Configuring VPLS

SYSTEM ADMINISTRATOR GUIDE

Copyright

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

SmartEdge	is a registered trademark of Telefonaktiebolaget LM Ericsson.
NetOp	is a trademark of Telefonaktiebolaget LM Ericsson.

Contents

1 Overview

This document provides an overview of Virtual Private LAN Services (VPLS) and describes the tasks and commands used to configure, operate, monitor, and troubleshoot VPLS features through the SmartEdge router.

This document applies to both the Ericsson SmartEdge® and SM family routers. However, the software that applies to the SM family of systems is a subset of the SmartEdge OS; some of the functionality described in this document may not apply to SM family routers.

For information specific to the SM family chassis, including line cards, refer to the SM family chassis documentation.

For specific information about the differences between the SmartEdge and SM family routers, refer to the Technical Product Description *SM Family of Systems* (part number 5/221 02-CRA 119 1170/1) in the **`Product Overview`** folder of this Customer Product Information library.

VPLS enables networks at separate geographical locations to communicate with each other across a WAN as if they were directly attached to each other in a LAN. Creating VPLS pseudowires (PWs) makes the WAN transparent.

A PW emulates the attributes and function of Ethernet connectivity over a WAN. Any required switching functionality or service translation is outside the scope of the pseudowire and of the transport network. Pseudowires are carried over Multiprotocol Label Switching (MPLS) tunnels on the network.

MPLS signaling protocols are used to automatically provision a service on a PW end-to-end, so you can provision a PW by pointing to its two endpoints, and MPLS automatically negotiates the path.

Figure 1 displays the network topology for a typical VPLS configuration.

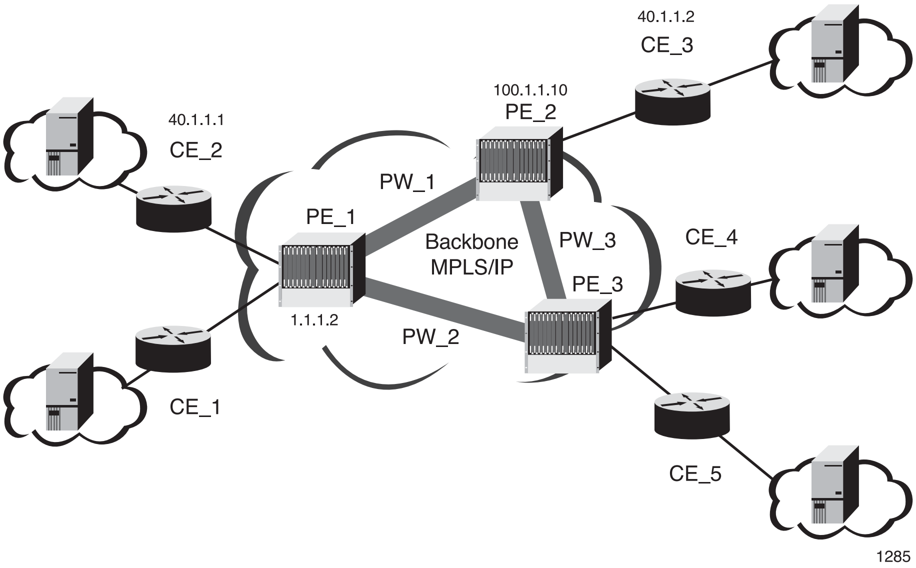


Figure 1 Typical VPLS Network Topology

Customer edge (CE) routers, which are on the edge of geographically separate customer networks, are connected by Ethernet to provider edge (PE) routers on an MPLS provider network. A PW is established for each pair of CE routers that are to be connected into a virtual private LAN. For example, the PW_1 pseudowire is used to connect the CE_1 and CE_3 routers. The CE_4, and CE_5 routers connect with the CE_1 router over the VPLS network, but not necessarily to each other.

To create PWs, a VPLS-enabled bridge must first be configured on each PE router, and then peering (neighbor) sessions can be established across that bridge. The PW is the circuit across which the peering session occurs. A VPLS-enabled bridge can have multiple peering sessions.

VPLS supports pseudowire load balancing, RSVP object tracking, and VCCV on pseudowires, as described in the sections that follow.

For procedures to troubleshoot issues with VPLS, see *Troubleshooting VPLS*.

1.1 Pseudowire Load Balancing

Pseudowire load balancing is supported for VPLS traffic. By default, load balancing is disabled, and traffic from all channels of a pseudowire traverse the same path. When load balancing is enabled, traffic from individual IPTV channels in a pseudowire is distributed among the links in the link group. The load balancing scheme for each pseudowire is based on source and destination information. Use the **`pseudowire multi-path`** command in global configuration mode to enable load balancing on all VPLS pseudowires configured on the router or in the current context. Be aware that load balancing is enabled and disabled in global configuration mode; this means that load balancing is globally enabled or disabled for all pseudowires that are configured on the router.

The SmartEdge router supports the mapping of traffic from a PW carrying traffic from a given VPLS instance to a specific RSVP tunnel. The SmartEdge router maps all MPLS traffic exiting the PW to a specific tunnel destined for the peer of the VPLS instance.

1.2 RSVP Object Tracking

In a typical hub-and-spoke network configuration, link aggregation nodes communicate with maximum transmission units (MTUs) through access nodes in a pseudowire-to-RSVP-tunnel configuration. In VPLS networks that use RSVP, an RSVP object tracking feature can be enabled to ensure that a MTU does not lose connectivity to the required link aggregation nodes if a link failure occurs between the MTU and its primary access node. When a primary access node loses connectivity to its associated aggregation nodes, the RSVP object tracking feature transparently changes MTU connectivity to a standby access node, and traffic flow continues uninterrupted.

Configuring RSVP object tracking is a three-step process:

1. Determine which interfaces need to be tracked. To track these interfaces, you must include them in an RSVP tracking object, as described in step 2.

2. On the primary access node, create an RSVP tracking object. A tracking object typically contains all the interfaces that exist between an access node and its associated link aggregation nodes.

3. On the primary access node, apply the tracking object name to the primary RSVP LSP or LSPs that must track the interfaces in that object.

The state of an RSVP object is reflected by its associated LSPs. If an RSVP tracking object is up, then the LSPs that reference that tracking object remain up. If an RSVP tracking object goes down, then the LSPs that reference that tracking object go down. When an RSVP tracking object is down, the RSVP LSPs that are tracking that object switch over to the standby spoke. In this case, the MTU switches over from the primary access node to the standby access node so that it does not lose connectivity to the aggregation nodes. After an access node loses connectivity to the aggregation nodes, RSVP continuously attempts to restore the down interfaces between the aggregation node and the access node. When at least one of the aggregation-facing interfaces comes back up and remains stable for five minutes, the tracking object that contains that interface comes back up, and connectivity transparently changes back to the primary access node.

