7450 Ethernet Service Switch
7750 Service Router
7950 Extensible Routing System
Virtualized Service Router

Release 23.3.R1

Layer 2 Services and EVPN Guide

3HE 19221 AAAA TQZZA 01
Edition 01
March 2023

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

Table of contents

1 Getting started

1.1 About this guide

This guide describes Layer 2 service and EVPN functionality provided by the SR-series routers and presents examples to configure and implement various protocols and services.

This guide is organized into functional chapters and provides concepts and descriptions of the implementation flow, as well as Command Line Interface (CLI) syntax and command usage.

Note: Unless otherwise indicated, this guide uses classic CLI command syntax and configuration examples.

The topics and commands described in this document apply to the:

- 7450 ESS
- 7750 SR
- 7950 XRS
- Virtualized Service Router

For a list of unsupported features by platform and chassis, see the SR OS R23.*x*.R*x* Software Release Notes, part number 3HE 19269 000 *x* TQZZA.

Command outputs shown in this guide are examples only; actual displays may differ depending on supported functionality and user configuration.

Note: The SR OS CLI trees and command descriptions can be found in the following guides:

- *7450 ESS, 7750 SR, 7950 XRS, and VSR Classic CLI Command Reference Guide*
- *7450 ESS, 7750 SR, 7950 XRS, and VSR Clear, Monitor, Show, and Tools Command Reference Guide* (for both MD-CLI and Classic CLI)
- *7450 ESS, 7750 SR, 7950 XRS, and VSR MD-CLI Command Reference Guide*

Note: This guide generically covers 23.*x*.R*x* content and may contain some content that will be released in later maintenance loads. See the SR OS R23.*x*.R*x* Software Release Notes, part number 3HE 19269 000 *x* TQZZA, for information about features supported in each load of the Release 23.*x*.R*x* software.

1.2 Layer 2 services and EVPN configuration process

Table 1: Configuration process lists the tasks related to configuring and implementing Layer 2 Services and EVPN functionality.

This guide is presented in an overall logical configuration flow. Each section describes a software area and provides CLI syntax and command usage to configure parameters for a functional area.

Table 1: Configuration process

Area	Task	Section
VLL Services	Configure services for MPLS-TP	Services configuration for MPLS-TP
	Configure VCCV BFD	VCCV BFD support for VLL, spoke-SDP termination on IES and VPRN, and VPLS services
	Configure pseudowire switching	Pseudowire switching
	Configure a VLL service	Configuring a VLL service with CLI
Virtual Private LAN Service (VPLS)	Configure a VPLS service	Configuring a VPLS service with CLI
	VPLS service management	Service management tasks
Ethernet Virtual Private Networks (EVPNs)	Configure EVPN-VXLAN and EVPN-MPLS in the same VPLS service	Multi-Instance EVPN: Two instances of different encapsulation in the same VPLS/R-VPLS service
	Configure an EVPN service	Configuring an EVPN service with CLI

1.3 Conventions

This section describes the general conventions used in this guide.

1.3.1 Precautionary and information messages

The following information symbols are used in the documentation.

DANGER: Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.

WARNING: Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.

Caution: Caution indicates that the described activity or situation may reduce your component or system performance.

Note: Note provides additional operational information.

Tip: Tip provides suggestions for use or best practices.

1.3.2 Options or substeps in procedures and sequential workflows

Options in a procedure or a sequential workflow are indicated by a bulleted list. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform one of the listed options to complete the step.

Example: Options in a procedure

1. User must perform this step.
2. This step offers three options. User must perform one option to complete this step.
   - This is one option.
   - This is another option.
   - This is yet another option.

Substeps in a procedure or a sequential workflow are indicated by letters. In the following example, at step 1, the user must perform the described action. At step 2, the user must perform two substeps (a. and b.) to complete the step.

Example: Substeps in a procedure

1. User must perform this step.
2. User must perform all substeps to complete this action.
   a. This is one substep.
   b. This is another substep.

2 VLL services

2.1 Circuit emulation services

This section provides information about Circuit Emulation (Cpipe) services. Cpipe is supported for the 7450 ESS and 7750 SR only.

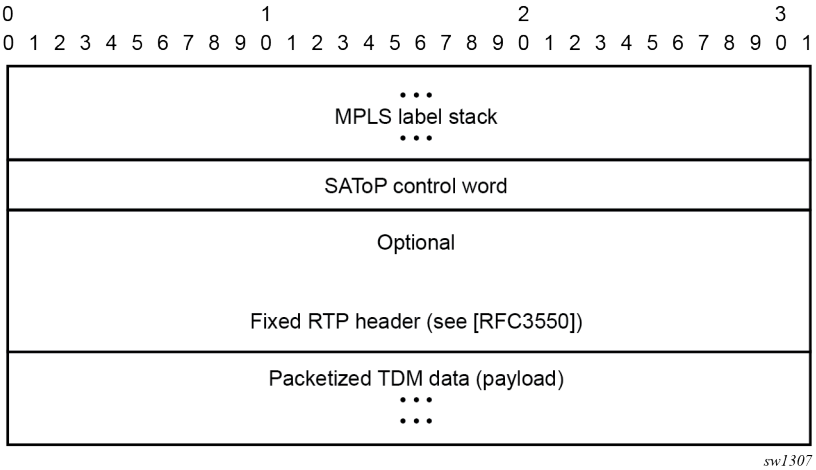
Note: Cpipe VLL is not supported in System Profile B. To determine if Cpipes are currently provisioned, use the **show service service-using cpipe** command before configuring profile B.

2.1.1 Circuit emulation modes

Two modes of circuit emulation are supported: unstructured and structured. Unstructured mode is supported for DS1 and E1 channels per RFC 4553, *Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)*. Structured mode is supported for N*64 kb/s circuits as per RFC 5086, *Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)*. Also, DS1, E1, and N*64 kb/s circuits are supported (per MEF8). TDM circuits are optionally encapsulated in MPLS or Ethernet as per the referenced standards in the following examples.

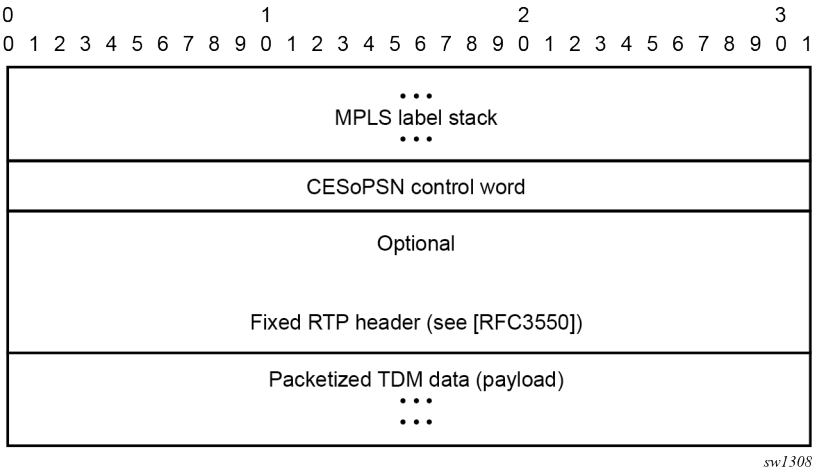
The following figure shows an example of RFC 4553 (SAToP) MPLS PSN encapsulation.

Figure 1: RFC 4553 (SAToP) MPLS PSN encapsulation



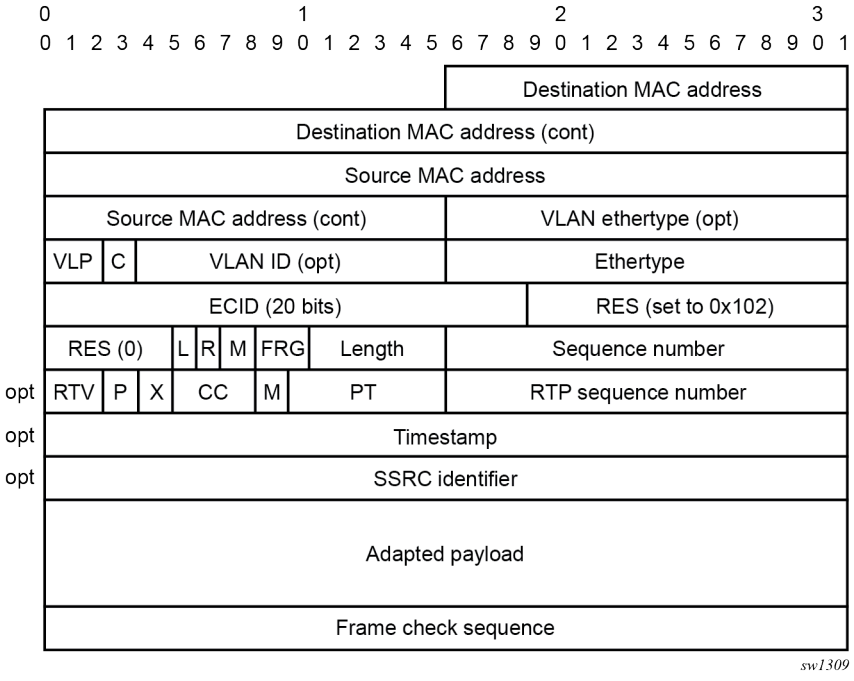
The following figure shows an example of CESoPSN packet format for an MPLS PSN.

Figure 2: CESoPSN packet format for an MPLS PSN



The following figure shows an example of MEF8 PSN encapsulation.

Figure 3: MEF8 PSN encapsulation



2.1.2 Circuit emulation parameters

2.1.2.1 Circuit emulation modes

All channels on the CES MDA are supported as circuits to be emulated across the packet network. Structure-aware mode is supported for N*64 kb/s channel groups in DS1 and E1 carriers. Fragmentation is not supported for circuit emulation packets (structured or unstructured).

For DS1 and E1 unstructured circuits, the framing can be set to unframed. When channel group 1 is created on an unframed DS1 or E1, it is automatically configured to contain all 24 or 32 channels, respectively.

N*64 kb/s circuit emulation supports basic and Channel Associated Signaling (CAS) options for timeslots 1 to 31 (channels 2 to 32) on E1 carriers and channels 1 to 24 on DS1 carriers. CAS in-band is supported; therefore, no separate pseudowire support for CAS is provided. CAS option can be enabled or disabled for all channel groups on a specific DS1 or E1. If CAS operation is enabled, timeslot 16 (channel 17) cannot be included in the channel group on E1 carriers. Control channel signaling (CCS) operation is not supported.

2.1.2.2 Absolute mode option

For all circuit emulation channels except those with differential clock sources, RTP headers in absolute mode can be optionally enabled (disabled by default). For circuit emulation channels that use differential clock sources, this configuration is blocked. All channel groups on a specific DS1 or E1 can be configured for the same mode of operation.

When enabled for absolute mode operation, an RTP header is inserted. On transmit, the CES IWF inserts an incrementing (by 1 for each packet) timestamp into the packets. All other fields are set to zero. The RTP header is ignored on receipt. This mode is enabled for interoperability purposes only for devices that require an RTP header to be present.

2.1.2.3 Payload size

For DS3, E3, DS1, and E1 circuit emulation, the payload size can be configurable in number of octets. The default values for this parameter are shown in Table 2: Unstructured payload defaults. Unstructured payload sizes can be set to a multiple of 32 octets and minimally be 64 octets. TDM satellite supports only unstructured payloads.

Table 2: Unstructured payload defaults

TDM circuit	Default payload size
DS1	192 octets
E1	256 octets

For N*64 kb/s circuits, the number of octets or DS1/E1 frames to be included in the TDM payload needs to be configurable in the range 4 to 128 DS1/E1 frames in increments of 1 or the payload size in octets. The default number of frames is shown in Table 3: Structured number of default frames with associated packet sizes. For the number of 64 kb/s channels included (N), the following number of default frames apply for no CAS: n=1, 64 frames; 2≤N≤ 4, 32 frames; 5≤N≤ 15, 16 frames; N≥16, 8 frames.

For CAS circuits, the number of frames can be 24 for DS1 and 16 for E1, which yields a payload size of N*24 octets for T1 and N*16 octets for E1. For CAS, the signaling portion is an additional ((N+1)/2) bytes,

where N is the number of channels. The additional signaling bytes are not included in the TDM payload size, although they are included in the actual packet size shown in Table 3: Structured number of default frames.

The full ABCD signaling value can be derived before the packet is sent. This occurs for every 24 frames for DS1 ESF and every 16 frames for E1. For DS1 SF, ABAB signaling is actually sent because SF framing only supports AB signaling every 12 frames.

Table 3: Structured number of default frames

Num timeslots	No CAS			DS1 CAS		E1 CAS	
	num-frames default	Default payload	Minimum payload	Payload (24 frames)	Packet size	Payload (16 frames)	Packet size
1	64	64	40	24	25	16	17
2	32	64	64	48	49	32	33
3	32	96	96	72	74	48	50
4	32	128	128	96	98	64	66
5	16	80	80	120	123	80	83
6	16	96	96	144	147	96	99
7	16	112	112	168	172	112	116
8	16	128	128	192	196	128	132
9	16	144	144	216	221	144	149
10	16	160	160	240	245	160	165
11	16	176	176	264	270	176	182
12	16	192	192	288	294	192	198
13	16	208	208	312	319	208	215
14	16	224	224	336	343	224	231
15	16	240	240	360	368	240	248
16	8	128	128	384	392	256	264
17	8	136	136	408	417	272	281
18	8	144	144	432	441	288	297
19	8	152	152	456	466	304	314
20	8	160	160	480	490	320	330

Num timeslots	No CAS			DS1 CAS		E1 CAS	
	num-frames default	Default payload	Minimum payload	Payload (24 frames)	Packet size	Payload (16 frames)	Packet size
21	8	168	168	504	515	336	347
22	8	176	176	528	539	352	363
23	8	184	184	552	564	368	380
24	8	192	192	576	588	384	396
25	8	200	200	—	—	400	413
26	8	208	208	—	—	416	429
27	8	216	216	—	—	432	446
28	8	224	224	—	—	448	462
29	8	232	232	—	—	464	479
30	8	240	240	—	—	480	495
31	8	248	248	—	—	—	—

Note: The num-frames DS1 CAS are multiples of 24; num-frames E1 is a multiple of 16.

2.1.2.4 Jitter buffer

For each circuit, the maximum receive jitter buffer is configurable. Packet delay from this buffer starts when the buffer is 50% full, to give an operational packet delay variance (PDV) equal to 75% of the maximum buffer size. The default value for the jitter buffer is nominally 5 ms. However, for lower-speed N*64 kb/s circuits and CAS circuits, the following default values are used to align with the default number of frames (and resulting packetization delay) to allow at least two frames to be received before starting to playout the buffer. The jitter buffer is at least four times the packetization delay. The following default jitter buffer values for structured circuits apply:

Basic CES (DS1 and E1):

N=1, 32 ms

2≤N≤4, 16 ms

5≤N≤15, 8 ms

N≥16, 5 ms

2.1.2.5 CES circuit operation

The circuit status can be tracked to be either up, loss of packets, or administratively down. Statistics are available for the number of in-service seconds and the number of out-of-service seconds when the circuit is administratively up.

Jitter buffer overrun and underrun counters are available by statistics and optionally logged while the circuit is up. On overruns, excess packets are discarded and counted. On underruns, all ones are sent for unstructured circuits. For structured circuits, all ones or a user-defined data pattern is sent based on configuration. Also, if CAS is enabled, all ones or a user-defined signaling pattern is sent based on configuration.

For each CES circuit, alarms can be optionally disabled/enabled for stray packets, malformed packets, packet loss, receive buffer overrun, and remote packet loss. An alarm is raised if the defect persists for 3 seconds, and cleared when the defect no longer persists for 10 seconds. These alarms are logged and trapped when enabled.

2.1.3 Services for transporting CES circuits

Each circuit can be optionally encapsulated in MPLS, Ethernet packets. Circuits encapsulated in MPLS use circuit pipes (Cpipes) to connect to the far-end circuit. Cpipes support either SAP spoke-SDP or SAP-SAP connections. Cpipes are supported over MPLS and GRE tunnels. The Cpipe default service MTU is set to 1514 bytes.

Circuits encapsulated in Ethernet can be selected as a SAP in Epipes. Circuits encapsulated in Ethernet can be SAP spoke-SDP connections or Ethernet CEM SAP-to-Ethernet SAP for all valid Epipe SAPs. Circuits requiring CEM SAP-to-CEM SAP connections use Cpipes. A local and remote EC-ID and far-end destination MAC address can be configurable for each circuit. The MDA MAC address is used as the source MAC address for these circuits.

For all service types, there are deterministic PIR=CIR values with class=EF parameters based on the circuit emulation parameters.

All circuit emulation services support the display of status of up, loss of packet (LOP), or admin down. Also, any jitter buffer overruns or underruns are logged.

Non-stop services are supported for Cpipes and CES over Epipes.

2.1.4 Network synchronization considerations

Each OC-3/STM-1 port can be independently configured to be loop-timed or node-timed. Each OC-3/STM-1 port can be configured to be a timing source for the node. TDM satellites only support node-timed mode.

Each DS-1 or E-1 channel without CAS signaling enabled can be independently configured to be loop-timed, node-timed, adaptive-timed, or differential-timed. Each DS-1 or E-1 channel with CAS signaling enabled can be independently configured to be loop-timed or node-timed. Adaptive timing and differential timing are not supported on DS-1 or E-1 channels with CAS signaling enabled. For the TDM satellite, each DS1/E1 channel can be loop-timed, node-timed, or differential-timed.

The adaptive recovered clock of a CES circuit can be used as a timing reference source for the node (ref1 or ref2). This is required to distribute network timing to network elements that only have packet connectivity to the network. One timing source on the MDA can be monitored for timing integrity. Both timing sources

can be monitored if they are configured on separate MDAs while respecting the timing subsystem slot requirements.

If a CES circuit is being used for adaptive clock recovery at the remote end (such that the local end is now an adaptive clock master), Nokia recommends setting the DS-1/E-1 to be node-timed to prevent potential jitter issues in the recovered adaptive clock at the remote device. This is not applicable to TDM satellites.

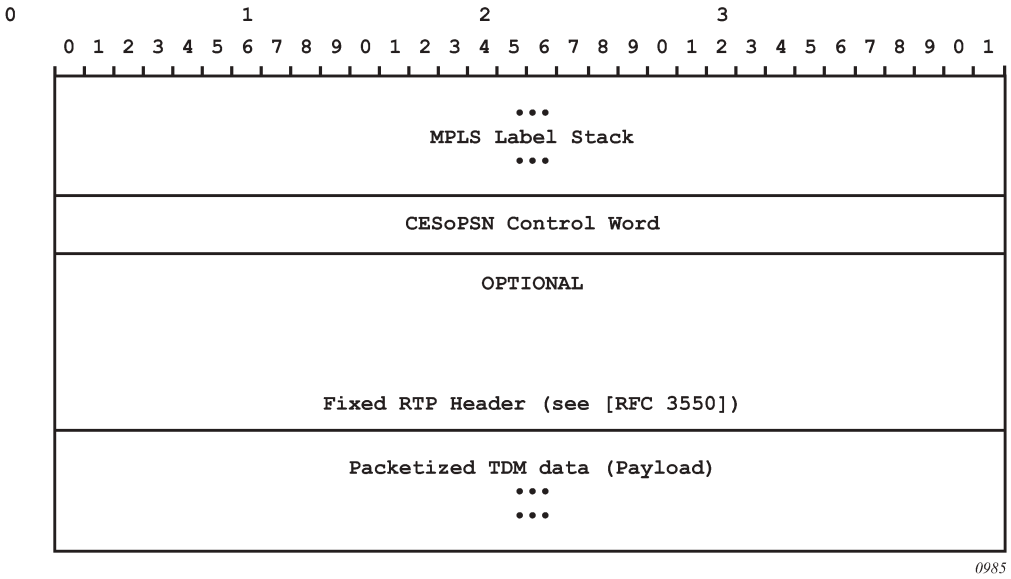
For differential-timed circuits, the following timestamp frequencies are supported: 103.68 MHz (for recommended >100 MHz operation), 77.76 MHz (for interoperability with SONET/SDH-based systems such as TSS-5) and 19.44 MHz (for Y.1413 compliance). TDM satellite supports only 77.76 MHz.

Adaptive and differential timing recovery must comply with published jitter and wander specifications (G.823, G.824, and G.8261) for traffic interfaces under typical network conditions and for synchronous interfaces under specified packet network delay, loss, and delay variance (jitter) conditions. The packet network requirements to meet the synchronous interface requirements are to be determined during the testing phase.

2.1.5 Cpipe payload

Figure 4: CESoPSN MPLS payload format shows the format of the CESoPSN TDM payload (with and without CAS) for packets carrying trunk-specific 64 kb/s service. In CESoPSN, the payload size is dependent on the number of timeslots used. This is not applicable to TDM satellite because only unstructured DS1/E1 is supported.

Figure 4: CESoPSN MPLS payload format



2.2 Ethernet pipe services

This section provides information about the Epipe service and implementation.

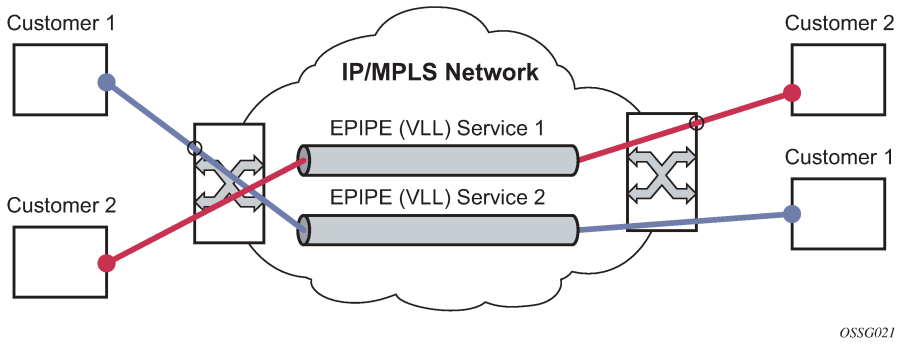
2.2.1 Epipe service overview

An Epipe service is the Nokia implementation of an Ethernet VLL based on the IETF "Martini Drafts" (*draft-martini-l2circuit-trans-mpls-08.txt* and *draft-martini-l2circuit-encapmpls-04.txt*) and the IETF Ethernet Pseudowire Draft (*draft-so-pwe3-ethernet-00.txt*).

An Epipe service is a Layer 2 point-to-point service where the customer data is encapsulated and transported across a service provider IP, MPLS, or Provider Backbone Bridging (PBB) VPLS network. An Epipe service is completely transparent to the customer data and protocols. The Epipe service does not perform any MAC learning. A local Epipe service consists of two SAPs on the same node, whereas a distributed Epipe service consists of two SAPs on different nodes. SDPs are not used in local Epipe services.

Each SAP configuration includes a specific port or channel on which service traffic enters the router from the customer side (also called the access side). Each port is configured with an encapsulation type. If a port is configured with an IEEE 802.1Q (referred to as dot1q) encapsulation, a unique encapsulation value (ID) must be specified.

Figure 5: Epipe/VLL service



2.2.2 Epipe service pseudowire VLAN tag processing

Distributed Epipe services are connected using a pseudowire, which can be provisioned statically or dynamically and is represented in the system as a spoke-SDP. The spoke-SDP can be configured to process zero, one, or two VLAN tags as traffic is transmitted and received; see Table 4: Epipe spoke-SDP VLAN tag processing: ingress and Table 5: Epipe-spoke SDP VLAN tag processing: egress for the ingress and egress tag processing. In the transmit direction, VLAN tags are added to the frame being sent. In the received direction, VLAN tags are removed from the frame being received. This is analogous to the SAP operations on a null, dot1q, and QinQ SAP.

The system expects a symmetrical configuration with its peer; specifically, it expects to remove the same number of VLAN tags from received traffic as it adds to transmitted traffic. When removing VLAN tags from a spoke-SDP, the system attempts to remove the configured number of VLAN tags. If fewer tags are found, the system removes the VLAN tags found and forwards the resulting packet.

Because some of the related configuration parameters are local and not communicated in the signaling plane, an asymmetrical behavior cannot always be detected and so cannot be blocked. With an asymmetrical behavior, a protocol extraction does not necessarily function as it would with a symmetrical configuration, resulting in an unexpected operation.

The VLAN tag processing is configured as follows on a spoke-SDP in an Epipe service:

- **zero VLAN tags processed**

  This requires the configuration of **vc-type ether** under the spoke-SDP, or in the related PW template.
- **one VLAN tag processed**

  This requires one of the following configurations:
  - **vc-type vlan** under the spoke-SDP or in the related PW template
  - **vc-type ether** and **force-vlan-vc-forwarding** under the spoke-SDP or in the related PW template
- **two VLAN tags processed**

  This requires the configuration of **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** under the spoke-SDP or in the related PW template.

The PW template configuration provides support for BGP VPWS services.

The following restrictions apply to VLAN tag processing:

- The configuration of **vc-type vlan** and **force-vlan-vc-forwarding** is mutually exclusive.
- **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** can be configured with the spoke-SDP signaled as either **vc-type ether** or **vc-type vlan**.
- The following are not supported with **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** configured under the spoke-SDP, or in the related PW template:
  - Multi-segment pseudowires.
  - PBB-Epipe services
  - force-vlan-vc-forwarding under the same spoke-SDP or PW template
  - Eth-CFM LM tests are NOT supported on UP MEPs when force-qinq-vc-forwarding is enabled.

Table 4: Epipe spoke-SDP VLAN tag processing: ingress and Table 5: Epipe-spoke SDP VLAN tag processing: egress describe the VLAN tag processing with respect to the zero, one, and two VLAN tag configuration described for the VLAN identifiers, Ethertype, ingress QoS classification (dot1p or DE), and QoS propagation to the egress (which can be used for egress classification or to set the QoS information, or both, in the innermost egress VLAN tag).

Table 4: Epipe spoke-SDP VLAN tag processing: ingress

Ingress (received on spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
VLAN identifiers	—	Ignored	Both inner and outer ignored
Ethertype (to determine the presence of a VLAN tag)	N/A	0x8100 or value configured under **sdp vlan-vc-etype**	Both inner and outer VLAN tags: 0x8100, or outer VLAN tag value configured under **sdp vlan-vc-etype** (inner VLAN tag value must be 0x8100)
Ingress QoS (dot1p/DE) classification	—	Ignored	Both inner and outer ignored
QoS (dot1p/DE) propagation to egress	Dot1p/DE=0	Dot1p/DE taken from received VLAN tag	Dot1p/DE taken as follows:

Ingress (received on spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
			• If the egress encapsulation is a Dot1q SAP, Dot1p/DE bits are taken from the outer received VLAN tag • If the egress encapsulation is QinQ SAP, the s-tag bits are taken from the outer received VLAN tag and the c-tag bits from the inner received VLAN tag The egress cannot be a spoke-sdp because **force-qinq-vc-forwarding** does not support multi-segment PWs.

Table 5: Epipe-spoke SDP VLAN tag processing: egress

Egress (sent on mesh or spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
VLAN identifiers (set in VLAN tags)	—	The tag is derived from one of the following: • the **vlan-vc-tag** value configured in PW template or under the spoke-SDP • value from the inner tag received on a QinQ SAP or QinQ spoke-SDP • value from the VLAN tag received on a dot1q SAP or spoke-SDP (with **vc-type vlan** or **force-vlan-vc-forwarding**) • value from the outer tag received on a qtag.* SAP • 0 if there is no service delimiting VLAN tag at the ingress SAP or spoke-SDP	The inner and outer VLAN tags are derived from one of the following: • vlan-vc-tag value configured in PW template or under the spoke-SDP: – If c-tag-c-tag is configured, both inner and outer tags are taken from the vlan-vc-tag value – If s-tag-c-tag is configured, only the s-tag value is taken from vlan-vc-tag • value from the inner tag received on a QinQ SAP for the c-tag-c-tag option and value from outer/inner tag received on a QinQ SAP for the s-tag-c-tag configuration option • value from the VLAN tag received on a dot1q SAP for

Egress (sent on mesh or spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
			the c-tag-c-tag option and value from the VLAN tag for the outer tag and zero for the inner tag • value from the outer tag received on a qtag.* SAP for the c-tag-c-tag option and value from the VLAN tag for the outer tag and zero for the inner tag • value 0 if there is no service delimiting VLAN tag at the ingress SAP or spoke-SDP Ethertype (set in VLAN tags).
Ethertype (set in VLAN tags)	—	0x8100 or value configured under **sdp vlan-vc-etype**	Both inner and outer VLAN tags: 0x8100, or outer VLAN tag value configured under **sdp vlan-vc-etype** (inner VLAN tag value is 0x8100)
Egress QoS (dot1p/DE) (set in VLAN tags)	—	The tag taken from the innermost ingress service delimiting tag can be one of the following: • The inner tag received on a QinQ SAP or QinQ spoke-SDP • value from the VLAN tag received on a dot1q SAP or spoke-SDP (with **vc-type vlan** or **force-vlan-vc-forwarding**) • value from the outer tag received on a qtag.* SAP	Inner and outer dot1p/DE: If **c-tag-c-tag** is configured, the inner and outer dot1p/DE bits are both taken from the innermost ingress service delimiting tag. It can be one of the following: • inner tag received on a QinQ SAP • value from the VLAN tag received on a dot1q SAP • value from the outer tag received on a qtag.* SAP • value 0 if there is no service delimiting VLAN tag at the ingress SAP
		0 if there is no service delimiting VLAN tag at the ingress SAP or spoke-SDP Note that neither the inner nor outer dot1p/DE values can be explicitly set.	If **s-tag-c-tag** is configured, the inner and outer dot1p/DE bits are taken from the inner and outer ingress service delimiting tag (respectively). They can be:

Egress (sent on mesh or spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
			• inner and outer tags received on a QinQ SAP • value from the VLAN tag received on a dot1q SAP for the outer tag and zero for the inner tag • value from the outer tag received on a qtag.* SAP for the outer tag and zero for the inner tag • value 0 if there is no service delimiting VLAN tag at the ingress SAP Note that neither the inner nor outer dot1p/DE values can be explicitly set.

Any non-service delimiting VLAN tags are forwarded transparently through the Epipe service. SAP egress classification is possible on the outermost customer VLAN tag received on a spoke-SDP using the **ethernet-ctag** parameter in the associated SAP egress QoS policy.

2.2.3 Epipe up operational state configuration option

By default, the operational state of the Epipe is tied to the state of the two connections that comprise the Epipe. If either of the connections in the Epipe are operationally down, the Epipe service that contains that connection is also operationally down. The operator can configure a single SAP within an Epipe that does not affect the operational state of that Epipe, using the optional **ignore-oper-state** command. Within an Epipe, if a SAP that includes this optional command becomes operationally down, the operational state of the Epipe does not transition to down. The operational state of the Epipe remains up. This does not change that the SAP is down and no traffic can transit an operationally down SAP. Removing and adding this command on the fly evaluates the operational state of the service, based on the SAPs and the addition or deletion of this command.

Service OAM (SOAM) designers may consider using this command if an operationally up MEP configured on the operationally down SAP within an Epipe is required to receive and process SOAM PDUs. When a service is operationally down, this is not possible. For SOAM PDUs to continue to arrive on an operationally up, MEP configured on the failed SAP, the service must be operationally up. Consider the case where an operationally up MEP is placed on a UNI-N or E-NNI and the UNI-C on E-NNI peer is shutdown in such a way that it causes the SAP to become operationally down.

Two connections must be configured within the Epipe; otherwise, the service is operationally down regardless of this command. The **ignore-oper-state** functionality only operates as intended when the Epipe has one ingress and one egress. This command is not to be used for Epipe services with redundant connections that provide alternate forwarding in case of failure, even though the CLI does not prevent this configuration.

Support is available on Ethernet SAPs configured on ports or Ethernet SAPs configured on LAG. However, it is not allowed on SAPs using LAG profiles or if the SAP is configured on a LAG that has no ports.

2.2.4 Epipe with PBB

A PBB tunnel may be linked to an Epipe to a B-VPLS. MAC switching and learning is not required for the point-to-point service. All packets that ingress the SAP are PBB encapsulated and forwarded to the PBB tunnel to the backbone destination MAC address. Likewise, all the packets that ingress the B-VPLS destined for the ISID are PBB de-encapsulated and forwarded to the Epipe SAP. A fully specified backbone destination address must be provisioned for each PBB Epipe instance to be used for each incoming frame on the related I-SAP. If the backbone destination address is not found in the B-VPLS FDB, packets may be flooded through the B-VPLSs.

All B-VPLS constructs may be used including B-VPLS resiliency and OAM. Not all generic Epipe commands are applicable when using a PBB tunnel.

2.2.5 Epipe over L2TPv3

The L2TPv3 feature provides a framework to transport Ethernet pseudowire services over an IPv6-only network without MPLS. This architecture relies on the abundance of address space in the IPv6 protocol to provide unique far-end and local-end addressing that uniquely identify each tunnel and service binding.

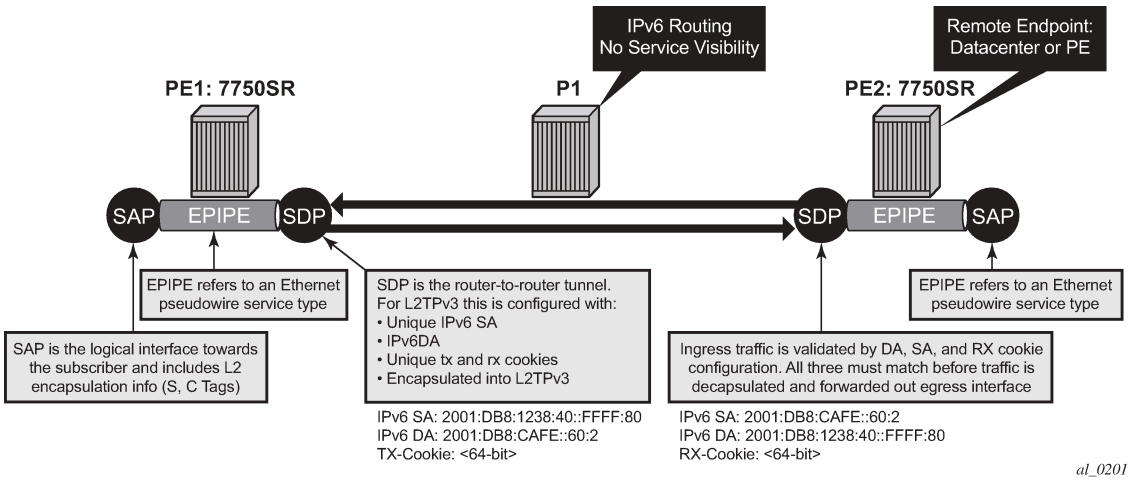
L2TPv3 provides the capability of transporting multiple Epipes (up to 16K per system), by binding multiple IPv6 addresses to each node and configuring one SDP per Epipe.

Because the IPv6 addressing uniqueness identifies the customer and service binding, the L2TPv3 control plane is disabled in this mode.

L2TPv3 is supported on non-12e 7750 SR, 7450 ESS, and 7950 XRS platforms.

ETH-CFM is supported for OAM services.

Figure 6: L2TPv3 SDP



2.2.6 VLL CAC

The VLL Connection Admission Control (CAC) is supported for the 7750 SR only and provides a method to administratively account for the bandwidth used by VLL services inside an SDP that consists of RSVP LSPs.

The service manager keeps track of the available bandwidth for each SDP. The SDP available bandwidth is applied through a configured booking factor. An administrative bandwidth value is assigned to the spoke-SDP. When a VLL service is bound to an SDP, the amount of bandwidth is subtracted from the adjusted available SDP bandwidth. When the VLL service binding is deleted from the SDP, the amount of bandwidth is added back into the adjusted SDP available bandwidth. If the total adjusted SDP available bandwidth is overbooked when adding a VLL service, a warning is issued and the binding is rejected.

This feature does not guarantee bandwidth to a VLL service because there is no change to the data path to enforce the bandwidth of an SDP by means such as shaping or policing of constituent RSVP LSPs.

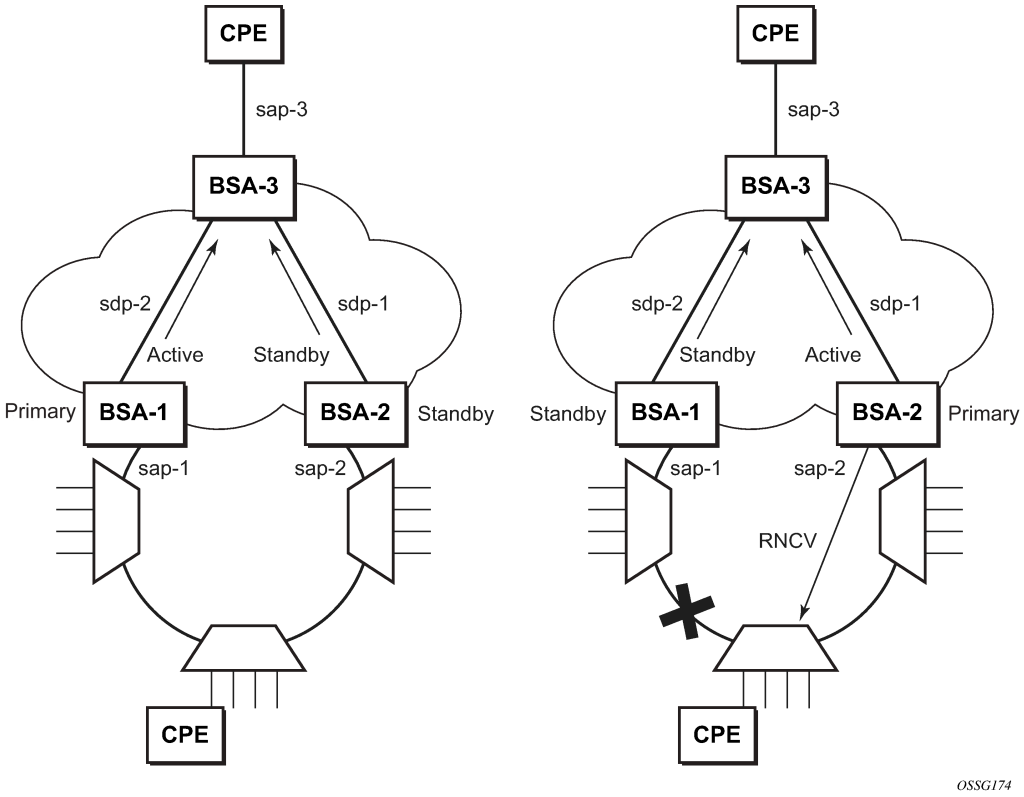
2.2.7 MC-Ring and VLL

To support redundant VLL access in ring configurations, the multi-chassis ring (MC-Ring) feature is applicable to VLL SAPs. A conceptual drawing of the operation is shown in Figure 7: MC-Ring in a combination with VLL service. The specific CPE that is connected behind the ring node has access to both BSAs through the same VLAN provisioned in all ring nodes. There are two SAPs (with the same VLAN) provisioned on both nodes.

If a closed ring status occurs, one of the BSAs becomes the primary BSA and signals an active status bit on the corresponding VLL pseudowire. Similarly, the standby BSA signals a standby status. With this information, the remote node can choose the correct path to reach the CPE. In case of a broken ring, the node that can reach the ring node, to which the CPE is connected by RNCV check, becomes the primary and signals corresponding status on its pseudowire.

The mapping of individual SAPs to the ring nodes is done statically through CLI provisioning. To keep the convergence time to a minimum, MAC learning must be disabled on the ring node so all CPE originated traffic is sent in both directions. If the status is operationally down on the SAP on the standby BSA, that part of the traffic is blocked and not forwarded to the remote site.

Figure 7: MC-Ring in a combination with VLL service



For further information about Multi-Chassis Ring Layer 2 (with ESM), see the *7450 ESS, 7750 SR, and 7950 XRS Advanced Configuration Guide*.

