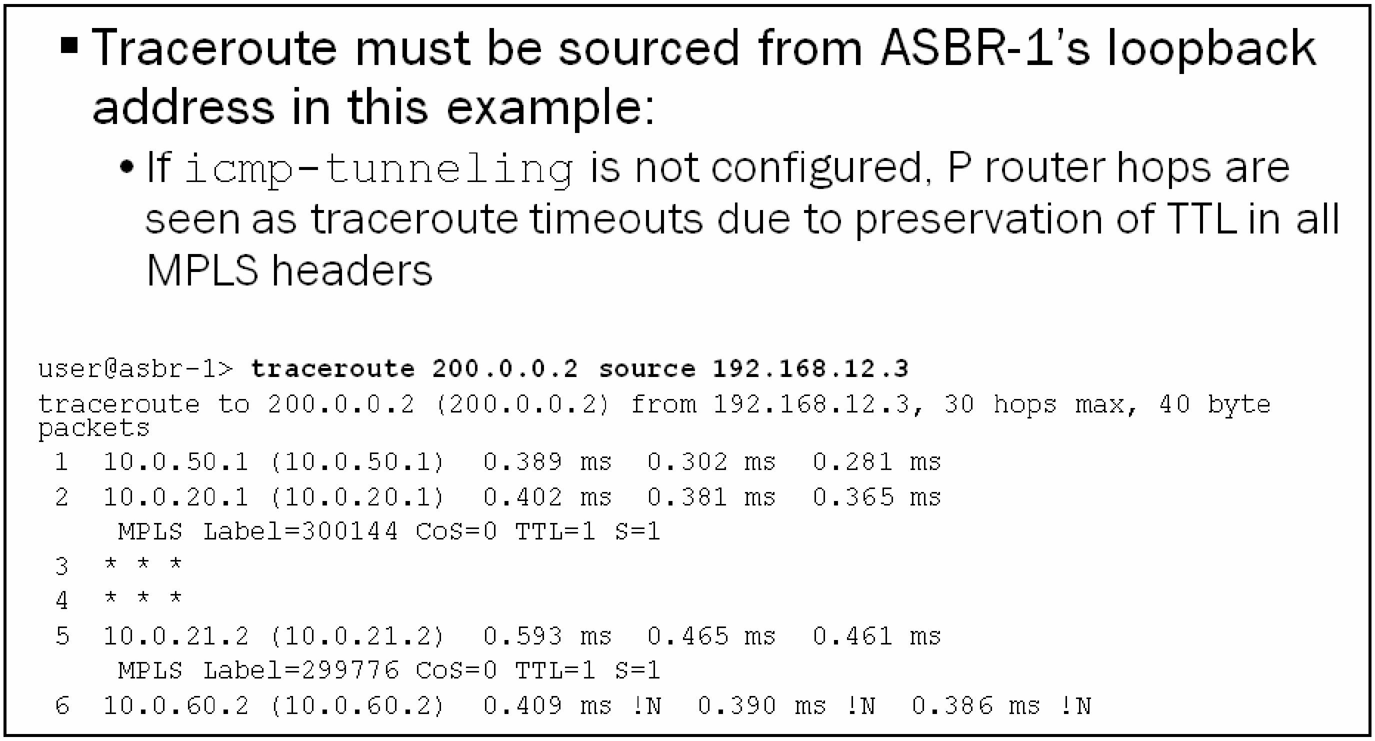
- PE-1's VPN MPLS forwarding table:
  - Swap/push operations create two-level label stack in provider core

```
user@pe-1> show route table vpn.mpls.0 detail

vpn.mpls.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
300112 (1 entry, 1 announced)
        *VPN    Preference: 170
                Next hop type: Indirect
                Next-hop reference count: 2
                Source: 192.168.2.2
                Next hop type: Router, Next hop index: 776
                Next hop: 172.22.221.2 via ge-1/0/1.221 weight 0x1, selected
                Label-switched-path pe1-to-pe2
                Label operation: Swap 299904, Push 302368(top)
                Protocol next hop: 192.168.2.2
                Swap 299904
                Indirect next hop: 28aab40 1048583
                State: <Active Int Ext>
                Local AS: 65512
                Age: 20:44      Metric2: 4
                Task: BGP RT Background
                Announcement bits (1): 0-KRT
```

This graphic shows a portion of PE-1's `vpn.mpls.0` switching table for the VRF instance called `vpn`. When PE-1 receives a packet with Label 300112, it swaps the top label with Label 299904 and then pushes an RSVP label (Label 302368) onto the top of the stack.

After PHP, PE-2 receives a packet with Label 299904, which it swaps with the label learned from CE-2 (labeled unicast route) before forwarding the singly labeled packet to CE-2.

Carrier-of-Carriers Operation: ASBR Traceroute



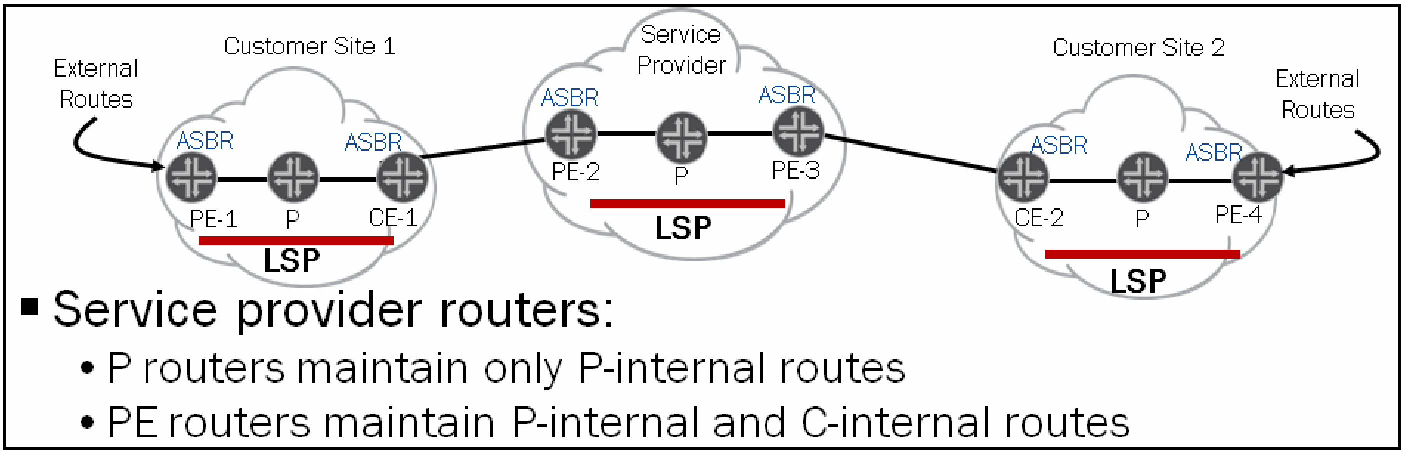
- Traceroute must be sourced from ASBR-1’s loopback address in this example:
  - If `icmp-tunneling` is not configured, P router hops are seen as traceroute timeouts due to preservation of TTL in all MPLS headers

```
user@asbr-1> traceroute 200.0.0.2 source 192.168.12.3
traceroute to 200.0.0.2 (200.0.0.2) from 192.168.12.3, 30 hops max, 40 byte
packets
 1  10.0.50.1 (10.0.50.1)  0.389 ms  0.302 ms  0.281 ms
 2  10.0.20.1 (10.0.20.1)  0.402 ms  0.381 ms  0.365 ms
     MPLS Label=300144 CoS=0 TTL=1 S=1
 3  * * *
 4  * * *
 5  10.0.21.2 (10.0.21.2)  0.593 ms  0.465 ms  0.461 ms
     MPLS Label=299776 CoS=0 TTL=1 S=1
 6  10.0.60.2 (10.0.60.2)  0.409 ms !N  0.390 ms !N  0.386 ms !N
```

This graphic shows a successful traceroute from ASBR-1 to the external route 200.0.0/24. Because only the /32 routes associated with customer loopback addresses are leaked, we must source the traceroute from the loopback address of ASBR-1.