Figure 2 shows an example of a hub-and-spoke network that has RSVP object tracking configured:

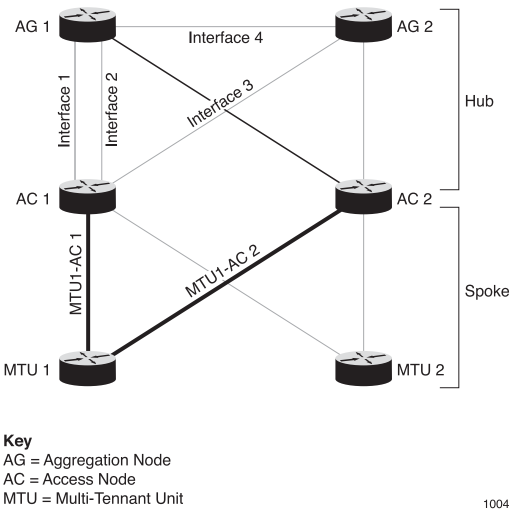


Figure 2 Example: RSVP Object Tracking in a Hub-and-Spoke Network Area

In Figure 2, the interfaces 1, 2, and 3 belong to a single tracking object. If all three of those interfaces go down, then `MTU-1` switches over from the a primary spoke (carried within tunnel `MTU1-AC1`) to the standby spoke (carried within tunnel `MTU1-AC2`) and accesses the aggregation routers (`AG-1` and `AG-2`) through the standby access node (`AC-2`). If any one of the down interfaces (interface 1, 2, or 3) comes back up and remains stable for five minutes, then `MTU-1` switches back over to `MTU1-AC1`, and accesses the aggregation nodes through the primary access server (`AC-1`) once again. In addition, if `MTU1-AC1` itself goes down, then `MTU-1` switches over to the standby spoke until `MTU1-AC1` becomes available again.

1.3 VCCV on Pseudowires

LDP uses the Virtual Circuit Connectivity Verification (VCCV) protocol during PW setup for connectivity verification. VCCV provides a control channel on a pseudowire over which connectivity verification messages are sent.

1.3.1 Including Control Words

You can choose to include a control word in the PW cell header. The control word is used to detect packet reordering and packet loss, and to perform ECMP path avoidance and various OAM tasks.

The control word can be enabled for both ATM and Ethernet PWs. However, control-word functionality is somewhat different between ATM and Ethernet PWs. These differences are described in the sections that follow.

1.3.1.1 Control Word on ATM PWs

All ATM pseudowire cross-connections allow you to enable or disable the inclusion of a four-byte PW control word after the MPLS labels. The control word contains the following fields:

- 16-bit packet sequence number used to detect packet reordering and packet loss
- Payload length

The sequence number is 16-bits and ranges from 1 through 65535. Sequence numbers are added to the control word incrementally, and the number wraps to 1 after reaching the maximum value. The inclusion of a sequence number ensures that packets are reordered properly if they become disassembled while traveling over the pseudowire.

Note: With *n*-to-1 cell encapsulation, the control word is optional.

Use the **`control-word`** command in L2VPN XC group configuration mode to enable inclusion of a control word in an ATM PW cell header. After you enable inclusion of the control word, the SmartEdge OS includes a single control word for the specified cross-connection group. The control word is applied to all pseudowires configured under the particular group.

To disable inclusion of a sequence number, enter the **`control-word sequence-number`** `0` command in the appropriate command mode.

1.3.1.2 Control Word on Ethernet PWs

All Ethernet pseudowire cross-connections that are not statically defined now allow you to enable or disable the inclusion of a control word between the Ethernet Frame and the MPLS label. The use of the control word ensures that all traffic for a given label stack follows the same path if hashing occurs while ECMP exists. When the control word is present, all traffic follows a single path because further lookups for the packet do not occur. The control word also permits the VCCV packet to follow the same path through the data plane that is taken by the PW data packets. Because sequencing is not required for Ethernet PWs, the Ethernet control word is made up of all zeroes. The control word is supported on both tagged and untagged Ethernet frames.

During Ethernet PW setup, LDP advertises the control word capability to the underlying PW. If the peer does not have control word capabilities enabled, then the control word is not included in the header of the packets that are sent over the PW.

Note: The control word is not available for Ethernet PWs that are statically defined.

Use the `control-word` command in the appropriate mode to enable the inclusion of a sequence number in an Ethernet PW cell header, as follows:

- VPLS profile neighbor configuration mode—Configures the control word attribute for a VPLS neighbor in a VPLS profile. The attributes in the VPLS profile are automatically applied to the XC to which the profile is attached.
- L2VPN profile peer configuration mode—Configures the control word setting in an L2VPN profile. The attributes in the L2VPN profile are automatically applied to the XC to which the profile is attached.

Note: This feature is available on VPLS and L2VPN Ethernet and VLAN PWs only. Be aware that this feature is not operational on L2VPN or VPLS PWs that are configured over a GRE tunnel.

1.3.2 VCCV Capability Negotiation During PW Setup

The VCCV-capable PW setup feature provides the following functionality for L2VPN or VPLS PWs:

- When a user first configures VPLS peers or L2VPNs, LDP advertises the following local VCCV capabilities to the underlying PWs:
  - CC-Type=0x01—Pseudowire Emulation Edge to Edge (pwe3) control word with 0001 as the first bit. Be aware that CC-Type=0x01 is advertised only if the control word is enabled in an L2VPN XC group or in a L2VPN or VPLS profile with the `control-word` command.
  - CC-Type=0x02—Router Alert Label
  - CV-Type=0x02—LSP ping

  Note: If the control word is not enabled in an L2VPN XC group or in an L2VPN or VPLS profile, then only CC-Type=0x02 and CV-Type=0x02 are advertised.

- During PW setup, LDP engages in the following VCCV capability negotiation:
  - If only CC-Type=0x01 is received from the peer, the PW uses CC-Type=0x01 for VCCV.
  - If only CC-Type=0x02 is received from the peer, the PW uses CC-Type=0x02 for VCCV
  - If both CC-Type=0x01 and CC-Type=0x02 are received from the peer, the PW uses CC-Type=0x01 for VCCV. Be aware that CC-Type=0x01 takes precedence over CC-Type=0x02.
  - If no VCCV capability is received from the peer, the PW is set up without VCCV capabilities.

- When LM processes are restarted, the PW objects within the SEOS are recreated with the existing VCCV capabilities. The VCCV capabilities of the PWs that are being recreated are not renegotiated.

- When the primary XCRP fails over to the backup XCRP, PWs are recreated from the synchronized shared memory. The VCCV capabilities of those PWs are maintained, and no VCCV capability negotiation occurs.