2.3 IP interworking VLL services

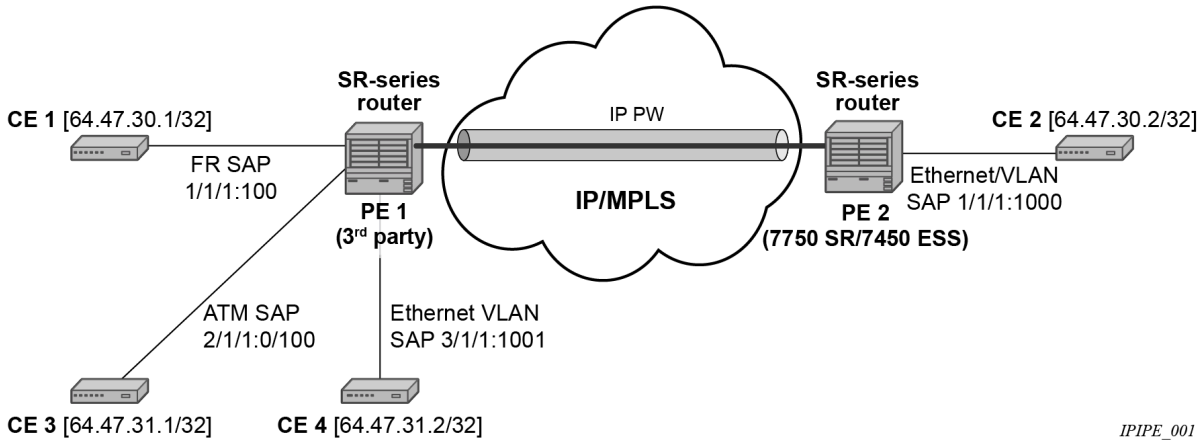
This section provides information about IP Interworking VLL (Ipipe) services.

2.3.1 Ipipe VLL

Figure 8: IP interworking VLL (Ipipe) provides an example of IP connectivity between a host attached to a point-to-point access circuit (FR, ATM, PPP) with routed PDU IPv4 encapsulation and a host attached to an Ethernet interface. Both hosts appear to be on the same LAN segment. This feature is supported on the 7450 ESS and 7750 SR with Ethernet access circuit only and enables service interworking between different link layer technologies when connecting over MPLS to a remote third party PE router implementation that supports the Ipipe VLL service with Frame Relay, ATM, PPP, or Cisco HDLC access circuits. A typical use of this application is in a Layer 2 VPN when upgrading a hub site to Ethernet while keeping the spoke sites with their existing Frame Relay or ATM IPv4 routed encapsulation.

Note: Ipipe VLL is not supported in System Profile B. To determine if Ipipes are currently provisioned, use the **show service service-using ipipe** command before configuring profile B.

Figure 8: IP interworking VLL (Ipipe)



Note: The Ipipe is a point-to-point Layer 2 service. All packets received on one SAP of the Ipipe are forwarded to the other SAP. No IP routing of customer packets occurs.

2.3.2 IP interworking VLL datapath

In Figure 8: IP interworking VLL (Ipipe), PE 2 is manually configured with both CE 1 and CE 2 IP addresses. These are host addresses and are entered in /32 format. PE 2 maintains an ARP cache context for each IP interworking VLL. PE 2 responds to ARP request messages received on the Ethernet SAP. PE 2 responds with the Ethernet SAP configured MAC address as a proxy for any ARP request for CE 1 IP address. PE 2 silently discards any ARP request message received on the Ethernet SAP for an address other than that of CE 1. Likewise, PE 2 silently discards any ARP request message with the source IP address other than that of CE 2. In all cases, PE 2 keeps track of the association of IP to MAC addresses for ARP requests it receives over the Ethernet SAP.

To forward unicast frames destined for CE 2, PE 2 needs to know the CE 2 MAC address. When the Ipipe SAP is first configured and administratively enabled, PE2 sends an ARP request message for CE 2 MAC address over the Ethernet SAP. Until an ARP reply is received from CE2, providing the CE2 MAC address, unicast IP packets destined for CE2 are discarded at PE2. IP broadcast and IP multicast packets are sent on the Ethernet SAP using the broadcast or direct-mapped multicast MAC address.

To forward unicast frames destined for CE 1, PE 2 validates the MAC destination address of the received Ethernet frame. The MAC address should match that of the Ethernet SAP. PE 2 then removes the Ethernet header and encapsulates the IP packet directly into a pseudowire without a control word. PE 1 removes the pseudowire encapsulation and forwards the IP packet over the Frame Relay SAP using RFC 2427, *Multiprotocol Interconnect over Frame Relay*, routed PDU encapsulation.

To forward unicast packets destined for CE1, PE2 validates the MAC destination address of the received Ethernet frame. If the IP packet is unicast, the MAC destination must match that of the Ethernet SAP. If the IP packet is multicast or broadcast, the MAC destination address must be an appropriate multicast or broadcast MAC address.

A PE does not flush the ARP cache unless the SAP goes administratively or operationally down. The PE with the Ethernet SAP sends unsolicited ARP requests to refresh the ARP cache every “T” seconds. ARP requests are staggered at an increasing rate if no reply is received to the first unsolicited ARP request. The value of T is configurable by the user through the **mac-refresh** command.

2.3.3 Extension to IP VLL for discovery of Ethernet CE IP address

VLL services provide IP connectivity between a host attached to a point-to-point access circuit with routed PDU encapsulation and a host attached to an Ethernet interface. Both hosts appear to be on the same IP interface.

In deployments where it is not practical for operators to obtain and configure their customer CE address, the following behaviors apply:

- A service comes up without prior configuration of the CE address parameter under both the SAP and the spoke-SDP.
- Operators rely solely on received ARP messages from the Ethernet SAP-attached CE device to update the ARP cache with no further check of the validity of the source IP address of the ARP request message and the target IP address being resolved.
- The LDP address list TLV signaling the learned CE IP address to the remote PE is supported. This is to allow the PE with the FR SAP to respond to an invFR ARP request message received from the FR-attached CE device.

2.3.3.1 VLL Ethernet SAP processes

The operator can enable the following CE address discovery processes by configuring the **ce-address-discovery** in the **config**>**service**>**ipipe** context.

- The service is brought up without the CE address parameter configured at either the SAP or the spoke-SDP.
- The operator cannot configure the **ce-address** parameter under the **config**>**service**>**ipipe**>**sap** or **config**>**service**>**ipipe**>**spoke-sdp** context when the **ce-address-discovery** in the **config**>**service**>**ipipe** context is enabled. Conversely, the operator is not allowed to enable the **ce-address-discovery** option under the Ipipe service if it has a SAP or spoke-SDP with a user-entered **ce-address** parameter.
- While an ARP cache is empty, the PE does not forward unicast IP packets over the Ethernet SAP but forwards multicast/broadcast packets. target IP address being resolved.
- The PE waits for an ARP request from the CE to learn both IP and MAC addresses of the CE. Both entries are added into the ARP cache. The PE accepts any ARP request message received over Ethernet SAP and updates the ARP cache IP and MAC entries with no further check of the source IP address of the ARP request message or of the target IP address being resolved.
- The 7450 ESS, 7750 SR, and 7950 XRS routers always reply to a received ARP request message from the Ethernet SAP with the SAP MAC address and a source IP address of the target IP address being resolved without any further check of the latter.
- If the router received an address list TLV from the remote PE node with a valid IP address of the CE attached to the remote PE, the router does not check the CE IP address against the target IP address being resolved when replying to an ARP request over the Ethernet SAP.
- The ARP cache is flushed when the SAP bounces or when the operator manually clears the ARP cache. This results in the clearing of the CE address discovered on this SAP. However, when the SAP comes up initially or comes back up from a failure, an unsolicited ARP request is not sent over the Ethernet SAP.

- If the Ipipe service uses a spoke-SDP, the router includes the address list TLV in the interface parameters field of the pseudowire Forwarding Equivalent Class (FEC) TLV in the label mapping message. The address list TLV contains the current value of the CE address in the ARP cache. If no address was learned, an address value of 0.0.0.0 must be used.
- If the remote PE included the address list TLV in the received label mapping message, the local router updates the remote PE node with the most current IP address of the Ethernet CE using a T-LDP notification message with the TLV status code set to 0x0000002C and containing an LDP address list. The notification message is sent each time an IP address different from the current value in the ARP cache is learned. This includes when the ARP is flushed and the CE address is reset to the value of 0.0.0.0.
- If the remote PE did not include the address list TLV in the received label mapping message, the local router does not send any notification messages containing the address list TLV during the lifetime of the IP pseudowire.
- If the operator disables the **ce-address-discovery** option under the VLL service, service manager instructs LDP to withdraw the service label and the service is shutdown. The pseudowire labels are only signaled and the service comes up if the operator re-enters the option again or manually enters the **ce-address** parameter under SAP and spoke-SDP.

2.3.4 IPv6 support on IP interworking VLL

The 7450 ESS, 7750 SR, and 7950 XRS nodes support both the transport of IPv6 packets and the interworking of IPv6 Neighbor discovery/solicitation messages on an IP Interworking VLL. IPv6 capability is enabled on an Ipipe using the **ce-address-discovery ipv6** command.

2.3.4.1 IPv6 Datapath operation

The IPv6 Datapath operation uses ICMPv6 extensions to automatically resolve IP address and link address associations. These are IP packets, as compared to ARP and invARP in IPv4, which are separate protocols and not based on IP packets. Manual configuration of IPv6 addresses is not supported on the IP Interworking VLL.

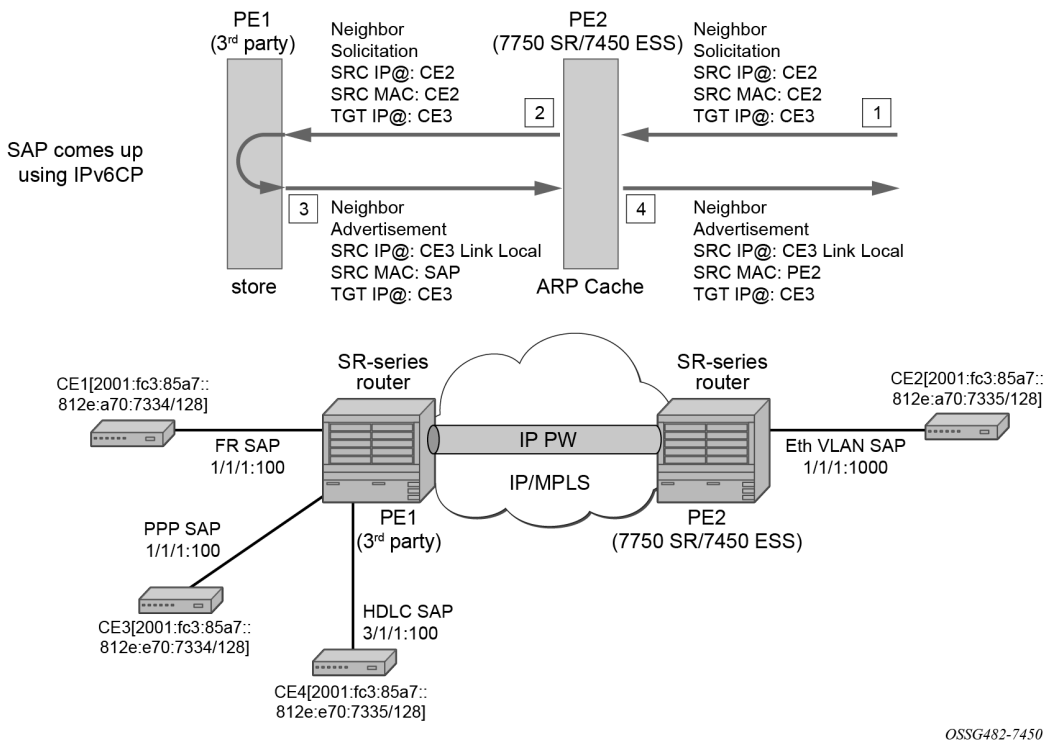
Each PE device intercepts ICMPv6 Neighbor Discovery (RFC 2461) packets, whether received over the SAP or over the pseudowire. The device inspects the packets to learn IPv6 interface addresses and CE link-layer addresses, modifies these packets as required according to the SAP type, then forwards them toward the original destination.

The PE device learns the IPv6 interface addresses for its directly-attached CE and other IPv6 interface addresses for the far-end CE. The PE device also learns the link-layer address of the local CE and uses it when forwarding traffic between the local and far-end CEs.

As with IPv4, the SAP accepts both unicast and multicast packets. For unicast packets, the PE checks that the MAC address/IP addresses are consistent with that in the ARP cache before forwarding; otherwise, the packet is silently discarded. Multicast packets are validated and forwarded. If more than one IP address is received per MAC address in a neighbor discovery packet, or if multiple neighbor discovery packets are received for a specific MAC address, the currently cached address is overwritten with the most recent value.

Figure 9: Data path for Ethernet CE to PPP attached CE shows the data path operation for IPv6 on an IP Interworking VLL between an Ethernet SAP on a PE router consisting of the 7750 SR or 7450 ESS and a PPP SAP on a third party PE router.

Figure 9: Data path for Ethernet CE to PPP attached CE



With reference to neighbor discovery between Ethernet and PPP CEs in Figure 9: Data path for Ethernet CE to PPP attached CE, the steps are as follows:

1. Ethernet-attached CE2 sends a Neighbor Solicitation message toward PE2 to begin the neighbor discovery process.
2. PE2 snoops this message, and the MAC address and IP address of CE2 is stored in the ARP cache of PE2 before forwarding the Neighbor Solicitation on the IP pseudowire to PE1.
3. PE1 snoops this message that arrives on the IP pseudowire and stores the IP address of the remote CE2. Because CE3 is attached to a PPP SAP, which uses IPv6CP to bring up the link, PE1 generates a neighbor advertisement message and sends it on the Ipipe toward PE2.
4. PE2 receives the neighbor advertisement on the Ipipe from PE1. It must replace the Layer 2 address in the neighbor advertisement message with the MAC address of the SAP before forwarding to CE2.

2.3.4.2 IPv6 stack capability signaling

The 7750 SR, 7450 ESS and 7950 XRS support IPv6 capability negotiation between PEs at the ends of an IP interworking VLL. Stack capability negotiation is performed if stack-capability-signaling is enabled in the CLI. Stack capability negotiation is disabled by default. Therefore, it must be assumed that the remote PE supports both IPv4 and IPv6 transport over an Ipipe.

A stack-capability sub-TLV is signaled by the two PEs using T-LDP so that they can agree on which stacks they should be using. By default, the IP pseudowire is always capable of carrying IPv4 packets. Therefore, this capability sub-TLV is used to indicate if other stacks need to be supported concurrently with IPv4.

The stack-capability sub-TLV is a part of the interface parameters of the pseudowire FEC. This means that any change to the stack support requires that the pseudowire be torn down and re-signaled.

A PE that supports IPv6 on an IP pseudowire must signal the stack-capability sub-TLV in the initial label mapping message for the pseudowire. For the 7750 SR, 7450 ESS, and 7950 XRS, this means that the stack-capability sub-TLV must be included if both the **stack-capability-signaling** and **ce-address-discovery ipv6** options are enabled under the VLL service.

If one PE of an IP interworking VLL supports IPv6, while the far-end PE does not support IPv6 (or ce-address-discovery ipv6 is disabled), the pseudowire does not come up.

If a PE that supports IPv6 (that is, stack-capability-signaling ipv6 is enabled) has already sent an initial label mapping message for the pseudowire, but does not receive a stack-capability sub-TLV from the far-end PE in the initial label mapping message, or one is received but it is set to a reserved value, then the PE assumes that a configuration error has occurred. That is, if the remote PE did not include the stack-capability sub-TLV in the received label mapping message, or it does include the sub-TLV but with the IPv6 bit cleared, and if stack-capability-signaling is enabled, the local node with ce-address-discovery ipv6 enabled withdraws its pseudowire label with the LDP status code "IP Address type mismatch”.

If a 7750 SR, 7450 ESS, and 7950 XRS PE that supports IPv6 (that is, stack-capability-signaling ipv6 is enabled) has not yet sent a label mapping message for the pseudowire and does not receive a stack-capability sub-TLV from the far-end PE in the initial label mapping message, or one is received but it is set to a reserved value, the PE assumes that a configuration error has occurred and does not send a label mapping message of its own.

If the IPv6 stack is not supported by both PEs, or at least one of the PEs does support IPv6 but does not have the **ce-address-discovery ipv6** option selected in the CLI, IPv6 packets received from the AC are discarded by the PE. IPv4 packets are always supported.

If IPv6 stack support is implemented by both PEs, but the **ce-address-discovery ipv6** command was not enabled on both so that the IP pseudowire came up with only IPv4 support, and one PE is later toggled to **ce-address-discovery ipv6**, then that PE sends a label withdraw with the LDP status code meaning "Wrong IP Address Type” (Status Code 0x0000004B9).

If the IPv6 stack is supported by both PEs and, therefore, the pseudowire is established with IPv6 capability at both PEs, but the **ce-address-discovery ipv6** command on one PE is later toggled to **no ce-address-discovery ipv6** so that a PE ceases to support the IPv6 stack, then that PE sends a label withdraw with the LDP status code meaning "Wrong IP Address Type”.

2.4 Services configuration for MPLS-TP

MPLS-TP PWs are supported in Epipe and Cpipe VLLs and Epipe spoke termination on IES/VPRN and VPLS, I-VPLS, and B-VPLS on the 7450 ESS and 7750 SR only.

This section describes how SDPs and spoke-SDPs are used with MPLS-TP LSPs and static pseudowires with MPLS-TP OAM. It also describes how to conduct test service throughput for PWs, using lock instruct messages and loopback configuration.

2.4.1 MPLS-TP SDPs

Only MPLS SDPs are supported.

An SDP used for MPLS-TP supports the configuration of an MPLS-TP identifier as the far-end address as an alternative to an IP address. IP addresses are used if IP/MPLS LSPs are used by the SDP, or if

MPLS-TP tunnels are identified by IPv4 source/destination addresses. MPLS-TP node identifiers are used if MPLS-TP tunnels are used.

Only static SDPs with signaling off support MPLS-TP spoke-SDPs.

The following CLI shows the MPLS-TP options:

```
config
   service
      sdp 10 [mpls | GRE | [ldp-enabled] [create]
         signaling <off | on>
         [no] lsp <xyz>
         [no] accounting-policy <policy-id>
         [no] adv-mtu-override
         [no] booking-factor <percentage>
         [no] class-forwarding
         [no] collect-stats
         [no] description <description-string>
         [no] far-end <ip-address> | [node-id
              {<ip-address> | <0…4,294,967,295>} [global-id <global-id>]]
         [no] tunnel-far-end <ip-address>
         [no] keep-alive
         [no] mixed-lsp-mode
         [no] metric <metric>
         [no] network-domain <network-domain-name>
         [no] path-mtu <mtu>
         [no] pbb-etype <ethertype>
         [no] vlan-vc-etype <ethertype>
         [no] shutdown
```

The **far-end node-id ip-address global-id global-id** command is used to associate an SDP far end with an MPLS-TP tunnel whose far-end address is an MPLS-TP node ID. If the SDP is associated with an RSVP-TE LSP, the far end must be a routable IPv4 address.

The system accepts the node-id being entered in either 4-octet IP address format (a.b.c.d) or unsigned integer format.

The SDP far end refers to an MPLS-TP node-id/global-id only if:

- delivery type is MPLS
- signaling is off
- keep-alive is disabled
- mixed-lsp-mode is disabled
- adv-mtu-override is disabled

An LSP can only be allowed to be configured if the far-end information matches the lsp far end information (whether MPLS-TP or RSVP).

- Only one LSP is allowed if the far end is an MPLS-TP node-id/global-id.
- MPLS-TP or RSVP-TE LSPs are supported. However, LDP and BG LSPs are not blocked in CLI.

Signaling LDP or BGP is blocked if:

- far-end node-id/global-id is configured
- control-channel-status is enabled on any spoke (or mate vc-switched spoke)
- pw-path-id is configured on any spoke (or mate vc-switched spoke)
- IES/VPRN interface spoke control-word is enabled

The following commands are blocked if a far-end node-id/global-id is configured:

- **class-forwarding**
- **tunnel-far-end**
- **mixed-lsp-mode**
- **keep-alive**
- **ldp** or **bgp-tunnel**
- **adv-mtu-override**

2.4.2 VLL spoke SDP configuration

The system can be a T-PE or an S-PE for a pseudowire (a spoke-SDP) supporting MPLS-TP OAM. MPLS-TP related commands are applicable to spoke-SDPs configured under all services supported by MPLS-TP pseudowires. All commands and functions that are applicable to spoke-SDPs are supported, except for those that explicitly depend on T-LDP signaling of the pseudowire, or as stated following. Likewise, all existing functions on a specified service SAP are supported if the spoke-SDP that it is mated to is MPLS-TP.

The **vc-switching** is supported.

The following describes how to configure MPLS-TP on an Epipe VLL. However, a similar configuration applies to other VLL types.

A spoke-SDP bound to an SDP with the **mpls-tp** keyword cannot be **no shutdown** unless the ingress label, the egress label, the control word, and the pw-path-id are configured, as follows:

```
config
   service
      epipe
         [no] spoke-sdp sdp-id[:vc-id]
              [no] hash-label
              [no] standby-signaling-slave

         [no] spoke-sdp sdp-id[:vc-id] [vc-type {ether | vlan}]
             [create] [vc-switching] [no-endpoint | {endpoint [icb]}]
            egress
               vc-label <out-label>
            ingress
               vc-label <in-label>
            control-word
            bandwidth <bandwidth>
            [no] pw-path-id
               agi <agi>
               saii-type2 <global-id:node-id:ac-id>
               taii-type2 <global-id:node-id:ac-id>
               exit
            [no] control-channel-status
[no] refresh-timer <value>
request-timer <request-timer-secs> retry-timer <retry-timer-secs> timeout-
multiplier <multiplier>
no request-timer
               [no] acknowledgment
               [no] shutdown
               exit
```

The **pw-path-id** context is used to configure the end-to-end identifiers for an MS-PW. These may not coincide with those for the local node if the configuration is at an S-PE. The SAII and TAII are consistent with the source and destination of a label mapping message for a signaled PW.

The **control-channel-status** command enables static pseudowire status signaling. This is valid for any spoke-SDP where **signaling none** is configured on the SDP (for example, where T-LDP signaling is not in use). The refresh timer is specified in seconds, from 10-65535, with a default of 0 (off). This value can only be changed if **control-channel-status** is **shutdown**.

Commands that rely on PW status signaling are allowed if control-channel-status is configured for a spoke-SDP bound to an SDP with signaling off, but the system uses control channel status signaling instead of T-LDP status signaling. The ability to configure control channel status signaling on a specified spoke-SDP is determined by the credit-based algorithm described earlier. Control channel status for a pseudowire only counts against the credit-based algorithm if the pseudowire is in a **no shutdown** state and has a non-zero refresh timer and a non-zero request timer.

A shutdown of a service results in the static PW status bits for the corresponding PW being set.

The spoke-SDP is held down unless the **pw-path-id** is complete.

The system accepts the node-id of the pw-path-id saii or taii being entered in either 4-octet IP address format (a.b.c.d) or unsigned integer format.

The control-word must be enabled to use MPLS-TP on a spoke-SDP.

The optional acknowledgment to a static PW status message is enabled using the **acknowledgment** command. The default is **no acknowledgment**.

The **pw-path-id** is only configurable if all of the following are true:

- in network mode D
- sdp signaling is off
- control-word is enabled (control-word is disabled by default)
- on service type Epipe, VPLS, Cpipe, or IES/VPRN interface
- An MPLS-TP node-id/global-id is configured under the **config**>**router**>**mpls**>**mpls-tp** context. This is required for OAM to provide a reply address.

In the vc-switching case, if configured to make a static MPLS-TP spoke SDP to another static spoke SDP, the TAII of the spoke-SDP must match the SAII of its mate, and the SAII of the spoke-SDP must match the TAII of its mate.

A control-channel-status no shutdown is allowed only if all of the following are true:

- in network-mode D
- sdp signaling is off
- control-word is enabled (control-word by default is disabled)
- the service type is Epipe, VPLS, Cpipe, or IES/VPRN interface
- pw-status-signaling is enabled (as follows)
- pw-path-id is configured for this spoke

The **hash-label** option is only configurable if SDP far end is not node-id/global-id.

The control channel status request mechanism is enabled when the **request-timer timer** parameter is non-zero. When enabled, this overrides the normal RFC-compliant refresh timer behavior. The refresh timer value in the status packet defined in RFC 6478 is always set to zero. The refresh-timer in the sending node is taken from the request-timer <timer1> timer. The two mechanisms are not compatible with each

other. One node sends a request timer while the other is configured for refresh timer. In a specified node, the request timer can only be configured with both acknowledgment and refresh timers disabled.

When configured, the procedures following are used instead of the RFC 6478 procedures when a PW status changes.

The CLI commands to configure control channel status requests are as follows:

```
[no] control-channel-status
     [no] refresh-timer <value> //0,10-65535, default:0
     [no] request-timer <timer1> retry-timer <timer2>
                  [timeout-multiplier <value>]
     [no] shutdown
     exit
```

request-timer <timer1>: 0, 10-65535, defaults: 0.

This parameter determines the interval at which PW status messages are sent, including a reliable delivery TLV, with the "request" bit set (as follows). This cannot be enabled if refresh-timer is not equal to zero (0).

retry-timer <timer2>: 3 to 60s

This parameter determines the timeout interval if no response to a PW status is received. This defaults to zero (0) when **no retry-timer**.

timeout-multiplier <value>: 3 to 15

If a requesting node does not get a response after retry-timer × multiplier, the node must assume that the peer is down. This defaults to zero (0) when **no retry-timer**.

2.4.2.1 Epipe VLL spoke SDP termination on IES, VPRN, and VPLS

All existing commands (except for those explicitly specified following) are supported for spoke-SDP termination on IES, VPRN, and VPLS (VPLS, I-VPLS and B-VPLS and routed VPLS) services. Also, the MPLS-TP commands listed preceding are supported. The syntax, default values, and functional behavior of these commands is the same as for Epipe VLLs, as specified preceding.

Also, the PW Control Word is supported on spoke-SDP termination on IES/VPRN interfaces for pseudowires of type "Ether" with statically assigned labels (signaling off) for spoke-SDPs configured with MPLS-TP Identifiers.

The following CLI commands under spoke-SDP are blocked for spoke-SDPs with statically assigned labels (and the SDP has signaling off) and MPLS-TP identifiers:

- **no status-signaling**

  This command causes the spoke-SDP to fall back to using PW label withdrawal as a status signaling method. However, T-LDP is not supported on MPLS-TP SDPs. Control channel status signaling should always be used for signaling PW status. Because active/standby dual-homing into a routed VPLS requires the use of T-LDP label withdrawal as the method for status signaling, active/standby dual-homing into routed VPLS is not supported if the spoke-SDPs are MPLS-TP.

- **propagate-mac-flush**

  This command requires the ability to receive MAC Flush messages using T-LDP signaling and is blocked.

2.4.3 Configuring MPLS-TP lock instruct and loopback

MPLS-TP supports lock instruct and loopback for PWs.

2.4.3.1 MPLS-TP PW lock instruct and loopback overview

The lock instruct and loopback capability for MPLS-TP PWs includes the ability to:

- administratively lock a spoke-SDP with MPLS-TP identifiers
- divert traffic to and from an external device connected to a SAP
- create a data path loopback on the corresponding PW at a downstream S-PE or T-PE that was not originally bound to the spoke-SDP being tested
- forward test traffic from an external test generator into an administratively locked PW, while simultaneously blocking the forwarding of user service traffic

MPLS-TP provides the ability to conduct test service throughput for PWs, using lock instruct messages and loopback configuration. To conduct a service throughput test, you can apply an administrative lock at each end of the PW. This creates a test service that contains the SAP connected to the external device. Lock request messaging is not supported. You can also configure a MEP to send a lock instruct message to the far-end MEP. The lock instruct message is carried in a G-ACh on Channel 0x0026. A lock can be applied using the CLI or NMS. The forwarding state of the PW can be either active or standby.

After locking a PW, you can put it into loopback mode (for two-way tests) so the ingress data path in the forward direction is cross-connected to the egress data path in the reverse direction of the PW. This is accomplished by configuring the source MEP to send a loopback request to an intermediate MIP or MEP. A PW loopback is created at the PW level, so everything under the PW label is looped back. This distinguishes a PW loopback from a service loopback, where only the native service packets are looped back. The loopback is also configured through CLI or NMS.

The following MPLS-TP lock instruct and loopback functionality is supported:

- An MPLS-TP loopback can be created for an Epipe or Cpipe VLL.
- Test traffic can be inserted at an Epipe or Cpipe VLL endpoint or at an Epipe spoke-sdp termination on a VPLS interface.

2.4.3.2 Lock PW endpoint model

You can administratively lock a spoke-SDP by locking the host service using the **admin-lock** parameter of the **tools** command. The following conditions and constraints apply:

- Both ends of a PW or MS-PW represented by a spoke-SDP must be administratively locked.
- Test traffic can be injected into the spoke-SDP using a SAP defined within a test service. The test service must be identified in the **tools** command at one end of the locked PW.
- All traffic is forwarded to and from the test SAP defined in the test service, which must be of a type that is compatible with the spoke-SDP.
- Traffic to and from a non-test SAP is dropped. If no test SAP is defined, all traffic received on the spoke-SDP is dropped, and all traffic received on the paired SAP is also dropped.
- If a spoke-SDP is administratively locked, it is treated as operationally down. If a VLL SAP is paired with a spoke-SDP that is administratively locked, the SAP OAM treats this as if the spoke-SDP is operationally down.

- If a VPLS interface is paired to a spoke-SDP that is administratively locked, the L2 interface is taken down locally.
- The **control-channel-status** must be shutdown before administratively locking a spoke-SDP.

2.4.3.3 PW redundancy and lock instruct and loopback

It is possible to apply an administrative lock and loopback to one or more spoke-SDPs within a redundant set. That is, it is possible to move a spoke-SDP from an existing endpoint to a test service. When an administrative lock is applied to a spoke-SDP, it becomes operationally down and cannot send or receive traffic from the normal service SAP or spoke interface. If the lock is applied to all the spoke-SDPs in a service, all the spoke-SDPs become operationally down.

2.4.3.4 Configuring a test SAP for an MPLS-TP PW

A test SAP is configured under a unique test service type. This looks similar to a normal service context, but normally only contain a SAP configuration:

```
config
   service
      epipe <service-id> [test] [create]
         [no] sap <sap-id>
         [no] shutdown
      [no] shutdown
config
   service
      cpipe <service-id> [vc-type {satop-e1 | satop-t1 | cesopsn | cesopsncas}
       [test][create]
         [no] sap <sap-id>
         [no] shutdown
      [no] shutdown
```

You can define test SAPs appropriate to any service or PW type supported by MPLS-TP, including a Cpipe or Epipe. The following test SAP types are supported:

- Ethernet NULL, 1q, Q-in-Q
- TDM E1, E3, DS0, DS3, and so on

The following constraints and conditions apply:

- Up to a maximum of 16 test services can be configured per system.
- It is possible to configure access ingress and access egress QoS policies on a test SAP, as well as any other applicable SAP-specific commands and overrides.
- Vc-switching and spoke-SDP are blocked for services configured under the test context.
- The **test** keyword is mutually exclusive vc-switching and customer.
- Valid commands under a compatible test service context do not need to be blocked just because the service is a test service.

2.4.3.5 Configuring an administrative lock

An administrative lock is configured on a spoke-SDP using the **admin-lock** option of the **tools perform** command, as follows:

```
tools
    perform
      service-id <svc-id>
         admin-lock
            pw
               sdp <sdp-id> admin-lock [test-svc-id <id>]
```

The following conditions and constraints apply for configuring an administrative lock:

- The lock can be configured either on a spoke-SDP that is bound to a SAP, another spoke-SDP or a VPLS interface.
- The lock is only allowed if a PW path ID is defined (for example, for static PWs with MPLS-TP identifiers).
- The lock cannot be configured on spoke-SDPs that are an Inter-Chassis Backup (ICB) or if the vc-switching keyword is present.
- The control-channel-status must be shutdown. The operator should also shutdown control-channel-status on spoke-SDPs belonging to an MS-PW at an S-PE whose far ends are administratively locked at its T-PEs. This should be enforced throughout the network management if using the 5620 SAM.
- When enabled, all traffic on the spoke-SDP is sent to and from a paired SAP that has the **test** keyword present, if such a SAP exists in the X endpoint (see Pseudowire redundancy service models). Otherwise, all traffic to and from the paired SAP is dropped.
- The lock can be configured at a spoke-SDP that is bound to a VLL SAP or a VPLS interface.
- The **test-svc-id** parameter refers to the test service that should be used to inject test traffic into the service. The test service must be of a compatible type to the existing spoke-SDP under test (see Table 6: Mapping of real services to test service types).
- If the **test-svc-id** parameter is not configured on an admin-locked spoke-SDP, user traffic is blocked on the spoke-SDP.

The service manager should treat an administrative lock as a fault from the perspective of a paired SAP that is not a test SAP. This causes the appropriate SAP OAM fault indication.

Table 6: Mapping of real services to test service types maps supported real services to their corresponding test services.

Table 6: Mapping of real services to test service types

Service	Test service
Cpipe	Cpipe
Epipe	Epipe
VPLS	Epipe
PBB VPLS	Epipe

2.4.3.6 Configuring a loopback

If a loopback is configured on a spoke-SDP, all traffic on the ingress direction of the spoke-sdp and associated with the ingress vc-label is forwarded to the egress direction of the spoke-SDP. A loopback may be configured at either a T-PE or an S-PE. It is recommended that an administrative lock is configured before configuring the loopback on a spoke-SDP. This is enforced by the NMS.

A data path loopback is configured using a **tools perform** command, as follows:

```
tools
   perform
      service-id <svc-id>
         loopback
            pw
               sdp <sdp-id>:<vc-id> {start | stop}
```

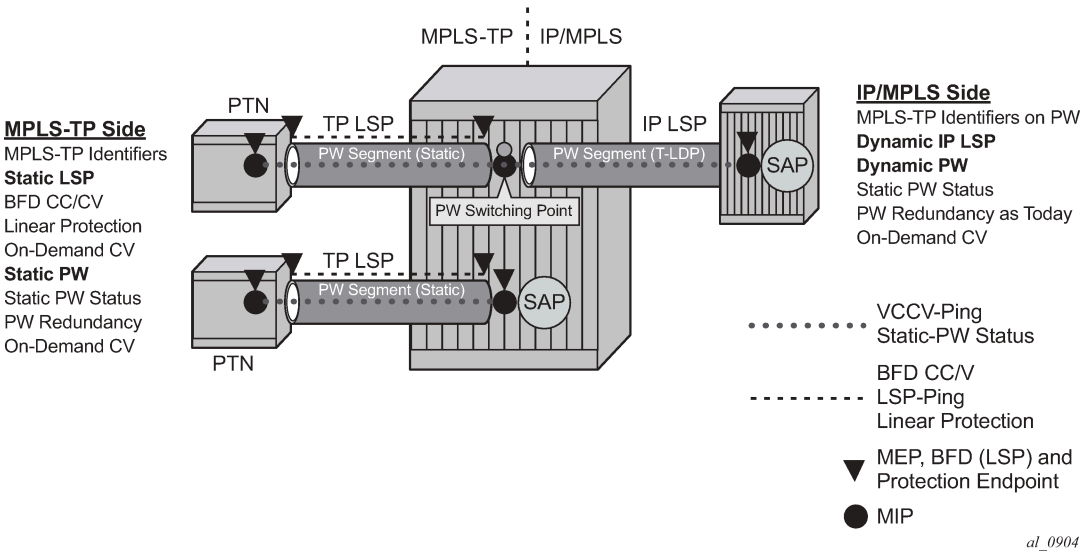
The following constraints and conditions apply for PW loopback configuration:

- The spoke-SDP cannot be an ICB or be bound to a VPLS interface.
- A PW path ID must be configured, that is, the spoke-SDP must be static and use MPLS-TP identifiers.
- The spoke-SDP must be bound to a VLL mate SAP or another spoke-SDP that is not an ICB.
- The control-channel-status must be shutdown.
- The following are disabled on a spoke-SDP for which a loopback is configured:
  - Filters
  - PW shaping
- Only network port QoS is supported.

2.4.4 Switching static MPLS-TP to dynamic T-LDP signaled PWs

Some use cases for MPLS-TP require an MPLS-TP based aggregation network and an IP-based core network to interoperate, so providing the seamless transport of packet services across static MPLS-TP and dynamically signaled domains using an MS-PW. In this environment, end-to-end VCCV Ping and VCCV Trace may be used on the MS-PW, as shown in Figure 10: Static - dynamic PW switching with MPLS-TP.

Figure 10: Static - dynamic PW switching with MPLS-TP



Services are backhauled from the static MPLS-TP network on the left to the dynamic IP/MPLS network on the right. The router acts as an S-PE interconnecting the static and dynamic domains.

The router implementation supports such use cases through the ability to mate a static MPLS-TP spoke SDP, with a defined **pw-path-id**, to a FEC128 spoke SDP. The dynamically signaled spoke SDP must be MPLS; GRE PWs are not supported, but the T-LDP signaled PW can use any supported MPLS tunnel type (for example, LDP, RSVP-TE, static, BGP). The control-word must be enabled on both mate spoke SDPs.

Mapping of control channel status signaling to and from T-LDP status signaling at the router S-PE is also supported.

The use of VCCV Ping and VCCV Trace on an MS-PW composed of a mix of static MPLS-TP and dynamic FEC128 segments is described in more detail in the *7450 ESS, 7750 SR, 7950 XRS, and VSR OAM and Diagnostics Guide*.

2.5 VCCV BFD support for VLL, spoke-SDP termination on IES and VPRN, and VPLS services

This section provides information about VCCV Bidirectional Forwarding Detection (BFD) support for VLL, spoke-SDP termination on IES and VPRN, and VPLS services.

2.5.1 VCCV BFD support

The SR OS supports RFC 5885, which specifies a method for carrying BFD in a pseudowire-associated channel. This enables BFD to monitor the pseudowire between its terminating PEs, regardless of how many P routers or switching PEs the pseudowire may traverse. This makes it possible for faults that are local to individual pseudowires to be detected, whether they also affect forwarding for other pseudowires, LSPs, or IP packets. VCCV BFD is ideal for monitoring specific high-value services, where detecting forwarding failures (and potentially restoring from them) in the minimal amount of time is critical.

VCCV BFD is supported on VLL services using T-LDP spoke-SDPs or BGP VPWS. It is supported for Cpipe, Epipe, and Ipipe VLL services.

VCCV BFD is supported on IES/VPRN services with T-LDP spoke -SDP termination (for Epipes and Ipipes).

VCCV BFD is supported on LDP- and BGP-signaled pseudowires, and on pseudowires with statically configured labels, whether signaling is off or on for the SDP. VCCV BFD is not supported on MPLS-TP pseudowires.

VCCV BFD is supported on VPLS services (both spoke-SDPs and mesh SDPs). VCCV BFD is configured by:

1. configuring generic BFD session parameters in a BFD template
2. applying the BFD template to a spoke-SDP or pseudowire-template binding, using the **bfd-template bfd-template-name** command
3. enabling the template on that spoke-SDP, mesh SDP, or pseudowire-template binding using the **bfd-enable** command

2.5.2 VCCV BFD encapsulation on a pseudowire

The SR OS supports IP/UDP encapsulation for BFD. With this encapsulation type, the UDP headers are included on the BFD packet. IP/UDP encapsulation is supported for pseudowires that use router alert (VCCV Type 2), and for pseudowires with a control word (VCCV Type 1). In the control word case, the IPv4 channel (channel type 0x0021) is used. On the node, the destination IPv4 address is fixed at 127.0.0.1 and the source address is 127.0.0.2.

VCCV BFD sessions run end-to-end on a switched pseudowire. They do not terminate on an intermediate S-PE; therefore, the TTL of the pseudowire label on VCCV BFD packets is always set to 255 to ensure that the packets reach the far-end T-PE of an MS-PW.

2.5.3 BFD session operation

BFD packets flow along the full length of a PW, from T-PE to T-PE. Because they are not intercepted at an S-PE, single-hop initialization procedures are used.

A single BFD session exists per pseudowire.

BFD runs in asynchronous mode.

BFD operates as a simple connectivity check on a pseudowire. The BFD session state is displayed in the MIBs and in the **show**>**service id**>**sdp**>**vccv-bfd session** command. By default, BFD is not used to change the operational state of the pseudowire, to modify pseudowire redundancy, or to map the BFD state to SAP OAM. However, the router may be optionally configured to take into account the BFD session state.

VCCV BFD runs in software with a minimum supported timer interval of 100 ms for Epipe LDP spoke-SDP; H-VPLS spoke-SDP and mesh-SDP; and Epipe spoke-SDP termination on IES and VPRN interfaces, and inter-chassis backup spoke-SDPs.

BFD is used only for fault detection. While RFC 5885 provides a mode in which VCCV BFD can be used to signal pseudowire status, this mode is applicable only to pseudowires that have no other status signaling mechanism in use. LDP status and static pseudowire status signaling always take precedence over BFD-

signaled PW status, and BFD-signaled pseudowire status is not used on pseudowires that use LDP status or static pseudowire status signaling mechanisms.