In this example, the external route is a static route on ASBR-2, so hops beyond ASBR-2 are not present. Also, because the provider core routers (main routing instances) do not have routes associated with the customer networks, core router hops show up as timeouts.

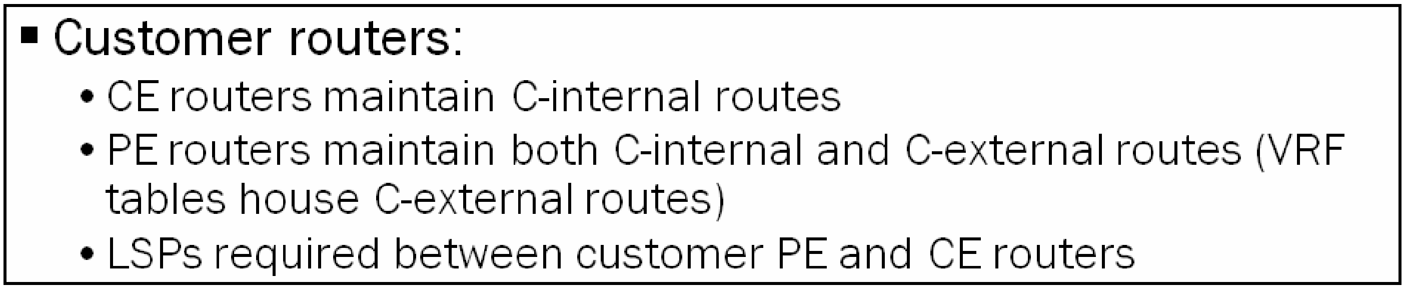
Service Provider Routers



- Service provider routers:
  - P routers maintain only P-internal routes
  - PE routers maintain P-internal and C-internal routes

The service provider’s P routers only maintain routes internal to the provider’s network (P-routes). The PE routers maintain both P-routes and customer internal routes (C-routes). The C-routes are housed in customer-specific VRF tables on the PE routers and normally consist of at least the customer’s /32 loopback addresses. The provider’s PE routers do not carry the customer’s external routes (C-external), which is a critical aspect of the overall scalability of this model.

Customer Routers



- Customer routers:
  - CE routers maintain C-internal routes
  - PE routers maintain both C-internal and C-external routes (VRF tables house C-external routes)
  - LSPs required between customer PE and CE routers

The customer's routers must maintain both C-internal and C-external routes. External routes are those learned from the customer's downstream subscribers and are now stored in site-specific VRF tables. Unlike the previous examples, the support of VPN routes requires that LSPs be established between customer PE and CE routers. These can be established using either RSVP or LDP. The use of LSP-based forwarding within the customer networks accommodates private/local use addressing.

ASBRs

ASBRs can be PE or CE routers and are used to connect with other autonomous systems. ASBRs advertise labeled routes between autonomous systems and maintain switching tables that allow them to stitch together LSPs existing in adjacent networks.

Three-Level Label Stack Required

- Three-level label stack in provider and customer networks
  - MP-I/EBGP needed for labeled route exchange

The presence of VRF-related labels results in the need to support three levels of label stacking in the provider and customer networks. In the case of PE-1, the three labels have the following functions:

1. The bottom label is the VRF label assigned using MP-BGP. This label does not change as the packet is forwarded.
2. The middle label is assigned by the downstream ASBR (CE-1, in the case of PE-1) and is used by the ASBR to associate the packet with the LSP leading to the next ASBR in the path.
3. The top label associates the packet with the LSP connecting PE-1 to CE-1 and is assigned by RSVP or LDP.

Because an LSP must be established across AS boundaries to interconnect customer PE routers, labels must be communicated along with the NLRI advertised by ASBRs. Although a protocol such as LDP could be used for this purpose, the Junos OS currently supports MP-BGP for this purpose.

RFC 3107, *Carrying Label Information in BGP-4*, specifies labeled routes. Labeled route advertisements use SAFI 4 and differ from VPN-labeled routes in that there is no route distinguisher or route target communities in the advertised NLRI. Simply put, labeled routes allow the binding of an MPLS label to the advertised IPv4 NLRI. ASBRs use the advertised labels to build MPLS switching tables that result in an end-to-end LSP spanning multiple autonomous networks.

Within an AS, labeled routes are sent using MP-IBGP while MP-EBGP is used across AS boundaries.

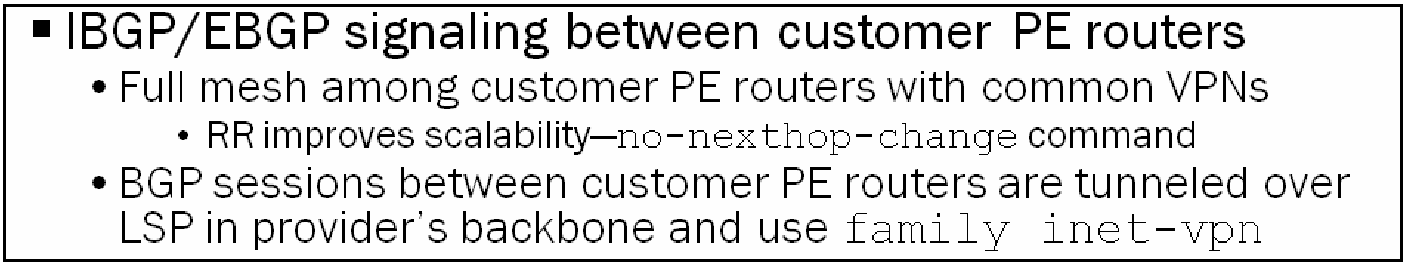
LSP Signaling Needed in Service Provider and Customer Networks

Because MPLS forwarding is now used end to end, LSPs must be signaled in both the customer and provider networks. The LSP signaling protocol need not be the same; the customer can use LDP while the provider uses RSVP.

MP-BGP Signaling Between Provider PE and Customer CE Routers

As with the previous application, the customer's CE routers use EBGP with `labeled-unicast` NLRI to exchange labeled routes with the provider's PE routers. The use of labeled routes allows the provider to extend its LSPs to the customer CE router and thereby eliminate the need to carry customer internal routes in its P routers.

IBGP/EBGP Signaling Between Customer ASBRs

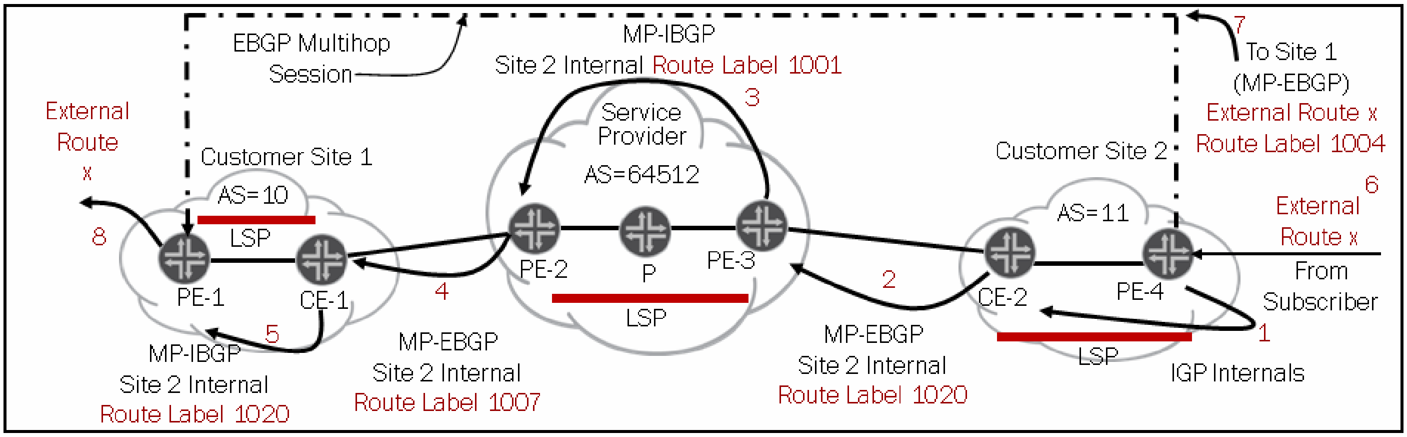


- IBGP/EBGP signaling between customer PE routers
  - Full mesh among customer PE routers with common VPNs
    - RR improves scalability—`no-nexthop-change` command
  - BGP sessions between customer PE routers are tunneled over LSP in provider's backbone and use `family inet-vpn`