Note: If the PW is negotiated with only VCCV CC Type 2 for MPLS PSNs, VCCV traffic may not follow the same path as the data traffic if ECMP is applied to the outer LSP label.

1.3.3 PW Connectivity and VCCV Capability Verification

When end-user packets do not reach their destination, you can verify the connectivity of the PWs associated with the target L2VPN or L2VPN instance. Use the **ping mpls pw** *pw-id* command to verify connectivity of PWs in the following situations:

- When end-user packets do not reach their destination

- When the remote end experiences loss of packet or high RTT

If the SmartEdge router is on the middle of a PW and the PW failure is at that SmartEdge router, the SmartEdge router responds to the ping and indicates that it is not the egress point for the PW. The ping response helps the administrator identify the point of failure.

You can also use the **ping mpls pw pw-id** command on the data plane to verify whether a target PW is VCCV capable. If a PW is not VCCV capable, the following error message is displayed and the operation is terminated:

```
Pseudo-wire is not VCCV capable.  Ping aborted.
```

If the PW is VCCV capable, the VCCV ping retrieves the following information from the PW:

- PW encapsulation details

- Next-hop and adjacency details

Note: When using the **ping mpls pw pw-id** command to verify VCCV capabilities, do not include the **control-plane** keyword in the command string.

Note: If a router at the remote end of the ping is running a version of the SmartEdge OS earlier than Release 6.1.3, you must include the **send-mode** and **control-plane** keywords in the **ping mpls pw pw-id** command on the local router to ensure that the ping is sent over the control plane. In releases prior to 6.1.3, the router could send pings only over the control plane.

To verify the remote and local VCCV capabilities for all L2VPN LDP PWs currently configured on your system, use the **`show ldp l2vpn fec detail`** command in any mode.

2 Configuration and Operations Tasks

Configuration of VPLS in a SmartEdge router context requires the following basic steps:

1. Setup of the MPLS core background

   The MPLS core must be up and running before configuring VPLS. For instructions on configuring MPLS, see *Configuring MPLS*.

2. Setup of Label Distribution Protocol (LDP) between PE peers

   LDP targeted discovery must be enabled between PE peers before configuring VPLS. For more information on configuring LDP targeted discovery, see “Targeted LDP” in *Configuring LDP*.

3. Setup of a VPLS profile in which the IP addresses of the VPLS neighbors must be entered

   Detailed instructions and notes are found in Section 2.2 on page 10.

4. Setup of the VPLS-enabled bridge

   For procedures to troubleshoot issues with VPLS, see *Troubleshooting VPLS*.

   Detailed configuration instructions are found in Section 2.3 on page 14. You must include the following in this step:

   - The name of the VPLS profile (from step 3)

   - The pseudowire IDs that pass through the LAN to the other routers in the MPLS backbone

   - The names of the bridging-capable interfaces bound to CE devices that use the VPLS bridge

Note: Additional configuration options might be required depending on the application for which you are configuring VPLS.

2.1 Configuring a Bridge Profile

You can assign a named bridge profile to a neighbor. When the subscriber circuit is bound to a bridged interface, the attribute values in the named bridge profile assigned to the neighbor override those in the default bridge profile for the circuit, unless the circuit is also assigned a named bridge profile.

The configuration of bridge profiles is described in the "*Configure Bridge Profiles*" section of the *Configuring Bridging* document.

2.2 Configuring a VPLS Profile

A VPLS profile contains one or more neighbors, with each neighbor defining the attributes necessary to establish a separate peer instance (pseudowire) to a remote PE device. When a VPLS profile is assigned to a VPLS-enabled bridge, the bridge uses the neighbors in the profile to establish the peer instances and enable bridging over the pseudowires.

To configure a VPLS profile (with one or more neighbors), perform the tasks described in Table 1. Enter all commands in VPLS profile neighbor configuration mode, unless otherwise noted.

Table 1 Configuring a VPLS Profile

Task	**Root Command**	**Notes**
Create a new VPLS profile, or select an existing one for modification, and enter VPLS profile configuration mode.	*vpls profile*	Enter this command in global configuration mode. VPSL profiles are used to configure one or more neighbors to which a VPLS instance can establish peering connections. All neighbors configured in a VPLS profile are referenced by the VPLS profile name. The VPLS profile name is unique in the system. The VPLS profile is referenced from the VPLS instance configuration. Multiple VPLS instances can apply (share) the same VPLS profile. If a profile is updated, all applied instances use the changed attributes. Conflicts arising due to the updated VPLS profile in the VPLS instances do not result in rejection of the VPLS profile or the updates; the individual VPLS instances handle these conditions.
Optional. Enable SNMP trap notifications per VPLS profile.	*snmp trap*	Enables trap notifications for cross-connect state change events.

Table 1 Configuring a VPLS Profile

Task	Root Command	Notes
Create a new neighbor, or select an existing one for modification, and enter VPLS profile neighbor configuration mode.	*neighbor (VPLS)*	Enter this command in VPLS profile configuration mode. The neighbor is identified by the IP address of the remote PE device. It is used with the pseudowire ID from the VPLS instance configuration to establish a pseudowire between the local and remote PE devices. Multiple peering sessions (created by VPLS profiles) can be established to the same PE device; different profiles can reference the same remote PE IP address.
Assign an existing named bridge profile to the neighbor.	*bridge profile*	Bridge profiles can be assigned to subscribers and VPLS neighbors associated with bridges. When the subscriber circuit is bound to a bridged interface, the attribute values in the named bridge profile assigned to the VPLS neighbor override those in the default bridge profile for the circuit, unless the circuit is also assigned a named bridge profile. For more information about this command, see *Configuring Bridging*.
Optional. Enable the inclusion of a control word in the packet header that enables or disables the inclusion of incremental sequence numbers that ensure disassembled packets are reassembled properly.	*control-word* [**sequence-number** [**zero**]]	Include the optional **sequence-number** keyword to enable sequencing support on all packets. Include the optional **zero** keyword to disable sequencing support on all packets
Enable circuit statistics for VPLS circuits.	*counters (VPLS)*	When enabled, packet receive and transmit statistics are collected for each pseudowire circuit associated with this neighbor. Use the **no** form of this command to disable circuit statistics for VPLS circuits.