2.5.4 Using VCCV BFD to set SDP binding operational state

Use the **failure-action down** command to configure the router to use the VCCV BFD session state to affect the operational status of its SDP bindings (spoke SDPs or mesh SDPs). This behavior is only supported for Epipe LDP spoke SDPs, H-VPLS spoke SDPs, inter-chassis backup spoke SDPs, and Epipe spoke SDP termination on IES and VPRN interfaces.

If this behavior is configured on a spoke SDP bound to a VPLS or VPLS component of the R-VPLS, the SDP binding operational state is treated as an operationally down PW by the VPLS. This means that if all SDP bindings in the VPLS instance go operationally down because the VCCV BFD is going down, the VPLS goes down. In the case of R-VPLS, this handling is required to take down the associated IP interface if all SDP bindings are down. An operational down state that is caused by the BFD session going down removes the SDP binding from the eligible set used for PW redundancy in a VPLS or VLL service, including spoke SDP termination on IES or VPRN interfaces.

An operational down state of the SDP binding, because the VCCV BFD session is going down, is mapped to any SAP OAM mechanisms. It also contributes to the operational state of an operational group that is monitoring the SDP binding on which VCCV BFD with **failure-action down** is configured.

2.5.5 Configuring VCCV BFD

Generic BFD session parameters are configured for VCCV using the **bfd-template** command, in the **config**>**router**>**bfd** context. However, there are some restrictions.

For VCCV, the BFD session cannot terminate on the CPM network processor. Therefore, an error is generated if the user tries to bind a BFD template using the **type cpm-np** command within the **config**>**router**>**bfd**>**bfd-template** context.

Attempting to bind a BFD template with any unsupported transmit or receive interval generates an error.

Finally, attempting to commit changes to a BFD template that is already bound to a PW where the new values are invalid for VCCV BFD results in an error.

If the preceding BFD timer values are changed in a specified template, any BFD sessions on PWs to which that template is bound try to renegotiate their timers to the new values.

Commands within the BFD-template use a **begin-commit** model. To edit any value within the BFD template, a **begin** command needs to be executed after the template context has been entered. However, a value is still stored temporarily in the template-module until the **commit** command is issued. When the **commit** is issued, values are used by other modules such as the MPLS-TP module and BFD module.

For PWs where the PW template does not apply, a named BFD template is configured on the spoke-SDP using the **config service [epipe | cpipe | ipipe] spoke-sdp bfd bfd-template** command and then enabled using the **config service [epipe | cpipe | ipipe] spoke-sdp bfd bfd-enable** command. For example, LDP-signaled spoke-SDPs for a VLL service that uses the PW ID FEC (FEC128) or spoke-SDPs with static PW labels.

Configuring and enabling a BFD template on a static PW already configured with MPLS-TP identifiers (that is, with a pw-path-id) or on a spoke-SDP with a configured pw-path-id is not supported. Likewise, if a BFD template is configured and enabled on a spoke-SDP, a **pw-path-id** cannot be configured on the spoke-SDP.

The **bfd-enable** command is blocked on a spoke-SDP configured with VC-switching. This is because VCCV BFD always operates end-to-end on an MS-pseudowire. It is not possible to extract VCCV BFD packets at the S-PE.

For IES and VPRN spoke-SDP termination where the PW template does not apply (that is, where the spoke-SDP is signaled with LDP and uses the PW ID FEC (FEC128)), the BFD template is configured using the **config service ies | vprn if spoke-sdp bfd bfd-template** command, then enabled using the **config service ies | vprn if spoke-sdp bfd bfd-enable** command.

For H-VPLS, where the PW template does not apply (that is, LDP-VPLS spoke SDPs that use the PW ID FEC(FEC128)) the BFD template is configured using the **configure service vpls spoke-sdp bfd bfd-template** or the **configure service vpls mesh-sdp bfd bfd-template** command. VCCV BFD is then enabled with the **bfd-enable** command under the **vpls spoke-sdp bfd** or **vpls mesh-sdp bfd** context.

PWs where the PW template does apply and that support VCCV BFD are as follows:

- BGP-AD, which is signaled using the Generalized PW ID FEC (FEC129) with Attachment Individual Identifier (AII) type I
- BGP VPLS
- BGP VPWS

For these PW types, a named BFD template is configured and enabled from the PW template binding context.

For BGP VPWS, the BFD template is configured using the **config service epipe bgp pw-template-binding bfd-template name** command, then enabled using the **config service epipe bgp pw-template-binding bfd-enable** command.

The ability to determine PW forwarding state from VCCV BFD on the pseudowire is configured using the **failure-action down** command. It is possible to configure **failure-action** whether a spoke-SDP is administratively shutdown or not. It is therefore recommended to first configure VCCV BFD to ensure the spoke-SDP is forwarding, and then configure the **failure-action** command. If the **failure-action down** command is configured, the router continues to send VCCV BFD packets on a spoke-SDP or mesh-SDP that is operationally down because of the VCCV BFD session being in the down state. The router can then rapidly detect when connectivity is restored.

The **wait-for-up-timer** can be configured when **failure-action down** is configured. The **wait-for-up-timer** timer is triggered when a spoke-SDP or mesh-SDP is first administratively enabled and when a VCCV BFD session transitions from up to down. It is useful to allow time for BFD sessions to come up when the spoke-SDP is initially **no shutdown**. This provides the BFD session time to settle before it selects the active spoke-SDP for use in a redundant set. It also prevents excessive flapping of the operation state of a spoke-SDP if a VCCV BFD session is bouncing.

2.6 Pseudowire switching

The pseudowire switching feature provides the user with the ability to create a VLL service by cross-connecting two spoke-SDPs. This feature allows the scaling of VLL and VPLS services in a large network in which the otherwise full mesh of PE devices would require thousands of Targeted LDP (T-LDP) sessions per PE node.

Services with one SAP and one spoke-SDP are created normally on the PE; however, the target destination of the SDP is the pseudowire switching node instead of what is normally the remote PE. Also, the user configures a VLL service on the pseudowire switching node using the two SDPs.

The pseudowire switching node acts in a passive role with respect to signaling of the pseudowires. It waits until one or both of the PEs sends the label mapping message before relaying it to the other PE. This is because it needs to pass the interface parameters of each PE to the other.

A pseudowire switching point TLV is inserted by the switching pseudowire to record its system address when relaying the label mapping message. This TLV is useful in a few situations:

- It allows for troubleshooting of the path of the pseudowire especially if multiple pseudowire switching points exist between the two PEs.
- It helps in loop detection of the T-LDP signaling messages where a switching point would receive back a label mapping message it had already relayed.
- The switching point TLV is inserted in pseudowire status notification messages when they are sent end-to-end or from a pseudowire switching node toward a destination PE.

Pseudowire OAM is supported for the manual switching pseudowires and allows the pseudowire switching node to relay end-to-end pseudowire status notification messages between the two PEs. The pseudowire switching node can generate a pseudowire status and send it to one or both of the PEs by including its system address in the pseudowire switching point TLV. This allows a PE to identify the origin of the pseudowire status notification message.

In the following example, the user configures a regular Epipe VLL service PE1 and PE2. These services each consist of a SAP and a spoke-SDP. However, the target destination of the SDP is not the remote PE, but the pseudowire switching node. Also, the user configures an Epipe VLL service on the pseudowire switching node using the two SDPs.

```
|7450 ESS, 7750 SR, and 7950 XRS PE1 (Epipe)|---sdp 2:10---|7450 ESS, 7750 SR, and
7950 XRS PW SW (Epipe)|---sdp 7:15---|7450 ESS, 7750 SR, and 7950 XRS PE2 (Epipe)|
```

Configuration examples are in Configuring two VLL paths terminating on T-PE2.

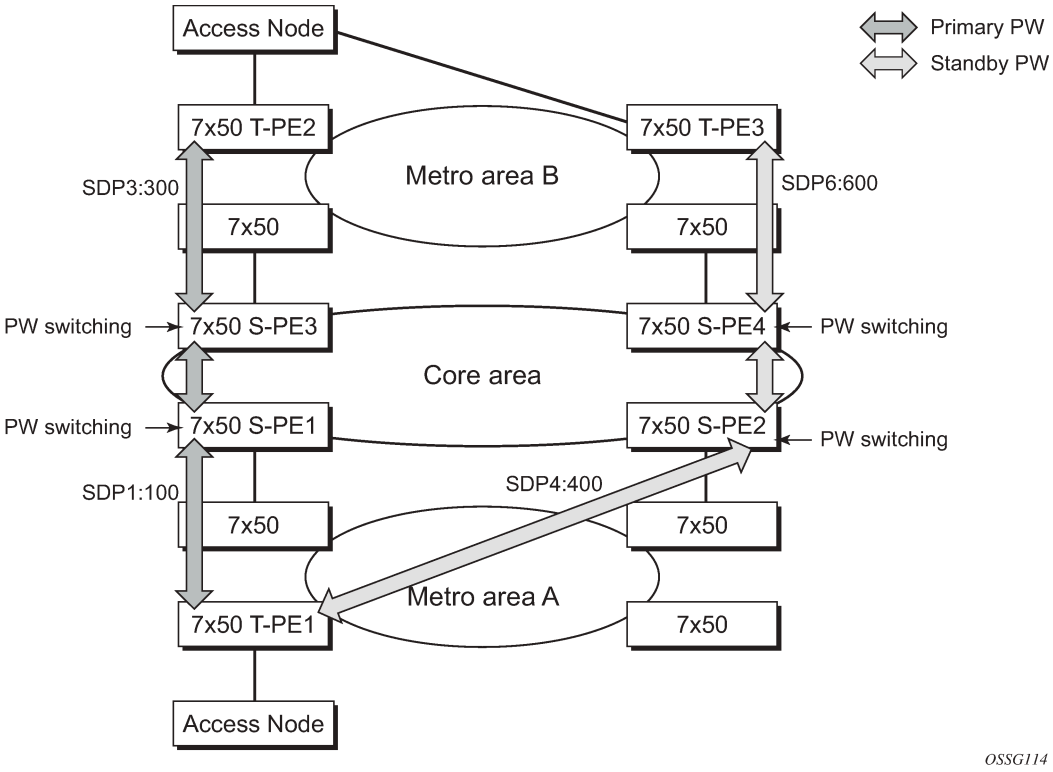
2.6.1 Pseudowire switching with protection

Pseudowire switching scales VLL and VPLS services over a multi-area network by removing the need for a full mesh of targeted LDP sessions between PE nodes. Figure 11: VLL resilience with pseudowire redundancy and switching shows the use of pseudowire redundancy to provide a scalable and resilient VLL service across multiple IGP areas in a provider network.

In the network in Figure 11: VLL resilience with pseudowire redundancy and switching, PE nodes act as leading nodes, and pseudowire switching nodes act as followers for the purpose of pseudowire signaling. A switching node needs to pass the SAP interface parameters of each PE to the other PEs. T-PE1 sends a label mapping message for the Layer 2 FEC to the peer pseudowire switching node; for example, S-PE1. The label mapping message includes the SAP interface parameters, such as MTU, in the label mapping message. S-PE1 checks the FEC against the local information and, if a match exists, appends the optional pseudowire switching point TLV to the FEC TLV in which it records its system address. T-PE1 then relays the label mapping message to S-PE2. S-PE2 performs similar operations and forwards a label mapping message to T-PE2.

The same procedures are followed for the label mapping message in the reverse direction; for example, from T-PE2 to T-PE1. S-PE1 and S-PE2 make the spoke-SDP cross-connect only when both directions of the pseudowire are signaled and matched.

Figure 11: VLL resilience with pseudowire redundancy and switching



The pseudowire switching TLV is useful in a few situations. First, it allows for troubleshooting of the path of the pseudowire, especially if multiple pseudowire switching points exist between the two T-PE nodes. Second, it helps in loop detection of the T-LDP signaling messages where a switching point receives back a label mapping message that the point already relayed. Finally, it can be inserted in pseudowire status messages when they are sent from a pseudowire switching node toward a destination PE.

Pseudowire status messages can be generated by the T-PE nodes or the S-PE nodes, or both. Pseudowire status messages received by a switching node are processed and passed on to the next hop. An S-PE node appends the optional pseudowire switching TLV, with the S-PEs system address added to it, to the FEC in the pseudowire status notification message, only if that S-PE originated the message or the message was received with the TLV in it. Otherwise, the message was originated by a T-PE node and the S-PE should process and pass the message without changes, except for the VC-ID value in the FEC TLV.

2.6.2 Pseudowire switching behavior

In the network in Figure 11: VLL resilience with pseudowire redundancy and switching, PE nodes act as leading nodes and pseudowire switching nodes act as followers for the purpose of pseudowire signaling. This is because a switching node needs to pass the SAP interface parameters of each PE to the other. T-PE1 sends a label mapping message for the Layer 2 FEC to the peer pseudowire switching node; for example, S-PE1. It includes the SAP interface parameters, such as MTU, in the label mapping message. S-PE1 checks the FEC against the local information and, if a match exists, appends the optional pseudowire switching point TLV to the FEC TLV in which it records its system address. T-PE1 then relays the label mapping message to S-PE2. S-PE2 performs similar operation and forwards a label mapping message to T-PE2.

The same procedures are followed for the label mapping message in the reverse direction; for example, from T-PE2 to T-PE1. S-PE1 and S-PE2 affect the spoke-SDP cross-connect only when both directions of the pseudowire are signaled and matched.

Pseudowire status messages can be generated by the T-PE nodes or the S-PE nodes, or both. Pseudowire status messages received by a switching node are processed, then passed on to the next hop. An S-PE node appends the optional pseudowire switching TLV, with its system address added to it, to the FEC in the pseudowire status notification message, only if it originated the message or the message was received with the TLV in it. Otherwise, the message was originated by a T-PE node and the S-PE should process and pass the message without changes, except for the VC-ID value in the FEC TLV.

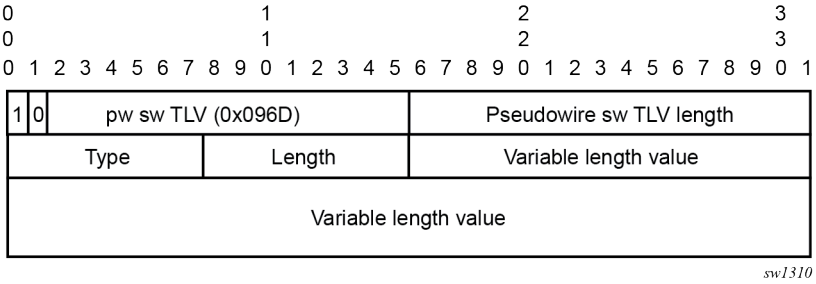
The merging of the received T-LDP status notification message and the local status for the spoke-SDPs from the service manager at a PE complies with the following rules:

- When the local status for both spoke-SDPs is up, the S-PE passes any received SAP or SDP binding generated status notification message unchanged; for example, the status notification TLV is unchanged but the VC-ID in the FEC TLV is set to value of the pseudowire segment to the next hop.
- When the local operational status for any of the spokes is down, the S-PE always sends an SDP-binding down status bits regardless of whether the received status bits from the remote node indicated SAP up or down or SDP-binding up or down.

2.6.2.1 Pseudowire switching TLV

The format of the pseudowire switching TLV is as follows.

Figure 12: Pseudowire switching TLV format



PW sw TLV Length

specifies the total length of all the following pseudowire switching point TLV fields in octets

Type

encodes how the Value field is to be interpreted

Length

specifies the length of the Value field in octets

Value

octet string of Length octets that encodes information to be interpreted as specified by the Type field

2.6.2.2 Pseudowire switching point sub-TLVs

Following is information specific to pseudowire switching point sub-TLVs:

- **pseudowire ID of last pseudowire segment traversed**

  This is the sub-TLV type that contains a pseudowire ID in the format of the pseudowire ID.
- **pseudowire switching point description string**

  This is an optional description text string of up to 80 characters.
- **IP address of pseudowire switching point**

  This is an options sub-TLV; IP V4 or V6 address of the pseudowire switching point.
- **MH VCCV capability indication**

2.6.3 Static-to-dynamic pseudowire switching

When one segment of the pseudowire cross-connect at the S-PE is static while the other is signaled using T-LDP, the S-PE operates much like a T-PE from a signaling perspective and as an S-PE from a data plane perspective.

The S-PE signals a label mapping message as soon as the local configuration is complete. The control word C-bit field in the pseudowire FEC is set to the value configured on the static spoke-SDP.

When the label mapping for the egress direction is also received from the T-LDP peer, and the information in the FEC matches that of the local configuration, the static-to-dynamic cross-connect is created.

It is possible that end nodes of a static pseudowire segment can be misconfigured. In this case, an S-PE or T-PE node may be receiving packets with the wrong encapsulation, so that an invalid payload could be forwarded over the pseudowire or the SAP, respectively. Also, if the S-PE or T-PE node is expecting the control word in the packet encapsulation and the received packet comes with no control word, but the first nibble below the label stack is 0x0001, the packet may be mistaken for a VCCV OAM packet and may be forwarded to the CPM. In that case, the CPM performs a check of the IP header fields such as version, IP header length, and checksum. If any of these fail the VCCV packet is discarded.

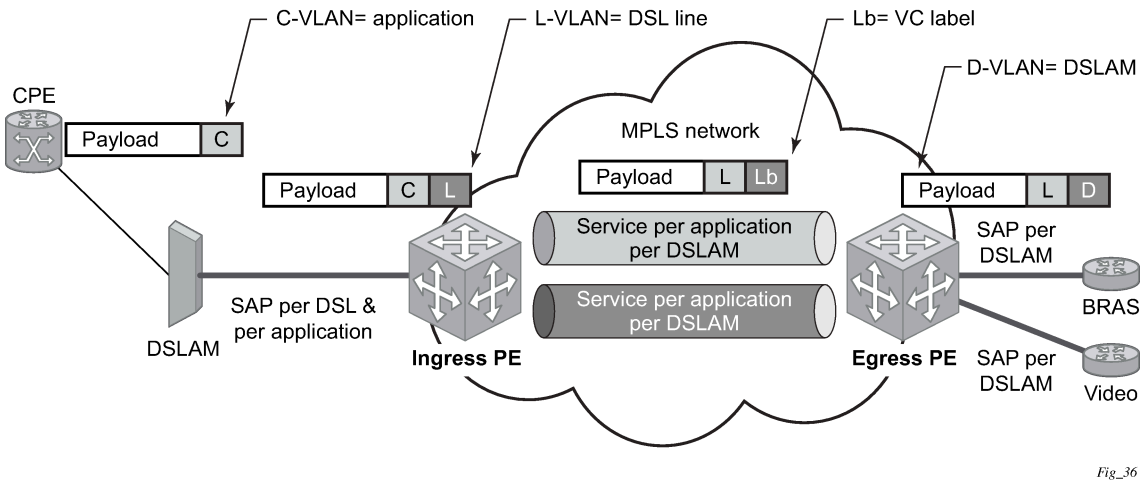
2.6.4 Ingress VLAN swapping

This feature is supported on VPLS and VLL services where the end-to-end solution is built using two node solutions (requiring SDP connections between the nodes).

In VLAN swapping, only the VLAN ID value is copied to the inner VLAN position. The Ethertype of the inner tag is preserved and all consecutive nodes work with that value. Similarly, the dot1p bits value of outer tag is not preserved.

Figure 13: Ingress VLAN swapping shows a network where, at user-access side (DSLAM-facing SAPs), every subscriber is represented by several QinQ SAPs with inner-tag encoding service and outer tag encoding subscriber (DSL line). At the aggregation side (BRAS- or PE-facing SAPs) every subscriber is represented by DSL line number (inner VLAN tag) and DSLAM (outer VLAN tag). The effective operation on the VLAN tag is to drop the inner tag at the access side and push another tag at the aggregation side.

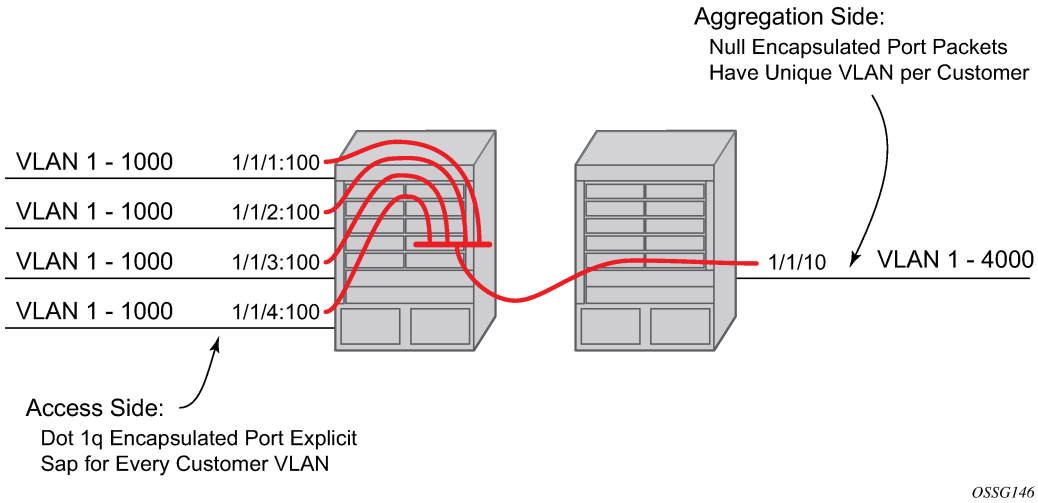
Figure 13: Ingress VLAN swapping



2.6.4.1 Ingress VLAN translation

Figure 14: Ingress VLAN translation shows an application where different circuits are aggregated in the VPLS-based network. The access side is represented by an explicit do1q encapsulated SAP. Because the VLAN ID is port specific, those connected to different ports may have the same VLAN. The aggregation side is aggregated on the same port; therefore, a unique VLAN ID is required.

Figure 14: Ingress VLAN translation



2.6.5 Pseudowire redundancy

Pseudowire redundancy provides the ability to protect a pseudowire with a pre-provisioned secondary standby pseudowire and to switch traffic over to that secondary standby pseudowire in case of a SAP or network failure condition, or both. Normally, pseudowires are redundant by the virtue of the SDP

redundancy mechanism. For instance, if the SDP is an RSVP LSP and is protected by a secondary standby path or by Fast-Reroute paths (FRR), or both, the pseudowire is also protected. However, there are two applications in which SDP redundancy does not protect the end-to-end pseudowire path:

- There are two different destination PE nodes for the same VLL service. The main use case is the provision of dual-homing of a CPE or access node to two PE nodes located in different POPs. The other use case is the provision of a pair of active and standby BRAS nodes, or active and standby links to the same BRAS node, to provide service resiliency to broadband service subscribers.
- The pseudowire path is switched in the middle of the network and the pseudowire switching node fails.

Pseudowire and VPLS link redundancy extends link-level resiliency for pseudowires and VPLS to protect critical network paths against physical link or node failures. These innovations enable the virtualization of redundant paths across the metro or core IP network to provide seamless and transparent fail-over for point-to-point and multi-point connections and services. When deployed with multi-chassis LAG, the path for return traffic is maintained through the pseudowire or VPLS switchover, which enables carriers to deliver "always on" services across their IP/MPLS networks.

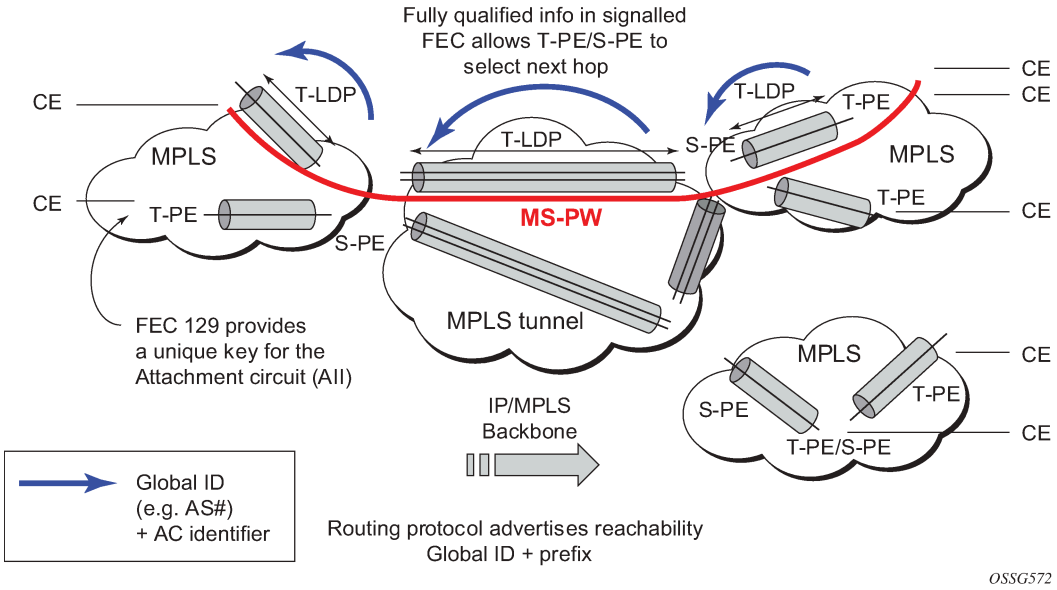
2.6.6 Dynamic multi-segment pseudowire routing

2.6.6.1 Overview

Dynamic Multi-Segment Pseudowire Routing (Dynamic MS-PWs) enable a complete multi-segment pseudowire to be established, while only requiring per-pseudowire configuration on the T-PEs. No per-pseudowire configuration is required on the S-PEs. End-to-end signaling of the MS-PW is achieved using T-LDP, while multi-protocol BGP is used to advertise the T-PEs, allowing dynamic routing of the MS-PW through the intervening network of S-PEs. Dynamic multi-segment pseudowires are described in the IETF Draft *draft-ietf-pwe3-dynamic-ms-pw-13.txt*.

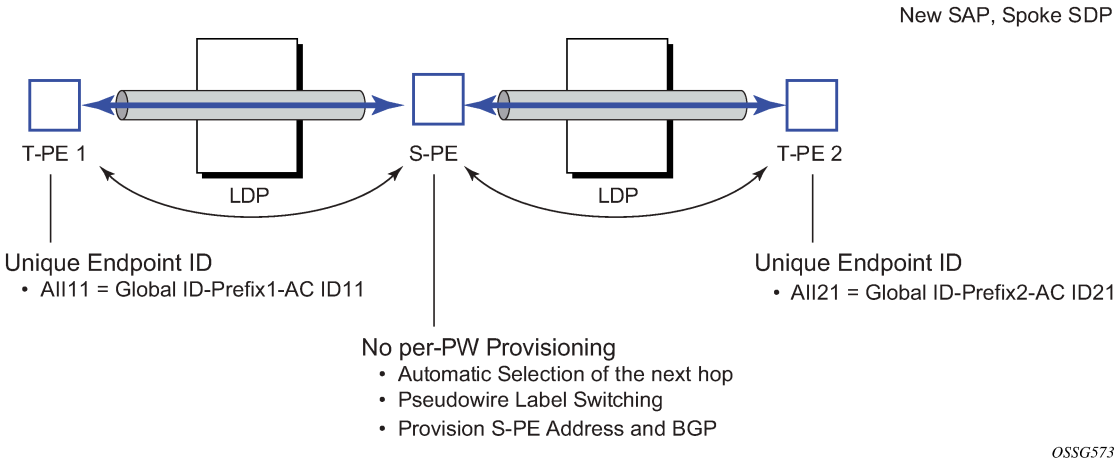
Figure 15: Dynamic MS-PW overview shows the operation of dynamic MS-PWs.

Figure 15: Dynamic MS-PW overview



The FEC 129 AII Type 2 structure depicted in Figure 16: MS-PW addressing using FEC129 AII Type 2 is used to identify each individual pseudowire endpoint:

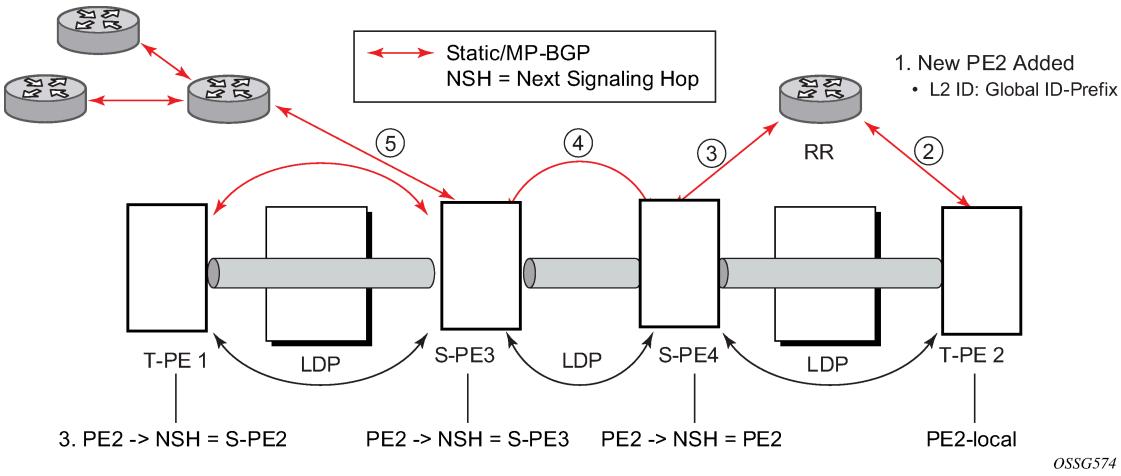
Figure 16: MS-PW addressing using FEC129 AII Type 2



A 4-byte global-id followed by a 4-byte prefix and a 4-byte attachment circuit ID are used to provide for hierarchical, independent allocation of addresses on a per-service provider network basis. The first 8 bytes (global-id + prefix) may be used to identify each individual T-PE or S-PE as a loopback Layer 2 address.

The AII type is mapped into the MS-PW BGP NLRI (a BGP AFI of L2VPN, and SAFI for network layer reachability information for dynamic MS-PWs). As soon as a new T- PE is configured with a local prefix address of global id: prefix, pseudowire routing proceeds to advertise this new address to all the other T-PEs and S-PEs in the network, as depicted in Figure 17: Advertisement of PE addresses by PW routing.

Figure 17: Advertisement of PE addresses by PW routing



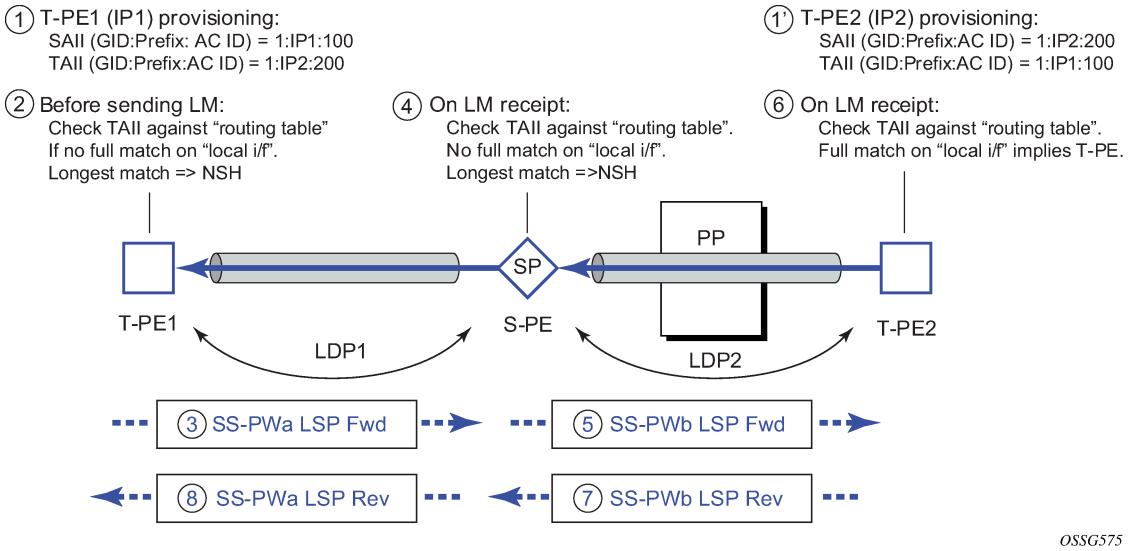
In step 1 of Figure 17: Advertisement of PE addresses by PW routing, a new T-PE (T-PE2) is configured with a local prefix.

Next, in steps 2 to 5, MP-BGP uses the NLRI for the MS-PW routing SAFI to advertise the location of the new T-PE to all the other PEs in the network. Alternatively, static routes may be configured on a per T-PE/S-PE basis to accommodate non-BGP PEs in the solution.

As a result, pseudowire routing tables for all the S-PEs and remote T-PEs are populated with the next hop to be used to reach T-PE2.

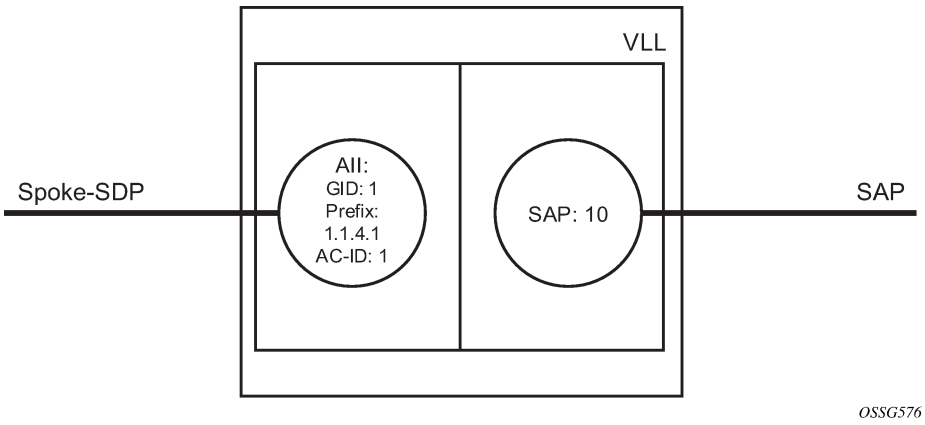
VLL services can then be established, as illustrated in Figure 18: Signaling of dynamic MS-PWs using T-LDP.

Figure 18: Signaling of dynamic MS-PWs using T-LDP



In step 1 and 1' of Figure 18: Signaling of dynamic MS-PWs using T-LDP the T-PEs are configured with the local and remote endpoint information: source AII (SAII) and Target AII (TAII). On the router, the AIIs are locally configured for each spoke-SDP, according to the model shown in Figure 19: Mapping of AII to SAP. Therefore the router provides for a flexible mapping of the AII to SAP. That is, the values used for the AII are through local configuration, and it is the context of the spoke-SDP that binds it to a specific SAP.

Figure 19: Mapping of AII to SAP



Before T-LDP signaling starts, the two T-PEs decide on an active and passive relationship using the highest AII (comparing the configured SAII and TAII) or the configured precedence. Next, the active T-PE (in the IETF draft, this is referred to as the source T-PE or ST-PE) checks the PW routing table to determine the next signaling hop for the configured TAII using the longest match between the TAII and the entries in the PW routing table.

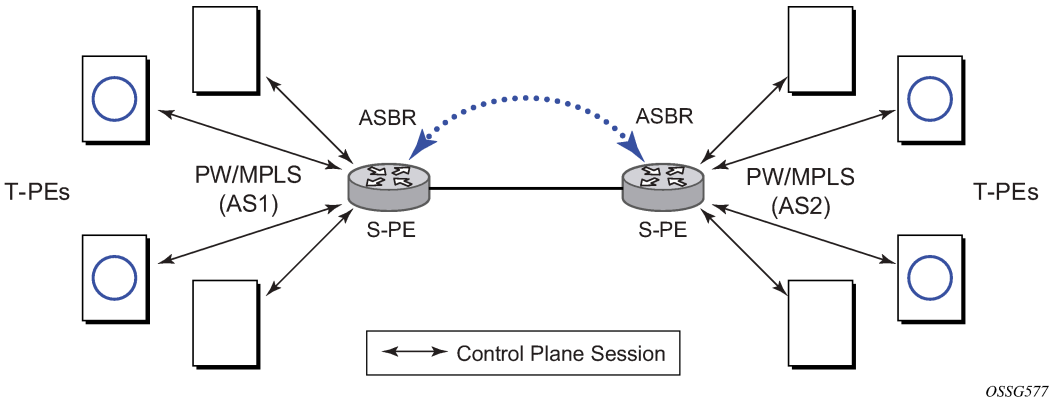
This signaling hop is then used to choose the T-LDP session to the chosen next-hop S-PE. Signaling proceeds through each subsequent S-PE using similar matching procedures to determine the next signaling hop. Otherwise, if a subsequent S-PE does not support dynamic MS-PW routing, so uses a statically configured PW segment, the signaling of individual segments follows the procedures already implemented in the PW Switching feature.

BGP can install a PW AII route in the PW routing table with ECMP next-hops. However, when LDP needs to signal a PW with matching TAII, it chooses only one next-hop from the available ECMP next-hops. PW routing supports up to 4 ECMP paths for each destination.

The signaling of the forward path ends when the PE matches the TAII in the label mapping message with the SAII of a spoke-SDP bound to a local SAP. The signaling in the reverse direction can now be initiated, which follows the entries installed in the forward path. The PW routing tables are not consulted for the reverse path. This ensures that the reverse direction of the PW follows exactly the same set of S-PEs as the forward direction.

This solution can be used in either a MAN-WAN environment or in an Inter-AS/Inter-Provider environment as depicted in Figure 20: VLL using dynamic MS-PWs, inter-AS scenario.

Figure 20: VLL using dynamic MS-PWs, inter-AS scenario



Data plane forwarding at the S-PEs uses pseudowire service label switching, as per the pseudowire switching feature.

2.6.6.2 Pseudowire routing

Each S-PE and T-PE has a pseudowire routing table that contains a reference to the T-LDP session to use to signal to a set of next hop S-PEs to reach a specific T-PE (or the T-PE if that is the next hop). For VLLs, this table contains aggregated AII Type 2 FECs and may be populated with routes that are learned through MP-BGP or that are statically configured.

MP-BGP is used to automatically distribute T-PE prefixes using the new MS-PW NLRI, or static routes can be used. The MS-PW NLRI is composed of a Length, an 8-byte route distinguisher (RD), a 4-byte global-id, a 4-byte local prefix, and (optionally) a 4-byte AC-ID. Support for the MS-PW address family is configured in CLI under the **config**>**router**>**bgp**>**family ms-pw** context.

MS-PW routing parameters are configured in the **config**>**service**>**pw-routing** context.

To enable support for dynamic MS-PWs on a 7750 SR, 7450 ESS, or 7950 XRS node to be used as a T-PE or S-PE, a single, globally unique, S-PE ID, known as the S-PE address, is first configured under **config**>**service**>**pw-routing** on each node to be used as a T-PE or S-PE. The S-PE address has the

format global-id:prefix. It is not possible to configure any local prefixes used for pseudowire routing or to configure spoke-SDPs using dynamic MS-PWs at a T-PE unless an S-PE address has already been configured. The S-PE address is used as the address of a node used to populate the switching point TLV in the LDP label mapping message and the pseudowire status notification sent for faults at an S-PE.

Each T-PE is also configured with the following parameters:

- **global-id**

  This is a 4-byte identifier that uniquely identifies an operator or the local network.

- **local prefix**

  One or more local (Layer 2) prefixes (up to a maximum of 16), which are formatted in the style of a 4-octet IPv4 address. A local prefix identifies a T-PE or S-PE in the PW routing domain.

  For each local prefix, at least one 8-byte RD can be configured. It is also possible to configure an optional BGP community attribute.

For each local prefix, BGP then advertises each global-id/prefix tuple and unique RD and community pseudowire using the MS-PW NLRI, based on the aggregated FEC129 AII Type 2 and the Layer 2 VPN/PW routing AFI/SAFI 25/6, to each T-PE/S-PE that is a T-LDP neighbor, subject to local BGP policies.

The dynamic advertisement of each of these pseudowire routes is enabled for each prefix and RD using the **advertise-bgp** command.

An export policy is also required to export MS-PW routes in MP-BGP. This can be done using a default policy, such as the following:

```
*A:lin-123>config>router>policy-options# info
----------------------------------------------
            policy-statement "ms-pw"
                default-action accept
                exit
            exit
----------------------------------------------
```

However, this would export all routes. A recommended choice is to enable filtering per-family, as follows:

```
*A:lin-123>config>router>policy-options# info
----------------------------------------------
            policy-statement "to-mspw"
                entry 1
                    from
                        family ms-pw
                    exit
                    action accept
                    exit
                exit
            exit
----------------------------------------------
```

The following command is then added in the **config**>**router**>**bgp** context:

```
export "to-mspw"
```

Local-preference for IBGP and BGP communities can be configured under such a policy.