Once the customer's internal routes are exchanged across the provider's backbone, the ASBRs (PE-1 and PE-4) can establish IBGP (same AS numbers) sessions or multihop EBGP (different AS numbers) sessions through the provider's backbone for the purposes of exchanging external routes. In this case, the routes are exchanged using MP-BGP and are labeled VPN routes.

To improve scalability, the customer networks can use route reflection. The two route reflectors peer using MP-EBGP. A command called `no-nexthop-change` is required to tell the route reflectors to pass—unchanged—the third party next hops to their clients.

Signaling: Step by Step



The details of the signaling exchanges shown on the graphic are:

1. The IGP at customer Site 2 exchanges internal reachability with CE-2. External (VRF) routes are not sent to PE-3.
2. CE-2 selectively advertises Site 2's internal routes to the provider's PE-3 router using MP-EBGP with support of `labeled-unicast` routes. The route to PE-4 is sent with a label value used to associate packets with the LSP to PE-4 in Site 2 (1020 in this example).
3. PE-3 houses Site 2's internal routes in a VRF table and uses MP-IBGP to send labeled VPN-IPv4 routes to PE-2. The route to PE-4 is assigned Label 1001 in this example.
4. PE-2 uses MP-EBGP to send Site 2's internal routes to CE-1. Because PE-2 has changed the BGP next hop (as is always the case with ASBRs), it must assign a new label to the prefix advertised (Label 1007 in this example).
5. After receiving the labeled route, CE-1 distributes Site 2's internal routes to PE-1 using MP-IBGP. Unlike the carrier-of-carriers application, this exchange involves labeled-unicast routes, and therefore requires MP-IBGP. Because CE-1 is also an ASBR, it rewrites the BGP next hop and must therefore assign a new label (Label 1020 in this example).

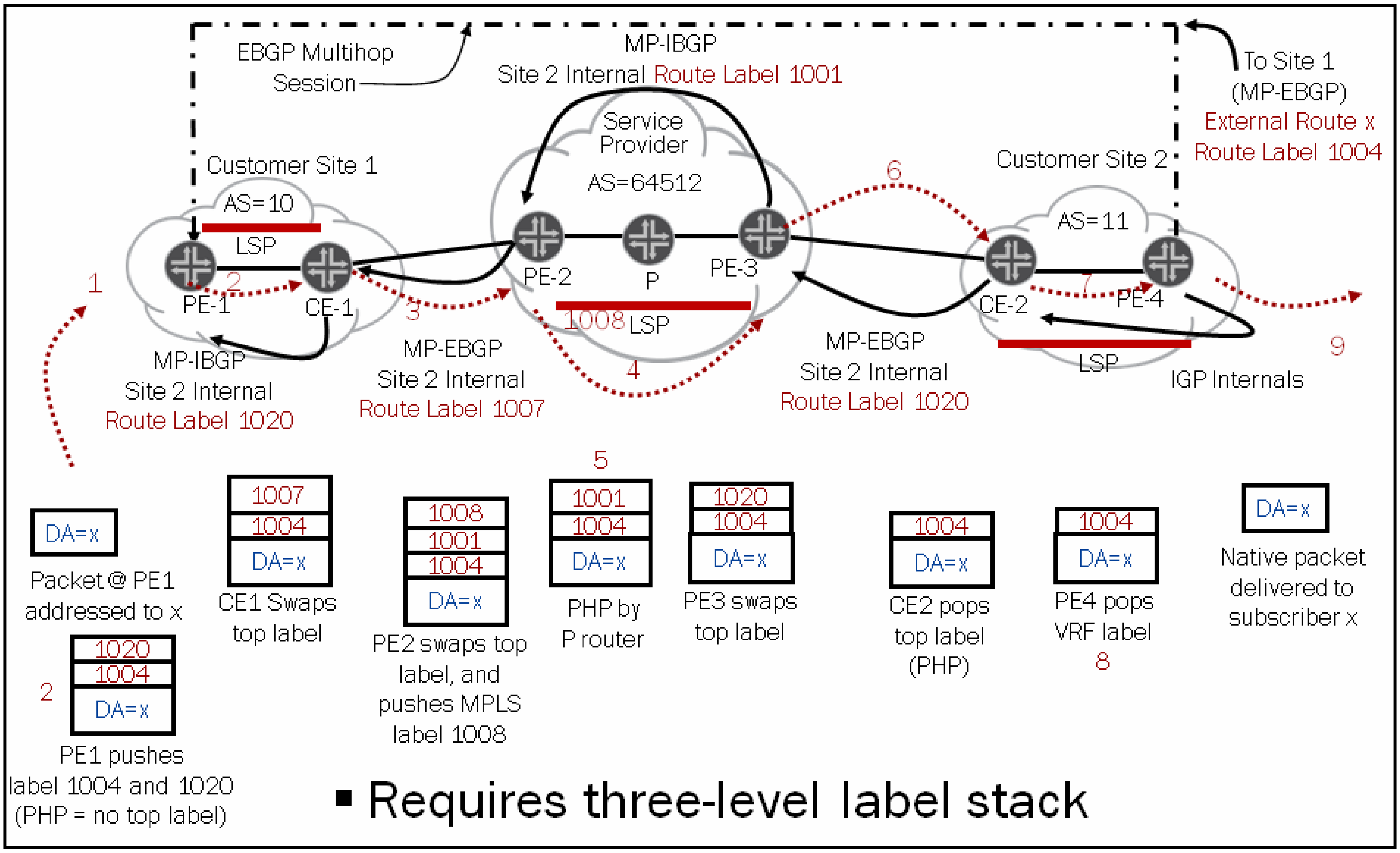
At this point, the ASBRs (PE-1 and PE-4) establish an MP-EBGP multihop session through the provider's backbone. This BGP session is tunneled through the LSP in the provider's network and is used to carry labeled VPN routes. This session should be contrasted to the carrier-of-carriers application, in which MP-I/EBGP was not needed due to native IP forwarding within the customer networks.

6. Here, PE-4 learns an external route *x* from one of its VPN subscribers.

7. The external route x is now advertised to PE-1 using the MP-EBGP session established at Step 5. This NLRI advertisement includes the VRF label that PE-4 expects to receive for routes associated with this particular VRF instance.

8. PE-1 advertises the external route to its downstream VPN subscribers.

Carrier-of-Carriers VPN Data Forwarding



- Requires three-level label stack

This graphic uses steps to describe the forwarding operations between PE-1 and PE-4 in an interprovider VPN application. The result is the need for a three-level label stack.

Forwarding: Step by Step

The details of the forwarding operation shown in the preceding graphic are:

1. A packet addressed to external route x arrives at PE-1.

2. PE-1 pushes two labels onto the packet: the inner label is the VRF label assigned by PE-4, and a second label assigned by CE-1 (to associate the packet with the LSP to PE-4). In this one-hop LSP example, PHP results in the absence of a third RSVP/LSP label used to associate the packet with the LSP between PE and CE routers.