Table 1 Configuring a VPLS Profile

Task	Root Command	Notes
Associate a description with the neighbor.	*description (VPLS profiles)*	This command does not affect the neighbor and is used only as a comment in the configuration. The neighbor is identified by the IP address of the remote PE device.
Set the local mode of operation for the neighbor connection.	*local-mode*	This command applies only if a spoke connection type is configured for the neighbor. With a spoke connection type, one end of the connection must be set to MTU-s mode, and the other must be set to PE-rs mode. For proper VPLS operation, ensure that the local mode at both ends is set correctly.
Configure a native VLAN tag for transporting an untagged 802.1Q permanent virtual circuit (PVC) traffic across a pseudowire.	*native-vlan-tag*	The native VLAN tag value is configurable on the SmartEdge router to enable interoperability with the native VLAN tag used by other devices in the network. When the native VLAN tag is configured for a pseudowire instance: • All untagged ingress packets are prepended with the configured native VLAN tag. • For all ingress packets with a VLAN tag value of 0, the tag value is rewritten to the configured native VLAN tag value. The original dot1q bits are not preserved. • At egress, when a packet is received over a pseudowire, the VLAN tag is removed if its value matches the native VLAN tag value associated with the pseudowire. If the pseudowire is configured with a different native VLAN tag value, or is not configured, then the packet retains its VLAN tag. Only one native VLAN tag per pseudowire is supported.

Table 1 Configuring a VPLS Profile

Task	Root Command	Notes
Set the connection type used between the local and remote PE devices.	*pe-type*	Currently, hub-and-spoke connection types are supported. For proper VPLS peering, both ends of the peer must be configured with the same connection type.
Specify the next-hop attribute for a VPLS neighbor.	*preferred-nhop*	This attribute dictates that packets are forwarded to the specified next-hop (if that next-hop is available).
Specify the pseudowire encapsulation type.	*pw-encap*	You can specify Ethernet or Ethernet VLAN encapsulation.
Configure pseudowire labels for a static pseudowire.	*pw-label*	When the pseudowire labels are configured, the pseudowire is not signaled using a targeted LDP session to the neighbor. Instead, a static mapping for the pseudowire is created using the specified pseudowire labels. A pseudowire label can be used only once. Trying to configure a pseudowire label that is already in use causes the **`pw-label`** command to be rejected. Pseudowire labels must be configured on both ends of the VPLS peering session for the static pseudowire to operate properly. Static pseudowires (inner tunnels) can be configured in either static or signaled outer tunnels, including static, LDP, and RSVP LSPs and GRE tunnels. When the outer tunnel is broken or when no next hop to the peer exists, the static pseudowire is marked down, and a standby pseudowire is used if it has been configured. MAC flush TLVs sent using the **`clear vpls mac-flush`** command (in exec mode) can be sent over both signaled and static pseudowires. Use the **`no`** form of this command to delete the pseudowire labels.

Table 1 *Configuring a VPLS Profile*

Task	Root Command	Notes
Set the maximum transmission unit (MTU) for a static pseudowire.	*pw-mtu*	
Assign a spanning-tree profile to the neighbors.	*spanning-tree-profile* `profile-name`	
Specify that the L2TPv3 tunnel uses static encapsulation and provide the session ID and cookie for the VPLS pseudowire.	*static encapsulation* l2tpv3	Use this command only for the scenario described in Section 2.6 on page 19.
Enable a neighbor as a standby neighbor for a primary neighbor.	*standby-for*	A neighbor can serve as a standby for only one primary neighbor. This method of configuring a standby neighbor to reference a primary neighbor allows for establishing the primary and standby pseudowires using independent sets of attributes. Before a standby neighbor can be enabled, the following conditions must be met: • A spoke connection type must be set for the neighbor. • Local mode must be set to MTU-s. • No other standby neighbor in the VPLS profile can reference the same primary neighbor IP address.

2.3 Configuring a VPLS-Enabled Bridge

A VPLS-enabled bridge is used to establish peer instances to neighbors.

To configure a VPLS-enabled bridge, perform the tasks described in Table 2. Enter all commands in VPLS configuration mode, unless otherwise noted.

Table 2 *Configuring a VPLS-Enabled Bridge*

Task	Root Command	Notes
Create a bridge, or select one for modification, and enter bridge configuration mode.	*bridge*	Enter this command in context configuration mode. For more information about this command, see *Configuring Bridging*.

Table 2   Configuring a VPLS-Enabled Bridge

Task	Root Command	Notes
Configure the parameters of the bridge, such as the `aging type` parameter.	Numerous commands	For more information about the commands that configure bridges, see the *Create a Named Bridge* section in the *Configuring Bridging* document. Enter these commands in bridge configuration mode.
Enable VPLS on a bridge and enter VPLS configuration mode.	*vpls*	Enter this command in bridge configuration mode.
Optional: Disable the operation of an enabled VPLS instance.	*disable (VPLS)*	If the VPLS instance has been disabled, you can use the **`no`** form of this command to enable it.
Apply an existing VPLS profile to a VPLS instance.	*profile (VPLS)*	When a VPLS profile is applied, a VPLS peer instance is created for each neighbor defined in the profile, and a pseudowire connection is established using the attributes defined for the neighbor. A VPLS profile must be configured using the **`vpls profile`** command (in global configuration mode) before it can be applied. Multiple VPLS profiles can be applied to the same VPLS instance. If two or more profiles reference the same neighbor (same IP address), then the neighbor from the first profile is used. The same profile cannot be applied multiple times, even if the pseudowire IDs are different.

Table 2 Configuring a VPLS-Enabled Bridge

Task	Root Command	Notes
Configure a default pseudowire number for use with all the pseudowires signaled by the VPLS instance.	*pw-id*	The default pseudowire number is used for VPLS profiles that do not have a pseudowire ID (number or name) specified. Remote PE devices use the pseudowire ID and the local IP address to identify the pseudowire and the associated VPLS instance. A VPLS instance can have only one default pseudowire ID number. If a default pseudowire ID number exists for a VPLS instance, and a new pseudowire ID number is configured, the new pseudowire ID replaces the previous pseudowire ID.
Send a MAC withdrawal message to VPLS hub peers when the spoke pseudowire state on the PE-RS node changes to active state.	*neighbor mac-flush*	Enabling this option improves MAC forwarding convergence time.

2.4 Configuring a Pseudowire-to-RSVP LSP Tunnel

The SmartEdge router supports the mapping of traffic from a PW carrying traffic from a given VPLS instance to a specific RSVP tunnel. You can administratively select an RSVP tunnel to carry traffic that is sent to a specific neighbor. To configure the mapping of traffic from a PW carrying traffic from a given VPLS instance to a specific RSVP tunnel, perform the tasks described in Table 3.

Table 3 Configuinge Pseudowire-to-RSVP LSP Tunnel

Task	Root Command	Notes
Select an existing VPLS profile, and enter VPLS profile configuration mode.	*vpls profile*	Enter this command in global configuration mode. VPSL profiles are used to configure one or more neighbors to which a VPLS instance can establish peering connections. All neighbors configured within a VPLS profile are referenced by the VPLS profile name. The VPLS profile name is unique in the system.