2.6.6.2.1 Static routing

As well as support for BGP routing, static MS-PW routes may also be configured using the **config services pw-routing static-route** command. Each static route comprises the target T-PE global-id and prefix, and the IP address of the T-LDP session to the next hop S-PE or T-PE that should be used.

If a static route is set to 0, this represents the default route. If a static route exists to a specified T-PE, this default route is used in preference to any BGP route that may exist.

2.6.6.2.2 Explicit paths

A set of default explicit routes to a remote T-PE or S-PE prefix may be configured on a T-PE under **config**>**services**>**pw-routing** using the **path name** command. Explicit paths are used to populate the explicit route TLV used by MS-PW T-LDP signaling. Only strict (fully qualified) explicit paths are supported.

It is possible to configure explicit paths independently of the configuration of BGP or static routing.

2.6.6.3 Configuring VLLs using dynamic MS-PWs

One or more spoke-SDPs may be configured for distributed Epipe VLL services. Dynamic MS-PWs use FEC129 (also known as the Generalized ID FEC) with Attachment Individual Identifier (AII) Type 2 to identify the pseudowire, as opposed to FEC128 (also known as the PW ID FEC) used for traditional single segment pseudowires and for pseudowire switching. FEC129 spoke-SDPs are configured under the **spoke-sdp-fec** command.

FEC129 AII Type 2 uses a Source Attachment Individual Identifier (SAII) and a Target Attachment Individual Identifier (TAII) to identify the end of a pseudowire at the T-PE. The SAII identifies the local end, while the TAII identifies the remote end. The SAII and TAII are each structured as follows:

- **global-id**

  This is a 4-byte identifier that uniquely identifies an operator or the local network.

- **prefix**

  This is a 4-byte prefix, which should correspond to one of the local prefixes assigned under pw-routing.

- **AC-ID**

  This is a 4-byte identifier for the local end of the pseudowire. This should be locally unique within the scope of the global-id:prefix.

2.6.6.3.1 Active/passive T-PE selection

Dynamic MS-PWs use single-sided signaling procedures with double-sided configuration; a fully qualified FEC must be configured at both endpoints. That is, one T-PE (the source T-PE, ST-PE) of the MS-PW initiates signaling for the MS-PW, while the other end (the terminating T-PE, TT-PE) passively waits for the label mapping message from the far end. This termination end only responds with a label mapping message to set up the opposite direction of the MS-PW when it receives the label mapping from the ST-PE. By default, the router determines which T-PE is the ST-PE (the active T-PE) and which is the TT-PE (the passive T-PE) automatically, based on comparing the SAII with the TAII as unsigned integers. The T-PE with SAII>TAII assumes the active role. However, it is possible to override this behavior using the signaling {**master | auto**} command under **spoke-sdp-fec**. If master is selected at a specified T-PE, that T-PE assumes the active role. If a T-PE is at the endpoint of a spoke-SDP that is bound to an VLL SAP and

single-sided auto-configuration is used, then that endpoint is always passive. For more information, see Automatic endpoint configuration. Therefore, signaling master should only be used when it is known that the far end assumes a passive behavior.

2.6.6.3.2 Automatic endpoint configuration

Automatic endpoint configuration allows the configuration of an endpoint without specifying the TAII associated with that **spoke-sdp-fec**. It allows a single-sided provisioning model where an incoming label mapping message with a TAII that matches the SAII of that spoke-SDP is automatically bound to that endpoint. This is useful in scenarios where a service provider wants to separate service configuration from the service activation phase.

Automatic endpoint configuration is supported for Epipe VLL **spoke-sdp-fec** endpoints bound to a VLL SAP. It is configured using the **spoke-sdp-fec auto-config** command, and excluding the TAII from the configuration. When auto-configuration is used, the node assumes passive behavior from a point of view of T-LDP signaling. See Active/passive T-PE selection for more information. Therefore, the far-end T-PE must be configured as the signaling master for that **spoke-sdp-fec**.

2.6.6.3.3 Selecting a path for an MS-PW

Path selection for signaling occurs in the outbound direction (ST-PE to TT-PE) for an MS-PW. In the TT-PE to ST-PE direction, a label mapping message follows the reverse of the path already taken by the outgoing label mapping.

A node can use explicit paths, static routes, or BGP routes to select the next hop S-PE or T-PE. The order of preference used in selecting these routes is:

1. Explicit Path
2. Static route
3. BGP route

To use an explicit path for an MS-PW, an explicit path must have been configured in the **config**>**services**>**pw-routing**>**path path-name** context. The user must then configure the corresponding **path path-name** under **spoke-sdp-fec**.

If an explicit path name is not configured, the TT-PE or S-PE performs a longest match lookup for a route (static if it exists, and BGP if not) to the next hop S-PE or T-PE to reach the TAII.

Pseudowire routing chooses the MS-PW path in terms of the sequence of S-PEs to use to reach a specified T-PE. It does not select the SDP to use on each hop, which is instead determined at signaling time. When a label mapping is sent for a specified pseudowire segment, an LDP SDP is used to reach the next-hop S-PE/T-PE if such an SDP exists. If not, and an RFC 8277 labeled BGP SDP is available, then that is used. Otherwise, the label mapping fails and a label release is sent.

2.6.6.3.4 Pseudowire templates

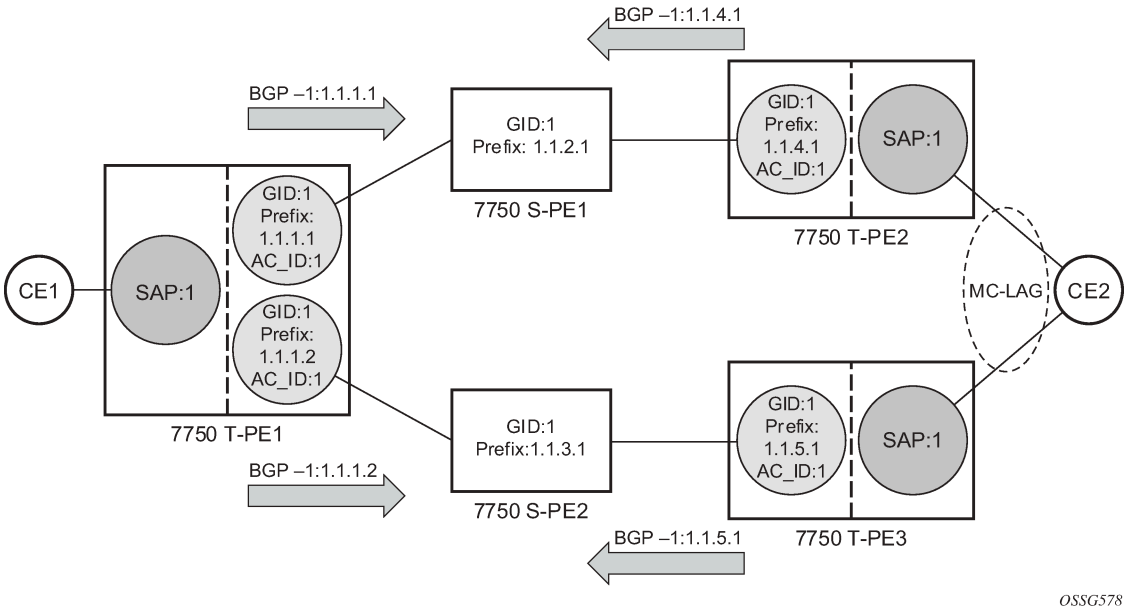
Dynamic MS-PWs support the use of the pseudowire template for specifying generic pseudowire parameters at the T-PE. The pseudowire template to use is configured in the **spoke-sdp-fec**>**pw-template-bind policy-id** context. Dynamic MS-PWs do not support the provisioned SDPs specified in the pseudowire template. Auto-created GRE SDPs are supported with dynamic MS-PWs by creating the PW template used within the **spoke-sdp-fec** with the parameter **auto-gre-sdp**.

2.6.6.4 Pseudowire redundancy

Pseudowire redundancy is supported on dynamic MS-PWs used for VLLs. It is configured in a similar manner to pseudowire redundancy on VLLs using FEC128, whereby each spoke-sdp-fec within an endpoint is configured with a unique SAII/TAII.

Figure 21: Pseudowire redundancy shows the use of pseudowire redundancy.

Figure 21: Pseudowire redundancy



The following is a summary of the key points to consider in using pseudowire redundancy with dynamic MS-PWs:

- Each MS-PW in the redundant set must have a unique SAII/TAII set and is signaled separately. The primary pseudowire is configured in the **spoke-sdp-fec**>**primary** context.
- Each MS-PW in the redundant set should use a diverse path (from the point of view of the S-PEs traversed) from every other MS-PW in that set if path diversity is possible in a specific network topology. There are a number of possible ways to achieve this:
  - Configure an explicit path for each MS-PW.
  - Allow BGP routing to automatically determine diverse paths using BGP policies applied to different local prefixes assigned to the primary and standby MS-PWs.
  - Path diversity can be further provided for each primary pseudowire through the use of a BGP RD.

If the primary MS-PW fails, fail-over to a standby MS-PW occurs, as per the normal pseudowire redundancy procedures. A configurable retry timer for the failed primary MS-PW is then started. When the timer expires, attempts to reestablish the primary MS-PW using its original path occur, up to a maximum number of attempts as per the retry count parameter. On successful reestablishment, the T-PE may then optionally revert to the primary MS-PW.

Because the SDP ID is determined dynamically at signaling time, it cannot be used as a tie breaker to choose the primary MS-PW between multiple MS-PWs of the same precedence. The user should,

therefore, explicitly configure the precedence values to determine which MS-PW is active in the final selection.

2.6.6.5 VCCV OAM for dynamic MS-PWs

The primary difference between dynamic MS-PWs and those using FEC128 is support for FEC129 AII type 2. As in PW Switching, VCCV on dynamic MS-PWs requires the use of the VCCV control word on the pseudowire. Both the **vccv-ping** and **vccv-trace** commands support dynamic MS-PWs.

2.6.6.6 VCCV-ping on dynamic MS-PWs

VCCV-ping supports the use of FEC129 AII type 2 in the target FEC stack of the ping echo request message. The FEC to use in the echo request message is derived in one of two ways, either the user can specify only the **spoke-sdp-fec-id** of the MS-PW in the **vccv-ping** command, or the user can explicitly specify the SAII and TAII to use.

If the SAII:TAII is entered by the user in the **vccv-ping** command, those values are used for the vccv-ping echo request, but their order is reversed before being sent so that they match the order for the downstream FEC element for an S-PE, or the locally configured SAII:TAII for a remote T-PE of that MS-PW. If SAII:TAII is entered as well as the **spoke-sdp-fec-id**, the system verifies the entered values against the values stored in the context for that **spoke-sdp-fec-id**.

Otherwise, if the SAII:TAII to use in the target FEC stack of the vccv-ping message is not entered by the user, and if a switching point TLV was previously received in the initial label mapping message for the reverse direction of the MS-PW (with respect to the sending PE), then the SAII:TAII to use in the target FEC stack of the vccv-ping echo request message is derived by parsing that switching point TLV based on the user-specified TTL (or a TTL of 255 if none is specified). In this case, the order of the SAII:TAII in the switching point TLV is maintained for the vccv-ping echo request message.

If no pseudowire switching point TLV was received, then the SAII:TAII values to use for the vccv-ping echo request are derived from the MS-PW context, but their order is reversed before being sent so that they match the order for the downstream FEC element for an S-PE, or the locally configured SAII:TAII for a remote T-PE of that MS-PW.

The use of **spoke-sdp-fec-id** in **vccv-ping** is only applicable at T-PE nodes because it is not configured for a specified MS-PW at S-PE nodes.

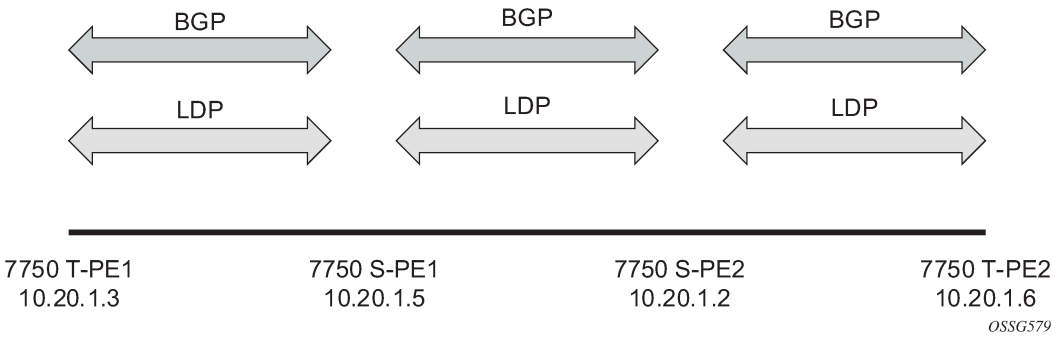
2.6.6.7 VCCV-trace on dynamic MS-PWs

The 7750 SR, 7450 ESS, and 7950 XRS support the MS-PW path trace mode of operation for VCCV trace, as per pseudowire switching, but using FEC129 AII type 2. As in the case of vccv-ping, the SAII:TAII used in the VCCV echo request message sent from the T-PE or S-PE from which the VCCV trace command is executed is specified by the user or derived from the context of the MS-PW. The use of **spoke-sdp-fec-id** in **vccv-trace** is only applicable at T-PE nodes because it is not configured for a specified MS-PW at S-PE nodes.

2.6.7 Example dynamic MS-PW configuration

This section describes an example of how to configure Dynamic MS-PWs for a VLL service between a set of Nokia nodes. The network consists of two T-PEs and two nodes, in the role of S-PEs, as shown in the following figure. Each 7750 SR, 7450 ESS, or 7950 XRS peers with its neighbor using LDP and BGP.

Figure 22: Dynamic MS-PW example



The example uses BGP to route dynamic MS-PWs and T-LDP to signal them. Therefore, each node must be configured to support the MS-PW address family under BGP, and BGP and LDP peerings must be established between the T-PEs/S-PEs. The appropriate BGP export policies must also be configured.

Next, pseudowire routing must be configured on each node. This includes an S-PE address for every participating node, and one or more local prefixes on the T-PEs. MS-PW paths and static routes may also be configured.

When this routing and signaling infrastructure is established, spoke-sdp-fecs can be configured on each of the T-PEs, as follows:

```
config
   router
      ldp
         targeted-session
            peer 10.20.1.5
            exit
         exit
      policy-options
         begin
         policy-statement "exportMsPw"
            entry 10
               from
                  family ms-pw
               exit
               action accept
               exit
            exit
         exit
         commit
      exit
      bgp
         family ms-pw
         connect-retry 1
         min-route-advertisement 1
         export "exportMsPw"
         rapid-withdrawal
         group "ebgp"
            neighbor 10.20.1.5
```

```
               multihop 255
               peer-as 200
            exit
         exit
      exit
config
   service
      pw-routing
         spe-address 3:10.20.1.3
         local-prefix 3:10.20.1.3 create
         exit
         path "path1_to_F" create
            hop 1 10.20.1.5
            hop 2 10.20.1.2
            no shutdown
         exit
      exit
      epipe 1 name "XYZ Epipe 1" customer 1 create
         description "Default epipe
              description for service id 1"
         service-mtu 1400
         sap 2/1/1:1 create
         exit
         spoke-sdp-fec 1 fec 129 aii-type 2 create
            retry-timer 10
            retry-count 10
            saii-type2 3:10.20.1.3:1
            taii-type2 6:10.20.1.6:1
            no shutdown
         exit
         no shutdown
      exit
config
   router
      ldp
         targeted-session
            peer 10.20.1.2
            exit
         exit
         …
      policy-options
         begin
         policy-statement "exportMsPw"
            entry 10
               from
                  family ms-pw
               exit
               action accept
               exit
            exit
         exit
         commit
      exit

      bgp
         family ms-pw
         connect-retry 1
         min-route-advertisement 1
         export "exportMsPw"
         rapid-withdrawal
         group "ebgp"
            neighbor 10.20.1.2
               multihop 255
               peer-as 300
```

```
            exit
         exit
     exit
config
   service
      pw-routing
         spe-address 6:10.20.1.6
         local-prefix 6:10.20.1.6 create
         exit
         path "path1_to_F" create
            hop 1 10.20.1.2
            hop 2 10.20.1.5
            no shutdown
         exit
     exit
     epipe 1 name "XYZ Epipe 1" customer 1 create
        description "Default epipe
             description for service id 1"
service-mtu 1400
        sap 1/1/3:1 create
        exit
        spoke-sdp-fec 1 fec 129 aii-type 2 create
           retry-timer 10
           retry-count 10
           saii-type2 6:10.20.1.6:1
           taii-type2 3:10.20.1.3:1
           no shutdown
        exit
        no shutdown
     exit
```

```
config
   router
      ldp
         targeted-session
            peer 10.20.1.3
            exit
            peer 10.20.1.2
            exit
         exit
         …
      bgp
         family ms-pw
         connect-retry 1
         min-route-advertisement 1
         rapid-withdrawal
         group "ebgp"
            neighbor 10.20.1.2
               multihop 255
               peer-as 300
            exit
            neighbor 10.20.1.3
               multihop 255
               peer-as 100
            exit
         exit
     exit

   service
      pw-routing
         spe-address 5:10.20.1.5
```

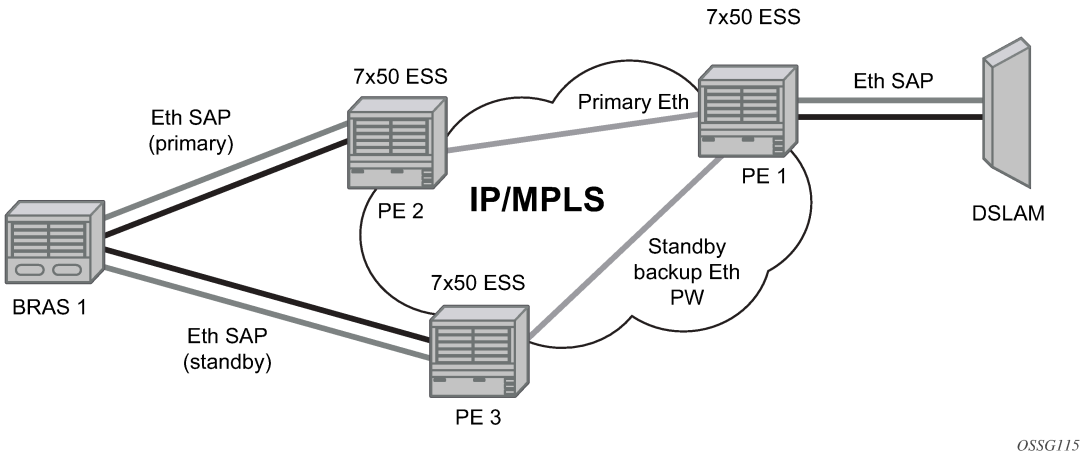
```
      exit
```

```
config
   router
      ldp
         targeted-session
            peer 10.20.1.5
            exit
            peer 10.20.1.6
            exit
         exit
         …
      bgp
         family ms-pw
         connect-retry 1
         min-route-advertisement 1
         rapid-withdrawal
         group "ebgp"
            neighbor 10.20.1.5
               multihop 255
               peer-as 200
            exit
            neighbor 10.20.1.6
               multihop 255
               peer-as 400
            exit
         exit
     exit
service
      pw-routing
         spe-address 2:10.20.1.2
      exit
```

2.6.8 VLL resilience with two destination PE nodes

Figure 23: VLL resilience shows the application of pseudowire redundancy to provide Ethernet VLL service resilience for broadband service subscribers accessing the broadband service on the service provider BRAS.

Figure 23: VLL resilience



If the Ethernet SAP on PE2 fails, PE2 notifies PE1 of the failure by either withdrawing the primary pseudowire label it advertised or by sending a pseudowire status notification with the code set to indicate a SAP defect. PE1 receives it and immediately switches its local SAP to forward over the secondary standby spoke-SDP. To avoid black holing of packets during the switching of the path, PE1 accepts packets received from PE2 on the primary pseudowire while transmitting over the backup pseudowire. However, in other applications such as those described in Access node resilience using MC-LAG and pseudowire redundancy, it is important to minimize service outage to end users.

When the SAP at PE2 is restored, PE2 updates the new status of the SAP by sending a new label mapping message for the same pseudowire FEC or by sending pseudowire status notification message indicating that the SAP is back up. PE1 then starts a timer and reverts to the primary at the expiry of the timer. By default, the timer is set to 0, which means PE1 reverts immediately. A special value of the timer (infinity) means that PE1 should never revert to the primary pseudowire.

The behavior of the pseudowire redundancy feature is the same if PE1 detects or is notified of a network failure that brought the spoke-SDP status to operationally down. The following events cause PE1 to trigger a switchover to the secondary standby pseudowire.

- The T-LDP peer (remote PE) node withdraws the pseudowire label.
- The T-LDP peer signals a FEC status indicating a pseudowire failure or a remote SAP failure.
- The T-LDP session to peer node times out.
- The SDP binding or VLL service goes down as a result of a network failure condition, such as the SDP to the peer node going operationally down, or the SDP binding going operationally down because the VCCV BFD session is going down.

The SDP type for the primary and secondary pseudowires need not be the same. That is, the user can protect an RSVP-TE based spoke-SDP with an LDP or GRE based one. This provides the ability to route the path of the two pseudowires over different areas of the network. All VLL service types, for example, Epipe and Ipipe, are supported on the 7750 SR.

Nokia routers support the ability to configure multiple secondary standby pseudowire paths. For example, PE1 uses the value of the user-configurable precedence parameter associated with each spoke-SDP to select the next available pseudowire path after the failure of the current active pseudowire (whether it is the primary or one of the secondary pseudowires). The revertive operation always switches the path of the VLL back to the primary pseudowire though. There is no revertive operation between secondary paths, meaning that the path of the VLL is not switched back to a secondary pseudowire of higher precedence when the latter comes back up again.

Nokia routers support the ability for a user-initiated manual switchover of the VLL path to the primary or any of the secondary, be supported to divert user traffic in case of a planned outage such as in node upgrade procedures.

On the 7750 SR, this application can make use of all types of VLL supported on SR-series routers. However, if a SAP is configured on an MC-LAG instance, only the Epipe service type is allowed.

2.6.8.1 Master-slave operation

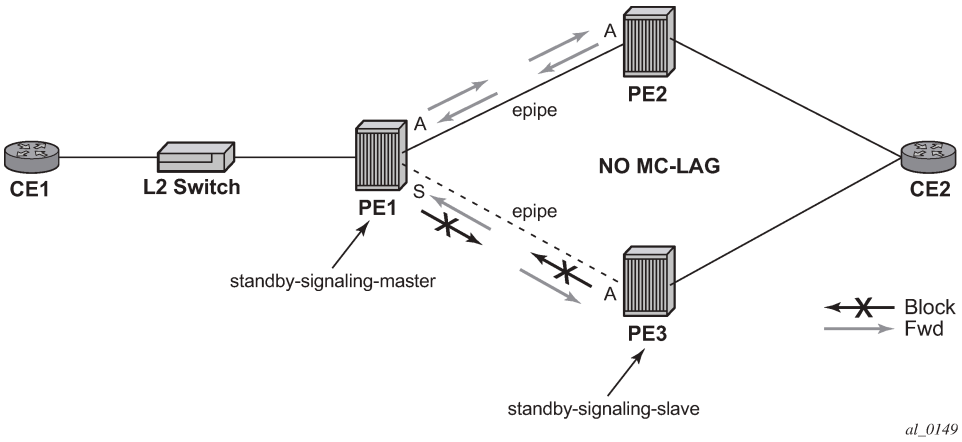
This section describes a mechanism in which one end on a pseudowire (the "master") dictates the active PW selection, which is followed by the other end of the PW (the "slave"). This mechanism and associated terminology is specified in RFC6870.

This section describes master-slave pseudowire redundancy. It adds the ability for the remote peer to react to the pseudowire standby status notification, even if only one spoke-SDP terminates on the VLL endpoint on the remote peer, by blocking the transmit (Tx) direction of a VLL spoke-SDP when the far-end PE

signals standby. This solution enables the blocking of the Tx direction of a VLL spoke-SDP at both master and slave endpoints when standby is signaled by the master endpoint. This approach satisfies a majority of deployments where bidirectional blocking of the forwarding on a standby spoke-SDP is required.

Figure 24: Master-slave pseudowire redundancy shows the operation of master-slave pseudowire redundancy. In this scenario, an Epipe service is provided between CE1 and CE2. CE2 is dual-homed to PE2 and PE3; therefore, PE1 is dual-homed to PE2 and PE3 using Epipe spoke-SDPs. The objective of this feature is to ensure that only one pseudowire is used for forwarding in both directions by PE1, PE2, and PE3 in the absence of a native dual homing protocol between CE2 and PE2/PE3, such as MC-LAG. In normal operating conditions (the SAPs on PE2 and PE3 toward CE2 are both up and there are no defects on the ACs to CE2), PE2 and PE3 cannot choose which spoke-SDP to forward on, based on the status of the AC redundancy protocol.

Figure 24: Master-slave pseudowire redundancy



Master-slave pseudowire redundancy adds the ability for the remote peer to react to the pseudowire standby status notification, even if only one spoke-SDP terminates on the VLL endpoint on the remote peer. When the CLI command **standby-signaling-slave** is enabled at the spoke-SDP or explicit endpoint level in PE2 and PE3, then any spoke-SDP for which the remote peer signals PW FWD Standby is blocked in the transmit direction.

This is achieved as follows. The **standby-signaling-master** state is activated on the VLL endpoint in PE1. In this case, a spoke-SDP is blocked in the transmit direction at this primary endpoint if it is either in operDown state, or it has lower precedence than the highest precedence spoke-SDP, or the specific peer PE signals one of the following pseudowire status bits:

- Pseudowire not forwarding (0x01)
- SAP (ingress) receive fault (0x02)
- SAP (egress) transmit fault (0x04)
- SDP binding (ingress) receive fault (0x08)
- SDP binding (egress) transmit fault (0x10)

That the specified spoke-SDP has been blocked is signaled to the LDP peer through the pseudowire status bit (PW FWD Standby (0x20)). This prevents traffic being sent over this spoke-SDP by the remote peer, but only if that remote peer supports and reacts to pseudowire status notification. Previously, this applied only if the spoke-SDP terminates on an IES, VPRN, or VPLS. However, if **standby-signaling-slave** is enabled at the remote VLL endpoint, the Tx direction of the spoke-SDP is also blocked, according to the rules in Operation of master-slave pseudowire redundancy with existing scenarios.

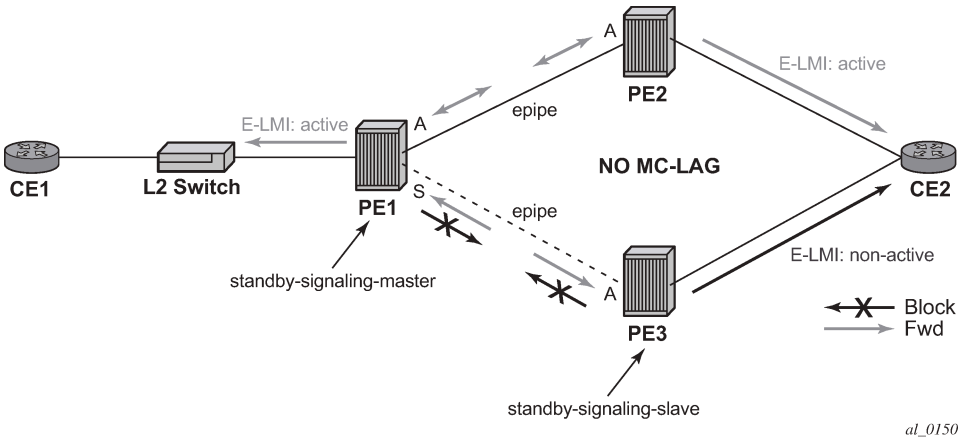
Although master-slave operation provides bidirectional blocking of a standby spoke-SDP during steady-state conditions, it is possible that the Tx directions of more than one slave endpoint can be active for transient periods during a fail-over operation. This is because of the slave endpoints transitioning a spoke-SDP from standby to active receiving or processing a pseudowire preferential forwarding status message, or both, before those endpoints transitioning a spoke-SDP to standby. This transient condition is most likely when a forced switchover is performed, or the relative preferences of the spoke-SDPs are changed, or the active spoke-SDP is shutdown at the primary endpoint. During this period, loops of unknown traffic may be observed. Fail-overs because of common network faults that can occur during normal operation, or a failure of connectivity on the path of the spoke-SDP or the SAP, would not result in such loops in the data path.

2.6.8.1.1 Interaction with SAP-specific OAM

If all of the spoke-SDPs bound to a SAP at a slave PE are selected as standby, then this should be treated from a SAP OAM perspective in the same manner as a fault on the service: an SDP binding down or remote SAP down. That is, a fault should be indicated to the service manager. If SAP-specific OAM is enabled toward the CE, such as Ethernet Continuity Check Message (CCM), Ethernet Link Management Interface (E-LMI), or FR LMI, then this should result in the appropriate OAM message being sent on the SAP. This can enable the remote CE to avoid forwarding traffic toward a SAP that drops it.

Figure 25: Example of SAP OAM interaction with master-slave pseudowire redundancy shows an example for the case of Ethernet LMI.

Figure 25: Example of SAP OAM interaction with master-slave pseudowire redundancy



2.6.8.1.2 Local rules at slave VLL PE

It is not possible to configure standby-signaling-slave on endpoints or spoke-SDPs that are bound to an IES, VPRN, ICB, or MC-EP, or that are part of an MC-LAG or MC-APS.

If **standby-signaling-slave** is configured on a specific spoke-SDP or explicit endpoint, then the following rules apply. The rules describe the case of several spoke-SDPs in an explicit endpoint. The same rules apply to the case of a single spoke-SDP outside of an endpoint where no endpoint exists:

- Rule for processing endpoint SAP active/standby status bits:

Because the SAP in endpoint X is never a part of an MC-LAG/MC-APS instance, a forwarding status of active is always advertised.

- Rules for processing and merging local and received endpoint objects with an up or down operational status:
- Endpoint X is operationally up if at least one of its objects is operationally up. It is Down if all of its objects are operationally down.
- If all objects in endpoint X performed any or all of the following operations, the node must send status bits of SAP down over all Y endpoint spoke-SDPs:
  - transitioned locally to down state
  - received a SAP down notification via remote T-LDP or via SAP-specific OAM signal
  - received status bits of SDP-binding down
  - received states bits of PW not forwarding
- Endpoint Y is operationally up if at least one of its objects is operationally up. It is down if all its objects are operationally down.
- If a spoke-SDP in endpoint Y, including the ICB spoke-SDP, transitions locally to down state, the node must send T-LDP SDP-binding down status bits on this spoke-SDP.
- If a spoke-SDP in endpoint Y received T-LDP SAP down status bits, T-LDP SDP-binding down status bits, or status bits of PW not forwarding, the node saves this status and takes no further action. The saved status is used for selecting the active transmit endpoint object.
- If all objects in endpoint Y, or a single spoke-SDP that exists outside of an endpoint (and no endpoint exists), the node must send a SAP down notification on the X endpoint SAP via the SAP-specific OAM signal, if applicable:
  - transitioned locally to down state
  - received status bits of T-LDP SAP down
  - received status bits of T-LDP SDP-binding down
  - received status bits of PW not forwarding
  - received status bits of PW FWD standby
- If the peer PE for a specified object in endpoint Y signals PW FWD standby, the spoke-SDP must be blocked in the transmit direction and the spoke-SDP is not eligible for selection by the active transmit selection rules.
- If the peer PE for a specified object in endpoint Y does not signal PW FWD standby, then spoke-SDP is eligible for selection.

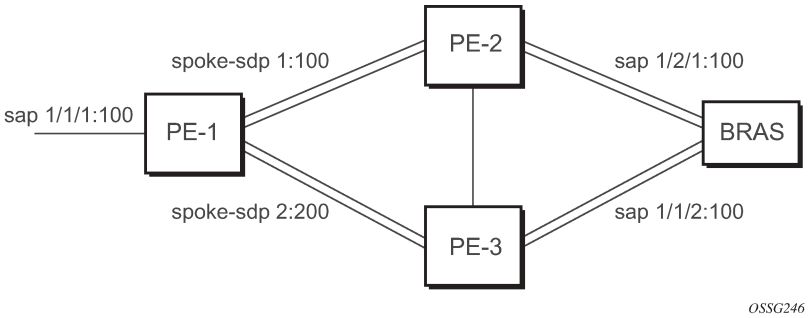
2.6.8.1.3 Operation of master-slave pseudowire redundancy with existing scenarios

This section discusses how master-slave pseudowire redundancy could operate.

2.6.8.1.3.1 VLL resilience path example

Figure 26: VLL resilience shows a VLL resilience path example. An example configuration follows.

Figure 26: VLL resilience



A **revert-time** value of zero (default) means that the VLL path is switched back to the primary immediately after it comes back up.

```
PE-1
configure service epipe 1
endpoint X
exit
endpoint Y
   revert-time 0
   standby-signaling-master
   exit
   sap 1/1/1:100 endpoint X
   spoke-sdp 1:100 endpoint Y
precedence primary
   spoke-sdp 2:200 endpoint Y
precedence 1

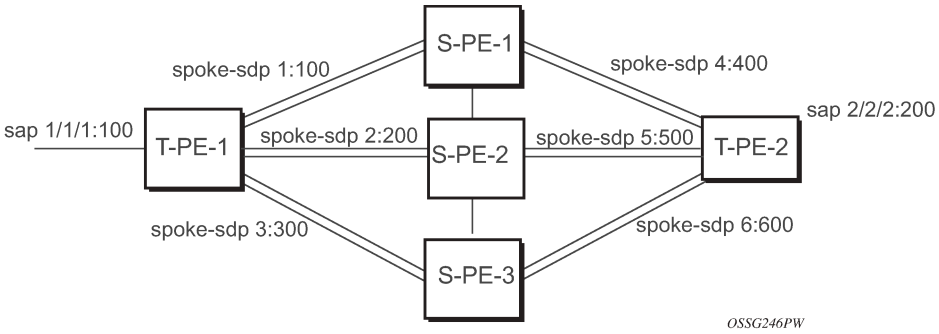
PE-2
configure service epipe 1
   endpoint X
   exit
   sap 2/2/2:200 endpoint X
   spoke-sdp 1:100
      standby-signaling-slave

PE-3
configure service epipe 1
   endpoint X
   exit
   sap 3/3/3:300 endpoint X
   spoke-sdp 2:200
      standby-signaling-slave
```

2.6.8.1.4 VLL resilience for a switched pseudowire path

Figure 27: VLL resilience with pseudowire switching displays VLL resilience for a switched pseudowire path example. An example configuration follows.

Figure 27: VLL resilience with pseudowire switching




```
T-PE-1
configure service epipe 1
   endpoint X
   exit
   endpoint Y
   revert-time 100
   standby-signaling-master
   exit
   sap 1/1/1:100 endpoint X
   spoke-sdp 1:100 endpoint Y
      precedence primary
   spoke-sdp 2:200 endpoint Y
      precedence 1
   spoke-sdp 3:300 endpoint Y
      precedence 1

T-PE-2
configure service epipe 1
   endpoint X
   exit
   endpoint Y
   revert-time 100
   standby-signaling-slave
   exit
   sap 2/2/2:200 endpoint X
   spoke-sdp 4:400 endpoint Y
      precedence primary
   spoke-sdp 5:500 endpoint Y
      precedence 1
   spoke-sdp 6:600 endpoint Y
      precedence 1
```

VC switching indicates a VC cross-connect so that the service manager does not signal the VC label mapping immediately but puts S-PE-1 into passive mode, as follows:

```
configure service epipe 1 vc-switching
   spoke-sdp 1:100
   spoke-sdp 4:400
```

2.6.9 Pseudowire SAPs

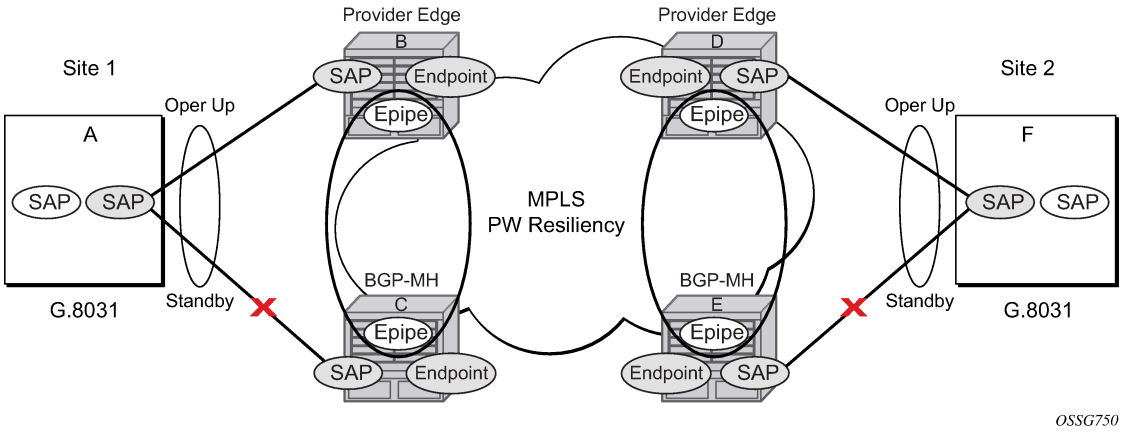
For information about how to use pseudowire SAPs with Layer 2 services, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN*.

2.6.10 Epipe using BGP-MH site support for Ethernet tunnels

Using Epipe in combination with G.8031 and BGP multihoming in the same manner as VPLS offers a multi-chassis resiliency option for Epipe services that is a non-learning and non-flooded service. MC-LAG (see Access node resilience using MC-LAG and pseudowire redundancy) offers access node redundancy with active/stand-by links while Ethernet tunnels offer per service redundancy with all active links and active or standby services. G.8031 offers an end-to-end service resiliency for Epipe and VPLS services. BGP-MH site support for Ethernet tunnels offers Ethernet edge resiliency for Epipe services that integrates with MPLS pseudowire redundancy.

Figure 28: BGP-MH site support for Ethernet tunnels shows the BGP-MH site support for Ethernet tunnels, where a G.8031 edge device (A) is configure with two provider edge switches (B and C). G.8031 is configured on the Access devices (A and F). An Epipe endpoint service is configured along with BGP Multihoming and Pseudowire Redundancy on the provider edge nodes (B/C and D/E). This configuration offers a fully redundant Epipe service.

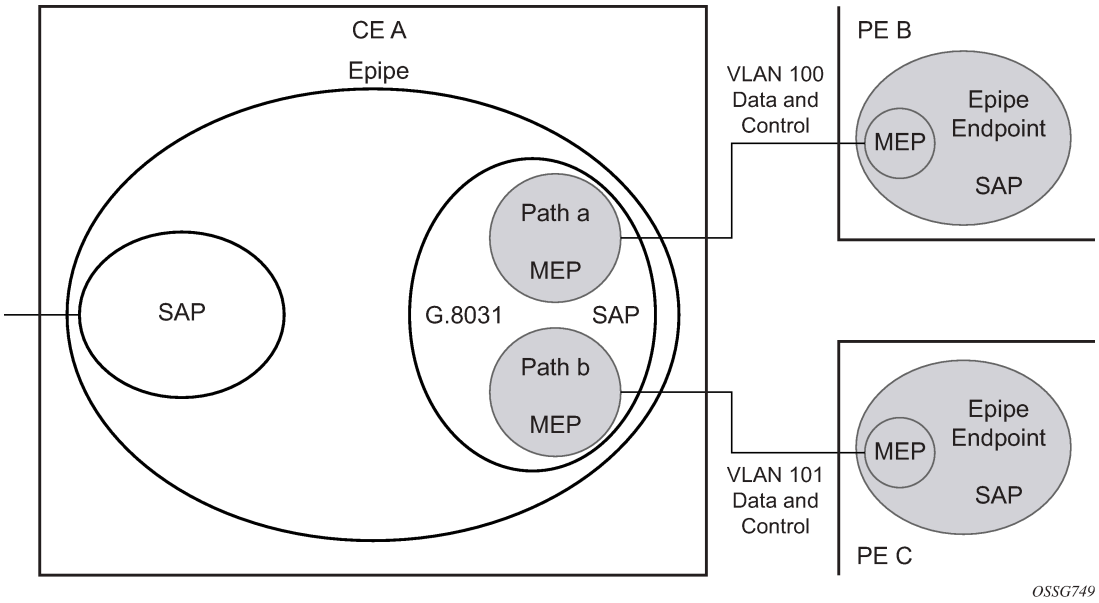
Figure 28: BGP-MH site support for Ethernet tunnels



2.6.10.1 Operational overview

G.8031 offers a number of redundant configurations. Normally, it offers the ability to control two independent paths for 1:1 protection. In the BGP-MH site support for Ethernet tunnels case, BGP drives G.8031 as a slave service. In this case, the provider edge operates using only standard 802.1ag MEPs with CCM to monitor the paths. Figure 29: G.8031 for slave operation shows an Epipe service on a Customer Edge (CE) device that uses G.8031 with two paths and two MEPs. The paths can use a single VLAN of dot1q or QinQ encapsulation.