3. CE-1 receives the labeled packet and swaps the top label.

4. PE-2 receives the labeled packet and swaps the top label with the value received from PE-3 while also pushing the MPLS label (Label 1008 in this example) onto the stack.

5. The P router pops the top label (PHP) so that PE-3 receives a packet with only two labels.

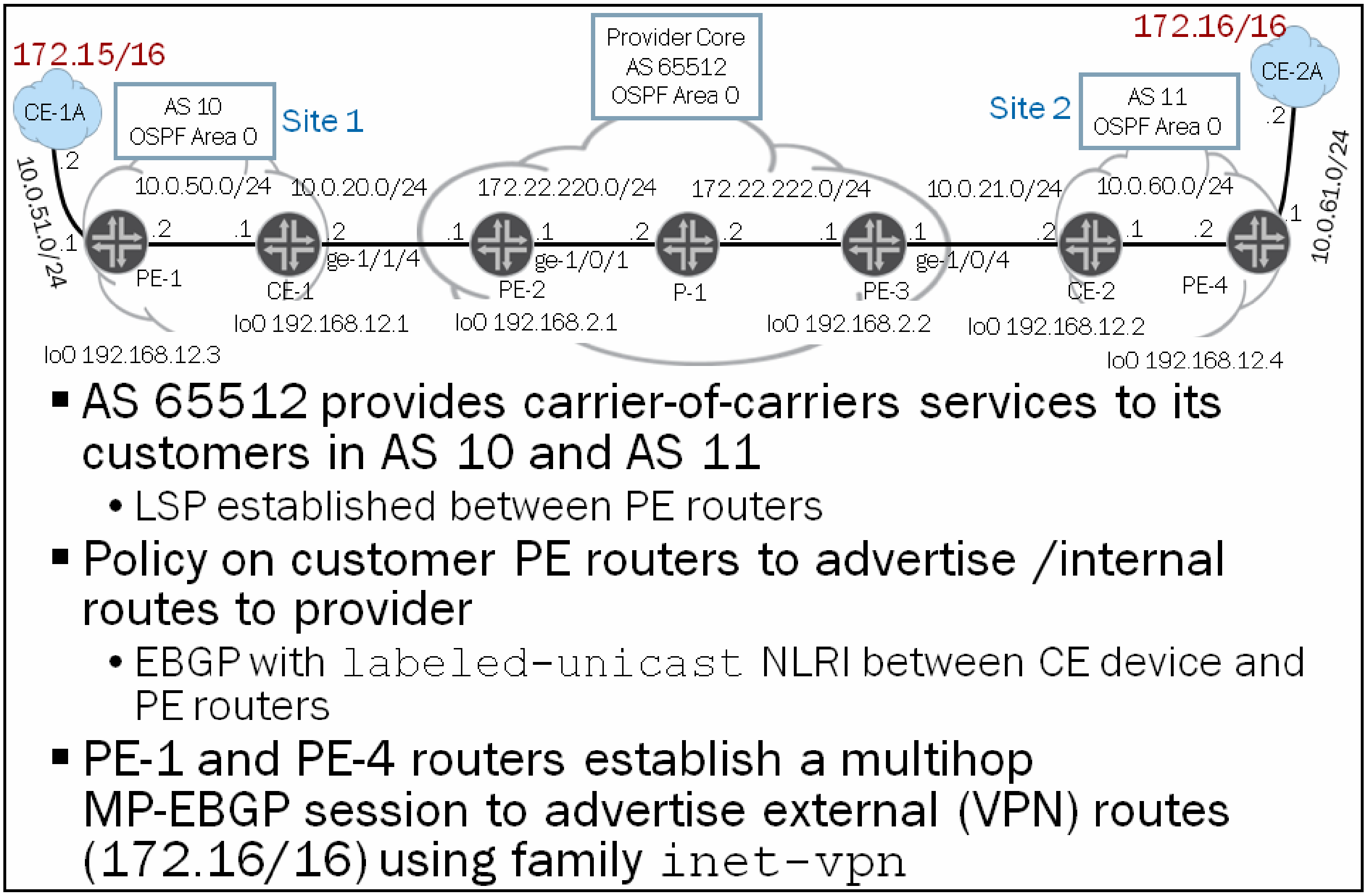
6. PE-3 also performs a swap on the top label before forwarding the packet to CE-2.

7. Being the penultimate router for the LSP to PE-4, CE-2 pops the label stack and sends the resulting VRF-labeled packet to PE-4.

8. PE-4 pops the VRF label and consults the corresponding VRF table to perform a longest-match lookup on the now unlabeled packet.

9. The native packet is forwarded out PE-4's VRF interface towards the subscriber to which it is addressed.

AS 65512 Provides Carrier-of-Carriers Services



- AS 65512 provides carrier-of-carriers services to its customers in AS 10 and AS 11
  - LSP established between PE routers
- Policy on customer PE routers to advertise /internal routes to provider
  - EBGP with `labeled-unicast` NLRI between CE device and PE routers
- PE-1 and PE-4 routers establish a multihop MP-EBGP session to advertise external (VPN) routes (172.16/16) using family `inet-vpn`

This graphic provides a sample network. The following sections show various configuration-mode and operational-mode screen captures relating to this graphic.

The provider's network is assigned AS 65512. It already has established an LSP between PE-2 and PE-3 using RSVP. The PE routers have a VRF table configured, along with the necessary VRF target and route distinguishers. PE-2 and PE-3 function as ASBRs in this application.