Table 3   Configuinge Pseudowire-to-RSVP LSP Tunnel

Task	Root Command	Notes
Select an existing neighbor for modification, and enter VPLS profile neighbor configuration mode.	*neighbor (VPLS)*	Enter this command in VPLS profile configuration mode. The neighbor is identified by the IP address of the remote PE device. It is used along with the pseudowire ID from the VPLS instance configuration to establish a pseudowire between the local and remote PE devices. Multiple peering sessions (created by VPLS profiles) can be established to the same PE device; different profiles can reference the same remote PE IP address.
Map traffic exiting this VPLS to a specific RSVP tunnel	*tunnel lsp*	Enter this command in VPLS profile neighbor configuration mode. Use the **`tunnel lsp`** command in VPLS profile neighbor configuration mode to map traffic exiting a VPLS to a specific RSVP tunnel (specified with the ***`tunnel-name`*** argument). By default, the exiting PW traffic is carried on any available tunnel.

2.5 Configuring RSVP Object Tracking

This section provides information for configuring RSVP object tracking on a SmartEdge router in a hub-and-spoke area.

Before you can configure RSVP object tracking in a hub-and-spoke area, you must perform the following tasks:

- You must have VPLS with RSVP enabled in the appropriate hub-and-spoke area of your network.
- You need to have configured all the required pseudowires between the access nodes and the aggregation nodes.
- You need to have configured all of the required LSP tunnels between the MTUs and the access nodes.

Consider the following restrictions before configuring RSVP object tracking in a hub-and-spoke area:

- Backup LSPs cannot be configured to track objects.
- Only RSVP LSPs can monitor tracked objects.
- The maximum number of interfaces that can be tracked by a single object is 10.
- The maximum number of LSPs that can track a single RSVP tracking object is 50.

To configure RSVP object tracking in a hub-and-spoke area, perform the tasks described in Table 4.

Table 4 Configure RSVP Object Tracking

#	Task	Root Command	Notes
1.	Enter RSVP router configuration mode.	*router rsvp*	Enter this command in context configuration mode.
2.	Create a tracking object and enter RSVP tracking object configuration mode.	*track* ***object-name***	Replace the ***object-name*** argument with a name that identifies the tracking object.
3.	Specify an interface to be tracked by this object.	*interface* ***if-name***	Replace the ***if-name*** argument with the name of an interface you want to track with this object.
4.	Repeat step 3 to continue adding interfaces to the object you created in step 2.		
5.	Exit RSVP tracking object configuration mode and enter RSVP router configuration mode.	*exit*	—
6.	Enter RSVP LSP configuration mode for the specified RSVP LSP.	*lsp* ***lsp-name***	Replace the ***lsp-name*** argument with an existing LSP identifier. Use the **show rsvp lsp** command to see a list of all LSPs currently configured on your router.
7.	Configure the RSVP LSP to track the interfaces contained in the specified object.	*track* ***object-name***	Replace the ***object-name*** argument with the name of the tracking object whose interfaces you want this LSP to track.
8.	Repeat step 7 to configure more RSVP LSPs to track the interfaces contained in the specified object.		
9.	Verify that the object you created in step 2 contains the appropriate interface and is being tracked by the RSVP LSPs you specified in step 7 and step 8.	*show rsvp track* ***object-name***	Replace the ***object-name*** argument with the name of the tracking object you want to verify.

2.6 Configuring L2TPv3 Lite Tunnels on VPLS Pseudowires

This section describes the configuration of VPLS pseudowires to function as L2TPv3 lite tunnels that terminate in a VPLS instance. L2TPv3 lite tunnel service is a useful way to provide Layer 2 tunneling service over a non-MPLS based IP backhaul network.

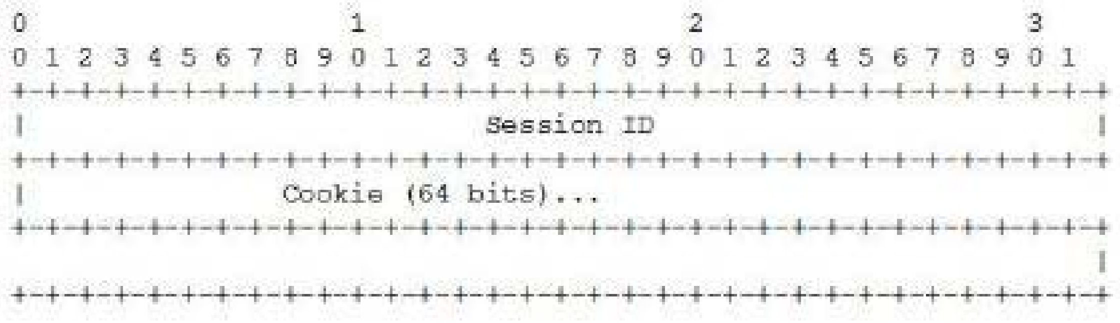
2.6.1 SmartEdge Router L2TPv3 Lite Tunnel Behavior

The configuration of the SmartEdge router for L2TPv3 lite tunnels encapsulated in VPLS pseudowires includes a step in which the `static encapsulation l2tpv3` command specifies a static L2TPv3 encapsulation consisting of a 32-bit session ID and 32-bit cookie.

Figure 4 shows how the SmartEdge router implements the L2TPv3 header with the embedded MPLS stack.

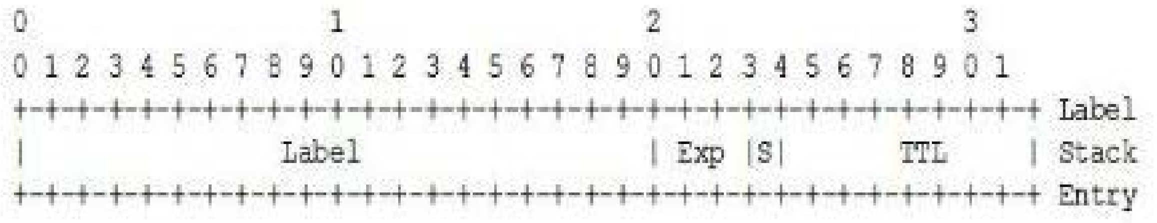
- When a packet is sent from the router over the VPLS pseudowire, it adds the L2TPv3 header to the inner MPLS packet. The L2TPv3 header is followed by a 32-bit MPLS label stack. When a non-MPLS remote peer receives a packet on the tunnel, it treats the packet as if it is an L2TPv3 packet sent with a 64-bit cookie.