Figure 29: G.8031 for slave operation



In a single-service deployment, the control (CFM) and data share the same port and VID. For multiple services for scaling, fate sharing is allowed between multiple SAPs, but all SAPs within a group must be on the same physical port.

To get fate sharing for multiple services with this feature, a dedicated G.8031 CE-based service (one VLAN) is connected to a Epipe SAP on a PE, which uses BGP-MH and operational groups to control other G.8031 tunnels. This dedicated G.8031 service still has data control capabilities, but the data Epipe service is not bearing user data packets. On the CE, this G.8031 is only used for group control. Making this a dedicated control (CFM) for a set of G.8031 tunnels is to simplify operation and allow individual disabling of services. Using a dedicated G.8031 service to both control and to carry data traffic is allowed.

Fate sharing from the PE side is achieved using BGP and operational groups. G.8031 Epipe services can be configured on the CE as regular non-fate shared G.8031 services, but because of the configuration on the PE side, these Ethernet tunnels are treated as a group following the one designated control service. The G.8031 control logic on the CE is a slave to the BGP-MH control.

On the CE, G.8031 allows independent configuration of VIDs on each path. On the PE, the Epipe or endpoint that connects to the G.8031 service must have a SAP with the corresponding VID. If the G.8031 service has a Maintenance End Point (MEP) for that VID, the SAP should be configured with a MEP. The MEPs on the paths on the CE signal standard interface status TLV (ifStatusTLV), No Fault (Up), and Fault (Down). The MEPs on the PE (Epipe or endpoint) also use signaling of ifStatusTLV No Fault (Up), and Fault (Down) to control the G.8031 SAP. However, in the 7750 SR, 7450 ESS, and 7950 XRS model, fate shared Ethernet tunnels with no MEP are allowed. In this case, it is up to the CE to manage these CE-based fate shared tunnels.

Interface status signaling (ifStatusTLV) is used to control the G.8031 tunnel from the PE side. Normally the CE signals No Fault (Up) in the path SAP MEP ifStatusTLV before the BGP-MH causes the SAP MEP to become active by signaling No Fault (Up).

2.6.10.2 Detailed operation

For this feature, BGP-MH is used as the master control and the Ethernet tunnel is a slave. The G.8031 on the CE is unaware that it is being controlled. While a single Epipe service is configured and serves as the control for the CE connection, allowing fate sharing, all signaling to the CE is based on the ifStatusTLV per G.8031 tunnel. By controlling G.8031 with BGP-MH, the G.8031 CE is forced to be a slave to the PE BGP-MH election. BGP-MH election is controlled by the received VPLS preference or BGP local-preference, or the PE ID (IP address of provider edge) if local-preference is equal to VPLS preference. There may be traps generated on the CE side for some G.8031 implementations, but these can be suppressed or filtered to allow this feature to operate.

There are two configuration options:

- Every G.8031 service SAP terminates on a single Epipe that has BGP-MH. These Epipes may use endpoints with or without ICBs.
- A control Epipe service monitors a single SAP that is used for group control of fate shared CE services. In this case, the Epipe service has an SAP that serves as the control termination for one Ethernet tunnel connection. The group fate sharing SAPs may or may not have MEPs if they use shared fate. In this case, the Epipe may have endpoints but does not support ICBs.

The MEP ifStatusTlv and CCM are used for monitoring the PE to CE SAP. MEP ifStatusTlv is used to signal that the Ethernet tunnel inactive and CCM is used as an aliveness mechanism. There is no G.8031 logic on the PE; the SAP is controlling the corresponding CE SAP.

2.6.10.2.1 Example operation of G.8031 BGP-MH

Any Ethernet tunnel actions (force, lock) on the CE (single site) do not control the action to switch paths directly, but they may influence the outcome of BGP-MH if they are on a control tunnel. If a path is disabled on the CE, the result may force the SAP with an MEP on the PE to eventually take the SAP down; Nokia recommends running commands from the BGP-MH side to control these connections.

Figure 30: Full redundancy G.8031 Epipe and BGP-MH

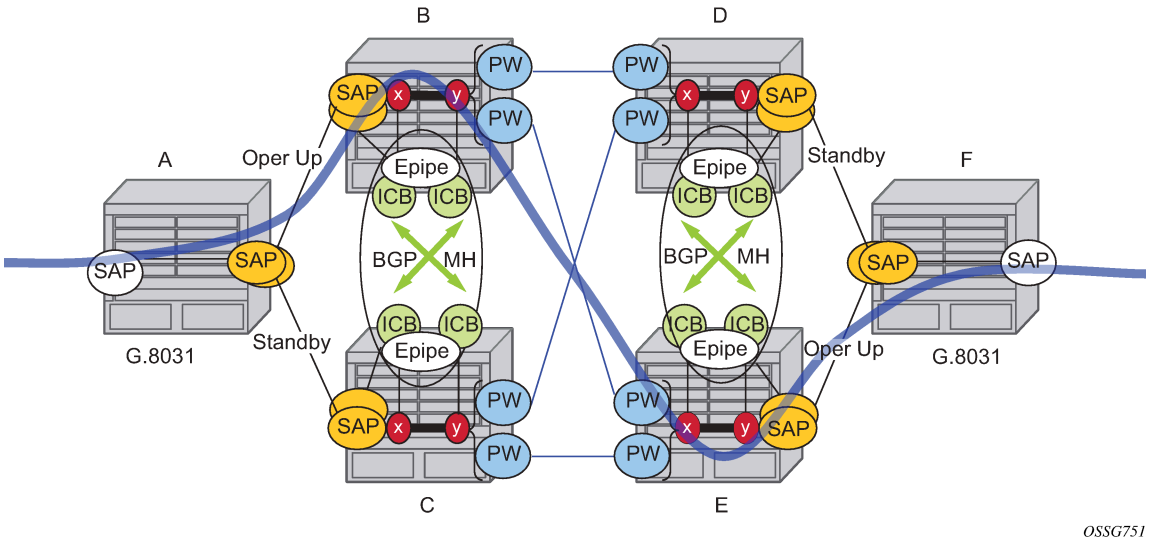


Table 7: SAP MEP signaling lists the SAP MEP signaling shown in Figure 30: Full redundancy G.8031 Epipe and BGP-MH. For a description of the events shown in this example operation, see Events in example operation.

Table 7: SAP MEP signaling

	G.8031 ET on CE	Path A MEP facing node B local ifStatus	Path B MEP facing node C local ifStatus	Path B PE MEP if Status	Path B PE MEP if Status
1	Down (inactive)	No Fault[1]	No Fault	Fault	Fault
2	Up use Path A	No Fault	No Fault	No Fault	Fault
3	Up use Path B	No Fault	No Fault	Fault	No Fault
4	Down Path A fault	Fault[2]	No Fault	Fault	Fault
5	Down Path A and B fault at A	Fault	No Fault	Fault	Fault
6	Partitioned Network Use Path Precedence Up use Path A	No Fault	No Fault	No Fault	No Fault

2.6.10.2.1.1 Events in example operation

The following describes the events for switchover in Figure 30: Full redundancy G.8031 Epipe and BGP-MH. This configuration uses operational groups. The nodes of interest are A, B, and C listed in Table 7: SAP MEP signaling.

1. A single G.8031 SAP that represents the control for a group of G.8031 SAPs is configured on the CE.
   - The Control SAP does not normally carry any data; however, it can if needed.
   - An Epipe service is provisioned on each PE node (B/C), only for control (no customer traffic flows over this service).
   - On CE A, there is an Epipe Ethernet tunnel (G.8031) control SAP.
   - The Ethernet tunnel has two paths:
     – one facing B
     – one facing C
   - PE B has an Epipe control SAP that is controlled by the BGP-MH site and PE C also has the corresponding SAP that is controlled by the same BGP-MH site.
2. At node A, there are MEPs configured under each path that check connectivity on the A-B and A-C links. At nodes B and C, there is a MEP configured under their respective SAPs with fault propagation enabled with the use of ifStatusTlv.

[1] No Fault = no ifStatusTlv transmit | CCM transmit normally

[2] Fault = ifStatusTlv transmit down | no CCM transmit

3. Initially, assume there is no link failure:
   - SAPs on node A have ifStatusTLV No Fault to B and C (no MEP fault detected at A); see Table 7: SAP MEP signaling row 1 (Fault is signaled in the other direction PE to CE).
   - BGP-MH determines which is the master or Designated Forwarder (DF).
   - Assume SAP on node B is picked as the DF.
   - The MEP at Path A-B signals ifStatusTlv No Fault. Because of this signal, the MEP under the node A path facing node B detects the path to node B is usable by the path manager on A.
4. At the CE node A, Path A-C becomes standby and is brought down; see Table 7: SAP MEP signaling row 2.
   - Because fault propagation is enabled under the SAP node C MEP, and ifStatusTLV is operationally Down, the Path remains in the present state.
   - Under these conditions, the MEP under the node A path facing node C detects the fault and informs Ethernet manager on node A.
   - Node A then considers bringing path A-C down.
   - ET port remains up because the path A-B is operationally up. This is a stable state.
5. On nodes B and C, each Epipe-controlled SAP is the sole (controlling) member of an operational group.
   - Other data SAPs may be configured for fate shared VLANs (Ethernet tunnels) and to monitor the control SAP.
   - The SAPs facing the CE node A share the fate of the control SAP and follow the operation.

If there is a break in path A-B connectivity (CCM timeout or LOS on the port for link A-B), the following actions take place:

1. On node A, the path MEP detects connectivity failure and informs Ethernet tunnel manager; see Table 7: SAP MEP signaling row 4.
2. At this point, the Ethernet tunnel is down because both path A-B and path A-C are down.
3. The CE node A Ethernet tunnel goes down.
4. At node B on the PE, the SAP also detects the failure and the propagation of fault status goes to BGP-MH; see Table 7: SAP MEP signaling row 4.
5. This in turn feeds into BGP-MH, which deems the site non-DF and makes the site standby.
6. Because the SAP at node B is standby, the service manager feeds this to CFM, which then propagates a Fault toward node A. This is a cyclic fault propagation. However, because path A-B is broken, the situation is stable; see Table 7: SAP MEP signaling row 5.
7. There is traffic loss during the BGP-MH convergence.
   - Load sharing mode is recommended when using a 7450 as a CE node A device.
   - BGP-MH signals that node C is now the DF; see Table 7: SAP MEP signaling row 3.
8. BGP-MH on node C elects a SAP and brings it up.
9. ET port transitions to port A-C, and is operationally up. This is a stable state. The A-C SAPs monitoring the operational group on C transitions to operationally up.

Unidirectional failures: at point 6 the failure was detected at both ends. In the case of a unidirectional failure, CCM times out on one side.

- In the case where the PE detects the failure, it propagates the failure to BGP-MH and the BGP-MH takes the site down causing the SAPs on the PE to signal a Fault to the CE.
- In the case where G.8031 on the CE detects the failure, it takes the tunnel down and signals a fault to the PE, and then the SAP propagates that to BGP-MH.

2.6.10.3 BGP-MH site support for Ethernet tunnels operational group model

For operational groups, one or more services follow the controlling service. On node A, there is an ET SAP facing nodes B/C, and on nodes B/C there are SAPs of the Epipe on physical ports facing node A. Each of the PE data SAPs monitor their respective operational groups, meaning they are operationally up or down based on the operational status of the control SAPs. On node A, because the data SAP is on the ET logical port, it goes operationally down whenever the ET port goes down and similarly for going operationally up.

Alternatively, an Epipe service may be provisioned on each node for each G.8031 data SAP (one-for-one service with no fate sharing). On CE node A, there is a G.8031 Ethernet tunnel. The Ethernet tunnel has two paths: one facing node B and one facing node C. This option is the same as the control SAP, but there are no operational groups. However, now there is a BGP-MH site per service. For large sites, operational groups are more efficient.

2.6.10.4 BGP-MH specifics for MH site support for Ethernet tunnels

BGP multihoming for VPLS describes the procedures for using BGP to control resiliency for VPLS. These procedures are the same except that an Epipe service can be configured for BGP-MH.

2.6.10.5 PW redundancy for BGP-MH site support for Ethernet tunnels

Pseudowire redundancy service models and Figure 33: VLL resilience with pseudowire redundancy and switching are used for the MPLS network resiliency. BGP MH site support for Ethernet tunnels reuses this model.

2.6.10.6 T-LDP status notification handling rules of BGP-MH Epipes

Using Figure 33: VLL resilience with pseudowire redundancy and switching as a reference, the following are the rules for generating, processing, and merging T-LDP status notifications in VLL service with endpoints.

2.6.10.6.1 Rules for processing endpoint SAP active/standby status bits

1. The advertised admin forwarding status of active/standby reflects the status of the local Epipe SAP in BGP-MH instance. If the SAP is not part of an MC-LAG instance or a BGP-MH instance, the forwarding status of Active is always advertised.
   - When the SAP in endpoint X is part of a BGP-MH instance, a node must send T-LDP forwarding status bit of SAP active/standby over all Y endpoint spoke-SDPs, except the ICB spoke-SDP, whenever this (BGP-MH designated forwarder) status changes. The status bit sent over the ICB is always zero (Active by default).

- When the SAP in endpoint X is not part of an MC-LAG instance or BGP-MH instance, then the forwarding status sent over all Y endpoint spoke-SDPs should always be set to zero (Active by default).

2. The received SAP active/standby status is saved and used for selecting the active transmit endpoint object Pseudowire Redundancy procedures.

2.6.10.6.2 Rules for processing, merging local, and received endpoint operational status

- Endpoint X is operationally up if at least one of its objects is operationally Up. It is Down if all its objects are operationally down.
- If the SAP in endpoint X transitions locally to the down state, or received a SAP Down notification via SAP-specific OAM signal (SAP MEP), the node must send T-LDP SAP down status bits on the Y endpoint ICB spoke-SDP only. BGP-MH SAPs support MEPs for ifStatusTLV signaling. All other SAP types cannot exist on the same endpoint as an ICB spoke-SDP because non Ethernet SAP cannot be part of an MC-LAG instance or a BGP-MH Instance.
- If the ICB spoke-SDP in endpoint X transitions locally to Down state, the node must send T-LDP SDP-binding down status bits on this spoke-SDP.
- If the ICB spoke-SDP in endpoint X received T-LDP SDP-binding down status bits or PW not forwarding status bits, the node saves this status and takes no further action. The saved status is used for selecting the active transmit endpoint object as per Pseudowire Redundancy procedures.
- If all objects in endpoint X performed any or all of the following operations, the node must send status bits of SAP Down over all Y endpoint spoke-SDPs, including the ICB:
  - transitioned locally to the down state because of the operator or BGP-MH DF election
  - received a SAP down notification via remote T-LDP status bits or via SAP-specific OAM signal (SAP MEP)
  - received status bits of SDP-binding down
  - received status bits of PW not forwarding
- Endpoint Y is operationally up if at least one of its objects is operationally Up. It is Down if all its objects are operationally down.
- If a spoke-SDP in endpoint Y, including the ICB spoke-SDP, transitions locally to down state, the node must send T-LDP SDP-binding down status bits on this spoke-SDP.
- If a spoke-SDP in endpoint Y, including the ICB spoke-SDP, performed any or all of the following operations, the node saves this status and takes no further action:
  - received T-LDP SAP down status bits
  - received T-LDP SDP-binding down status bits
  - received PW not forwarding status bits

  The saved status is used for selecting the active transmit endpoint object as per Pseudowire Redundancy procedures.
- If all objects in endpoint Y, except the ICB spoke-SDP, performed any or all of the following operations, the node must send status bits of SDP-binding down over the X endpoint ICB spoke-SDP only:
  - transitioned locally to the down state
  - received T-LDP SAP down status bits

  - received T-LDP SDP-binding down status bits
  - received PW not forwarding status bits
- If all objects in endpoint Y performed any or all of the following operations, the node must send status bits of SDP-binding down over the X endpoint ICB spoke-SDP only, and must send a SAP down notification on the X endpoint SAP via the SAP-specific OAM signal:
  - transitioned locally to down state
  - received T- LDP SAP down status bits
  - received T-LDP SDP-binding down status bits
  - received PW not forwarding status bits

  In this case the SAP MEP ifStatusTLV is operationally down and also signals the BGP-MH site, if this SAP is part of a BGP Site.

2.6.10.6.3 Operation for BGP-MH site support for Ethernet tunnels

A multihomed site can be configured on up to four PEs although two PEs are enough for most applications, with each PE having a single object SAP connecting to the multihomed site. SR OS G.8031 implementation with load sharing allows multiple PEs as well. The designated forwarder election chooses a single connection to be operationally up, with the other placed in standby. Only revertive behavior is supported.

Fate sharing (the status of one site can be inherited from another site) is achievable using monitor-groups.

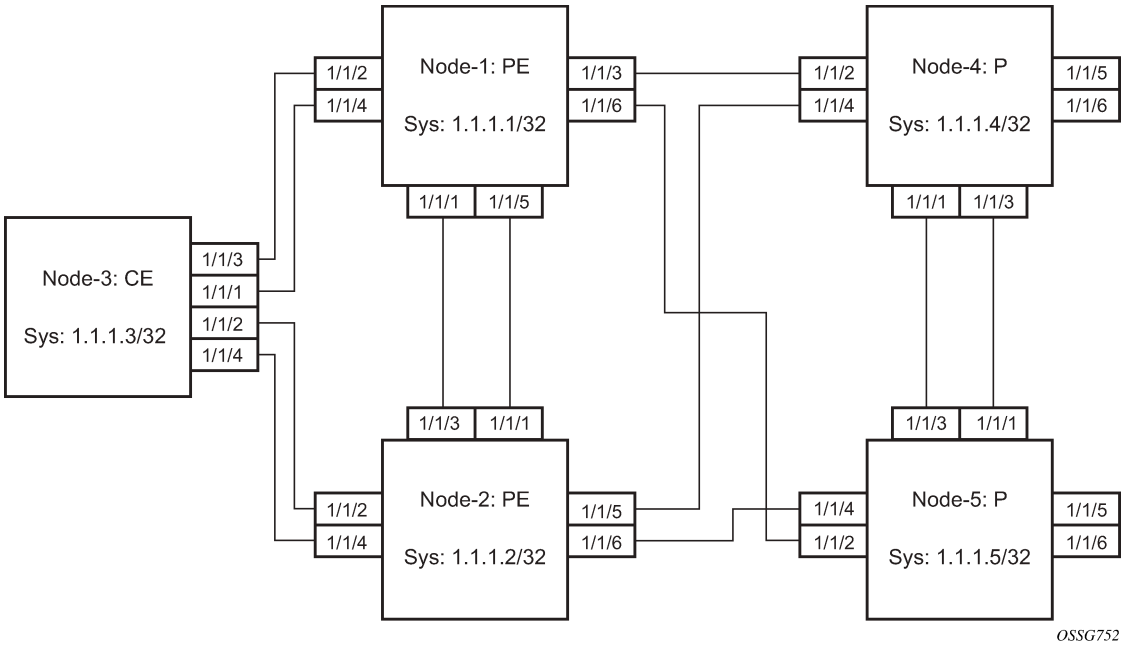
The following are supported:

- All Ethernet-tunnel G.8031 SAPs on CE:
  - 7750 SR, 7450 ESS, or 7950 XRS G.8031 in load sharing mode (recommended)
  - 7750 SR, 7450 ESS, or 7950 XRS G.8031 in non-load sharing mode
- Epipe and endpoint with SAPs on PE devices.
- Endpoints with PW.
- Endpoints with active/standby PWs.

There are the following constraints with this feature:

- Not supported with PBB Epipes.
- Spoke SDP (pseudowire).
  - BGP signaling is not supported.
  - Cannot use BGP MH for auto-discovered pseudowire. This is achieved in a VPLS service using SHGs, which are not available in Epipes.
- Other multi-chassis redundancy features are not supported on the multihomed site object, as follows:
  - MC-LAG
  - MC-EP
  - MC-Ring
  - MC-APS
- Master and slave pseudowire is not supported.

Figure 31: Example topology full redundancy



See the following Configuration examples for configuration examples derived from Figure 31: Example topology full redundancy.

2.6.10.6.3.1 Configuration examples

Node-1: Using operational groups and Ethernet CFM per SAP

```
#--------------------------------------------------
echo "Eth-CFM Configuration"
#--------------------------------------------------
    eth-cfm
        domain 100 format none level 3
            association 2 format icc-based name "node-3-site-1-0"
                bridge-identifier 1
                exit
                remote-mepid 310
            exit
            association 2 format icc-based name "node-3-site-1-1"
                bridge-identifier 100
                exit
                remote-mepid 311
            exit
        exit
    exit

#--------------------------------------------------
echo "Service Configuration"
#--------------------------------------------------
    service
        customer 1 create
            description "Default customer"
```

```
        exit
        sdp 2 mpls create
            far-end 10.1.1.4
            lsp "to-node-4-lsp-1"
            keep-alive
                shutdown
            exit
            no shutdown
        exit
        sdp 3 mpls create  //  Etcetera

        pw-template 1 create
            vc-type vlan
        exit
        oper-group "og-name-et" create
        exit
        oper-group "og-name-et100" create
        exit
        epipe 1 customer 1 create
            service-mtu 500
            bgp
                route-distinguisher 65000:1
                route-target export target:65000:1 import target:65000:1
            exit
            site "site-1" create
                site-id 1
                sap 1/1/2:1.1
                boot-timer 100
                site-activation-timer 2
                no shutdown
            exit
            endpoint "x" create
            exit
            endpoint "y" create
            exit
            sap 1/1/2:1.1 endpoint "x" create
                eth-cfm
                    mep 130 domain 100 association 2 direction down
                        fault-propagation-enable use-if-tlv
                        ccm-enable
                        no shutdown
                    exit
                exit
                oper-group "og-name-et"
            exit
            spoke-sdp 2:1 endpoint "y" create
                precedence primary
                no shutdown
            exit
            spoke-sdp 3:1 endpoint "y" create
                precedence 2
                no shutdown
            exit
            no shutdown
        exit
        epipe 100 customer 1 create
            description "Epipe 100 in separate opergroup"
            service-mtu 500
            bgp
                route-distinguisher 65000:2
                route-target export target:65000:2 import target:65000:2
            exit
            site "site-name-et100" create
                site-id 1101
```

```
                sap 1/1/4:1.100
                boot-timer 100
                site-activation-timer 2
                no shutdown
            exit

            endpoint "x" create
            exit
            endpoint "y" create
            exit
            sap 1/1/4:1.100 endpoint "x" create
                    eth-cfm
                    mep 131 domain 1 association 2 direction down
                        fault-propagation-enable use-if-tlv
                        ccm-enable
                        no shutdown
                    exit
                exit
                oper-group "og-name-et100"

            exit
            spoke-sdp 2:2 vc-type vlan endpoint "y" create
                precedence 1
                no shutdown
            exit
            spoke-sdp 3:2 vc-type vlan endpoint "y" create
                precedence 2
                no shutdown
            exit
            no shutdown
        exit

    exit
#--------------------------------------------------
echo "BGP Configuration"
#--------------------------------------------------
        bgp
            rapid-withdrawal
            rapid-update l2-vpn
            group "internal"
                type internal
                neighbor 10.1.1.2
                    family l2-vpn
                exit
            exit
        exit
    exit
```

Node-3: Using operational groups and Ethernet CFM per SAP

```
#--------------------------------------------------
echo "Eth-CFM Configuration"
#--------------------------------------------------
    eth-cfm
        domain 100 format none level 3
            association 2 format icc-based name "node-3-site-1-0"
                bridge-identifier 1
                exit
                ccm-interval 1
                remote-mepid 130
            exit
            association 2 format icc-based name "node-3-site-1-1"
                bridge-identifier 100
```

```
                exit
                ccm-interval 1
                remote-mepid 131
            association 3 format icc-based name "node-3-site-2-0"
                bridge-identifier 1
                exit
                ccm-interval 1
                remote-mepid 120
            exit
            association 3 format icc-based name "node-3-site-2-1"
                bridge-identifier 100
                exit
                ccm-interval 1
                remote-mepid 121
            exit
        exit
    exit

#--------------------------------------------------
echo "Service Configuration"
#--------------------------------------------------

  eth-tunnel 1
        description "Eth Tunnel loadsharing mode QinQ example"
        protection-type loadsharing
        ethernet
            encap-type qinq
        exit
        path 1
            member 1/1/3
            control-tag 1.1
            eth-cfm
                mep 310 domain 100 association 2
                    ccm-enable
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        path 2
            member 1/1/4
            control-tag 1.2
            eth-cfm
                mep 320 domain 100 association 3
                    ccm-enablepath
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        no shutdown
    exit
#--------------------------------------------------
echo "Ethernet Tunnel Configuration"
#--------------------------------------------------
    eth-tunnel 2
        description "Eth Tunnel QinQ"
        revert-time 10
        path 1
            precedence primary
            member 1/1/1
            control-tag 1.100
```

```
            eth-cfm
                mep 311 domain 100 association 2
                    ccm-enable
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        path 2
            member 1/1/2
            control-tag 1.100
            eth-cfm
                mep 321 domain 100 association 3
                    ccm-enable
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        no shutdown
    exit
#--------------------------------------------------
echo "Service Configuration"
#--------------------------------------------------
    service
        epipe 1 customer 1 create
            sap 2/1/2:1.1 create
            exit
            sap eth-tunnel-1 create
            exit
            no shutdown
        exit
        epipe 100 customer 1 create
            service-mtu 500
            sap 2/1/10:1.100 create
            exit
            sap eth-tunnel-2 create
            exit
            no shutdown
        exit
```

2.6.10.6.3.2 Configuration with fate sharing on node-3

In this example, the SAPs monitoring the operational groups do not need CFM if the corresponding SAP on the CE side is using fate sharing.

Node-1:

```
#--------------------------------------------------
echo "Service Configuration"  Oper-groups
#--------------------------------------------------
    service
        customer 1 create
            description "Default customer"
        exit
        sdp 2 mpls create
          ...
```

```
        exit
        pw-template 1 create
            vc-type vlan
        exit
        oper-group "og-name-et" create
        exit
        epipe 1 customer 1 create
            service-mtu 500
            bgp
                route-distinguisher 65000:1
                route-target export target:65000:1 import target:65000:1
            exit
            site "site-1" create
                site-id 1
                sap 1/1/2:1.1
                boot-timer 100
                site-activation-timer 2
                no shutdown
            exit
            endpoint "x" create
            exit
            endpoint "y" create
            exit
            sap 1/1/2:1.1 endpoint "x" create
                eth-cfm
                    mep 130 domain 100 association 1 direction down
                        fault-propagation-enable use-if-tlv
                        ccm-enable
                        no shutdown
                    exit
                exit
                oper-group "og-name-et"
            exit
            spoke-sdp 2:1 endpoint "y" create
                precedence primary
                no shutdown
            exit
            spoke-sdp 3:1 endpoint "y" create
                precedence 2
                no shutdown
            exit
            no shutdown
        exit
        epipe 2 customer 1 create
            description "Epipe 2 in opergroup with Epipe 1"
            service-mtu 500
            bgp
                route-distinguisher 65000:2
                route-target export target:65000:2 import target:65000:2
            exit
            endpoint "x" create
            exit
            endpoint "y" create
            exit
            sap 1/1/2:1.2 endpoint "x" create
                monitor-oper-group "og-name-et"
            exit
            spoke-sdp 2:2 vc-type vlan endpoint "y" create
                precedence 1
                no shutdown
            exit
            spoke-sdp 3:2 vc-type vlan endpoint "y" create
                precedence 2
                no shutdown
```

```
            exit
            no shutdown
        exit

    exit
```

Node-3:

```
#--------------------------------------------------
echo "Eth-CFM Configuration"
#--------------------------------------------------
    eth-cfm
        domain 100 format none level 3
            association 1 format icc-based name "node-3-site-1-0"
                bridge-identifier 1
                exit
                ccm-interval 1
                remote-mepid 130
            exit
            association 2 format icc-based name "node-3-site-2-0"
                bridge-identifier 2
                exit
                ccm-interval 1
                remote-mepid 120
            exit
        exit
    exit

#--------------------------------------------------
echo "Service Configuration"
#--------------------------------------------------

  eth-tunnel 2
        description "Eth Tunnel loadsharing mode QinQ example"
        protection-type loadsharing
        ethernet
            encap-type qinq
        exit
        path 1
            member 1/1/1
            control-tag 1.1
            eth-cfm
                mep 310 domain 100 association 1
                    ccm-enable
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        path 2
            member 1/1/2
            control-tag 1.1
            eth-cfm
                mep 320 domain 100 association 2
                    ccm-enablepath
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        no shutdown
```

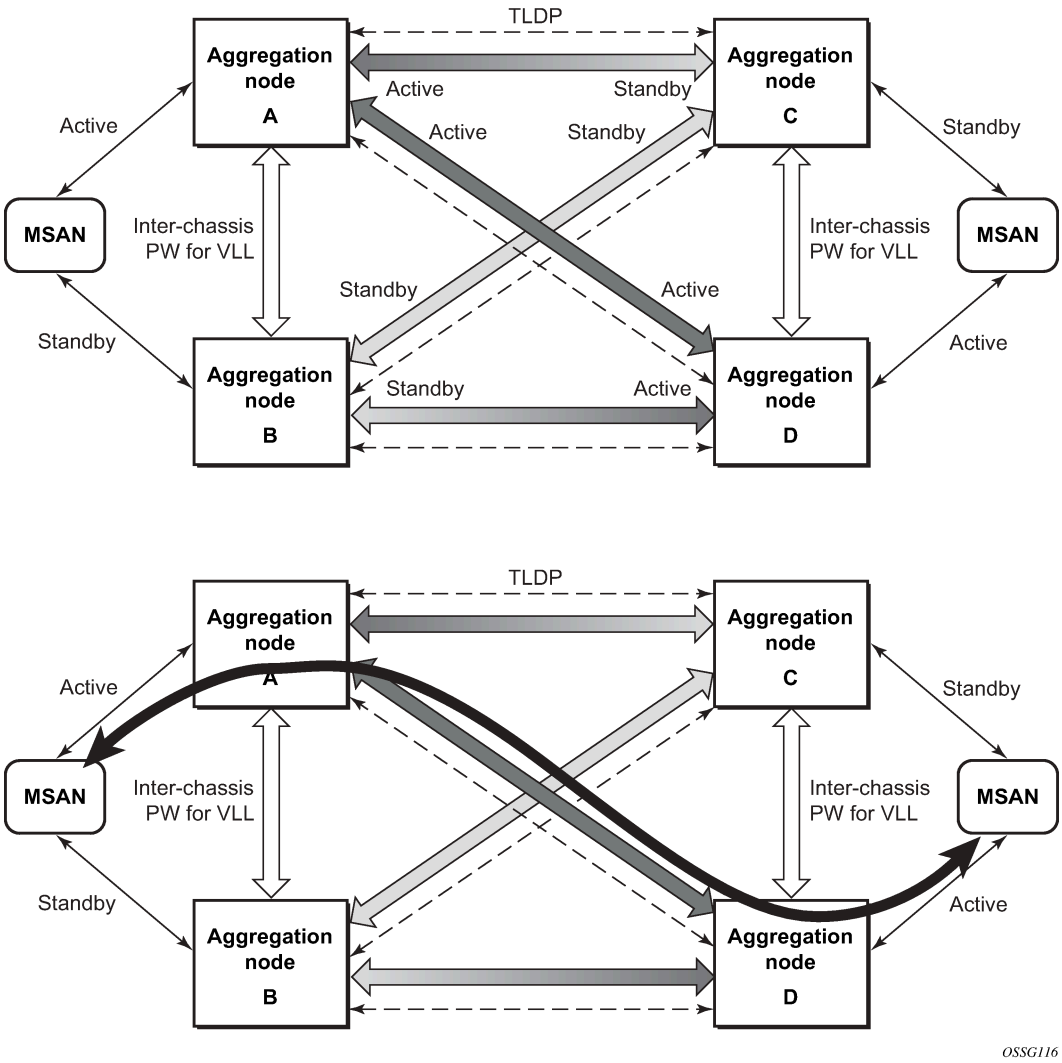
```
    exit


#--------------------------------------------------
echo "Service Configuration"
#--------------------------------------------------
    service
        epipe 1 customer 1 create
            sap 1/10/1:1 create
            exit
            sap eth-tunnel-1 create
            exit
            no shutdown
        exit
#--------------------------------------------------
echo "Service Configuration for a shared fate Ethernet Tunnel"
#--------------------------------------------------
        epipe 2 customer 1 create
            sap 1/10/2:3 create
            exit
            sap eth-tunnel-1:2 create
                eth-tunnel
                    path 1 tag 1.2
                    path 2 tag 1.2
                exit
            exit
            no shutdown
        exit
```

2.6.11 Access node resilience using MC-LAG and pseudowire redundancy

Figure 32: Access node resilience shows the use of both Multi-Chassis Link Aggregation (MC-LAG) in the access network and pseudowire redundancy in the core network to provide a resilient end-to-end VLL service to the customers.

Figure 32: Access node resilience



In this application, a new pseudowire status bit of active or standby indicates the status of the SAP in the MC-LAG instance in the SR-series aggregation node. All spoke-SDPs are of secondary type and there is no use of a primary pseudowire type in this mode of operation. Node A is in the active state according to its local MC-LAG instance and therefore advertises active status notification messages to both its peer pseudowire nodes; for example, nodes C and D. Node D performs the same operation. Node B is in the standby state according to the status of the SAP in its local MC-LAG instance, so advertises standby status notification messages to both nodes C and D. Node C performs the same operation.

An SR-series node selects a pseudowire as the active path for forwarding packets when both the local pseudowire status and the received remote pseudowire status indicate active status. However, an SR-series device in standby status according to the SAP in its local MC-LAG instance is capable of processing packets for a VLL service received over any of the pseudowires that are up. This is to avoid black holing of user traffic during transitions. The SR-series standby node forwards these packets to the active node by the Inter-Chassis Backup pseudowire (ICB pseudowire) for this VLL service. An ICB is a spoke-SDP used by an MC-LAG node to back up an MC-LAG SAP during transitions. The same ICB can also be used by the peer MC-LAG node to protect against network failures causing the active pseudowire to go down.

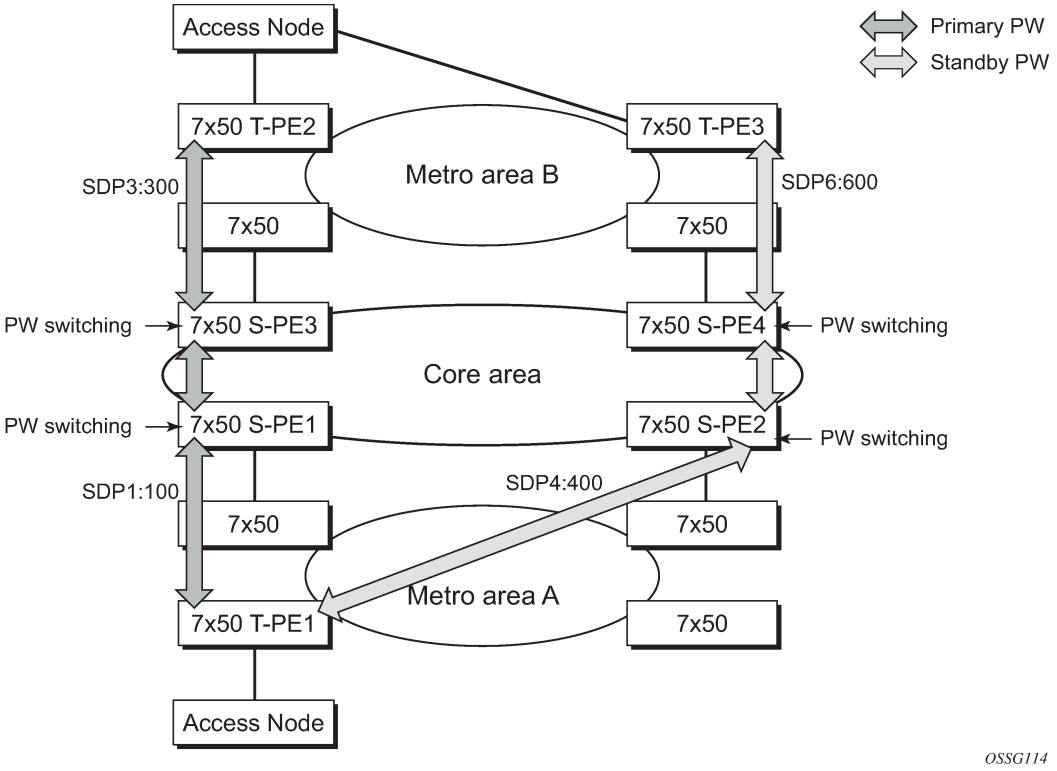
At configuration time, the user specifies a precedence parameter for each of the pseudowires that are part of the redundancy set, as described in the application in VLL resilience with two destination PE nodes. An SR-series node uses this to select which pseudowire to forward packets to in case both pseudowires show active/active for the local/remote status during transitions.

Only VLL service of type Epipe is supported in this application. Also, ICB spoke-SDP can only be added to the SAP side of the VLL cross-connect if the SAP is configured on an MC-LAG instance.

2.6.12 VLL resilience for a switched pseudowire path

Figure 33: VLL resilience with pseudowire redundancy and switching shows the use of both pseudowire redundancy and pseudowire switching to provide a resilient VLL service across multiple IGP areas in a provider network.

Figure 33: VLL resilience with pseudowire redundancy and switching



Pseudowire switching is a method for scaling a large network of VLL or VPLS services by removing the need for a full mesh of T-LDP sessions between the PE nodes as the number of these nodes grows over time.

Like in the application in VLL resilience with two destination PE nodes, the T-PE1 node switches the path of a VLL to a secondary standby pseudowire if a network side failure caused the VLL binding status to be operationally down or if T-PE2 notified it that the remote SAP went down. This application requires that pseudowire status notification messages generated by either a T-PE node or a S-PE node be processed and relayed by the S-PE nodes.

It is possible that the secondary pseudowire path terminates on the same target PE as the primary; for example, T-PE2. This provides protection against network side failures but not against a remote SAP

failure. When the target destination PE for the primary and secondary pseudowires is the same, T-PE1 normally does not switch the VLL path onto the secondary pseudowire upon receipt of a pseudowire status notification indicating the remote SAP is down, because the status notification is sent over both the primary and secondary pseudowires. However, the status notification on the primary pseudowire may arrive earlier than the one on the secondary pseudowire because of the differential delay between the paths. This causes T-PE1 to switch the path of the VLL to the secondary standby pseudowire and remain there until the status notification is cleared. Then, the VLL path is switched back to the primary pseudowire because of the revertive behavior operation. The path does not switch back to a secondary path when it comes up, even if it has a higher precedence than the currently active secondary path.

For the 7750 SR, this application can make use of all types of VLL supported on the routers; for example, Epipe and Ipipe services. A SAP can be configured on a SONET/SDH port that is part of an APS group. However, if a SAP is configured on an MC-LAG instance, only the Epipe service type is allowed.

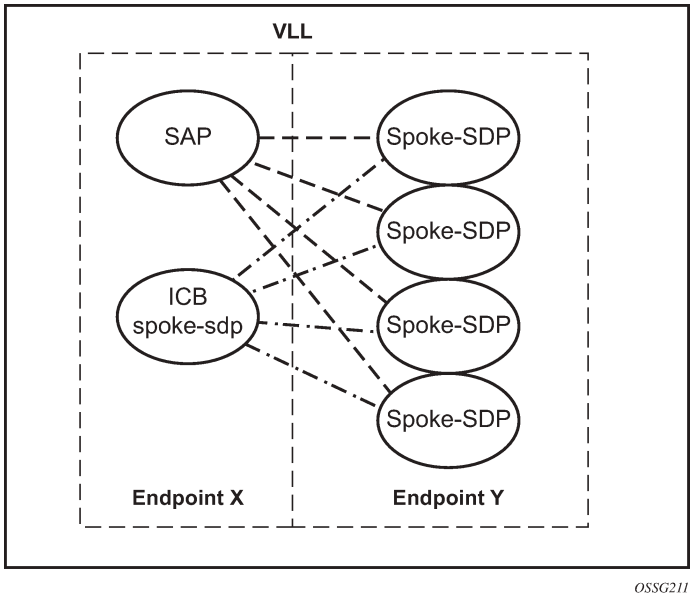
2.7 Pseudowire redundancy service models

This section describes the MC-LAG and pseudowire redundancy scenarios as, and the algorithm used to select the active transmit object in a VLL endpoint.