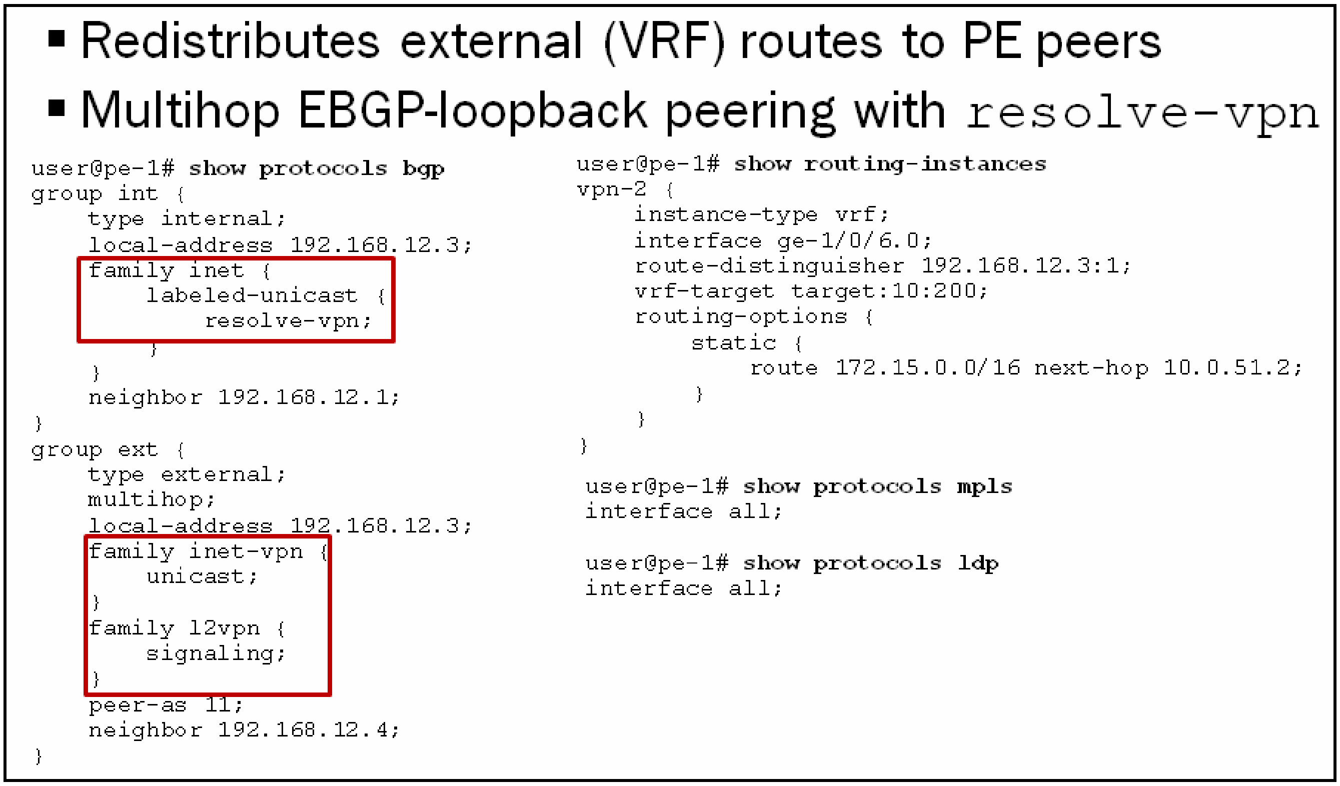
Policy on CE Routers

The CE routers are configured to run MP-EBGP (`family inet labeled-unicast`) with the PE routers and have a policy in place to ensure that only internal prefixes are advertised to the provider's PE routers.

PE-1 and PE-4 Routers Exchange External Routes

A multihop MP-EBGP session is configured between the PE-1 and PE-4, because the customer networks are assigned different AS numbers. The external route 172.16/16 is advertised as a labeled-VPN route by PE-4 to PE-1 using this MP-EBGP session. Customer routers CE-1 and CE-2 also function as ASBRs in this example.

PE-1 Configuration



- Redistributes external (VRF) routes to PE peers
- Multihop EBGP-loopback peering with `resolve-vpn`

```
user@pe-1# show protocols bgp
group int {
    type internal;
    local-address 192.168.12.3;
    family inet {
        labeled-unicast {
            resolve-vpn;
        }
    }
    neighbor 192.168.12.1;
}
group ext {
    type external;
    multihop;
    local-address 192.168.12.3;
    family inet-vpn {
        unicast;
    }
    family l2vpn {
        signaling;
    }
    peer-as 11;
    neighbor 192.168.12.4;
}
```

```
user@pe-1# show routing-instances
vpn-2 {
    instance-type vrf;
    interface ge-1/0/6.0;
    route-distinguisher 192.168.12.3:1;
    vrf-target target:10:200;
    routing-options {
        static {
            route 172.15.0.0/16 next-hop 10.0.51.2;
        }
    }
}
```

```
user@pe-1# show protocols mpls
interface all;
```

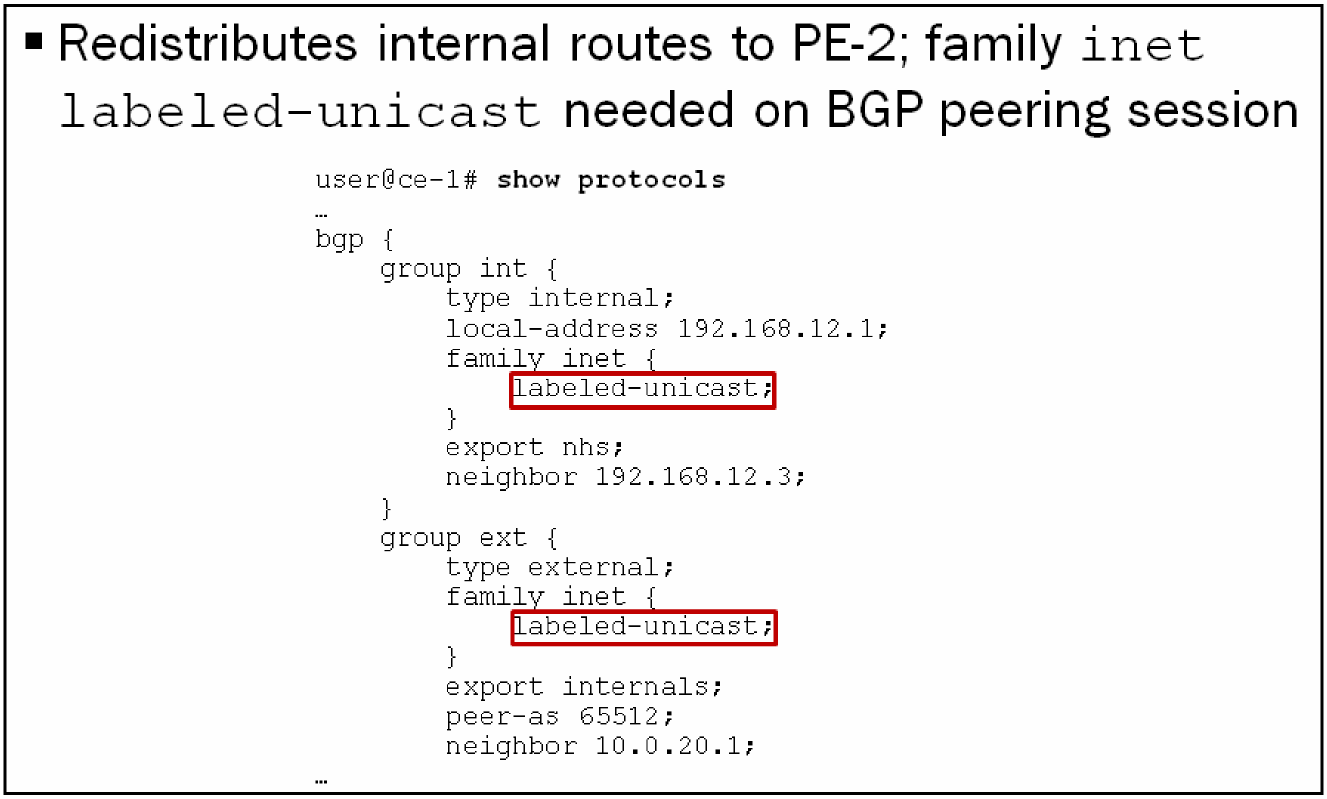
```
user@pe-1# show protocols ldp
interface all;
```

This graphic shows the key aspects of PE-1's configuration. Family `labeled-unicast` is configured for its MP-IBGP session to CE-1, and family `inet-vpn` is configured for the multihop MP-EBGP session to PE-4.

`resolve-vpn`

The `resolve-vpn` option causes PE-1 to copy the `labeled-unicast` routes it receives from CE-1 into `inet.3`, which allows VPN routes to resolve through the interprovider LSPs. Without this option, all the VPN routes received would be hidden, due to unusable next hops.