- When an L2TPv3 packet is received by the SmartEdge router, it treats the packet as a L2TPv3 packet with a 64-bit cookie. The SmartEdge router strips off the session ID and 32-bit cookie and extracts the 32-bit MPLS label stack embedded in the lower 32 bits of the cookie. The embedded MPLS label stack then identifies the VPLS pseudowire corresponding to the tunnel, and the packet is injected into the corresponding VPLS instance.



```
0                   1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Session ID                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Cookie (64 bits)...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 3 L2TPv3 Session Header Over IP (from RFC 3931)



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Label
|                Label                  | Exp |S|       TTL     | Stack
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+ Entry
```

Figure 4 Embedded MPLS label stack (lower 32-bits of the 64-bit cookie)

2.6.2 Requirements and Restrictions (L2TPv3 Tunnels on VPLS Pseudowires)

- Only statically configured L2TPv3 tunnels (RFC 3931) are supported, and only the data plane specifications recommended in RFC 3391 are supported. The signal plan specifications are not supported.

- If you configure a large number of L2TPv3 tunnels, the chassis bootup and switchover time are slowed.

- When QoS propagation is enabled for static L2Tpv3 encapsulation, the MPLS EXP values are updated as well. This can cause remote peers that validate the cookie to drop packets. As a workaround, configure class maps to set the EXP bits to 0.

2.6.3 Example (L2TPv3 Tunnel on a VPLS Pseudowire with QoS)

The following example shows the configuration of `rock1200` which is part of a VPLS bridge that includes two other PE routers. The VPLS pseudowire between `rock1200` and `jazz` has an L2TPv3 tunnel configured in it. The pseudowire ID is `20`. The L2TPv3 tunnel is statically configured in VPLS pseudowire `20` with a session-id of `30` for incoming packets and `30` for outgoing packets.

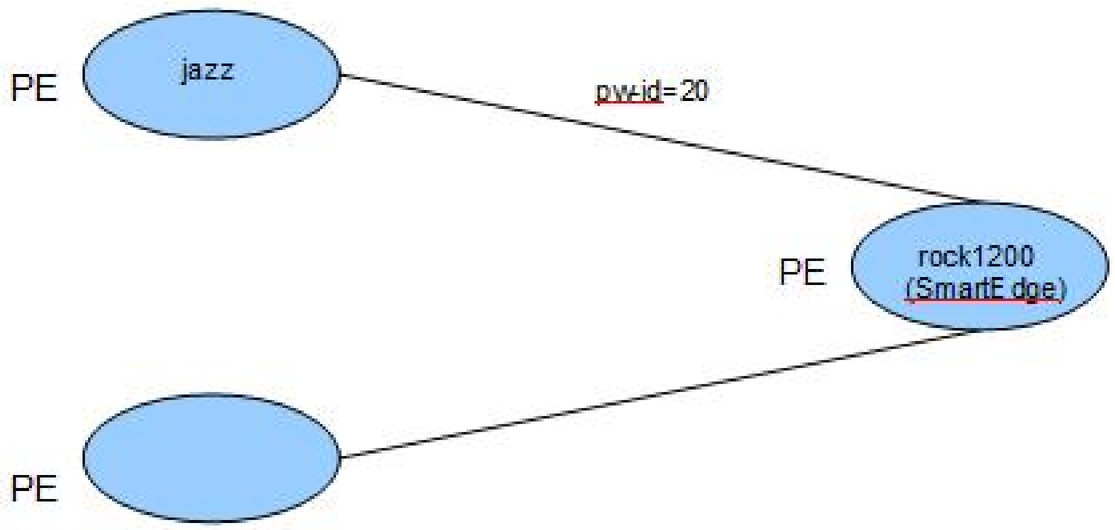


Figure 5 L2TPv3 Tunnel on a VPLS Pseudowire with QoS

```
[local]rock1200#show config
Building configuration...

Current configuration:
!
!  Configuration last changed by user 'test' at Wed May 26 12:07:04 2010
!
!
service multiple-contexts
!
!
!
! VPLS global configuration
vpls profile P11
  neighbor 67.1.1.2
    pw-label in 5001 out 5001
    static encapsulation l2tpv3 session-id in 30 out 30
    static encapsulation l2tpv3 cookie out 32 0 15
! Additional neighbors can be configured
!
qos class-map pd-to-exp mpls out
 qos 1 to mpls 0
 qos 2 to mpls 0
 qos 3 to mpls 0
 qos 4 to mpls 0
 qos 5 to mpls 0
 qos 6 to mpls 0
 qos 7 to mpls 0
 qos 8 to mpls 0
 qos 9 to mpls 0
 qos 10 to mpls 0
 qos 11 to mpls 0
 qos 12 to mpls 0
 qos 13 to mpls 0
 qos 14 to mpls 0
 qos 15 to mpls 0
 qos 16 to mpls 0
 qos 17 to mpls 0
 qos 18 to mpls 0
 qos 19 to mpls 0
 qos 20 to mpls 0
 qos 21 to mpls 0
 qos 22 to mpls 0
 qos 23 to mpls 0
 qos 24 to mpls 0
 qos 25 to mpls 0
 qos 26 to mpls 0
 qos 27 to mpls 0
 qos 28 to mpls 0
 qos 29 to mpls 0
 qos 30 to mpls 0
 qos 31 to mpls 0
 qos 32 to mpls 0
 qos 33 to mpls 0
 qos 34 to mpls 0
 qos 35 to mpls 0
 qos 36 to mpls 0
 qos 37 to mpls 0
 qos 38 to mpls 0
 qos 39 to mpls 0
 qos 40 to mpls 0
 qos 41 to mpls 0
 qos 42 to mpls 0
 qos 43 to mpls 0
 qos 44 to mpls 0
 qos 45 to mpls 0
 qos 46 to mpls 0
 qos 47 to mpls 0
 qos 48 to mpls 0
 qos 49 to mpls 0
 qos 50 to mpls 0
 qos 51 to mpls 0
 qos 52 to mpls 0
 qos 53 to mpls 0
 qos 54 to mpls 0
 qos 55 to mpls 0
 qos 56 to mpls 0
 qos 57 to mpls 0
 qos 58 to mpls 0
 qos 59 to mpls 0
 qos 60 to mpls 0
 qos 61 to mpls 0
 qos 62 to mpls 0
 qos 63 to mpls 0
!
qos class-map pd-to-ip ip out
!
!
!
!
context local
!
 no ip domain-lookup
!
 interface loopback_rock1200 loopback
  ip address 10.10.10.2/32
!
 interface mgmt
  ip address 10.18.17.103/24
!
 interface to_jazz
  ip address 192.168.1.2/24
  propagate qos to ip class-map pd-to-ip
 logging console
!
 ip soft-gre source 10.10.10.2
!
 router mpls
  propagate qos to mpls class-map pd-to-exp
  interface loopback_rock1200
  interface to_jazz
!
!
 administrator test encrypted 1 $1$........$kvQfdsjs0ACFMeDHQ7n/o.
   privilege start 15
!
!
 ip route 67.1.1.2/32 192.168.1.1
 ip route 155.53.0.0/16 10.18.17.254
 ip route 155.53.0.0/16 10.18.49.254
```

```
!
!
!
!
context yellow
!
 no ip domain-lookup
!
 interface bridge-if bridge
  bridge name A
 no logging console
!
 bridge A
  vpls
   profile P11 pw-id 20
!
!
!
!
!
! ** End Context **
 logging tdm console
 logging active
 logging standby short
!
!
!
!
port ethernet 7/1
! XCRP management ports on slot 7 and 8 are configured through 7/1
 no shutdown
 bind interface mgmt local
!
card ge-20-port 9
!
port ethernet 9/1
 no shutdown
 bind interface to_jazz local
!
 system hostname rock1200
 timeout session idle 9999
!
no service console-break
!
service crash-dump-dram
!
no service auto-system-recovery
!
end
```