2.7.1 Redundant VLL service model

To implement pseudowire redundancy, a VLL service accommodates more than a single object on the SAP side and on the spoke-SDP side. Figure 34: Redundant VLL endpoint objects shows the model for a redundant VLL service based on the concept of endpoints.

Figure 34: Redundant VLL endpoint objects



By default, a VLL service supports two implicit endpoints managed internally by the system. Each endpoint can only have one object: a SAP or a spoke-SDP.

To add more objects, up to two explicitly named endpoints may be created per VLL service. The endpoint name is locally significant to the VLL service. They are referred to as endpoint X and endpoint Y, as shown in the example in Figure 34: Redundant VLL endpoint objects.

In Figure 34: Redundant VLL endpoint objects, the Y endpoint can also have a SAP or an ICB spoke-SDP, or both. The following is a list of supported endpoint objects and the applicable rules to associate the object with an endpoint of a VLL service:

- **SAP**

  A maximum of one SAP per VLL endpoint is supported.

- **primary spoke-SDP**

  The VLL service always uses this pseudowire and only switches to a secondary pseudowire when this primary pseudowire is down; the VLL service switches the path to the primary pseudowire when it is back up. The user can configure a timer to delay reverting back to primary or to never revert. A maximum of one primary spoke-SDP per VLL endpoint is supported.

- **secondary spoke-SDP**

  There can be a maximum of four secondary spoke-SDPs per endpoint. The user can configure the precedence of a secondary pseudowire to indicate the order in which a secondary pseudowire is activated.

- **Inter-Chassis Backup (ICB) spoke-SDP**

  This special pseudowire is used for MC-LAG and pseudowire redundancy applications. Forwarding between ICBs is blocked on the same node. The user has to explicitly indicate that the spoke-SDP is an ICB, at creation time. However, following are a few scenarios where the user can configure the spoke-SDP as ICB or as a regular spoke-SDP on a specified node. The CLI for those cases indicates both options.

A VLL service endpoint can only use a single active object to transmit at a specific time, but it can receive from all endpoint objects.

An explicitly named endpoint can have a maximum of one SAP and one ICB. When a SAP is added to the endpoint, only one more object of type ICB spoke-SDP is allowed. The ICB spoke-SDP cannot be added to the endpoint if the SAP is not part of an MC-LAG instance. Conversely, a SAP that is not part of an MC-LAG instance cannot be added to an endpoint that already has an ICB spoke-SDP.

An explicitly named endpoint that does not have a SAP object can have a maximum of four spoke-SDPs and includes any of the following:

- a single primary spoke-SDP
- one or many secondary spoke-SDPs with precedence
- a single ICB spoke-SDP

2.7.2 T-LDP status notification handling rules

Using Figure 34: Redundant VLL endpoint objects as a reference, this section describes the rules for generating, processing, and merging T-LDP status notifications in VLL service with endpoints. Any allowed combination of objects as specified in Redundant VLL service model, can be used on endpoints X and Y. The following sections see the specific combination objects in Figure 34: Redundant VLL endpoint objects as an example to describe the more general rules.

2.7.2.1 Processing endpoint SAP active/standby status bits

The advertised administrative forwarding status bit of active/standby reflects the status of the local LAG SAP in MC-LAG application. If the SAP is not part of an MC-LAG instance, the forwarding status of active is always advertised.

If the SAP in endpoint X is part of an MC-LAG instance, a node must send a T-LDP forwarding status bit of SAP active/standby over all Y endpoint spoke-SDPs, except the ICB spoke-SDP, whenever this status changes. The status bit sent over the ICB is always zero (active by default).

If the SAP in endpoint X is not part of an MC-LAG instance, then the forwarding status sent over all Y endpoint spoke-SDPs should always be set to zero (active by default).

2.7.2.2 Processing and merging

Endpoint X is operationally up if at least one of its objects is operationally up. It is down if all of its objects are operationally down.

If the SAP in endpoint X transitions locally to the down state or received a SAP down notification by the SAP-specific OAM signal, the node must send T-LDP SAP down status bits on the Y endpoint ICB spoke-SDP only. Ethernet SAP does not support SAP OAM protocol. All other SAP types cannot exist on the same endpoint as an ICB spoke-SDP because a non-Ethernet SAP cannot be part of an MC-LAG instance.

If the ICB spoke-SDP in endpoint X transitions locally to down state, the node must send T-LDP SDP-binding down status bits on this spoke-SDP.

If the ICB spoke-SDP in endpoint X receives T-LDP SDP-binding down status bits or pseudowire not forwarding status bits, the node saves this status and takes no further action. The saved status is used for active transmit endpoint object selection.

If all objects in endpoint X perform any or all of the following operations, the node must send status bits of SAP down over all "Y" endpoint spoke-SDPs, including the ICB:

- transitioned locally to down state
- received a SAP down notification by remote T-LDP status bits or by SAP-specific OAM signal
- received SDP-binding down status bits
- received PW not forwarding status bits

Endpoint Y is operationally up if at least one of its objects is operationally up. It is down if all its objects are operationally down.

If a spoke-SDP in endpoint Y, including the ICB spoke-SDP, transitions locally to down state, the node must send T-LDP SDP-binding down status bits on this spoke-SDP.

If a spoke-SDP in endpoint Y, including the ICB spoke-SDP, performed any or all of the following operations, the node saves this status and takes no further action:

- received T-LDP SAP down status bits
- received T-LDP SDP-binding down status bits
- received PW not forwarding status bits

The saved status is used for selecting the active transmit endpoint object.

If all objects in endpoint Y, except the ICB spoke-SDP, performed any or all of the following operations, the node must send status bits of SDP-binding down over the X endpoint ICB spoke-SDP only:

- transitioned locally to the down state
- received T-LDP SAP down status bits
- received T-LDP SDP-binding down status bits
- received PW not forwarding status bits

If all objects in endpoint Y performed any or all of the following operations, he node must send status bits of SDP-binding down over the X endpoint ICB spoke-SDP, and must send a SAP down notification on the X endpoint SAP by the SAP-specific OAM signal if applicable:

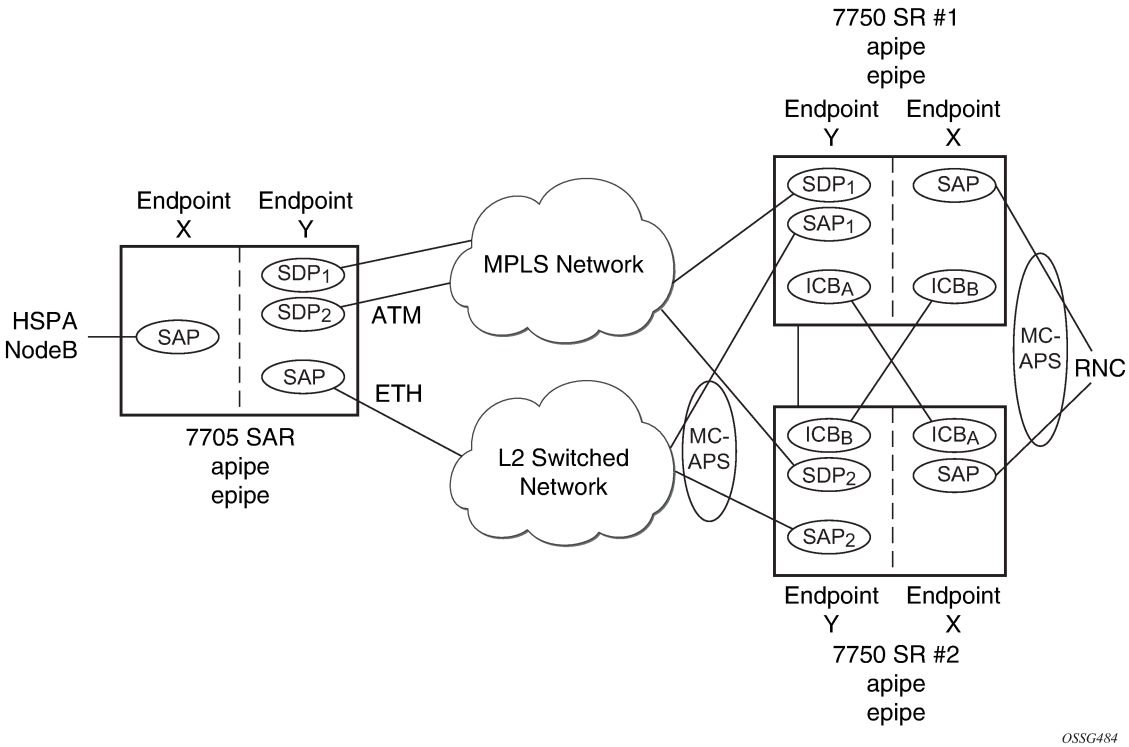
- transitioned locally to down state
- received T-LDP SAP down status bits
- received T-LDP SDP-binding down status bits
- received PW not forwarding status bits

An Ethernet SAP does not support signaling status notifications.

2.8 MC-APS and MC-LAG

In many cases, 7750 SRs are deployed in redundant pairs at the MSC. Figure 35: HSDPA off load fallback with MC-APS shows this case, assuming that MC-APS is deployed on the RNC connection. For MC-APS to be used, clear channel SONET or SDH connections should be used.

Figure 35: HSDPA off load fallback with MC-APS



In this scenario, endpoint Y allows an ICB spoke-SDP as well as the primary spoke-SDP and secondary SAP. ICB operation is maintained as the current redundant pseudowire operation and the ICB spoke-SDP is always provided an active status. The ICB spoke-SDP is only used if both the primary spoke-SDP and secondary SAP are not available. The secondary SAP is used if it is operationally up and the primary spoke-SDP pseudowire status is not active. Receive is enabled on all objects even though transmit is only enabled on one.

To allow correct operation in all failure scenarios, an ICB spoke-SDP must be added to endpoint X. The ICB spoke-SDP is only used if the SAP is operationally down.

The following is an example configuration of Epipes mapping to Figure 35: HSDPA off load fallback with MC-APS. A SAP can be added to an endpoint with a non-ICB spoke-SDP only if the precedence of the spoke is **primary**.

7750 SR #1

```
*A:ALA-A>config>service#  epipe 1
-----------------------------------------------
            endpoint X
            exit
            endpoint Y
            exit
            sap 1/1/2:0 endpoint X
              exit
            spoke-sdp 1:100 endpoint X icb
              exit
            spoke-sdp 10:500 endpoint Y
              precedence primary
              exit
            sap 1/1/3:0 endpoint Y
            exit
            spoke-sdp 1:200 endpoint Y icb
            exit
-----------------------------------------------
*A:ALA-A>config>service#
```

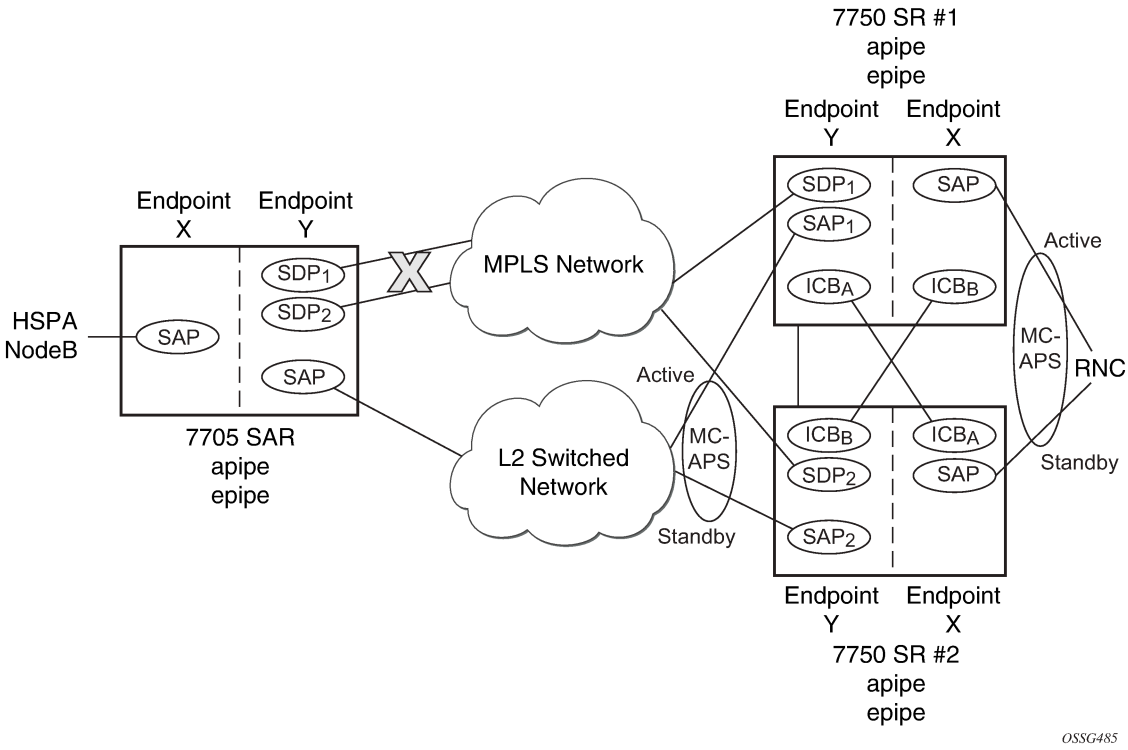
7750 SR #2

```
*A:ALA-B>config>service# epipe 1
-----------------------------------------------
            endpoint X
            exit
            endpoint Y
            exit
            sap 2/3/4:0 endpoint X
            exit
            spoke-sdp 1:200 endpoint X icb
            exit
            spoke-sdp 20:600 endpoint Y
              precedence primary
              exit
            sap 2/3/5:0 endpoint Y
            exit
            spoke-sdp 1:100 endpoint Y icb
            exit
-----------------------------------------------
*A:ALA-B>config>service#
```

2.8.1 Failure scenario

Based on the previously mentioned rules, the following is an example of a failure scenario. Assuming both links are active on 7750 SR #1 and the Ethernet connection to the cell site fails (most likely failure scenario because the connection would not be protected), SDP1 would go down and the secondary SAP would be used in 7750 SR #1 and 7705 SAR, as shown in Figure 36: Ethernet failure at cell site.

Figure 36: Ethernet failure at cell site



If the active link to the Layer 2 switched network was on 7750 SR #2 at the time of the failure, SAP1 would be operationally down (because the link is in standby) and ICBA would be used. Because the RNC SAP on 7750 SR #2 is on a standby APS link, ICBA would be active and it would connect to SAP2 because SDP2 is operationally down as well.

All APS link failures would be handled through the standard pseudowire status messaging procedures for the RNC connection and through standard ICB usage for the Layer 2 switched network connection.

2.9 VLL using G.8031 protected Ethernet tunnels

The use of MPLS tunnels provides the 7450 ESS and 7750 SR OS a way to scale the core while offering fast failover times using MPLS FRR. In environments where Ethernet services are deployed using native Ethernet backbones, Ethernet tunnels are provided to achieve the same fast failover times as in the MPLS FRR case.

The Nokia VLL implementation offers the capability to use core Ethernet tunnels compliant with ITU-T G.8031 specification to achieve 50 ms resiliency for backbone failures. This is required to comply with the stringent SLAs provided by service providers. Epipe and Ipipe services are supported.

When using Ethernet tunnels, the Ethernet tunnel logical interface is created first. The Ethernet tunnel has member ports, which are the physical ports supporting the links. The Ethernet tunnel control SAPs carry G.8031 and 802.1ag control traffic and user data traffic. Ethernet service SAPs are configured on the Ethernet tunnel. Optionally, when tunnels follow the same paths, end-to-end services may be configured with fate shared Ethernet tunnel SAPs, which carry only user data traffic and share the fate of the Ethernet tunnel port (if correctly configured).

Ethernet tunnels provide a logical interface that VLL SAPs may use just as regular interfaces. The Ethernet tunnel provides resiliency by providing end-to-end tunnels. The tunnels are stitched together by VPLS or Epipe services at intermediate points. Epipes offer a more scalable option.

For further information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*.

2.10 MPLS entropy label and hash label

The router supports the MPLS entropy label (RFC 6790) and the Flow Aware Transport label, known as the hash label (RFC 6391). LSR nodes in a network can load-balance labeled packets in a more granular way than by hashing on the standard label stack. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

The entropy label is supported for Epipe and Ipipe VLL services as well as BGP VPWS. To configure insertion of the entropy label on a spoke-SDP of a specific service, use the **entropy-label** command in the **spoke-sdp** or **pw-template** context. Note that the entropy label is only inserted if the far end of the MPLS tunnel is also entropy-label-capable.

The hash label is supported for Epipe and Ipipe VLL services. For TLDP based spoke SDPs, configure it using the following commands:

```
configure service epipe spoke-sdp hash-label
```

```
configure service ipipe spoke-sdp hash-label
```

and for BGP-VPWS spoke-SDPs configure it using the following command:

```
configure service pw-template hash-label
```

Optionally, the **hash-label signal-capability** command can be configured. If the user configures the **hash-label** command only, the hash label is sent (and it is expected to be received) in all the packets. However, if the **hash-label signal-capability** command is configured, the use of the hash label is signaled and only used in case the peer PE signals support for hash label in its TLDP signaling or BGP-VPLS route (RFC 8395).

Either the hash label or the entropy label can be configured on one object, but not both.

2.11 BGP VPWS

BGP Virtual Private Wire Service (VPWS) is a point-to-point Layer 2 VPN service based on RFC 6624 *Layer 2 Virtual Private Networks using BGP for Auto-Discovery and Signaling*, which in turn uses the BGP pseudowire signaling concepts described in RFC 4761, *Virtual Private LAN Service Using BGP for Auto-Discovery and Signaling*.

The BGP-signaled pseudowires created can use either automatic or preprovisioned SDPs over LDP- or BGP-signaled tunnels; the choice of tunnel depends on the tunnel's preference in the tunnel table, or over GRE. Preprovisioned SDPs must be configured when RSVP signaled transport tunnels are used.

The use of an automatically created GRE tunnel is enabled by creating the PW template used within the service with the parameter **auto-gre-sdp**. The GRE SDP and SDP binding are created after a matching BGP route is received.

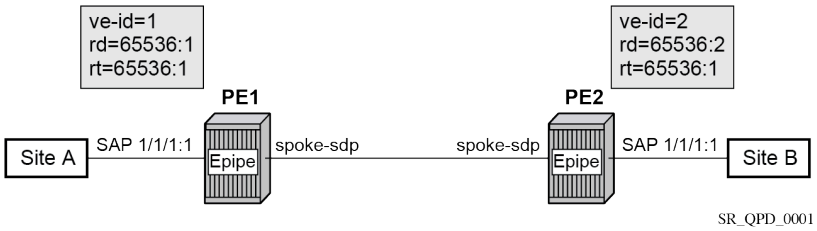
Inter-AS model C and dual-homing are supported.

2.11.1 Single-homed BGP VPWS

A single-homed BGP VPWS service is implemented as an Epipe connecting a SAP or static GRE tunnel (a spoke-SDP using a GRE SDP configured with static MPLS labels) and a BGP signaled pseudowire, maintaining the Epipe properties such as no MAC learning. The MPLS pseudowire data plane uses a two-label stack; the inner label is derived from the BGP signaling and identifies the Epipe service while the outer label is the tunnel label of an LSP transporting the traffic between the two end systems.

The following figures shows how this service would be used to provide a virtual leased line service (VLL) across an MPLS network between sites A and B.

Figure 37: Single-homed BGP-VPWS example



An Epipe is configured on PE1 and PE2 with BGP VPWS enabled. PE1 and PE2 are connected to site A and B, respectively, each using a SAP. The interconnection between the two PEs is achieved through a pseudowire that is signaled using BGP VPWS updates over a specific tunnel LSP.

2.11.2 Dual-homed BGP VPWS

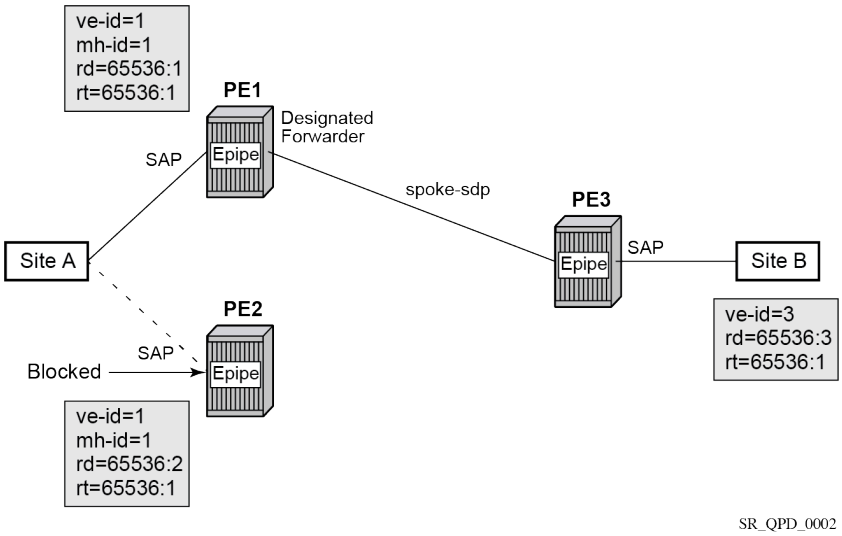
A BGP-VPWS service can benefit from dual-homing, as described in IETF Draft *draft-ietf-bess-vpls-multihoming-01*. When using dual-homing, two PEs connect to a site, with one PE being the DF for the site and the other blocking its connection to the site. On failure of the active PE, its pseudowire, or its connection to the site, the other PE becomes the designated forwarder and unblocks its connection to the site.

2.11.2.1 Single pseudowire example

A pseudowire is established between the designated forwarder of the dual-homed PEs and the remote PE. If a failure causes a change in the designated forwarder, the pseudowire is deleted and reestablished between the remote PE and the new designated forwarder. This topology requires that the VE IDs on the dual-homed PEs are set to the same value.

The following figure shows a dual-homed, single pseudowire topology example.

Figure 38: Dual-homed BGP VPWS with single pseudowire



An Epipe with BGP VPWS enabled is configured on each PE. Site A is dual-homed to PE1 and PE2 with the remote PE (PE3) connecting to site B. An Epipe service is configured on each PE in which there is a SAP connecting to the local site.

The pair of dual-homed PEs perform a designated forwarder election, which is influenced by BGP route selection, the site state, and configuration of the **site-preference**. A site is only eligible to be the designated forwarder if it is up (the site state is down if there is no pseudowire established or if the pseudowire is in an operationally down state). The winner, for example PE1, becomes the active switch for traffic sent to and from site A, while the loser blocks its connection to site A.

Pseudowires are signaled using BGP from PE1 and PE2 to PE3, but only from PE3 to the designated forwarder in the opposite direction (so only one bidirectional pseudowire is established). There is no pseudowire between PE1 and PE2; this is achieved by configuration.

Traffic is sent and received traffic on the pseudowire connected between PE3 and the designated forwarder, PE1.

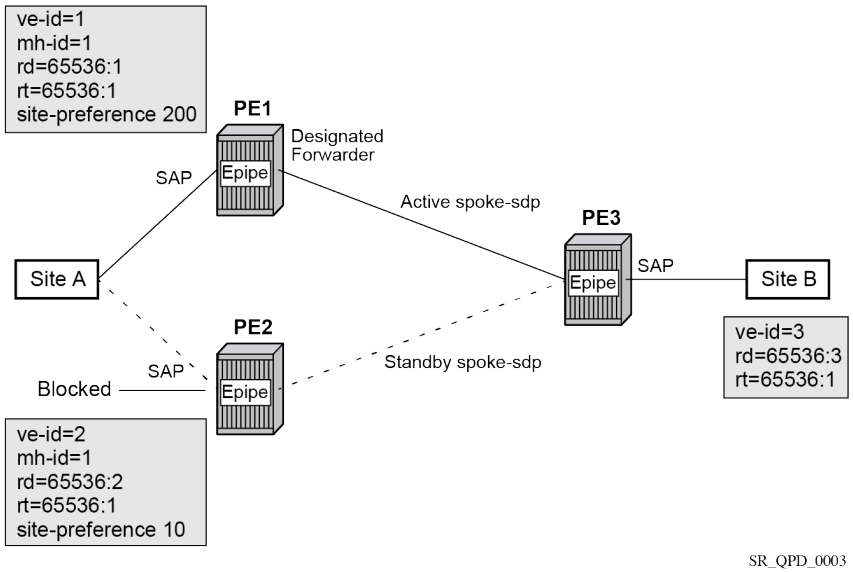
If the site state is operationally down, both the D and Circuit Status Vector (CSV) bits (see the following for more details) are set in the BGP-VPWS update, which causes the remote PE to use the pseudowire to the new designated forwarder.

2.11.2.2 Active/standby pseudowire example

Pseudowires are established between the remote PE and each dual-homed PE. The remote PE can receive traffic on either pseudowire, but only sends on the one to the designated forwarder. This creates an active/standby pair of pseudowires. At most, one standby pseudowire is established; this being determined using the tie-breaking rules defined in the multihoming draft. This topology requires each PE to have a different VE ID.

The following figure shows an example of a dual-homed, active/standby pseudowires topology.

Figure 39: Dual-homed BGP VPWS with active/standby pseudowires



An Epipe with BGP VPWS enabled is configured on each PE. Site A is dual-homed to PE1 and PE2 with the remote PE (PE3) connecting to site B. An Epipe service is configured on each PE in which there is a SAP connecting to the local site.

The pair of dual-homed PEs perform a designated forwarder election, which is influenced by configuring the **site-preference** value. The winner, PE1 (based on its higher **site-preference** value) becomes the active switch for traffic sent to and from site A, while the loser, PE2, blocks its connection to site A. Pseudowires are signaled using BGP between PE1 and PE3, and between PE2 and PE3. There is no pseudowire between PE1 and PE2; this is achieved by configuration. The active/standby pseudowires on PE3 are part of an endpoint automatically created in the Epipe service.

Traffic is sent and received on the pseudowire connected to the designated forwarder, PE1.

2.11.3 BGP VPWS pseudowire switching

Pseudowire switching is supported with a BGP VPWS service allowing the cross connection between a BGP VPWS signaled spoke-SDP and a static GRE tunnel, the latter being a spoke SDP configured with static MPLS labels using a GRE SDP. No other spoke SDP types are supported. Support is not included for BGP multihoming using an active and a standby pseudowire to a pair of remote PEs.

Operational state changes to the GRE tunnel are reflected in the state of the Epipe and propagated accordingly in the BGP VPWS spoke SDP status signaling, specifically using the BGP update D and CSV bits.

The following configuration is required:

1. The Epipe service must be created using the **vc-switching** parameter.
2. The GRE tunnel spoke SDP must be configured using a GRE SDP with **signaling off** and have the ingress and egress vc-labels statically configured.

Example: BGP VPWS service configured to allow pseudowire switching

```
configure
```

```
    service
        sdp 1 create
            signaling off
            far-end 192.168.1.1
            keep-alive
                shutdown
            exit
            no shutdown
        exit
        pw-template 1 create
        exit
        epipe 1 customer 1 vc-switching create
            description "BGP VPWS service"
            bgp
                route-distinguisher 65536:1
                route-target export target:65536:1 import target:65536:1
                pw-template-binding 1
                exit
            exit
            bgp-vpws
                ve-name "PE1"
                    ve-id 1
                exit
                remote-ve-name "PE2"
                    ve-id 2
                exit
                no shutdown
            exit
            spoke-sdp 1:1 create
                ingress
                    vc-label 1111
                exit
                egress
                    vc-label 1122
                exit
                no shutdown
            exit
            no shutdown
        exit
```

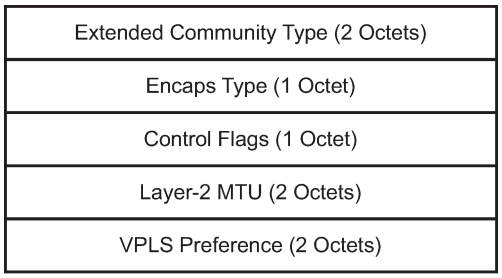
2.11.4 Pseudowire signaling

The BGP signaling mechanism used to establish the pseudowires is described in the BGP VPWS standards with the following differences:

- As stated in Section 3 of RFC 6624, there are two modifications of messages when compared to RFC 4761:
  - the Encaps Types supported in the associated extended community
  - the addition of a circuit status vector sub-TLV at the end of the VPWS NLRI
- The control flags and VPLS preference in the associated extended community are based on IETF Draft *draft-ietf-bess-vpls-multihoming-01*.

The following figure shows the format of the BGP VPWS update extended community.

Figure 40: BGP VPWS update extended community format



- **extended community type**

  This is the value allocated by IANA for this attribute is 0x800A.

- **encaps type**

  The encapsulation type identifies the type of pseudowire encapsulation. Ethernet VLAN (4) and Ethernet Raw mode (5), as described in RFC 4448, are the only values supported. If there is a mismatch between the Encaps Type signaled and the one received, the pseudowire is created but with the operationally down state.

- **control flags**

  This is control information concerning the pseudowires, see Figure 41: Control flags for more information.

- **Layer 2 MTU**

  This is the MTU to be used on the pseudowires. If the received Layer 2 MTU is zero, no MTU check is performed and the related pseudowire is established. If there is a mismatch between the local **service-mtu** and the received Layer 2 MTU, the pseudowire is created with the operationally down state and an MTU/Parameter mismatch indication.

- **VPLS preference**

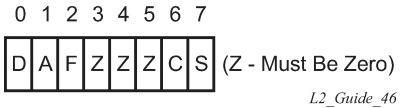
  VPLS preference has a default value of zero for BGP-VPWS updates sent by the system, indicating that it is not in use. If **site-preference** is configured, its value is used for the VPLS preference and is also used in the local designated forwarder election.

  On receipt of a BGP VPWS update containing a non-zero value, it is used to determine to which system the pseudowire is established, as part of the VPWS update process tie-breaking rules. The BGP local preference of the BGP VPWS update sent by the system is set to the same value as the VPLS preference if the latter is non-zero, as required by the draft (as long as the D bit in the extended community is not set to 1). Consequently, attempts to change the BGP local preference when exporting a BGP VPWS update with a non-zero VPLS preference is ignored. This prevents the updates being treated as malformed by the receiver of the update.

  For inter-AS, the preference information must be propagated between autonomous systems using the VPLS preference. Consequently, if the VPLS preference in a BGP-VPWS or BGP multihoming update is zero, the local preference is copied by the egress ASBR into the VPLS preference field before sending the update to the External Border Gateway Protocol (EBGP) peer. The adjacent ingress ASBR then copies the received VPLS preference into the local preference to prevent the update from being considered malformed.

The following figure shows the pseudowire control flags.

Figure 41: Control flags



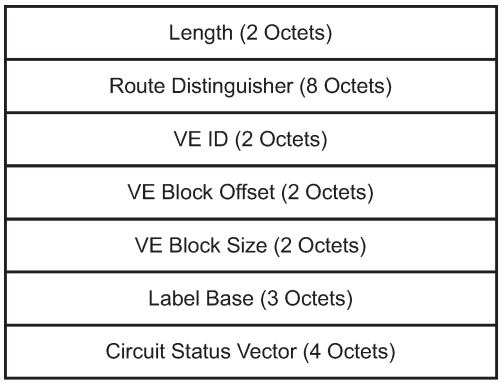
L2_Guide_46

The following bits in the Control Flags are defined:

D	Access circuit down indicator from IETF Draft *draft-kothari-l2vpn-auto-site-id-01*. D is 1 if all access circuits are down, otherwise D is 0.
A	Automatic site ID allocation, which is not supported. This is ignored on receipt and set to 0 on sending.
F	MAC flush indicator. This is not supported because it relates to a VPLS service. This is set to 0 and ignored on receipt.
C	Presence of a control word. Control word usage is supported. When this is set to 1, packets are sent and are expected to be received, with a control word. When this is set to 0, packets are sent and are expected to be received without a control word (by default).
S	Sequenced delivery. Sequenced delivery is not supported. This is set to 0 on sending (no sequenced delivery) and, if a non-zero value is received (indicating sequenced delivery required), the pseudowire is not created.

The BGP VPWS NLRI is based on that defined for BGP VPLS, but is extended with a circuit status vector as shown in the following figure.

Figure 42: BGP VPWS NLRI



L2_Guide_43

The VE ID value is configured within each BGP VPWS service, the label base is chosen by the system, and the VE block offset corresponds to the remote VE ID because a VE block size of 1 is always used.

The circuit status vector is encoded as a TLV, as shown in Figure 43: BGP VPWS NLRI TLV extension format and Figure 44: Circuit status vector TLV type.

Figure 43: BGP VPWS NLRI TLV extension format

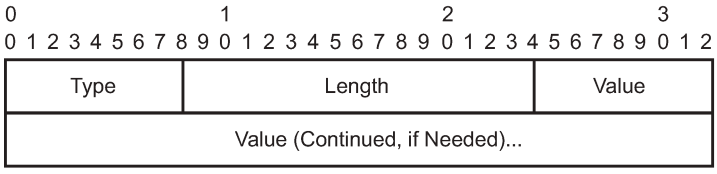


Figure 44: Circuit status vector TLV type

TLV Type	Description
1	Circuit Status Vector

L2_Guide_45

The circuit status vector is used to indicate the status of both the SAP/GRE tunnel and the status of the spoke-SDP within the local service. Because the VE block size used is 1, the most significant bit in the circuit status vector TLV value is set to 1 if either the SAP/GRE tunnel or spoke-SDP is down, otherwise it is set to 0. On receiving a circuit status vector, only the most significant byte of the CSV is examined for designated forwarder selection purposes.

If a circuit status vector length field of greater than 32 is received, the update is ignored and not reflected to BGP neighbors. If the length field is greater than 800, a notification message is sent and the BGP session restarts. Also, BGP VPWS services support a single access circuit, so only the most significant bit of the CSV is examined on receipt.

A pseudowire is established when a BGP VPWS update is received that matches the service configuration, specifically the configured route targets and remote VE ID. If multiple matching updates are received, the system to which the pseudowire is established is determined by the tie-breaking rules, as described in IETF Draft *draft-ietf-bess-vpls-multihoming-01*.

Traffic is sent on the active pseudowire connected to the remote designated forwarder. Traffic can be received on either the active or standby pseudowire, although no traffic should be received on the standby pseudowire because the SAP/GRE tunnel on the non-designated forwarder should be blocked.

The **adv-service-mtu** command can be used to override the MTU value used in BGP signaling to the far-end of the pseudowire. This value is also used to validate the value signaled by the far-end PE unless **ignore-l2vpn-mtu-mismatch** is also configured.

If the **ignore-l2vpn-mtu-mismatch** command is configured, the router does not check the value of the "Layer 2 MTU" in the "Layer2 Info Extended Community" received in a BGP update message against the local service MTU, or against the MTU value signaled by this router. The router brings up the BGP-VPWS service regardless of any MTU mismatch.

2.11.5 BGP-VPWS with inter-AS model C

BGP VPWS with inter-AS model C is supported both in a single-homed and dual-homed configuration.

When dual-homing is used, the dual-homed PEs must have different values configured for the **site-preference** (under the **site** within the Epipe service) to allow the PEs in a different AS to select the designated forwarder when all access circuits are up. The value configured for the **site-preference** is propagated between autonomous systems in the BGP VPWS and BGP multihoming update extended

community VPLS preference field. The receiving ingress ASBR copies the VPLS preference value into local preference of the update to ensure that the VPLS preference and local preference are equal, which prevents the update from being considered malformed.

2.11.6 BGP VPWS configuration procedure

In addition to configuring the associated BGP and MPLS infrastructure, the provisioning of a BGP VPWS service requires:

- configuring the BGP Route Distinguisher, Route Target

  The updates are accepted into the service only if they contain the configured import route-target.
- configuring a binding to the pseudowire template

  The multiple pseudowire template bindings can be configured with their associated route-targets used to control which is applied.
- configuring the SAP or static GRE tunnel
- configuring the name of the local VE and its associate VE ID
- configuring the name of the remote VE and its associated VE ID
- for a dual-homed PE:
  - enabling the site
  - configuring the site with non-zero site-preference
- for a remote PE, configure up to two remote VE names and associated VE IDs
- enabling BGP VPWS

2.11.7 Use of pseudowire template for BGP VPWS

The pseudowire template concept used for BGP AD is reused for BGP VPWS to dynamically instantiate pseudowires (SDP-bindings) and the related SDPs (provisioned or automatically instantiated).

The settings for the L2-Info extended community in the BGP update sent by the system are derived from the **pw-template** attributes. The following rules apply:

- If multiple **pw-template-bindings** (with or without **import-rt**) are specified for the VPWS instance, the first (numerically lowest ID) **pw-template** entry is used.
- Both Ethernet VLAN and Ethernet Raw Mode Encaps Types are supported; these are selected by configuring the **vc-type** in the pseudowire template to be either **vlan** or **ether**, respectively. The default is **ether**.

  The same value must be used by the remote BGP VPWS instance to ensure that the related pseudowire comes up.
- Layer 2 MTU is derived from the service VPLS **service-mtu** parameter.

  The same value must be used by the remote BGP VPWS instance to ensure that the related pseudowire comes up.
- Control Flag C can be 0 or 1, depending on the setting of the **controlword** parameter in the PW template 0.
- Control Flag S is always 0.

On reception, the values of the parameters in the L2-Info extended community of the BGP update are compared with the settings from the corresponding **pw-template**. The following steps are used to determine the local **pw-template**:

- The **route-target** values are matched to determine the **pw-template**. The binding configured with the first matching route target is chosen.
- If a match is not found from the previous step, the lowest **pw-template-binding** (numerically) without any **route-target** configured is used.
- If the values used for **encap-type** or Layer 2 MTU do not match, the pseudowire is created but with the operationally down state.

  To interoperate with existing implementations, if the received MTU value = 0, the MTU negotiation does not take place; the related pseudowire is set up ignoring the MTU.
- If the value of the S flag is not zero, the pseudowire is not created.

The following pseudowire template parameters are supported when applied within a BGP VPWS service; the remainder are ignored:

```
configure service pw-template policy-id [use-provisioned-sdp |
       [prefer-provisioned-sdp] [auto-sdp]] [create] [name name]
   accounting-policy acct-policy-id
   no accounting-policy
   [no] collect-stats
   [no] controlword
   egress
      filter ipv6 ipv6-filter-id
      filter ip ip-filter-id
      filter mac mac-filter-id
      no filter [ip ip-filter-id] [mac mac-filter-id] [ipv6 ipv6-filter-id]
      qos network-policy-id port-redirect-group queue-group-name instance instance
id
      no qos [network-policy-id]
   [no] force-vlan-vc-forwarding
   hash-label [signal-capability]
   no hash-label
   ingress
      filter ipv6 ipv6-filter-id
      filter ip ip-filter-id
      filter mac mac-filter-id
      no filter [ip ip-filter-id] [mac mac-filter-id] [ipv6 ipv6-filter-id]
      qos network-policy-id fp-redirect-group queue-group-name instance instance-id
      no qos [network-policy-id]
   [no] sdp-exclude
   [no] sdp-include
   vc-type {ether | vlan}
   vlan-vc-tag vlan-id
   no vlan-vc-tag
```

For more information about this command, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*.

The **use-provisioned-sdp** option is permitted when creating the pseudowire template if a preprovisioned SDP is to be used. Preprovisioned SDPs must be configured whenever RSVP-signaled transport tunnels are used.

When the **prefer-provisioned-sdp** option is specified, if the system finds an existing matching SDP that conforms to any restrictions defined in the pseudowire template (for example, **sdp-include** or **sdp-exclude group**), it uses this matching SDP (even if the existing SDP is operationally down); otherwise, it automatically creates an SDP.

When the **auto-gre-sdp** option is specified, a GRE SDP is automatically created.