CE-1 Configuration



- Redistributes internal routes to PE-2; family `inet labeled-unicast` needed on BGP peering session

```
user@ce-1# show protocols
…
bgp {
    group int {
        type internal;
        local-address 192.168.12.1;
        family inet {
            labeled-unicast;
        }
        export nhs;
        neighbor 192.168.12.3;
    }
    group ext {
        type external;
        family inet {
            labeled-unicast;
        }
        export internals;
        peer-as 65512;
        neighbor 10.0.20.1;
…
```

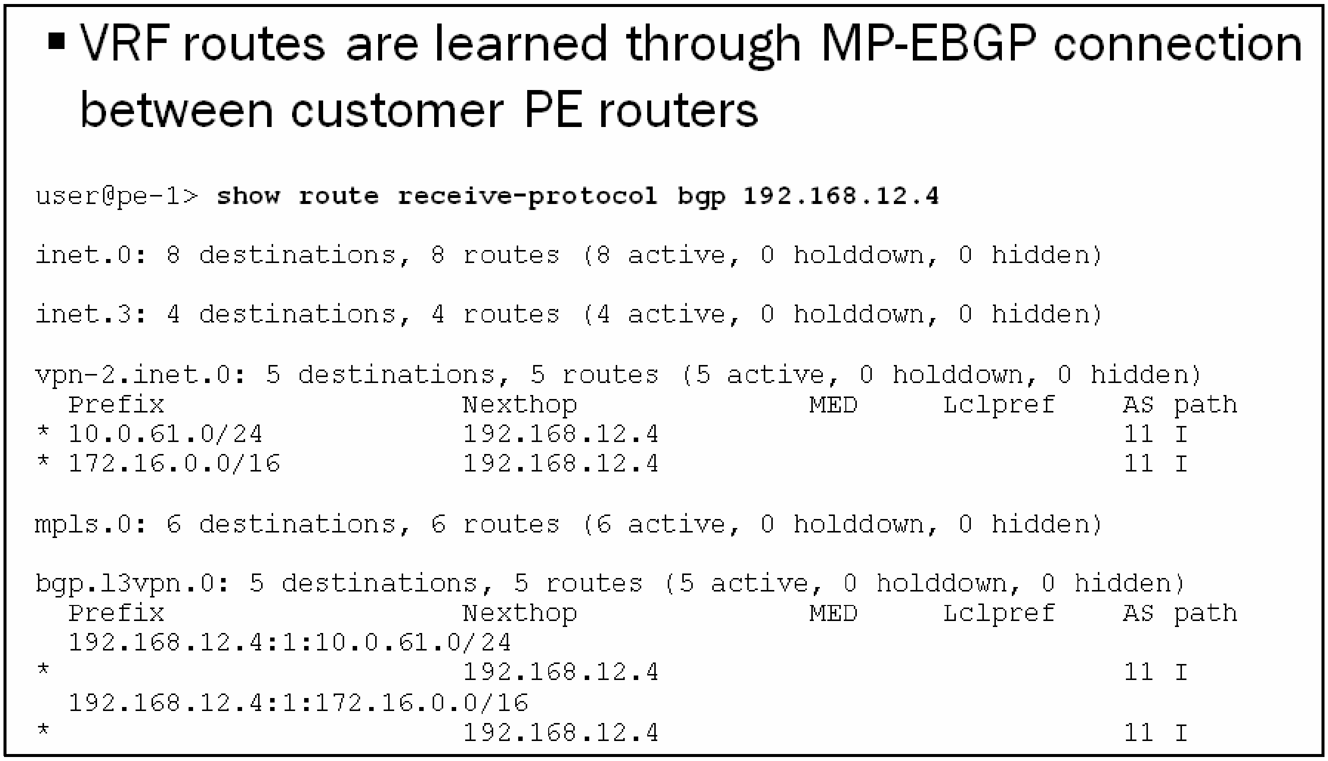
This graphic displays key portions of the configuration on CE-1. RSVP is enabled, and an LSP is defined back to PE-1 (not shown). The MP-IBGP session to PE-1 has the `labeled-unicast` family configured. This configuration is needed so that CE-1 can include `labeled-unicast` routes along with the advertisements for Site 2's internal routes.

CE-1 also has an MP-EBGP session configured for its connection to PE-2. This session must also support `labeled-unicast` routes.

The following policy is applied to CE-1's MP-EBGP session to PE-2. This policy ensures that Site 1 sends only internal routes to the provider:

```
lab@ce-1# show policy-options
policy-statement internals {
    term 1 {
        from protocol [ ospf direct ];
        then accept;
    }
    term 3 {
        then reject;
    }
}
```

Carrier-of-Carriers VPNs Operation: VPN Routes

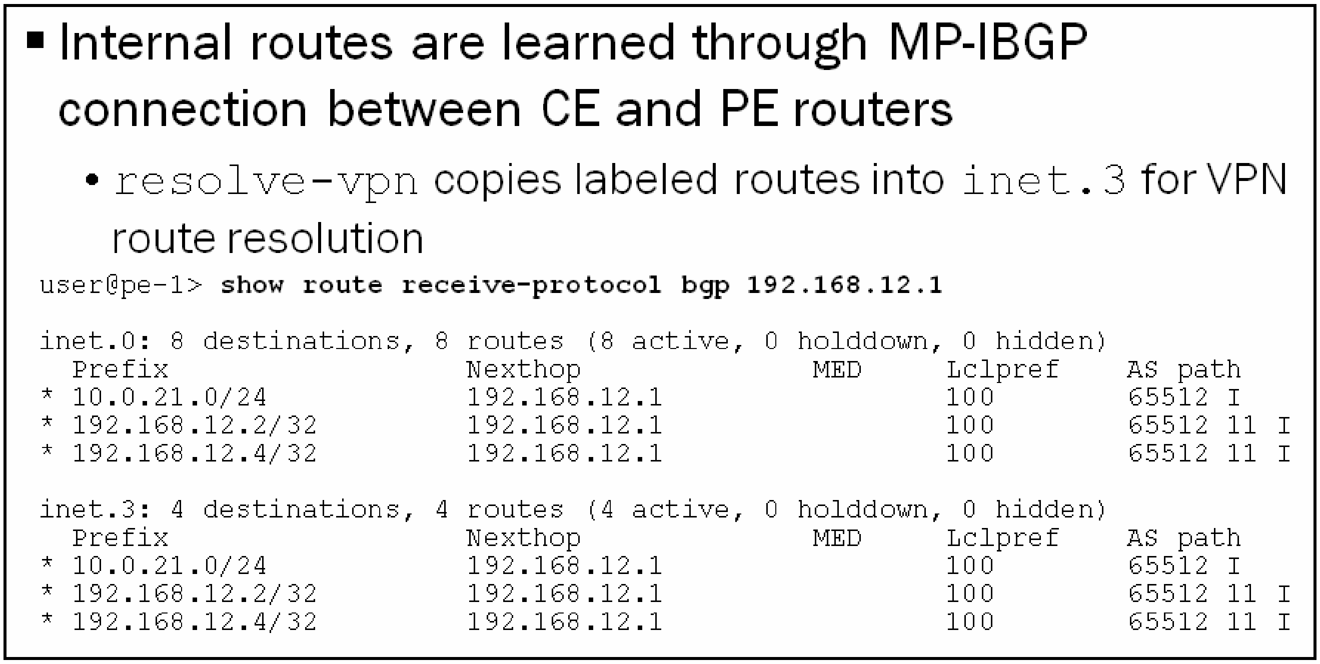


- VRF routes are learned through MP-EBGP connection between customer PE routers

```
user@pe-1> show route receive-protocol bgp 192.168.12.4

inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)

inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)

vpn-2.inet.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.61.0/24            192.168.12.4                            11 I
* 172.16.0.0/16           192.168.12.4                            11 I

mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)

bgp.l3vpn.0: 5 destinations, 5 routes (5 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
  192.168.12.4:1:10.0.61.0/24
*                         192.168.12.4                            11 I
  192.168.12.4:1:172.16.0.0/16
*                         192.168.12.4                            11 I
```

This graphic shows that PE-1 learns labeled VPN routes from PE-4 at Site 2. These routes are associated with a VRF label (not shown) used by the advertising router (PE-4) to associate the packets with the correct VRF table.