2.6.4 Configuring L2TPv3 Tunnels on VPLS Pseudowires with QoS

This configuration procedure constructs a QoS-provisioned L2TPv3 tunnel on the VPLS pseudowire connection between a local SmartEdge router and a remote PE router.

The configuration steps are specific to the `rock1200` SmartEdge router in the example configuration in Section 2.6.3 on page 20; however, you can substitute in your own network specifics.

To configure L2TPv3 Tunnels on VPLS Pseudowires with QoS, do the following:

1. Configure an interface to the remote PE router (`jazz`) and a static route to the loopback address of the remote customer edge (CE) router (`jazz`). The remote CE router could be either directly connected or multiple hops away.

```
context local
 ip route 67.1.1.2/32 192.168.1.1
 interface to_jazz
  ip address 192.168.1.2/24
```

2. Enable MPLS routing in the `local` context of the SmartEdge router.

    - Configure a loopback interface address for the local MPLS router.
    - Enable soft-GRE for that router using the loopback address.
    - Enable MPLS on all interfaces that bind to the local context with the following commands.

```
context local
 interface loopback_rock1200 loopback
  ip address 10.10.10.2/32
!
 router mpls
  ip soft-gre source 10.10.10.2
  interface to_jazz
  interface loopback_rock1200
```

3. Enter VPLS profile configuration mode and do the following:

    a Configure the IP address in the neighbor statement with the loopback address of the remote PE router.

    b Configure static inbound and outbound pseudowire labels.

    c Use the `static encapsulation l2tpv3` command to configure the L2TPv3 session ID and cookie.

    The session ID for incoming and outgoing packets must be identical and consistent with the configuration on the remote peer PE router. The cookie configuration must also be consistent with the configuration of the peer PE router.

    All other attributes in VPLS profile configuration attributes are optional.

```
vpls profile  P11
 neighbor 67.1.1.2
  pw-label in 5001 out 5001
  static encapsulation l2tpv3 session-id in 30 out 30
  static encapsulation l2tpv3 cookie out 32 0 15
```

4. In any context other than the `local` context, configure a bridge instance and bridge interface and apply the VPLS profile to the bridge.

The local context is conventionally used for interfaces to PE routers, while other contexts are conventionally used for interfaces to CE routers. This configuration deviates from that convention.

```
context yellow
 interface bridge-if bridge
  bridge name A
 bridge A
  vpls
   profile P11 pw-id 20
```

5. Bind the attachment circuits and other VPLS peers (using VPLS profiles) to the bridge instance.

   This step is not illustrated.

6. Before enabling QoS propagation for static L2TPv3 encapsulation, you must first configure two class maps for outgoing packets transmitted by the `rock1200` SmartEdge router.

   - The first class map (`pd-to-exp`) sets the EXP bits values to 0. This prevents QoS from updating the MPLS EXP values as explained in Section 2.6.2 on page 20.

   - The second class map (`pd-to-ip`) updates the outer IP DSCP values.

```
qos class-map pd-to-exp mpls out
 qos 1 to mpls 0
 qos 2 to mpls 0
 qos 3 to mpls 0
 qos 4 to mpls 0
 qos 5 to mpls 0
 qos 6 to mpls 0
 qos 7 to mpls 0
 qos 8 to mpls 0
 qos 9 to mpls 0
 qos 10 to mpls 0
 qos 11 to mpls 0
 qos 12 to mpls 0
 qos 13 to mpls 0
 qos 14 to mpls 0
 qos 15 to mpls 0
 qos 16 to mpls 0
 qos 17 to mpls 0
 qos 18 to mpls 0
 qos 19 to mpls 0
 qos 20 to mpls 0
 qos 21 to mpls 0
 qos 22 to mpls 0
 qos 23 to mpls 0
 qos 24 to mpls 0
 qos 25 to mpls 0
 qos 26 to mpls 0
 qos 27 to mpls 0
 qos 28 to mpls 0
 qos 29 to mpls 0
 qos 30 to mpls 0
 qos 31 to mpls 0
 qos 32 to mpls 0
 qos 33 to mpls 0
 qos 34 to mpls 0
 qos 35 to mpls 0
 qos 36 to mpls 0
 qos 37 to mpls 0
 qos 38 to mpls 0
 qos 39 to mpls 0
 qos 40 to mpls 0
 qos 41 to mpls 0
 qos 42 to mpls 0
 qos 43 to mpls 0
 qos 44 to mpls 0
 qos 45 to mpls 0
 qos 46 to mpls 0
 qos 47 to mpls 0
 qos 48 to mpls 0
 qos 49 to mpls 0
 qos 50 to mpls 0
 qos 51 to mpls 0
 qos 52 to mpls 0
 qos 53 to mpls 0
 qos 54 to mpls 0
 qos 55 to mpls 0
 qos 56 to mpls 0
 qos 57 to mpls 0
 qos 58 to mpls 0
 qos 59 to mpls 0
 qos 60 to mpls 0
 qos 61 to mpls 0
 qos 62 to mpls 0
 qos 63 to mpls 0
 exit
!
qos class-map pd-to-ip ip out
```

7. Enable QoS propagation on the MPLS router and the interface to the VPLS peer.

```
context local
 router mpls
  propagate qos to mpls class-map pd-to-exp
!
 interface to_jazz
```

```
propagate qos to ip class-map pd-to-ip
```

2.7 Verifying PW Connectivity and VCCV Capabilities

To verify PW connectivity and VCCV capabilities, perform the tasks described in Table 5. Enter the **`ping mpls pw`** command in exec mode.