The **tools perform** command can be used in the same way as for BGP-AD to apply changes to the pseudowire template using the following format:

```
tools perform service [id service-id] eval-pw-template policy-id [allow-service-impact]
```

If a user configures a service using a pseudowire template with the **prefer-provisioned-sdp** option, but without provisioning an applicable SDP and the system binds to an automatic SDP, and the user subsequently provisions an appropriate SDP, the system does not automatically switch to the new provisioned SDP. This only occurs if the pseudowire template is reevaluated using the **tools perform service id service-id eval-pw-template** command.

2.11.8 Use of endpoint for BGP VPWS

An endpoint is required on a remote PE connecting to two dual-homed PEs to associate the active/standby pseudowires with the Epipe service. An endpoint is automatically created within the Epipe service such that active/standby pseudowires are associated with that endpoint. The creation of the endpoint occurs when **bgp-vpws** is enabled (and deleted when it is disabled) and so exists in both a single- and dual-homed scenario. This simplifies converting a single-homed service to a dual-homed service. The naming convention used is _tmnx_BgpVpws-x, where x is the service identifier. The automatically created endpoint has the default parameter values, although all are ignored in a BGP-VPWS service with the description field being defined by the system.

The following command does not have any effect on an automatically created VPWS endpoint:

tools perform service id <service-id> endpoint <endpoint-name> force-switchover

2.12 VLL service considerations

This section describes the general 7450 ESS, 7750 SR, and 7950 XRS service features and any special capabilities or considerations as they relate to VLL services.

2.12.1 SDPs

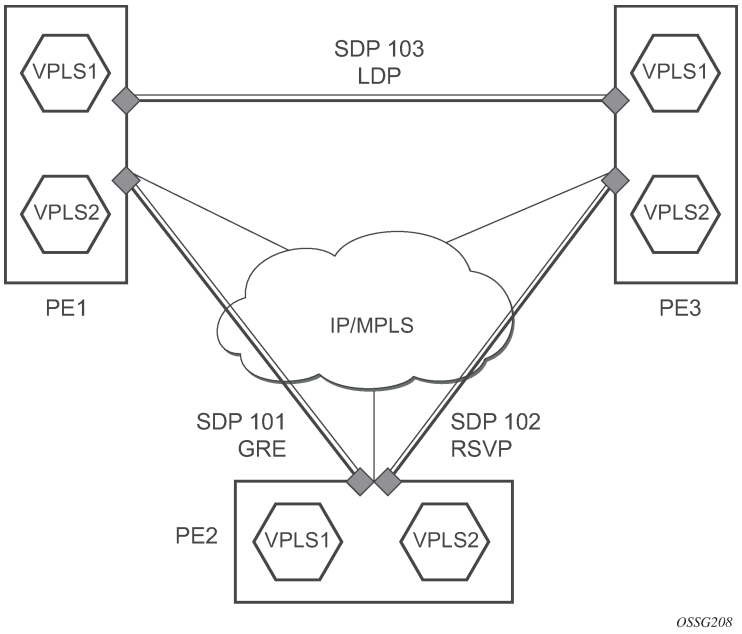
The most basic SDPs must have the following:

- A locally unique SDP identification (ID) number.
- The system IP address of the originating and far-end routers.
- An SDP encapsulation type, either GRE or MPLS.

2.12.1.1 SDP statistics for VPLS and VLL services

The three-node network in Figure 45: SDP statistics for VPLS and VLL services shows two MPLS SDPs and one GRE SDP defined between the nodes. These SDPs connect VPLS1 and VPLS2 instances that are defined in the three nodes. With this feature, the operator has local CLI-based and SNMP-based statistics collection for each VC used in the SDPs. This allows for traffic management of tunnel usage by the different services and with aggregation the total tunnel usage.

Figure 45: SDP statistics for VPLS and VLL services



2.12.2 SAP encapsulations and pseudowire types

The Epipe service is designed to carry Ethernet frame payloads, so it can provide connectivity between any two SAPs that pass Ethernet frames. The following SAP encapsulations are supported on the 7450 ESS, 7750 SR, and 7950 XRS Epipe service:

- Ethernet null
- Ethernet dot1q
- QinQ

While different encapsulation types can be used, encapsulation mismatching can occur if the encapsulation behavior is not understood by connecting devices, which are unable to send and receive the expected traffic. For example, if the encapsulation type on one side of the Epipe is dot1q and the other is null, tagged traffic received on the null SAP is double-tagged when it is transmitted out of the dot1q SAP.

One pseudowire encapsulation mode, that is, SDP vc-type, is available: PWE3 N-to-1 Cell Mode Encapsulation.

2.12.2.1 QoS policies

When applied to 7450 ESS, 7750 SR, or 7950 XRS Epipe services, service ingress QoS policies only create the unicast queues defined in the policy. The multipoint queues are not created on the service.

With Epipe services, egress QoS policies function as with other services where the class-based queues are created as defined in the policy. Both Layer 2 or Layer 3 criteria can be used in the QoS policies for traffic classification in a service.

2.12.2.2 Filter policies

7450 ESS, 7750 SR, and 7950 XRS Epipe and Ipipe services can have a single filter policy associated on both ingress and egress. Both MAC and IP filter policies can be used on Epipe services.

2.12.2.3 MAC resources

Epipe services are point-to-point Layer 2 VPNs capable of carrying any Ethernet payloads. Although an Epipe is a Layer 2 service, the 7450 ESS, 7750 SR, and 7950 XRS Epipe implementation does not perform any MAC learning on the service, so Epipe services do not consume any MAC hardware resources.

2.13 Configuring a VLL service with CLI

This section provides information to configure Virtual Leased Line (VLL) services using the command line interface.

2.13.1 Common configuration tasks

This section provides a brief overview of the tasks that must be performed and the CLI commands that must be executed to configure the VLL services:

1. Associate the service with a customer ID.
2. Optionally, define SAP parameters:
   - Select egress and ingress QoS or scheduler policies, or both (configured in the **config**>**qos** context).
   - Select accounting policy (configured in the **config**>**log** context).
3. Define spoke-SDP parameters.
4. Enable the service.

2.13.2 Configuring VLL components

This section provides VLL configuration examples for the VLL services.

2.13.2.1 Creating a Cpipe service

2.13.2.1.1 Basic configuration

Use the following CLI syntax to create a Cpipe service on a 7750 SR. A route distinguisher must be defined in order for Cpipe to be operationally active.

CLI syntax:

```
config>service# cpipe service-id [customer customer-id] [vpn vpn-id] [vc-type {satop-e1 |
 satop-t1 | cesopsn | cesopsn-cas}] [vc-switching] [create]
```

For the 7450 ESS platforms, the **vc-switching** option must be configured for Cpipe functionality, as follows:

```
cpipe 1 name "XYZ Cpipe 1" customer 1 vc-switching vc-type cesopsn create
            spoke-sdp 20:1 create
                description "Description for Sdp Bind 20 for Svc ID 1"
                ingress
                    vc-label 10002
                exit
                egress
                    vc-label 10001
                exit
            exit
            spoke-sdp 50:1 create
                description "Description for Sdp Bind 50 for Svc ID 1"
            exit
            no shutdown
        exit
```

The following displays a Cpipe service configuration example:

```
*A:ALA-1>config>service# info
----------------------------------------------
...
        cpipe 210 customer 1 vc-type cesopsn create
            service-mtu 1400
            sap 1/5/10.1.3.1 create
            exit
            spoke-sdp 1:210 create
            exit
            no shutdown
        exit
...
----------------------------------------------
*A:ALA-1>config>service#
```

2.13.2.1.2 Configuration requirements

Before a Cpipe service can be provisioned, a DS1 port and channel group must be configured. The subsequent sections provide example configurations for both.

2.13.2.1.2.1 Configuring a DS1 port

The following example shows a DS1 port configured for CES:

```
A:sim216#  show port 1/5/10.1.3.1
===============================================================================
TDM DS1 Interface
===============================================================================
Description        : DS1
Interface          : 1/5/10.1.3,1
```

```
Type               : ds1                      Framing              : esf
Admin Status       : up                       Oper Status          : up
Physical Link      : yes                      Clock Source         : loop-timed
Signal Mode        : none
Last State Change  : 10/31/2006 14:23:12      Channel IfIndex      : 580943939
Loopback           : none                     Invert Data          : false
Remote Loop respond: false                    In Remote Loop       : false
Load-balance-algo  : default                  Egr. Sched. Pol      : n/a
BERT Duration      : N/A                      BERT Pattern         : none
BERT Synched       : 00h00m00s                Err Insertion Rate   : 0
BERT Errors        : 0                        BERT Status          : idle
BERT Total Bits    : 0
Cfg Alarm          : ais los
Alarm Status       :
===============================================================================
A:sim216#
```

2.13.2.1.2.2 Configuring a channel group

The following example shows a DS1 channel group configured for CES:

```
A:sim216#  show port 1/5/10.1.3.1
===============================================================================
TDM DS0 Chan Group
===============================================================================
Description        : DS0GRP
Interface          : 1/5/10.1.3.1
TimeSlots          : 1-12
Speed              : 64                       CRC                  : 16
Admin Status       : up                       Oper Status          : up
Last State Change  : 10/31/2006 14:23:12      Chan-Grp IfIndex     : 580943940
Configured mode    : access                   Encap Type           : cem
Admin MTU          : 4112                     Oper MTU             : 4112
Physical Link      : Yes                      Bundle Number        : none
Idle Cycle Flags   : flags                    Load-balance-algo    : default
Egr. Sched. Pol    : n/a
===============================================================================
A:sim216#
```

2.13.2.1.3 Configuring Cpipe SAPs and spoke-SDPs

The following examples show Cpipe SAP and spoke-SDP configurations:

```
*A:ALA-49>config>service# info
#------------------------------------------------
echo "Service Configuration"
#------------------------------------------------
...
        cpipe 100 customer 1 vc-type cesopsn create
            service-mtu 1400
            sap 1/5/10.1.1.1 create
            exit
            spoke-sdp 1:100 create
            exit
            no shutdown
        exit
        cpipe 200 customer 1 vc-type cesopsn-cas create
            sap 1/5/10.2.1.1 create
```

```
            exit
            sap 1/5/10.2.2.1 create
            exit
            no shutdown
        exit
        cpipe 210 customer 1 vc-type cesopsn-cas create
            service-mtu 1400
            sap 1/5/10.1.3.1 create
            exit
            spoke-sdp 1:210 create
            exit
            no shutdown
        exit
        cpipe 300 customer 1 vc-type cesopsn create
            sap 1/5/10.3.4.1 create
            exit
            sap 1/5/10.3.6.1 create
            exit
            no shutdown
        exit
        cpipe 400 customer 1 vc-type satop-e1 create
            sap 1/5/10.2.3.1 create
            exit
            spoke-sdp 1:400 create
            exit
            no shutdown
        exit
...
#--------------------------------------------------
*A:ALA-49>config>service#
```

```
A:sim213>config>service>cpipe# info
----------------------------------------------
            description "cpipe-100"
            sap 1/5/10.1.1.1 create
                cem
                    packet jitter-buffer 16 payload-size 384
                    report-alarm rpktloss
                    no report-alarm stray
                    rtp-header
                exit
            exit
            spoke-sdp 1:100 create
            exit
            no shutdown
----------------------------------------------
A:sim213>config>service>cpipe#
```

2.13.2.2 Creating an Epipe service

Use the following CLI syntax to create an Epipe service.

CLI syntax:

```
config>service# epipe service-id [customer customer-id] [vpn vpn-id] [vc-switching]
description description-string
no shutdown
```

The following example shows an Epipe configuration:

```
A:ALA-1>config>service# info
----------------------------------------
...
        epipe 500 customer 5 vpn 500 create
            description "Local epipe service"
            no shutdown
        exit
----------------------------------------
A:ALA-1>config>service#
```

2.13.2.2.1 Configuring Epipe SAP parameters

A default QoS policy is applied to each ingress and egress SAP. Additional QoS policies can be configured in the **config**>**qos** context. Filter policies are configured in the config>filter context and explicitly applied to a SAP. There are no default filter policies.

Use the following CLI syntax to create:

- Local Epipe SAPs
- Distributed Epipe SAPs

The following example shows a configuration for the 7950 XRS.

CLI syntax:

```
config>service# epipe service-id [customer customer-id]
    – sap sap-id [endpoint endpoint-name]
    – sap sap-id [no-endpoint]
        – accounting-policy policy-id
        – collect-stats
        – description description-string
        – no shutdown
        – egress
            – filter {ip ip-filter-name | mac mac-filter-name}
            – qos sap-egress-policy-id
            – scheduler-policy scheduler-policy-name
        – ingress
            – filter {ip ip-filter-name | mac mac-filter-name}
            – match-qinq-dot1p {top | bottom}
            – qos policy-id
            – scheduler-policy scheduler-policy-name
```

The following example shows a configuration for the 7450 ESS and 7750 SR.

CLI syntax:

```
config>service# epipe service-id [customer customer-id]
    – sap sap-id [endpoint endpoint-name]
    – sap sap-id [no-endpoint]
        – accounting-policy policy-id
        – collect-stats
        – description description-string
        – no shutdown
        – egress
            – filter {ip ip-filter-name | mac mac-filter-name}
            – qos sap-egress-policy-id
            – scheduler-policy scheduler-policy-name
        – ingress
```

```
            – filter {ip ip-filter-name | mac mac-filter-name}
            – match-qinq-dot1p {top | bottom}
            – qos policy-id [shared-queuing]
            – scheduler-policy scheduler-policy-name
```

2.13.2.2.1.1 Local Epipe SAPs

To configure a basic local Epipe service, enter the **sap sap-id** command twice with different port IDs in the same service configuration.

By default, QoS policy ID 1 is applied to ingress and egress service SAPs. Existing filter policies or other existing QoS policies can be associated with service SAPs on ingress and egress ports. Table 8: Supported SAP types shows supported SAP types.

Table 8: Supported SAP types

Uplink type	Svc SAP type	Cust. VID	Access SAPs	Network SAPs
L2	Null-star	—	Null, dot1q *	Q.*
L2	Dot1q	—	Dot1q	Q.*
L2	Dot1q-preserve	—	Dot1q (encap = X)	Q1.Q2 (where Q2 = X)

An existing scheduler policy can be applied to ingress and egress SAPs to be used by the SAP queues and, at egress only, policers. The schedulers comprising the policy are created when the scheduler policy is applied to the SAP. If any policers or orphaned queues (with a non-existent local scheduler defined) exist on a SAP and the policy application creates the required scheduler, the status on the queue becomes non-orphaned at this time.

Ingress and egress SAP parameters can be applied to local and distributed Epipe service SAPs.

The following example shows the SAP configurations for local Epipe service 500 on SAP 1/1/2 and SAP 1/1/3 on ALA-1:

```
A:ALA-1>config>service# epipe 500 customer 5 create
    config>service>epipe$ description "Local epipe service"
    config>service>epipe# sap 1/1/2:0 create
    config>service>epipe>sap? ingress
    config>service>epipe>sap>ingress# qos 20
    config>service>epipe>sap>ingress# filter ip 1
    config>service>epipe>sap>ingress# exit
    config>service>epipe>sap# egress
    config>service>epipe>sap>egress# qos 20
    config>service>epipe>sap>egress# scheduler-policy test1
    config>service>epipe>sap>egress# exit
    config>service>epipe>sap# no shutdown
    config>service>epipe>sap# exit

    config>service>epipe# sap 1/1/3:0 create
    config>service>epipe>sap# ingress
    config>service>epipe>sap>ingress# qos 555
    config>service>epipe>sap>ingress# filter ip 1
    config>service>epipe>sap>ingress# exit
    config>service>epipe>sap# egress
```

```
    config>service>epipe>sap>egress# qos 627
    config>service>epipe>sap>egress# scheduler-policy alpha
    config>service>epipe>sap>egress# exit
    config>service>epipe>sap# no shutdown
    config>service>epipe>sap# exit
```

The following example shows the local Epipe configuration:

```
A:ALA-1>config>service# info
----------------------------------------------
...
        epipe 500 customer 5 vpn 500 create
            description "Local epipe service"
            sap 1/1/2:0 create
                ingress
                    qos 20
                    filter ip 1
                exit
                egress
                    scheduler-policy "test1"
                    qos 20
                exit
            exit
            sap 1/1/3:0 create
                ingress
                    qos 555
                    filter ip 1
                exit
                egress
                    scheduler-policy "alpha"
                    qos 627
                exit
            exit
            no shutdown
        exit
----------------------------------------------
A:ALA-1>config>service#
```

2.13.2.2.2 Distributed Epipe SAPs

To configure a distributed Epipe service, you must configure service entities on the originating and far-end nodes. You should use the same service ID on both ends (for example, Epipe 5500 on ALA-1 and Epipe 5500 on ALA-2). The **spoke-sdp sdp-id:vc-id** must match on both sides. A distributed Epipe consists of two SAPs on different nodes.

By default, QoS policy ID 1 is applied to ingress and egress service SAPs. Existing filter policies or other existing QoS policies can be associated with service SAPs on ingress and egress.

An existing scheduler policy can be applied to ingress and egress SAPs to be used by the SAP queues and, at egress only, policers. The schedulers comprising the policy are created when the scheduler policy is applied to the SAP. If any policers or orphaned queues (with a non-existent local scheduler defined) exist on a SAP and the policy application creates the required scheduler, the status on the queue becomes non-orphaned at this time.

Ingress and egress SAP parameters can be applied to local and distributed Epipe service SAPs.

For SDP configuration information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*. For SDP binding information, see Configuring SDP bindings.

The following example shows a configuration of a distributed service between ALA-1 and ALA-2:

```
A:ALA-1>epipe 5500 customer 5 create
    config>service>epipe$ description "Distributed epipe service to east coast"
    config>service>epipe# sap 221/1/3:21 create
    config>service>epipe>sap# ingress
    config>service>epipe>sap>ingress# qos 555
    config>service>epipe>sap>ingress# filter ip 1
    config>service>epipe>sap>ingress# exit
    config>service>epipe>sap# egress
    config>service>epipe>sap>egress# qos 627
    config>service>epipe>sap>egress# scheduler-policy alpha
    config>service>epipe>sap>egress# exit
    config>service>epipe>sap# no shutdown
    config>service>epipe>sap# exit
    config>service>epipe#

A:ALA-2>config>service# epipe 5500 customer 5 create
    config>service>epipe$ description "Distributed epipe service to west coast"
    config>service>epipe# sap 441/1/4:550 create
    config>service>epipe>sap# ingress
    config>service>epipe>sap>ingress# qos 654
    config>service>epipe>sap>ingress# filter ip 1020
    config>service>epipe>sap>ingress# exit
    config>service>epipe>sap# egress
    config>service>epipe>sap>egress# qos 432
    config>service>epipe>sap>egress# filter ip 6
    config>service>epipe>sap>egress# scheduler-policy test1
    config>service>epipe>sap>egress# exit
    config>service>epipe>sap# no shutdown
    config>service>epipe#
```

The following example shows the SAP configurations for ALA-1 and ALA-2:

```
A:ALA-1>config>service# info
----------------------------------------------
...
        epipe 5500 customer 5 vpn 5500 create
            description "Distributed epipe service to east coast"
            sap 221/1/3:21 create
                ingress
                    qos 555
                    filter ip 1
                exit
                egress
                    scheduler-policy "alpha"
                    qos 627
                exit
            exit
        exit
...
----------------------------------------------
A:ALA-1>config>service#


A:ALA-2>config>service# info
----------------------------------------------
...
        epipe 5500 customer 5 vpn 5500 create
            description "Distributed epipe service to west coast"
            sap 441/1/4:550 create
                ingress
                    qos 654
```

```
                    filter ip 1020
                exit
                egress
                    scheduler-policy "test1"
                    qos 432
                    filter ip 6
                exit
            exit
        exit
...
----------------------------------------------
A:ALA-2>config>service#
```

2.13.2.2.2.1 PBB Epipe configuration

The following example shows the PBB Epipe configuration:

```
*A:Wales-1>config>service>epipe# info
----------------------------------------------------------------------
...
    description "Default epipe description for service id 20000"
    pbb-tunnel 200 backbone-dest-mac 00:03:fa:15:d3:a8 isid 20000
    sap 1/1/2:1.1 create
        description "Default sap description for service id 20000"
        ingress
        filter mac 1
        exit
    exit
    no shutdown
----------------------------------------------------------------------
*A:Wales-1>config>service>epipe#
```

CLI syntax:

configure service vpls 200 customer 1 b-vpls create

```
*A:Wales-1>config>service>vpls# info
----------------------------------------------------------------------
...
    service-mtu 2000
    fdb-table-size 131071
    stp
    no shutdown
    exit
    sap 1/1/8 create
    exit
    sap 1/2/3:200 create
    exit
    mesh-sdp 1:200 create
    exit
    mesh-sdp 100:200 create
    exit
    mesh-sdp 150:200 create
    exit
    mesh-sdp 500:200 create
    exit
    no shutdown
----------------------------------------------------------------------
*A:Wales-1>config>service>vpls#
```

2.13.2.2.2.2 Configuring ingress and egress SAP parameters

By default, QoS policy ID 1 is applied to ingress and egress service SAPs. Existing filter policies or other existing QoS policies can be associated with service SAPs on ingress and egress ports.

An existing scheduler policy can be applied to ingress and egress SAPs to be used by the SAP queues and, at egress only, policers. The schedulers comprising the policy are created when the scheduler policy is applied to the SAP. If any policers or orphaned queues (with a non-existent local scheduler defined) exist on a SAP and the policy application creates the required scheduler, the status on the queue becomes non-orphaned at this time.

Ingress and egress SAP parameters can be applied to local and distributed Epipe service SAPs.

The following example shows the SAP ingress and egress parameters:

```
ALA-1>config>service# epipe 5500
    config>service>epipe# sap 2/1/3:21
    config>service>epipe>sap# ingress
    config>service>epipe>sap>ingress# qos 555
    config>service>epipe>sap>ingress# filter ip 1
    config>service>epipe>sap>ingress# exit
    config>service>epipe>sap# egress
    config>service>epipe>sap>egress# qos 627
    config>service>epipe>sap>egress# scheduler-policy alpha
    config>service>epipe>sap>egress# exit
    config>service>epipe>sap#
```

The following example shows the Epipe SAP ingress and egress configuration:

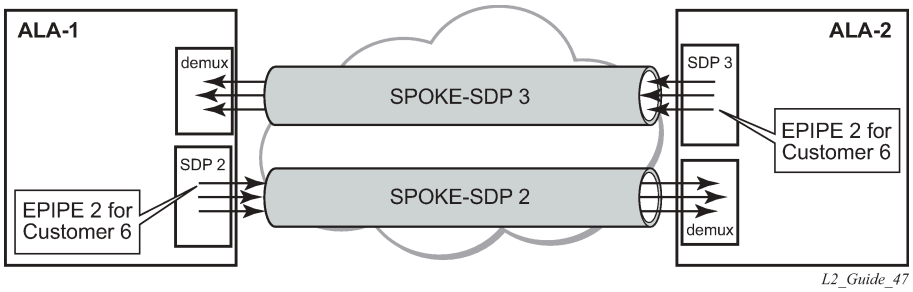
```
A:ALA-1>config>service#
------------------------------------------
...
        epipe 5500 customer 5 vpn 5500 create
            description "Distributed epipe service to east coast"
            sap 2/1/3:21 create
                ingress
                    qos 555
                    filter ip 1
                exit
                egress
                    scheduler-policy "alpha"
                    qos 627
                exit
            exit
            spoke-sdp 2:123 create
                ingress
                    vc-label 6600
                exit
                egress
                    vc-label 5500
                exit
            exit
            no shutdown
        exit
------------------------------------------
A:ALA-1>config>service#
```

2.13.2.2.3 Configuring SDP bindings

Figure 46: SDPs — unidirectional tunnels shows an example of a distributed Epipe service configuration between two routers, identifying the service and customer IDs, and the unidirectional SDPs required to communicate to the far-end routers.

A spoke-SDP is treated like the equivalent of a traditional bridge "port", where flooded traffic received on the spoke-SDP is replicated on all other "ports" (other spoke and mesh SDPs or SAPs) and not transmitted on the port it was received.

Figure 46: SDPs — unidirectional tunnels



Use the following CLI syntax to create a spoke-SDP binding with an Epipe service.

CLI syntax:

```
config>service# epipe service-id [customer customer-id]
    – spoke-sdp sdp-id:vc-id [vc-type {ether | vlan}]
        – vlan-vc-tag 0..4094
        – egress
            – filter {ip ip-filter-id}
            – vc-label egress-vc-label
        – ingress
            – filter {ip ip-filter-id}
            – vc-label ingress-vc-label
        – no shutdown
```

The following example shows the command usage to bind an Epipe service between ALA-1 and ALA-2. This example assumes the SAPs have already been configured (see Distributed Epipe SAPs).

```
A:ALA-1>config>service# epipe 5500
    config>service>epipe# spoke-sdp 2:123
    config>service>epipe>spoke-sdp# egress
    config>service>epipe>spoke-sdp>egress# vc-label 5500
    config>service>epipe>spoke-sdp>egress# exit
    config>service>epipe>spoke-sdp# ingress
    config>service>epipe>spoke-sdp>ingress# vc-label 6600
    config>service>epipe>spoke-sdp>ingress# exit
    config>service>epipe>spoke-sdp# no shutdown

    ALA-2>config>service# epipe 5500
    config>service>epipe# spoke-sdp 2:456
    config>service>epipe>spoke-sdp# egress
    config>service>epipe>spoke-sdp>egress# vc-label 6600
    config>service>epipe>spoke-sdp>egress# exit
    config>service>epipe>spoke-sdp# ingress
    config>service>epipe>spoke-sdp>ingress# vc-label 5500
    config>service>epipe>spoke-sdp>ingress# exit
    config>service>epipe>spoke-sdp# no shutdown
```

The following example shows the SDP binding for the Epipe service between ALA-1 and ALA-2:

```
A:ALA-1>config>service# info
----------------------------------------------
...
        epipe 5500 customer 5 vpn 5500 create
            description "Distributed epipe service to east coast"
            sap 2/1/3:21 create
                ingress
                    qos 555
                    filter ip 1
                exit
                egress
                    scheduler-policy "alpha"
                    qos 627
                exit
            exit
            spoke-sdp 2:123 create
                ingress
                    vc-label 6600
                exit
                egress
                    vc-label 5500
                exit
            exit
            no shutdown
        exit
...
----------------------------------------------
A:ALA-1>config>service#


A:ALA-2>config>service# info
----------------------------------------------
...
exit
        epipe 5500 customer 5 vpn 5500 create
            description "Distributed epipe service to west coast"
            sap 441/1/4:550 create
                ingress
                    qos 654
                    filter ip 1020
                exit
                egress
                    scheduler-policy "test1"
                    qos 432
                    filter ip 6
                exit
            exit
            spoke-sdp 2:456 create
                ingress
                    vc-label 5500
                exit
                egress
                    vc-label 6600
                exit
            exit
            no shutdown
        exit
...
----------------------------------------------
A:ALA-2>config>service#
```

2.13.2.3 Creating an Ipipe service

Use the following CLI syntax to create an Ipipe service on a 7450 ESS or 7750 SR.

CLI syntax:

```
config>service# ipipe service-id [customer customer-id] [vpn vpn-id] [vc-switching]
    – description description-string
    – no shutdown
```

The following example shows an Ipipe configuration:

```
A:ALA-1>config>service# info
----------------------------------------
...
        ipipe 202 customer 1 create
            description "eth_ipipe"
            no shutdown
        exit
----------------------------------------
A:ALA-1>config>service#
```

2.13.2.3.1 Configuring Ipipe SAP parameters

The following example shows an Ipipe SAP configuration:

```
A:ALA-48>config>service# info
----------------------------------------------
...
        ipipe 202 customer 1 create
            sap 1/1/2:444 create
                description "eth_ipipe"
                ce-address 31.31.31.1
            exit
            sap 1/3/2:445 create
                description "eth_ipipe"
                ce-address 31.31.31.2
            exit
            no shutdown
        exit
...
----------------------------------------------
A:ALA-48>config>service#
```

The following example shows the output:

```
A:ALA-48>config>service# info
----------------------------------------------
...
        ipipe 204 customer 1 create
            sap 1/1/2:444 create
                description "eth_ipipe"
                ce-address 32.32.32.1
            exit
            sap 2/2/2:445 create
                ce-address 32.32.32.2
            exit
            no shutdown
```

```
        exit
...
-------------------------------------------
A:ALA-48>config>service#
```

2.13.2.3.2 Configuring Ipipe SDP bindings

The following example shows an Ipipe SDP configuration:

```
A:ALA-48>config>service# info
-------------------------------------------
...
        sdp 16 mpls create
            far-end 10.4.4.4
            ldp
            path-mtu 1600
            keep-alive
                shutdown
            exit
            no shutdown
        exit
...
        ipipe 207 customer 1 create
            shutdown
            sap 1/1/2:449 create
                description "Remote_Ipipe"
                ce-address 10.34.34.1
            exit
            spoke-sdp 16:516 create
                ce-address 10.31.31.2
            exit
        exit
...

-------------------------------------------
A:ALA-48>config>service#
```

2.13.3 Using spoke SDP control words

The control word command provides the option to add a control word as part of the packet encapsulation for PW types for which the control word is optional. These are Ethernet pseudowire (Epipe). The control word may be needed because when ECMP is enabled on the network, packets of a specific pseudowire may be spread over multiple ECMP paths if the hashing router mistakes the PW packet payload for an IPv4 or IPv6 packet. This occurs when the first nibble following the service label corresponds to a value of 4 or 6.

The control word negotiation procedures described in Section 6.2 of RFC 4447 are not supported and, therefore, the service only comes up if the same C-bit value is signaled in both directions. If a spoke-SDP is configured to use the control word, but the node receives a label mapping message with a C-bit clear, the node releases the label with an "Illegal C-bit" status code per Section 6.1 of RFC 4447. As soon as the user enables control of the remote peer, the remote peer withdraws its original label and sends a label mapping with the C-bit set to 1 and the VLL service is up in both nodes.

When the control word is enabled, VCCV packets also include the VCCV control word. In that case, the VCCV CC type 1 (OAM CW) is signaled in the VCCV parameter in the FEC. If the control word is disabled on the spoke-SDP, the Router Alert label is used. In that case, VCCV CC type 2 is signaled. For a multi-

segment pseudowire (MS-PW), the CC type 1 is the only type supported; therefore, the control word must be enabled on the spoke SDP to be able to use VCCV-ping and VCCV-trace.

The following example shows a spoke SDP control word configuration:

```
-Dut-B>config>service>epipe# info
----------------------------------------------
    description "Default epipe description for service id 2100"
    sap 1/2/7:4 create
        description "Default sap description for service id 2100"
    exit
    spoke-sdp 1:2001 create
        control-word
    exit
    no shutdown
----------------------------------------------
*A:ALA-Dut-B>config>service>epipe#
To disable the control word on spoke-sdp 1:2001:
*A:ALA-Dut-B>config>service>epipe# info
----------------------------------------------
    description "Default epipe description for service id 2100"
    sap 1/2/7:4 create
        description "Default sap description for service id 2100"
    exit
    spoke-sdp 1:2001 create
    exit
    no shutdown
----------------------------------------------
*A:ALA-Dut-B>config>service>epipe#
```

2.13.4 Same-fate Epipe VLANs access protection

The following example shows a G.8031 Ethernet tunnel for Epipe protection configuration for 7450 ESS or 7750 SR using same-fate SAPs for each Epipe access (two Ethernet member ports 1/1/1 and 2/1/1/1 are used):

```
*A:7750_ALU>config>eth-tunnel 1
----------------------------------------------
        description "Protection is APS"
        protection-type 8031_1to1
        ethernet
            mac 00:11:11:11:11:12
            encap-type dot1q
        exit
        ccm-hold-time down 5 up 10 // 50 ms down, 1 second up
        path 1
            member 1/1/1
            control-tag 5 // primary control vlan 5
            precedence primary
            eth-cfm
                mep 2 domain 1 association 1
                    ccm-enable
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        path 2
            member 2/1/1
```

```
            control-tag 105 //secondary control vlan 105
            eth-cfm
                mep 2 domain 1 association 2
                    ccm-enable
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        no shutdown
-------------------------------------------------
# Configure Ethernet tunnel SAPs
-------------------------------------------------
*A:7750_ALU>config>service epipe 10 customer 5 create
        sap eth-tunnel-1 create // Uses control tags from the Ethernet tunnel port
            description "g8031-protected access ctl/data SAP for eth-tunnel 1"

        exit
        no shutdown
-----------------------------------------------
*A:7750_ALU>config>service epipe 11 customer 5 create
    sap eth-tunnel-1:1 create
        description "g8031-protected access same-fate SAP for eth-tunnel 1"

        // must specify tags for each corresponding path in Ethernet tunnel port
        eth-tunnel path 1 tag 6
        eth-tunnel path 2 tag 106
    exit
    …
-----------------------------------------------
*A:7750_ALU>config>service epipe 10 customer 5 create
    sap eth-tunnel-1:3 create
        description "g8031-protected access same-fate SAP for eth-tunnel 1"
        // must specify tags for each path for same-fate SAPs
        eth-tunnel path 1 tag 10
        eth-tunnel path 2 tag 110
    exit
    …
-----------------------------------------------
```

2.13.5 Pseudowire configuration notes

The **vc-switching** parameter must be specified when the VLL service is created. When the **vc-switching** parameter is specified, you are configuring an S-PE. This is a pseudowire switching point (switching from one pseudowire to another). Therefore, you cannot add a SAP to the configuration.

The following example shows the configuration when a SAP is added to a pseudowire. The CLI generates an error response if you attempt to create a SAP. VC switching is only needed on the pseudowire at the S-PE.

```
*A:ALA-701>config>service# epipe 28 customer 1 create vc-switching
*A:ALA-701>config>service>epipe$ sap 1/1/3 create
MINOR: SVCMGR #1311 SAP is not allowed under PW switching service
*A:ALA-701>config>service>epipe$
```

Use the following CLI syntax to create pseudowire switching VLL services. These are examples only. Different routers support different pseudowire switching VLL services.

CLI syntax:

```
config>service# epipe service-id [customer customer-id] [vpn vpn-id] [vc-switching]
description description-string
spoke-sdp sdp-id:vc-id
```

CLI syntax:

```
config>service# ipipe service-id [customer customer-id][vpn vpn-id] [vc-switching]
description description-string
spoke-sdp sdp-id:vc-id
```

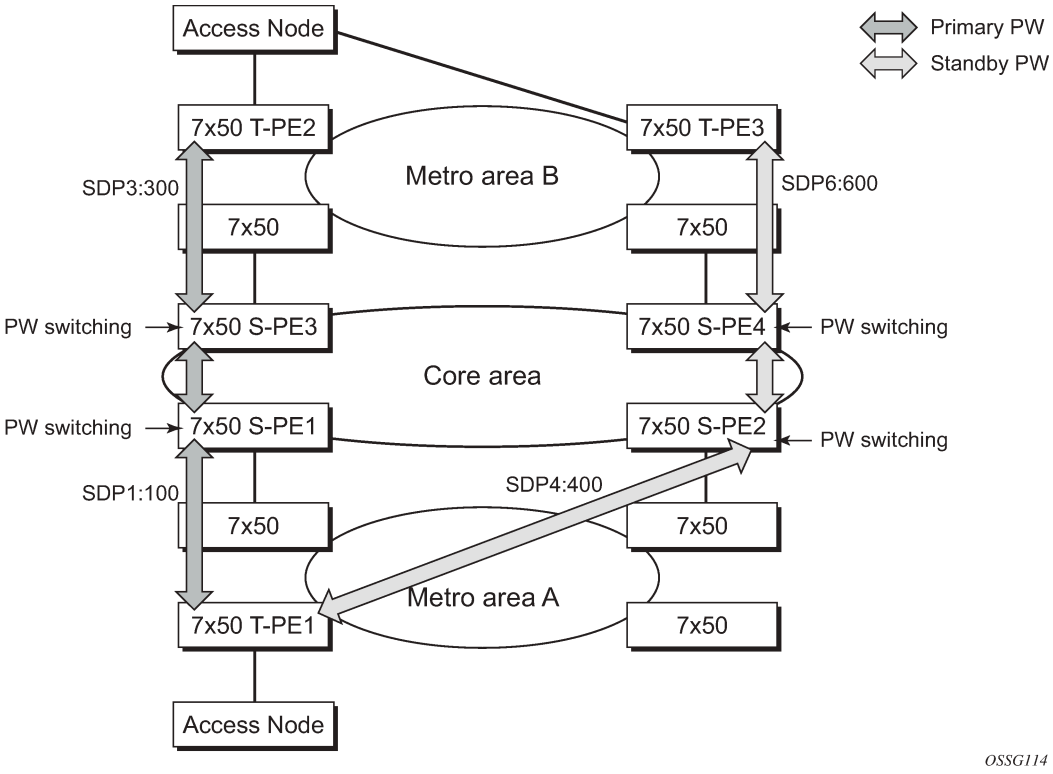
The following example shows configurations for each service:

```
*A:ALA-48>config>service# info
----------------------------------------------
...

        epipe 107 customer 1 vpn 107 vc-switching create
            description "Default epipe description for service id 107"
            spoke-sdp 3:8 create
            exit
            spoke-sdp 6:207 create
            exit
            no shutdown
        exit
...
        ipipe 108 customer 1 vpn 108 vc-switching create
            description "Default ipipe description for service id 108"
            spoke-sdp 3:9 create
            exit
            spoke-sdp 6:208 create
            exit
            no shutdown
        exit
...
```

2.13.6 Configuring two VLL paths terminating on T-PE2

Figure 47: VLL resilience with pseudowire redundancy and switching



T-PE1

The following shows an example of the T-PE1 configuration:

```
*A:ALA-T-PE1>config>service>epipe# info
----------------------------------------------
            endpoint "x" create
            exit
            endpoint "y" create
            exit
            spoke-sdp 1:100 endpoint "y" create
                precedence primary
                revert-time 0
            exit
            spoke-sdp 4:400 endpoint "y" create
                precedence 0
            exit
            no shutdown
----------------------------------------------
*A:ALA-T-PE1>config>service>epipe#
```

The following shows an example of the T-PE2 configuration for 7950 XRS.

T-PE2

```
*A:ALA-T-PE2>config>service>epipe# info
```

```
-----------------------------------------------
            endpoint "x" create
            exit
            endpoint "y" create
            exit
            sap endpoint "x" create
            exit
            spoke-sdp 3:300 endpoint "y" create
                precedence primary
                revert-time 0
            exit
            spoke-sdp 6:600 endpoint "y" create
                precedence 0
            exit
            no shutdown
-----------------------------------------------
*A:ALA-T-PE2>config>service>epipe#
```

The following shows and example of the T-PE2 configuration for 7450 ESS and 7750 SR.

T-PE2

```
*A:ALA-T-PE2>config>service>epipe# info
-----------------------------------------------
            endpoint "x" create
            exit
            endpoint "y" create
            exit
            sap 2/2/2:200 endpoint "x" create
            exit
            spoke-sdp 3:300 endpoint "y" create
                precedence primary
                revert-time 0
            exit
            spoke-sdp 6:600 endpoint "y" create
                precedence 0
            exit
            no shutdown
-----------------------------------------------
*A:ALA-T-PE2>config>service>epipe#
```

S-PE1

Specifying the **vc-switching** parameter enables a VC cross-connect, so the service manager does not signal the VC label mapping immediately, but puts this into passive mode.

The following example shows the configuration:

```
*A:ALA-S-PE1>config>service>epipe# info
-----------------------------------------------
...
            spoke-sdp 2:200 create
            exit
            spoke-sdp 3:300 create
            exit
            no shutdown
-----------------------------------------------
*A:ALA-S-PE1>config>service>epipe#
```

S-PE2

Specifying the **vc-switching** parameter enables a VC cross-connect, so the service manager does not signal the VC label mapping immediately, but puts this into passive mode.

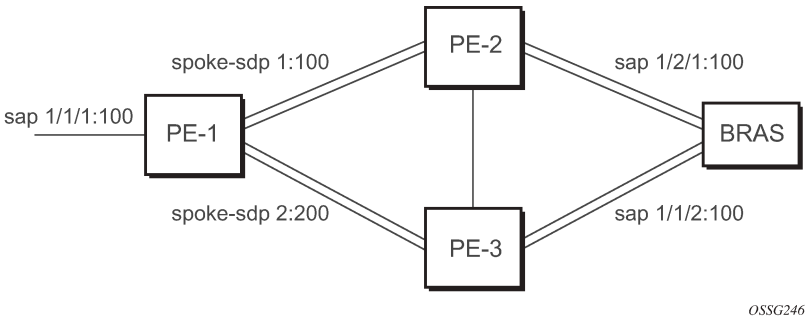
The following example shows the configuration:

```
*A:ALA-S-PE2>config>service>epipe# info
----------------------------------------------
...
            spoke-sdp 2:200 create
            exit
            spoke-sdp 3:300 create
            exit
            no shutdown
----------------------------------------------
*A:ALA-S-PE2>config>service>epipe#
```

2.13.7 Configuring VLL resilience

Figure 48: VLL resilience shows an example to create VLL resilience. The zero revert-time value means that the VLL path is switched back to the primary immediately after it comes back up.