Carrier-of-Carriers VPN Operation: Internal Routes



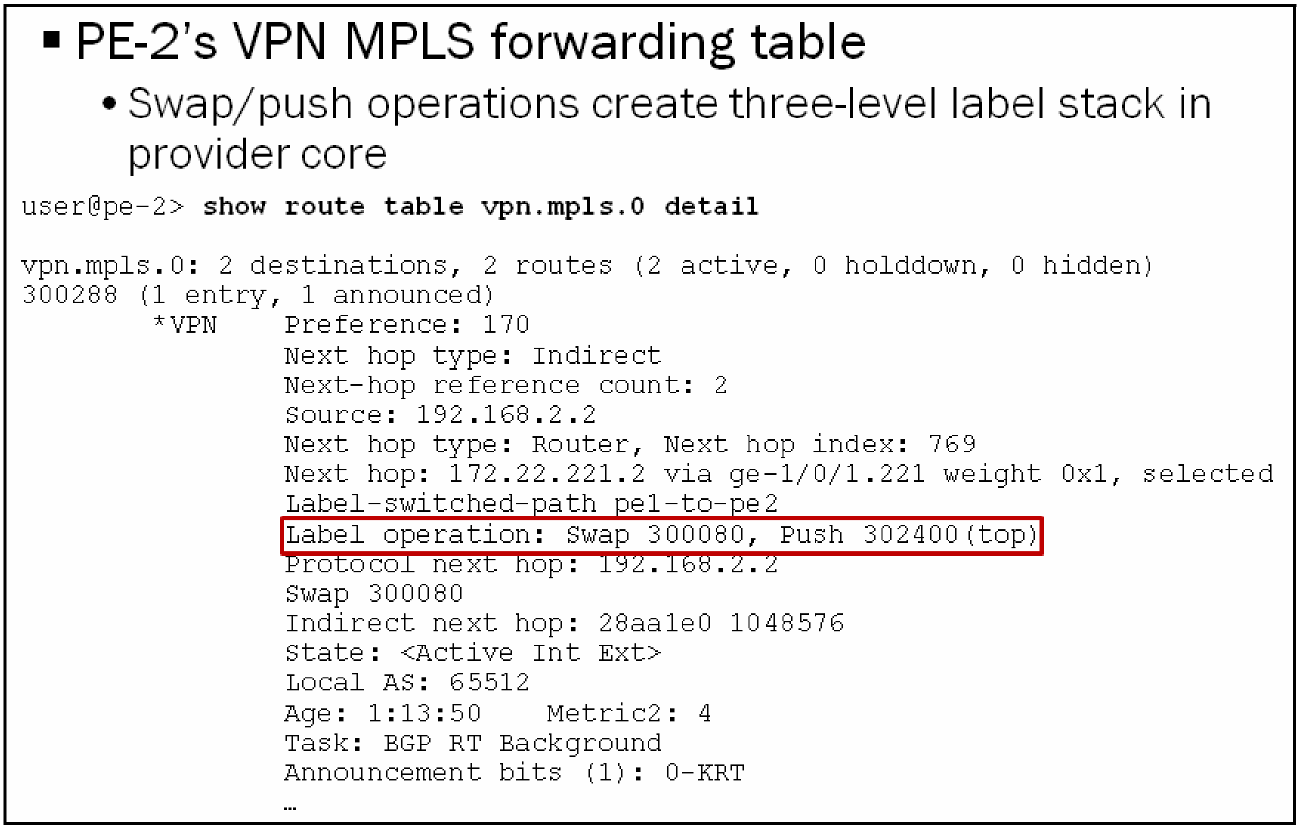
- Internal routes are learned through MP-IBGP connection between CE and PE routers
  - `resolve-vpn` copies labeled routes into `inet.3` for VPN route resolution

```
user@pe-1> show route receive-protocol bgp 192.168.12.1

inet.0: 8 destinations, 8 routes (8 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.21.0/24            192.168.12.1                 100        65512 I
* 192.168.12.2/32         192.168.12.1                 100        65512 11 I
* 192.168.12.4/32         192.168.12.1                 100        65512 11 I

inet.3: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
  Prefix                  Nexthop              MED     Lclpref    AS path
* 10.0.21.0/24            192.168.12.1                 100        65512 I
* 192.168.12.2/32         192.168.12.1                 100        65512 11 I
* 192.168.12.4/32         192.168.12.1                 100        65512 11 I
```

This graphic shows that PE-1 learns about Site 2's internal routes through its MP-IBGP connection to CE-1 (an ASBR).

The `labeled-unicast` routes received by PE-1 are copied into the main routing table (`inet.0`) as well as the `inet.3` table. This copying is the result of the `resolve-vpn` option on PE-1 and is critical to the operation of this network. Normally, VPN routes must resolve to an LSP that terminates on the egress PE router. Because PE-1 does not have an LSP terminating directly on PE-4, the VPN routes are unusable without the `labeled-unicast` entries to the remote PE routers in `inet.3`, which indicate a multinetwork LSP between PE-1 and PE-4 exists.

Carrier-of-Carriers VPN Operation: PE-2



- PE-2's VPN MPLS forwarding table
  - Swap/push operations create three-level label stack in provider core

```
user@pe-2> show route table vpn.mpls.0 detail

vpn.mpls.0: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
300288 (1 entry, 1 announced)
        *VPN    Preference: 170
                Next hop type: Indirect
                Next-hop reference count: 2
                Source: 192.168.2.2
                Next hop type: Router, Next hop index: 769
                Next hop: 172.22.221.2 via ge-1/0/1.221 weight 0x1, selected
                Label-switched-path pe1-to-pe2
                Label operation: Swap 300080, Push 302400(top)
                Protocol next hop: 192.168.2.2
                Swap 300080
                Indirect next hop: 28aa1e0 1048576
                State: <Active Int Ext>
                Local AS: 65512
                Age: 1:13:50    Metric2: 4
                Task: BGP RT Background
                Announcement bits (1): 0-KRT
                …
```

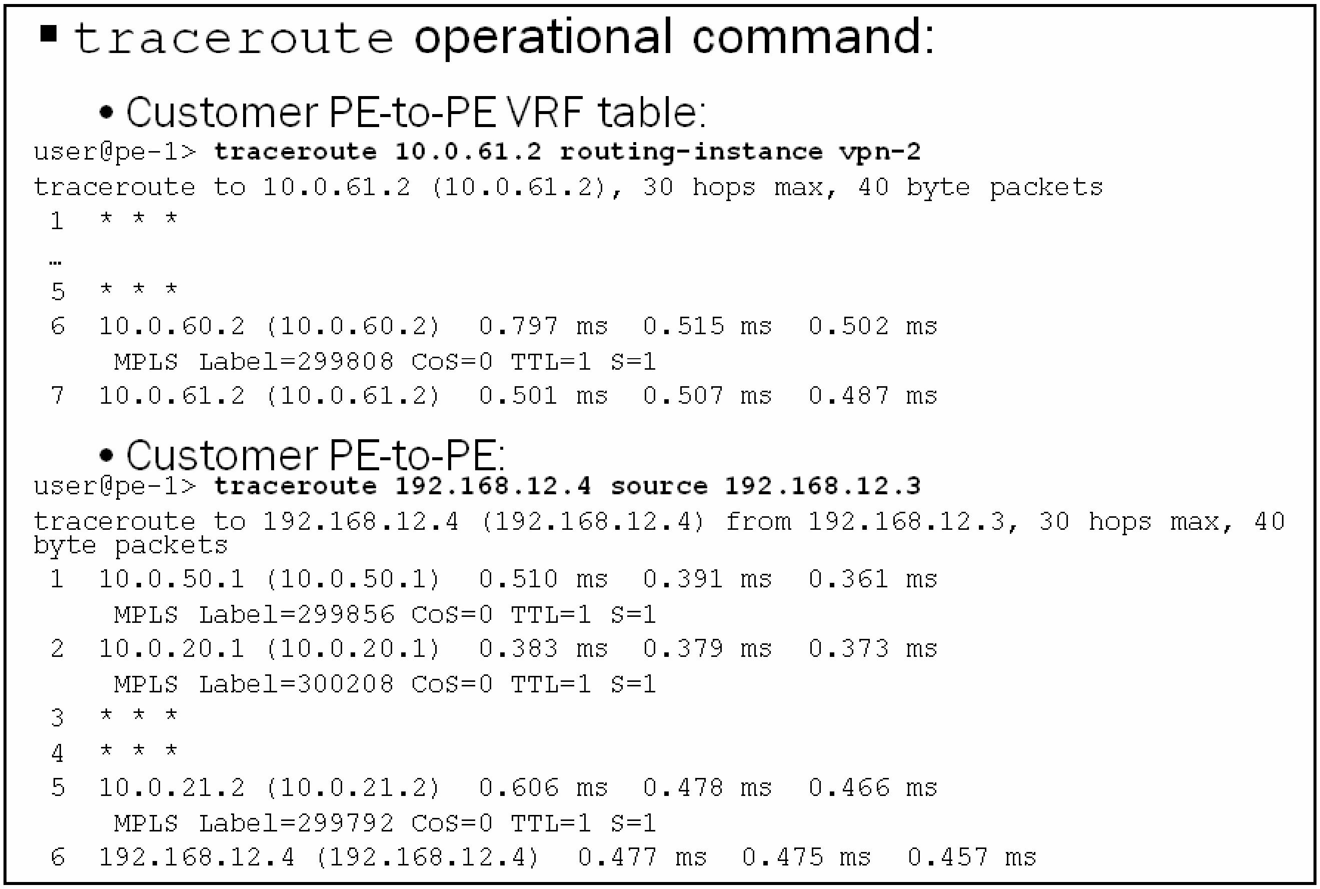
This graphic shows the VPN instance's `mpls.0` table that exists on PE-2. Here, packets received with a label of 300288 have their top label swapped. PE-2 pushes a new label (obtained from RSVP or LDP) onto the stack, creating a three-level label stack (VRF-label—300080—302400).