Table 5 Verifying PW Connectivity and VCCV Capabilities

Task	Root Command	Notes
Ping a particular PW to verify connectivity and display the VCCV capabilities of the PW.	*ping mpls pw* **`pw-id`** *`pw-num`* **`peer`** *`ip-addr`* [*`options`*]	Include the **`send-mode data-plane`** construct in the command to verify whether the PW is VCCV capable. Include any of the optional constructs, keywords, and arguments to configure various ping options as desired. If the ping is successful, various statistics and configuration information are displayed for the specified PW.

2.8 VPLS Operations

To manage VPLS functions, perform the appropriate tasks described in Table 6. Enter the **`show`** commands in any mode; enter the **`debug`** commands in exec mode.

Table 6 VPLS Operations Tasks

Task	Root Command
Clear circuit counters for VPLS circuits in the system.	*clear circuit counters vpls*
Reset VPLS peer connections on a specified VPLS bridge or profile.	*clear vpls*
Reset the counters for the VPLS peers on a specified VPLS bridge or profile.	*clear vpls counters*
Set the administrative state to admin down for VPLS peer connections on a specified VPLS bridge or profile, instead of resetting the peer connections.	*clear vpls disable*
Reset VPLS peers by sending a medium access control (MAC) flush type-length-value (TLV) over the pseudo-wires of VPLS peers on the specified VPLS bridge or profile to remove the MAC entries.	*clear vpls mac-flush*
Restart VPLS peer connections for the specified VPLS bridge or profile.	*clear vpls restart*

Table 6 VPLS Operations Tasks

Task	**Root Command**
Display packet counter information for VPLS circuits.	*show circuit counters vpls*
Display VPLS circuit information.	*show circuit vpls*
Display VPLS-enabled bridge information.	*show vpls*
Display VPLS peer information.	*show vpls peer*
Display VPLS profile information.	*show vpls profile*

3 Configuration Examples

The VPLS configuration examples provided in this section assume that the following conditions are true:

- MPLS core backbone configuration is up and running.

  For more information on configuring MPLS, see *Configuring MPLS*.

- LDP targeted discovery has been enabled between PE peers.

  For more information on configuring LDP targeted discovery, see *Targeted LDP* in *Configuring LDP*.

3.1 Bridge Profile

The following configuration example how to create two bridge profiles, **100Mbps-bc** and **120Mbps-mc**. The **100Mbps-bc** bridge profile sets a rate limit of 125 Mbps (12,500 kbps) for **broadcast** traffic on the VPLS pseudowire circuit to which this bridge profile is assigned. The **120Mbps-mc** bridge profile sets a rate limit of 150 Mbps (15,000 kbps) for **multicast** traffic on the VPLS pseudowire circuit to which this bridge profile is assigned. The attributes of these bridge profiles will be applied to VPLS neighbor configurations:

```
[local]Redback#config
[local]Redback(config)#bridge profile 100Mbps-bc
[local]Redback(config-bridge-profile)#broadcast rate-limit 12500000
[local]Redback(config-bridge-profile)#exit
[local]Redback(config)#bridge profile 120Mbps-mc
[local]Redback(config-bridge-profile)#multicast rate-limit 15000000
[local]Redback(config-bridge-profile)#end
```

3.2 VPLS Profile

The following configuration example shows how to create a VPLS profile, **vprofile1**, and two neighbors, **64.10.192.112** and **110.32.164.5**. The attributes from the bridge profile, **100Mbps-bc**, are applied to the neighbor given the description, **dallas-to-nyc**. The attributes from the bridge profile, **120Mbps-mc**, are applied to the neighbor given the description, **dallas-to-sfo**. The neighbor attributes in this bridge profile will be applied to VPLS-enabled bridge instance:

```
[local]Redback#config
[local]Redback(config)#vpls profile vprofile1
[local]Redback(config-vpls-profile)#neighbor 64.10.192.112
[local]Redback(config-vpls-profile-neighbor)#description dallas-to-nyc
[local]Redback(config-vpls-profile-neighbor)#bridge-profile 100Mbps-bc
[local]Redback(config-vpls-profile-neighbor)#exit
[local]Redback(config-vpls-profile)#neighbor 110.32.164.5
[local]Redback(config-vpls-profile-neighbor)#description dallas-to-sfo
[local]Redback(config-vpls-profile-neighbor)#bridge-profile 120Mbps-mc
[local]Redback(config-vpls-profile-neighbor)#end
```

3.3 VPLS-Enabled Bridge

The following configuration example shows how to create a VPLS-enabled bridge instance, **truecom.net**, configures a default pseudowire number, **100**, for this instance, and applies the attributes from the VPLS profile, **vprofile1**, to this instance:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge truecom.net
[local]Redback(config-bridge)#vpls
[local]Redback(config-vpls)#pw-id 100
[local]Redback(config-vpls)#profile vprofile1
[local]Redback(config-vpls)#end
```

3.4 Pseudowire-to-RSVP LSP Tunnel Traffic Mapping

The following example shows how to map traffic exiting a PW to a specific VPLS instance:

```
[local]Redback#config
[local]Redback(config)#vpls profile blue
[local]Redback(config-vpls-profile)#neighbor 110.32.164.5
[local]Redback(config-vpls-profile-neighbor)#tunnel lsp blue005
[local]Redback(config-vpls-profile-neighbor)#exit
[local]Redback(config-vpls-profile)#exit
[local]Redback(config)#context local
[local]Redback(config-ctx)#bridge local-x vpls
[local]Redback(config-bridge)#vpls profile blue
[local]Redback(config-bridge)#end
```

3.5 RSVP Object Tracking

The following example describes how to configure RSVP object tracking on the example network shown in Figure 2.

On the appropriate access node, create a tracking object called `san-jose-1` and add interfaces 1, 2, and 3 to `san-jose-1`:

```
[local]Redback#configure
[local]Redback(config)#context local
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#track san-jose-1
[local]Redback(config-rsvp-track_obj)#interface 1
[local]Redback(config-rsvp-track_obj)#interface 2
[local]Redback(config-rsvp-track_obj)#interface 3
```

On the access node, configure the RSVP LSP called `MTU1-AC1 to track the interfaces contained in the object called san-jose-1`:

```
[local]Redback#config
[local]Redback(config)#context local
[local]Redback(config-ctx)#router rsvp
[local]Redback(config-rsvp)#lsp MTU1-AC1
[local]Redback(config-rsvp-lsp)#track san-jose-1
[local]Redback(config-rsvp-lsp)#exit
[local]Redback(config-rsvp)#
```

After committing the configuration, verify that the tracking object is up and tracking the appropriate interfaces:

```
[local]Redback#show rsvp track san-jose-1
Track Object san-jose-1 is UP
   Number of members: 3      RSVP interface:1 is DOWN       RSVP interface:2 is DOWN
   RSVP interface:3 is UP    Tracked by 1 observers
    RSVP LSP:MTU1-AC1
```