The top label is popped by the provider P router (PHP), such that PE-3 receives a packet with a two-level label stack. PE-3 swaps the top label with the value assigned to the LSP to PE-4 by CE-2. PE-3's label operation is shown here:

```
user@pe-3> show route table mpls.0

mpls.0: 6 destinations, 6 routes (6 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

...
299904             *[VPN/170] 19:07:27
                    > to 10.0.21.2 via ge-1/0/4.0, Pop
299904(S=0)        *[VPN/170] 19:07:27
                    > to 10.0.21.2 via ge-1/0/4.0, Pop
300080             *[VPN/170] 01:15:43
                    > to 10.0.21.2 via ge-1/0/4.0, Swap 299872
```

The result is that CE-2 receives a packet with a two-level label stack (VRF-label—299872). CE-2 then swaps the top label with the value it associates with the LSP to the egress PE router. In this example, CE-2 pops the stack because it is the penultimate router for this LSP.

Carrier-of-Carriers VPN Operation: `traceroute`



- `traceroute` operational command:
  - Customer PE-to-PE VRF table:

```
user@pe-1> traceroute 10.0.61.2 routing-instance vpn-2
traceroute to 10.0.61.2 (10.0.61.2), 30 hops max, 40 byte packets
 1  * * *
 …
 5  * * *
 6  10.0.60.2 (10.0.60.2)  0.797 ms  0.515 ms  0.502 ms
     MPLS Label=299808 CoS=0 TTL=1 S=1
 7  10.0.61.2 (10.0.61.2)  0.501 ms  0.507 ms  0.487 ms
```

  - Customer PE-to-PE:

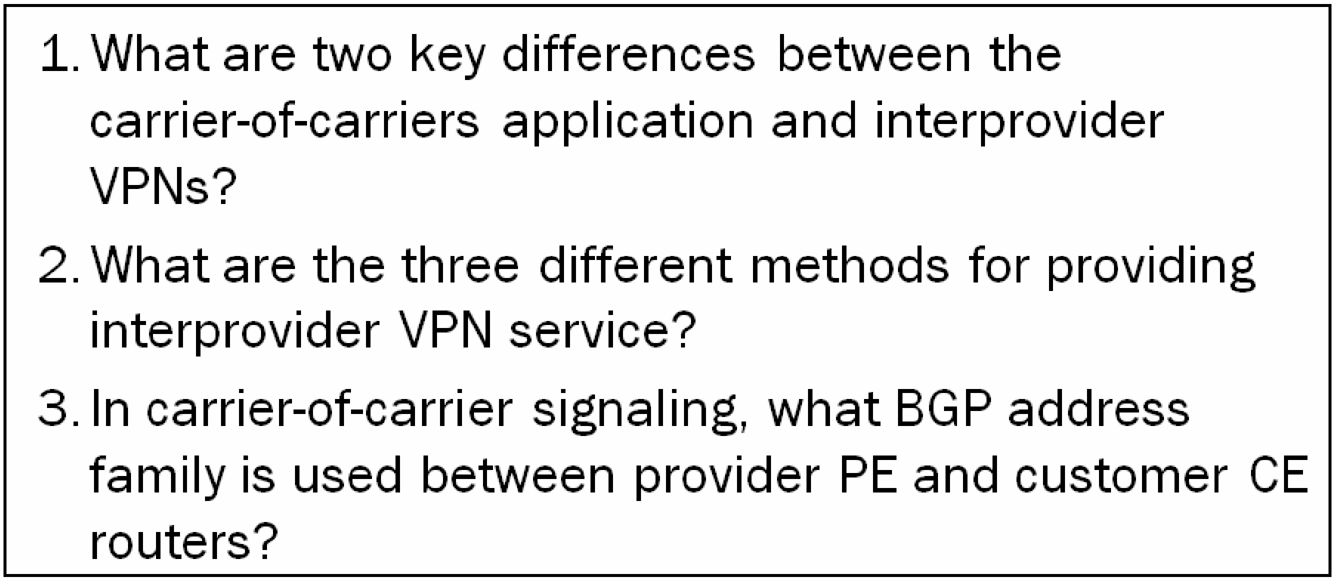
```
user@pe-1> traceroute 192.168.12.4 source 192.168.12.3
traceroute to 192.168.12.4 (192.168.12.4) from 192.168.12.3, 30 hops max, 40
byte packets
 1  10.0.50.1 (10.0.50.1)  0.510 ms  0.391 ms  0.361 ms
     MPLS Label=299856 CoS=0 TTL=1 S=1
 2  10.0.20.1 (10.0.20.1)  0.383 ms  0.379 ms  0.373 ms
     MPLS Label=300208 CoS=0 TTL=1 S=1
 3  * * *
 4  * * *
 5  10.0.21.2 (10.0.21.2)  0.606 ms  0.478 ms  0.466 ms
     MPLS Label=299792 CoS=0 TTL=1 S=1
 6  192.168.12.4 (192.168.12.4)  0.477 ms  0.475 ms  0.457 ms
```

This graphic shows the results of a VRF table-to-VRF table `traceroute` operational command as well as a `traceroute` operational command from ingress PE router to egress PE router.

All other router hops in the customer an provider networks are seen as traceroute timeouts.

The traceroute between PE-1 and PE-4 shows the outer MPLS label for the various hops in the path, except for the provider's routers which appear as timeouts. The provider routers are not able to generate traceroute responses, owing to their not carrying customer external or internal routes.

Review Questions



1. What are two key differences between the carrier-of-carriers application and interprovider VPNs?
2. What are the three different methods for providing interprovider VPN service?
3. In carrier-of-carrier signaling, what BGP address family is used between provider PE and customer CE routers?

Answers to Review Questions

1.

In a carrier-of-carrier application the customer routers maintain both customer internal and external routes. In an interprovider VPN, except for the ASBRs connect to VPN sites, the customer routers maintain customer internal routes only.

2.

Option A specifies the use of separate VRFs for every VPN on the ASBRs. Option B specifies the used of the EBGP exchange of VPN routes between ASBRs. Option C specifies the use of multihop EBGP (or IBGP) to exchange VPN routes between PEs in remote autonomous systems.

3.

The `labeled-unicast` address family is used between PE and CE.