Figure 48: VLL resilience



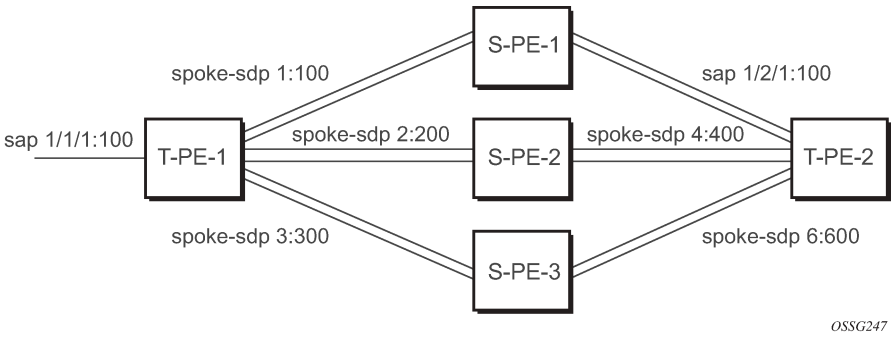
PE-1:

The following example shows the configuration on PE-1:

```
*A:ALA-48>config>service>epipe# info
----------------------------------------------
            endpoint "x" create
            exit
            endpoint "y" create
            exit
            spoke-sdp 1:100 endpoint "y" create
                precedence primary
            exit
            spoke-sdp 2:200 endpoint "y" create
                precedence 1
            exit
            no shutdown
----------------------------------------------
*A:ALA-48>config>service>epipe#
```

2.13.8 Configuring VLL resilience for a switched pseudowire path

Figure 49: VLL resilience with pseudowire switching



T-PE-1

The following example shows the configuration on T-PE-1.

```
*A:ALA-48>config>service>epipe# info
----------------------------------------------
            endpoint "x" create
            exit
            endpoint "y" create
            exit
            sap 1/1/1:100 endpoint "x" create
            exit
            spoke-sdp 1:100 endpoint "y" create
                precedence primary
            exit
            spoke-sdp 2:200 endpoint "y" create
                precedence 1
            exit
            spoke-sdp 3:300 endpoint "y" create
                precedence 1
            exit
            no shutdown
----------------------------------------------
*A:ALA-48>config>service>epipe#
```

T-PE-2

The following example shows the configuration on T-PE-2.

```
*A:ALA-49>config>service>epipe# info
----------------------------------------------
            endpoint "x" create
            exit
            endpoint "y" create
                revert-time 100
            exit
            spoke-sdp 4:400 endpoint "y" create
                precedence primary
            exit
            spoke-sdp 5:500 endpoint "y" create
                precedence 1
```

```
            exit
            spoke-sdp 6:600 endpoint "y" create
                precedence 1
            exit
            no shutdown
----------------------------------------------
*A:ALA-49>config>service>epipe#
```

S-PE-1

The following example shows the configuration on S-PE-1.

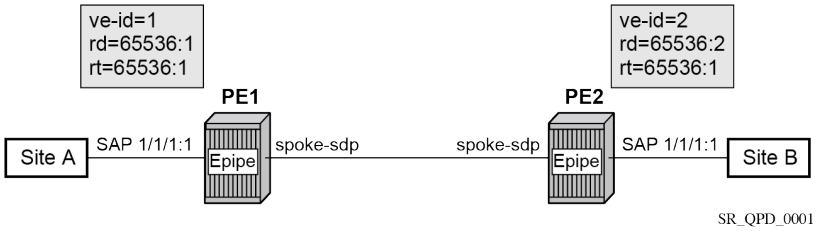
```
*A:ALA-50>config>service>epipe# info
----------------------------------------------
...
            spoke-sdp 1:100 create
            exit
            spoke-sdp 4:400 create
            exit
            no shutdown
----------------------------------------------
*A:ALA-49>config>service>epipe#
```

2.13.9 Configuring BGP VPWS

2.13.9.1 Single-homed BGP VPWS

The following figure shows an example topology for a BGP VPWS service used to create a virtual lease-line across an MPLS network between sites A and B.

Figure 50: Single-homed BGP VPWS configuration example



An Epipe is configured on PE1 and PE2 with BGP VPWS enabled. PE1 and PE2 are connected to site A and B, respectively, each using a SAP. The interconnection between the two PEs is achieved through a pseudowire, using Ethernet VLAN encapsulation, which is signaled using BGP VPWS over a tunnel LSP between PE1 and PE2. A MIP or MEP can be configured on a BGP VPWS SAP. However, fault propagation between a MEP and the BGP update state signaling is not supported. BGP VPWS routes are accepted only over an IBGP session.

The following example shows the BGP VPWS configuration on each PE.

```
PE1:
pw-template 1 create
    vc-type vlan
```

```
exit
epipe 1 customer 1 create
    bgp
        route-distinguisher 65536:1
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
        exit
    exit
    bgp-vpws
        ve-name PE1
            ve-id 1
        exit
        remote-ve-name PE2
            ve-id 2
        exit
        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    no shutdown
exit

PE2:

pw-template 1 create
    vc-type vlan
exit
epipe 1 customer 1 create
    bgp
        route-distinguisher 65536:2
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
        exit
    exit
    bgp-vpws
        ve-name PE2
            ve-id 2
        exit
        remote-ve-name PE1
            ve-id 1
        exit
        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    no shutdown
exit
```

The BGP-VPWS update can be displayed using the following command:

```
A:PE1# show service l2-route-table bgp-vpws detail
===============================================================================
Services: L2 Bgp-Vpws Route Information - Summary
===============================================================================
Svc Id         : 1
VeId           : 2
PW Temp Id     : 1
RD             : *65536:2
Next Hop       : 10.1.1.2
State (D-Bit)  : up(0)
Path MTU       : 1514
Control Word   : 0
Seq Delivery   : 0
```

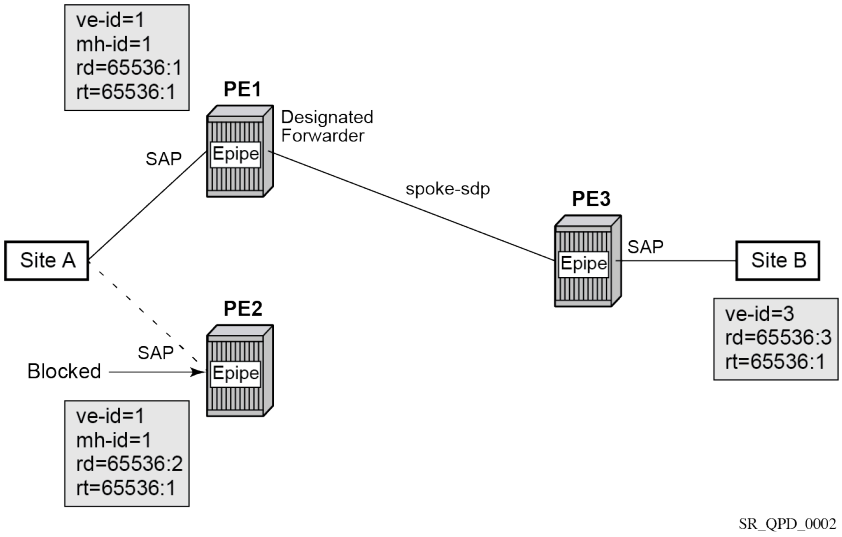
```
Status         : active
Tx Status      : active
CSV            : 0
Preference     : 0
Sdp Bind Id    : 17407:4294967295
===============================================================================
A:PE1#
```

2.13.9.2 Dual-homed BGP VPWS

Single pseudowire example:

The following figure shows an example topology for a dual-homed BGP VPWS service used to create a virtual lease line (VLL) across an MPLS network between sites A and B. A single pseudowire is established between the designated forwarder of the dual-homed PEs and the remote PE.

Figure 51: Example of dual-homed BGP VPWS with single pseudowire



An Epipe with BGP VPWS enabled is configured on each PE. Site A is dual-homed to PE1 and PE2 with a remote PE (PE3) connected to site B; each connection uses a SAP. A single pseudowire using Ethernet Raw Mode encaps connects PE3 to PE1. The pseudowire is signaled using BGP VPWS over a tunnel LSP between the PEs.

Site A is configured on PE1 and PE2 with the BGP route selection, the site state, and the site-preference used to ensure PE1 is the designated forwarder when the network is fully operational.

The following example shows the BGP VPWS configuration on each PE.

The following example shows the configuration on PE1:

```
pw-template 1 create
exit
epipe 1 customer 1 create
    bgp
        route-distinguisher 65536:1
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
```

```
        exit
    exit
    bgp-vpws
        ve-name PE1
            ve-id 1
        exit
        remote-ve-name PE3
            ve-id 3
        exit
        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    site "siteA" create
        site-id 1
        sap 1/1/1:1
        boot-timer 20
        site-activation-timer 5
        no shutdown
    exit
    no shutdown
exit
```

The following example shows the configuration on PE2:

```
pw-template 1 create
exit
epipe 1 customer 1 create
    bgp
        route-distinguisher 65536:2
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
        exit
    exit
    bgp-vpws
        ve-name PE2
            ve-id 1
        exit
        remote-ve-name PE3
            ve-id 3
        exit
        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    site "siteA" create
        site-id 1
        sap 1/1/1:1
        boot-timer 20
        site-activation-timer 5
        no shutdown
    exit
    no shutdown
exit
```

The following example shows the configuration on PE3:

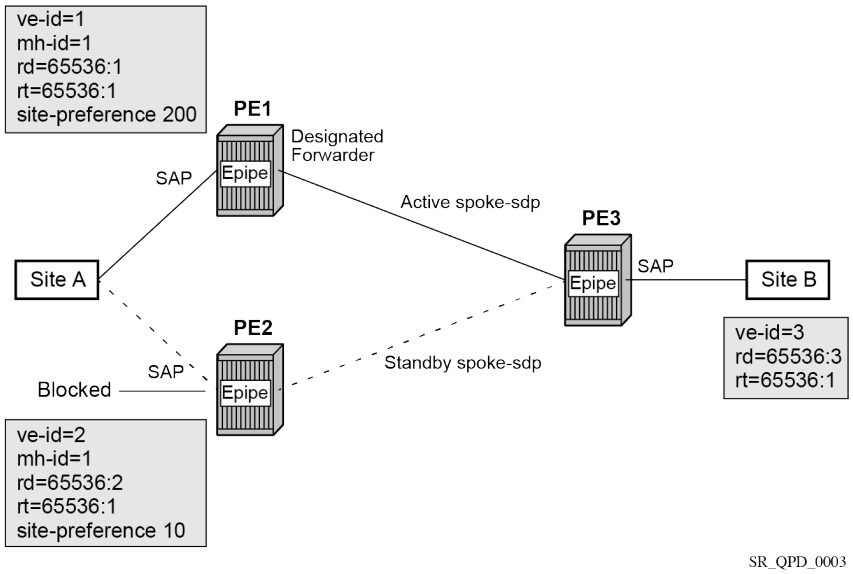
```
pw-template 1 create
exit
epipe 1 customer 1 create
    bgp
```

```
        route-distinguisher 65536:3
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
        exit
    exit
    bgp-vpws
        ve-name PE3
            ve-id 3
        exit
        remote-ve-name PE1orPE2
            ve-id 1
        exit
        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    no shutdown
exit
```

Active/standby pseudowire example:

The following figure shows an example topology for a dual-homed BGP VPWS service used to create a VLL across an MPLS network between sites A and B. Two pseudowires are established between the remote PE and the dual-homed PEs. The active pseudowire used for the traffic is the one connecting the remote PE to the designated forwarder of the dual-homed PEs.

Figure 52: Example of dual-homed BGP VPWS with active/standby pseudowires



An Epipe with BGP VPWS enabled is configured on each PE. Site A is dual-homed to PE1 and PE2 with a remote PE (PE3) connected to site B; each connection uses a SAP. Active/standby pseudowires using Ethernet Raw Mode encapsulation connect PE3 to PE1 and PE2, respectively. The pseudowires are signaled using BGP VPWS over a tunnel LSP between the PEs.

Site A is configured on PE1 and PE2 with the **site-preference** set to ensure that PE1 is the designated forwarder when the network is fully operational. An endpoint is automatically created on PE3 in which the active/standby pseudowires are created.

The following example shows the BGP VPWS configuration on each PE.

Example: Dual-homed BGP VPWS configuration with active/standby pseudowires on PE1:

```
pw-template 1 create
exit
epipe 1 customer 1 create
    bgp
        route-distinguisher 65536:1
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
        exit
    exit
    bgp-vpws
        ve-name PE1
            ve-id 1
        exit
        remote-ve-name PE3
            ve-id 3
        exit
        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    site "siteA" create
        site-id 1
        sap 1/1/1:1
        boot-timer 20
        site-activation-timer 5
        site-preference 200
        no shutdown
    exit
    no shutdown
exit
```

Example: Dual-homed BGP VPWS configuration with active/standby pseudowires on PE2:

```
pw-template 1 create
exit
epipe 1 customer 1 create
    bgp
        route-distinguisher 65536:2
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
        exit
    exit
    bgp-vpws
        ve-name PE2
            ve-id 2
        exit
        remote-ve-name PE3
            ve-id 3
        exit
        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    site "siteA" create
        site-id 1
        sap 1/1/1:1
        boot-timer 20
        site-activation-timer 5
        site-preference 10
        no shutdown
```

```
    exit
    no shutdown
exit
```

Example: Dual-homed BGP VPWS configuration with active/standby pseudowires on PE3:

```
pw-template 1 create
exit
epipe 1 customer 1 create
    bgp
        route-distinguisher 65536:3
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
        exit
    exit
    bgp-vpws
        ve-name PE3
            ve-id 3
        exit
        remote-ve-name PE1
            ve-id 1
        exit
        remote-ve-name PE2
            ve-id 2
        exit
        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    no shutdown
exit
```

2.14 Service management tasks

This section discusses VLL service management tasks.

2.14.1 Modifying a Cpipe service

The following example shows the Cpipe service configuration, supported on the 7750 SR only:

```
*A:ALA-1>config>service# info
----------------------------------------------
...
        cpipe 94002 customer 1 vc-type cesopsn create
            endpoint "to7705" create
            exit
            endpoint "toMC-APS" create
            exit
            sap aps-4.10.1.2.1 endpoint "toMC-APS" create
                ingress
                    qos 20
                exit
            exit
            spoke-sdp 14004:94002 endpoint "to7705" create
            exit
            spoke-sdp 100:294002 endpoint "toMC-APS" icb create
```

```
            exit
            spoke-sdp 100:194002 endpoint "to7705" icb create
            exit
            no shutdown
        exit
...
-----------------------------------------------
*A:ALA-1>config>service> Cpipe#
```

2.14.2 Deleting a Cpipe service

A Cpipe service cannot be deleted until SAPs are shut down and deleted. If a spoke-SDP is defined, it must be shut down and removed from the configuration as well.

Use the following CLI syntax to delete a Cpipe service.

CLI syntax:

```
config>service#
[no] cpipe service-id [customer customer-id]
   [no] spoke-sdp sdp-id
      [no] shutdown
   shutdown
```

2.14.3 Modifying Epipe service parameters

The following example shows how to add an accounting policy to an existing SAP:

```
config>service# epipe 2
config>service>epipe# sap 2/1/3:21
config>service>epipe>sap# accounting-policy 14
config>service>epipe>sap# exit
```

The following example shows the SAP configuration:

```
ALA-1>config>service# info
----------------------------------------------
    epipe 2 customer 6 vpn 2 create
            description "Distributed Epipe service to east coast"
            sap 2/1/3:21 create
              accounting-policy 14
            exit
            spoke-sdp 2:6000 create
            exit
            no shutdown
        exit
----------------------------------------------
ALA-1>config>service#
```

2.14.4 Disabling an Epipe service

You can shut down an Epipe service without deleting the service parameters.

CLI syntax:

```
config>service> epipe service-id
shutdown
```

Example:

```
config>service# epipe 2
config>service>epipe# shutdown
config>service>epipe# exit
```

2.14.5 Re-enabling an Epipe service

Use the following CLI syntax to re-enable an Epipe service that was shut down.

CLI syntax:

```
config>service# epipe service-id
no shutdown
```

Example:

```
config>service# epipe 2
config>service>epipe# no shutdown
config>service>epipe# exit
```

2.14.6 Deleting an Epipe service

About this task

Perform the following steps to delete an Epipe service:

Procedure

Step 1. Shut down the SAP and SDP.

Step 2. Delete the SAP and SDP.

Step 3. Shut down the service.

Step 4. Use the following CLI syntax to delete an Epipe service:

```
config>service
      [no] epipe service-id
          shutdown
          [no] sap sap-id
              shutdown
      [no] spoke-sdp sdp-id:vc-id
              shutdown
```

Example

```
config>service# epipe 2
config>service>epipe# sap 2/1/3:21
config>service>epipe>sap# shutdown
config>service>epipe>sap# exit
```

```
config>service>epipe# no sap 2/1/3:21
config>service>epipe# spoke-sdp 2:6000
config>service>epipe>spoke-sdp# shutdown
config>service>epipe>spoke-sdp# exit
config>service>epipe# no spoke-sdp 2:6000
config>service>epipe# epipe 2
config>service>epipe# shutdown
config>service>epipe# exit
config>service# no epipe 2
```

2.14.7 Modifying Ipipe service parameters

The following example shows the command usage to modify Ipipe parameters, supported on the 7450 ESS and 7750 SR only.

Example:

```
config>service# ipipe 202
config>service>ipipe# sap 1/1/2:444
config>service>ipipe>sap# shutdown
config>service>ipipe>sap# exit
config>service>ipipe# no sap 1/1/2:444
config>service>ipipe# sap 1/1/2:555 create
config>service>ipipe>sap$ description “eth_ipipe”
config>service>ipipe>sap$ ce-address 10.31.31.1
config>service>ipipe>sap$ no shutdown
config>service>ipipe>sap$ exit
config>service>ipipe# info
```

```
A:ALA-48>config>service# info
----------------------------------------------
...
        ipipe 202 customer 1 create
            sap 1/1/2:445 create
                description "eth_ipipe"
                ce-address 10.31.31.2
            exit
            sap 1/1/2:555 create
                description "eth_ipipe"
                ce-address 10.31.31.1
            exit
            no shutdown
        exit
...
----------------------------------------------
A:ALA-48>config>service#
```

2.14.8 Disabling an Ipipe service

An Ipipe service can be shut down without deleting any service parameters.

CLI syntax:

```
config>service#
ipipe service-id
  shutdown
```

Example:

```
A:ALA-41>config>service# ipipe 202
A:ALA-41>config>service>ipipe# shutdown
```

```
A:ALA-48>config>service# info
----------------------------------------------
...
       ipipe 202 customer 1 create
           shutdown
           sap 1/1/2:445 create
               description "eth_ipipe"
               ce-address 10.31.31.2
           exit
           sap 1/1/2:555 create
               description "eth_ipipe"
               ce-address 10.31.31.1
           exit
       exit
...
----------------------------------------------
A:ALA-48>config>service#
```

2.14.9 Re-enabling an Ipipe service

Use the following CLI syntax to re-enable an Ipipe service that was shut down.

CLI syntax:

```
config>service#
ipipe service-id
   no shutdown
```

Example:

```
A:ALA-41>config>service# ipipe 202
A:ALA-41>config>service>ipipe# no shutdown
```

2.14.10 Deleting an Ipipe service

An Ipipe service cannot be deleted until the SAP is shut down. If protocols or a spoke-SDP, or both are defined, they must be shut down and removed from the configuration as well.

Use the following CLI syntax to delete an Ipipe service.

CLI syntax:

```
config>service#
    – no ipipe service-id
        – shutdown
        – no sap sap-id
            – shutdown
        – no spoke-sdp [sdp-id:vc-id]
            – shutdown
```

Example:

```
config>service# ipipe 207
config>service>ipipe# sap 1/1/2:449
config>service>ipipe>sap# shutdown
config>service>ipipe>sap# exit
config>service>ipipe# no sap 1/1/2:449
config>service>ipipe# spoke-sdp 16:516
config>service>ipipe>spoke-sdp# shutdown
config>service>ipipe>spoke-sdp# exit
config>service>ipipe# no spoke-sdp 16:516
config>service>ipipe# exit
config>service# no ipipe 207
config>service#
```

3 Virtual private LAN service

3.1 VPLS service overview

VPLS as described in RFC 4905, *Encapsulation methods for transport of layer 2 frames over MPLS*, is a class of virtual private network service that allows the connection of multiple sites in a single bridged domain over a provider-managed IP/MPLS network. The customer sites in a VPLS instance appear to be on the same LAN, regardless of their location. VPLS uses an Ethernet interface on the customer-facing (access) side, which simplifies the LAN/WAN boundary and allows for rapid and flexible service provisioning.

VPLS offers a balance between point-to-point Frame Relay service and outsourced routed services (VPRN). VPLS enables each customer to maintain control of their own routing strategies. All customer routers in the VPLS service are part of the same subnet (LAN), which simplifies the IP addressing plan, especially when compared to a mesh constructed from many separate point-to-point connections. The VPLS service management is simplified because the service is not aware of nor participates in the IP addressing and routing.

A VPLS service provides connectivity between two or more SAPs on one (which is considered a local service) or more (which is considered a distributed service) service routers. The connection appears to be a bridged domain to the customer sites so protocols, including routing protocols, can traverse the VPLS service.

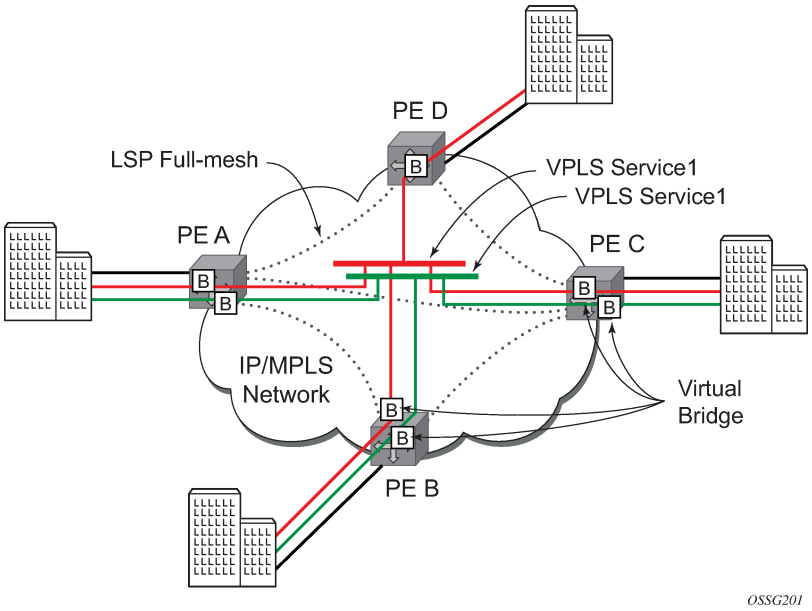
Other VPLS advantages include:

- VPLS is a transparent, protocol-independent service.
- There is no Layer 2 protocol conversion between LAN and WAN technologies.
- There is no need to design, manage, configure, and maintain separate WAN access equipment, which eliminates the need to train personnel on WAN technologies.

3.1.1 VPLS packet walkthrough

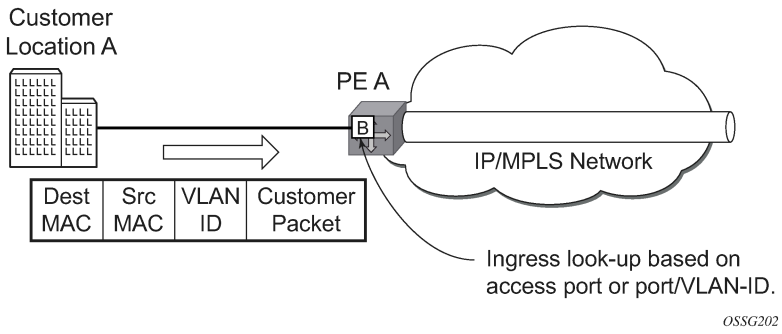
This section provides an example of VPLS processing of a customer packet sent across the network from site A, which is connected to PE Router A, to site B, which is connected to PE Router C (see Figure 53: VPLS service architecture).

Figure 53: VPLS service architecture



1. PE Router A (see Figure 54: Access port ingress packet format and lookup)
   a. Service packets arriving at PE Router A are associated with a VPLS service instance based on the combination of the physical port and the IEEE 802.1Q tag (VLAN ID) in the packet.

      Figure 54: Access port ingress packet format and lookup



   b. PE Router A learns the source MAC address in the packet and creates an entry in the FDB table that associates the MAC address with the service access point (SAP) on which it was received.
   c. The destination MAC address in the packet is looked up in the FDB table for the VPLS instance. There are two possibilities: either the destination MAC address has already been learned (known MAC address) or the destination MAC address is not yet learned (unknown MAC address).

      For a known MAC address, see Figure 55: Network port egress packet format and flooding and proceed to 1.d.

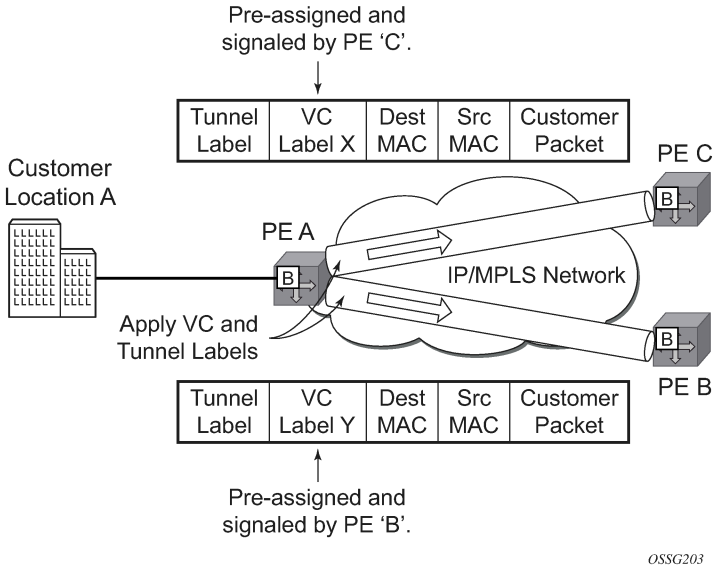
      For an unknown MAC address, see Figure 55: Network port egress packet format and flooding and proceed to 1.f.

d. If the destination MAC address has already been learned by PE Router A, an existing entry in the FDB table identifies the far-end PE router and the service VC-label (inner label) to be used before sending the packet to far-end PE Router C.

e. PE Router A chooses a transport LSP to send the customer packets to PE Router C. The customer packet is sent on this LSP after the IEEE 802.1Q tag is stripped and the service VC-label (inner label) and the transport label (outer label) are added to the packet.

f. If the destination MAC address has not been learned, PE Router A floods the packet to both PE Router B and PE Router C that are participating in the service by using the VC-labels that each PE Router previously added for the VPLS instance. The packet is not sent to PE Router D because this VPLS service does not exist on that PE router.

Figure 55: Network port egress packet format and flooding



2. Core Router Switching

All the core routers ("P" routers in IETF nomenclature) between PE Router A and PE Router B and PE Router C are Label Switch Routers (LSRs) that switch the packet based on the transport (outer) label of the packet until the packet arrives at the far-end PE Router. All core routers are unaware that this traffic is associated with a VPLS service.

3. PE router C

a. PE Router C strips the transport label of the received packet to reveal the inner VC-label. The VC-label identifies the VPLS service instance to which the packet belongs.

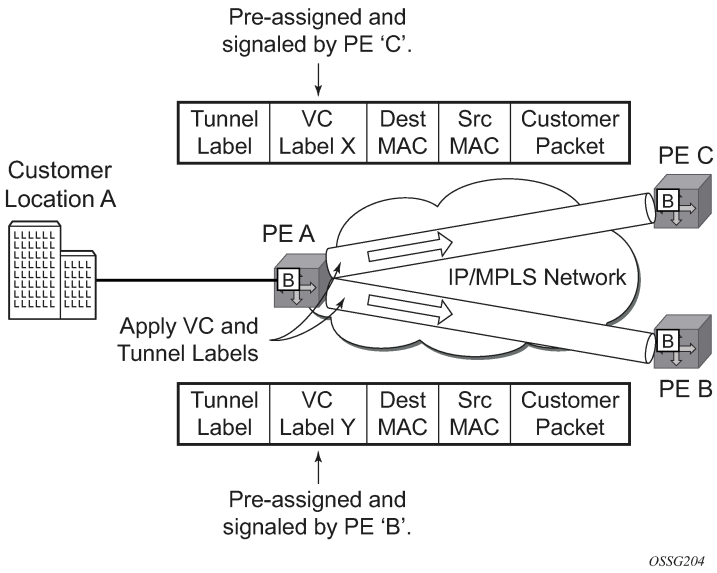
b. PE Router C learns the source MAC address in the packet and creates an entry in the FDB table that associates the MAC address with PE Router A, and the VC-label that PE Router A added for the VPLS service on which the packet was received.

c. The destination MAC address in the packet is looked up in the FDB table for the VPLS instance. Again, there are two possibilities: either the destination MAC address has already been learned (known MAC address) or the destination MAC address has not been learned on the access side of PE Router C (unknown MAC address).

For a known MAC address see Figure 56: Access port egress packet format and lookup.

If the destination MAC address has been learned by PE Router C, an existing entry in the FDB table identifies the local access port and the IEEE 802.1Q tag to be added before sending the packet to customer Location C. The egress Q tag may be different than the ingress Q tag.

Figure 56: Access port egress packet format and lookup



3.2 VPLS features

This section provides information about VPLS features.

3.2.1 VPLS enhancements

Nokia's VPLS implementation includes several enhancements beyond basic VPN connectivity. The following VPLS features can be configured individually for each VPLS service instance:

- Extensive MAC and IP filter support (up to Layer 4). Filters can be applied on a per-SAP basis.
- Forwarding Database (FDB) management features on a per service-level basis including:
  - Configurable FDB size limit. On the 7450 ESS, it can be configured on a per-VPLS, per-SAP, and per spoke-SDP basis.
  - FDB size alarms. On the 7450 ESS, it can be configured on a per-VPLS basis.
  - MAC learning disable. On the 7450 ESS, it can be configured on a per-VPLS, per-SAP, and per spoke-SDP basis.
  - Discard unknown. On the 7450 ESS, it can be configured on a per-VPLS basis.
  - Separate aging timers for locally and remotely learned MAC addresses.
- Ingress rate limiting for broadcast, multicast, and unknown destination flooding on a per-SAP basis.
- Implementation of STP parameters on a per-VPLS, per-SAP, and per spoke-SDP basis.

- A split horizon group on a per-SAP and per spoke-SDP basis.
- DHCP snooping and anti-spoofing on a per-SAP and per-SDP basis for the 7450 ESS or 7750 SR.
- IGMP snooping on a per-SAP and per-SDP basis.
- Optional SAP or spoke-SDP, or both, redundancy to protect against node failure.

3.2.2 VPLS over MPLS

The VPLS architecture proposed in RFC 4762, *Virtual Private LAN Services Using LDP Signaling* specifies the use of provider equipment (PE) that is capable of learning, bridging, and replication on a per-VPLS basis. The PE routers that participate in the service are connected using MPLS Label Switched Path (LSP) tunnels in a full-mesh composed of mesh SDPs or based on an LSP hierarchy (Hierarchical VPLS (H-VPLS)) composed of mesh SDPs and spoke-SDPs.

Multiple VPLS services can be offered over the same set of LSP tunnels. Signaling specified in RFC 4905, *Encapsulation methods for transport of layer 2 frames over MPLS* is used to negotiate a set of ingress and egress VC labels on a per-service basis. The VC labels are used by the PE routers for demultiplexing traffic arriving from different VPLS services over the same set of LSP tunnels.

VPLS is provided over MPLS by:

- connecting bridging-capable provider edge routers with a full mesh of MPLS LSP tunnels
- negotiating per-service VC labels using Draft-Martini encapsulation
- replicating unknown and broadcast traffic in a service domain
- enabling MAC learning over tunnel and access ports (see VPLS MAC learning and packet forwarding)
- using a separate FDB per VPLS service

3.2.3 VPLS service pseudowire VLAN tag processing

VPLS services can be connected using pseudowires that can be provisioned statically or dynamically and are represented in the system as either a mesh or a spoke-SDP. The mesh and spoke-SDP can be configured to process zero, one, or two VLAN tags as traffic is transmitted and received. In the transmit direction, VLAN tags are added to the frame being sent, and in the received direction, VLAN tags are removed from the frame being received. This is analogous to the SAP operations on a null, dot1q, and QinQ SAP.

The system expects a symmetrical configuration with its peer; specifically, it expects to remove the same number of VLAN tags from received traffic as it adds to transmitted traffic. When removing VLAN tags from a mesh or spoke-SDP, the system attempts to remove the configured number of VLAN tags (see the following configuration information); if fewer tags are found, the system removes the VLAN tags found and forwards the resulting packet. As some of the related configuration parameters are local and not communicated in the signaling plane, an asymmetrical behavior cannot always be detected and so cannot be blocked. With an asymmetrical behavior, protocol extractions do not necessarily function as they would with a symmetrical configuration, resulting in an unexpected operation.

The VLAN tag processing is configured as follows on a mesh or spoke-SDP in a VPLS service:

- **zero VLAN tags processed**

  VPLS Service Pseudowire VLAN Tag Processing. This requires the configuration of **vc-type ether** under the mesh-SDP or spoke-SDP, or in the related PW template.

- **one VLAN tag processed**

  This requires one of the following configurations:

  – **vc-type vlan** under the mesh-SDP or spoke-SDP, or in the related PW template

  – **vc-type ether** and **force-vlan-vc-forwarding** under the mesh-SDP or spoke-SDP, or in the related PW template

- **two VLAN tags processed**

  This requires the configuration of **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** under the mesh-SDP or spoke-SDP, or in the related PW template.

The PW template configuration provides support for BGP VPLS services and LDP VPLS services using BGP Auto-Discovery.

The following restrictions apply to VLAN tag processing:

- The configuration of **vc-type vlan** and **force-vlan-vc-forwarding** is mutually exclusive.
- BGP VPLS services operate in a mode equivalent to **vc-type ether**; consequently, the configuration of **vc-type vlan** in a PW template for a BGP VPLS service is ignored.
- **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** can be configured with the mesh-SDP or spoke-SDP signaled as either **vc-type ether** or **vc-type vlan**.
- The following are not supported with **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** configured under the mesh-SDP or spoke-SDP, or in the related PW template:
  – Routed, E-Tree, or PBB VPLS services (including B-VPLS and I-VPLS)
  – L2PT termination on QinQ mesh-SDP or spoke-SDPs
  – IGMP/MLD/PIM snooping within the VPLS service
  – force-vlan-vc-forwarding under the same spoke-SDP or PW template
  – Eth-CFM LM tests

Table 9: VPLS mesh and spoke-SDP VLAN tag processing: ingress and Table 10: VPLS mesh and spoke-SDP VLAN tag processing: egress describe the VLAN tag processing with respect to the zero, one, and two VLAN tag configuration described for the VLAN identifiers, Ethertype, ingress QoS classification (dot1p/DE), and QoS propagation to the egress (which can be used for egress classification or to set the QoS information, or both, in the innermost egress VLAN tag).

Table 9: VPLS mesh and spoke-SDP VLAN tag processing: ingress

Ingress (received on mesh or spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
VLAN identifiers	—	Ignored	Both inner and outer ignored
Ethertype (to determine the presence of a VLAN tag)	—	0x8100 or value configured under **sdp vlan-vc-etype**	Both inner and outer VLAN tags: 0x8100, or outer VLAN tag value configured under **sdp** **vlan-vc-etype** (inner VLAN tag value must be 0x8100)

Ingress (received on mesh or spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
Ingress QoS (dot1p/DE) classification	—	Ignored	Both inner and outer ignored
QoS (dot1p/DE) propagation to egress	Dot1p/DE=0	Dot1p/DE taken from received VLAN tag	Dot1p/DE taken as follows: • If the egress encapsulation is a Dot1q SAP, Dot1p/DE bits are taken from the outer received VLAN tag. • If the egress encapsulation is QinQ SAP, the s-tag bits are taken from the outer received VLAN tag and the c-tag bits from the inner received VLAN tag.

Table 10: VPLS mesh and spoke-SDP VLAN tag processing: egress

Egress (sent on mesh or spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
VLAN identifiers (set in VLAN tags)	—	For one VLAN tag, one of the following applies: • the **vlan-vc-tag** value configured in PW template or value under the mesh/spoke-SDP • value from the inner tag received on a QinQ SAP or QinQ mesh/spoke-SDP • value from the VLAN tag received on a dot1q SAP or mesh/spoke-SDP (with **vc-type vlan** or **force-vlan-vc-forwarding**) • value from the outer tag received on a qtag.* SAP • 0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP	The inner and outer VLAN tags are derived from one of the following: • vlan-vc-tag value configured in PW template or under the mesh/spoke-SDP: – If c-tag-c-tag is configured, both inner and outer tags are taken from the vlan-vc-tag value. – If s-tag-c-tag is configured, only the s-tag value is taken from vlan-vc-tag. • value from the inner tag received on a QinQ SAP or QinQ mesh/spoke-SDP for the c-tag-c-tag option and value from outer/inner tag received on a QinQ SAP or QinQ mesh/spoke-SDP for the s-tag-c-tag configuration option

Egress (sent on mesh or spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
			• value from the VLAN tag received on a dot1q SAP or mesh/spoke-SDP (with vc-type vlan or force-vlan-vc-forwarding) for the c-tag-c-tag option and value from the VLAN tag for the outer tag and zero for the inner tag
			• value from the outer tag received on a qtag.* SAP for the c-tag-c-tag option and value from the VLAN tag for the outer tag and zero for the inner tag • value 0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP Ethertype (set in VLAN tags)
Ethertype (set in VLAN tags)	—	0x8100 or value configured under **sdp vlan-vc-etype**	Both inner and outer VLAN tags: 0x8100, or outer VLAN tag value configured under **sdp vlan-vc-etype** (inner VLAN tag value is 0x8100)
Egress QoS (dot1p/DE) (set in VLAN tags)	—	Taken from the innermost ingress service delimiting tag, one of the following applies: • the inner tag received on a QinQ SAP or QinQ mesh/spoke-SDP • value from the VLAN tag received on a dot1q SAP or mesh/spoke-SDP (with **vc-type** vlan or **force-vlan-vc-forwarding**) • value from the outer tag received on a qtag.* SAP	Inner and outer dot1p/DE: If **c-tag-c-tag** is configured, the inner and outer dot1p/DE bits are both taken from the innermost ingress service delimiting tag. It can be one of the following: • inner tag received on a QinQ SAP • value from the VLAN tag received on a dot1q SAP or spoke-SDP (with **vc-type vlan** or **force-vlan-vcforwarding**) • value from the outer tag received on a qtag.* SAP

Egress (sent on mesh or spoke-SDP)	Zero VLAN tags	One VLAN tag	Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]
Egress QoS (dot1p/DE) (set in VLAN tags)	—	0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP Note that neither the inner nor outer dot1p/DE values can be explicitly set.	0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP If **s-tag-c-tag** is configured, the inner and outer dot1p/DE bits are taken from the inner and outer ingress service delimiting tag (respectively). They can be: • inner and outer tags received on a QinQ SAP or QinQ mesh/spoke-SDP • value from the VLAN tag received on a dot1q SAP or mesh/spoke-SDP (with **vc-type vlan** or **force-vlan-vc-forwarding**) for the outer tag and zero for the inner tag • value from the outer tag received on a qtag.* SAP for the outer tag and zero for the inner tag • value 0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP

Any non-service delimiting VLAN tags are forwarded transparently through the VPLS service. SAP egress classification is possible on the outermost customer VLAN tag received on a mesh or spoke-SDP using the **ethernet-ctag** parameter in the associated SAP egress QoS policy.

3.2.4 VPLS MAC learning and packet forwarding

The 7950 XRS, 7750 SR, and 7450 ESS perform the packet replication required for broadcast and multicast traffic across the bridged domain. MAC address learning is performed by the router to reduce the amount of unknown destination MAC address flooding.

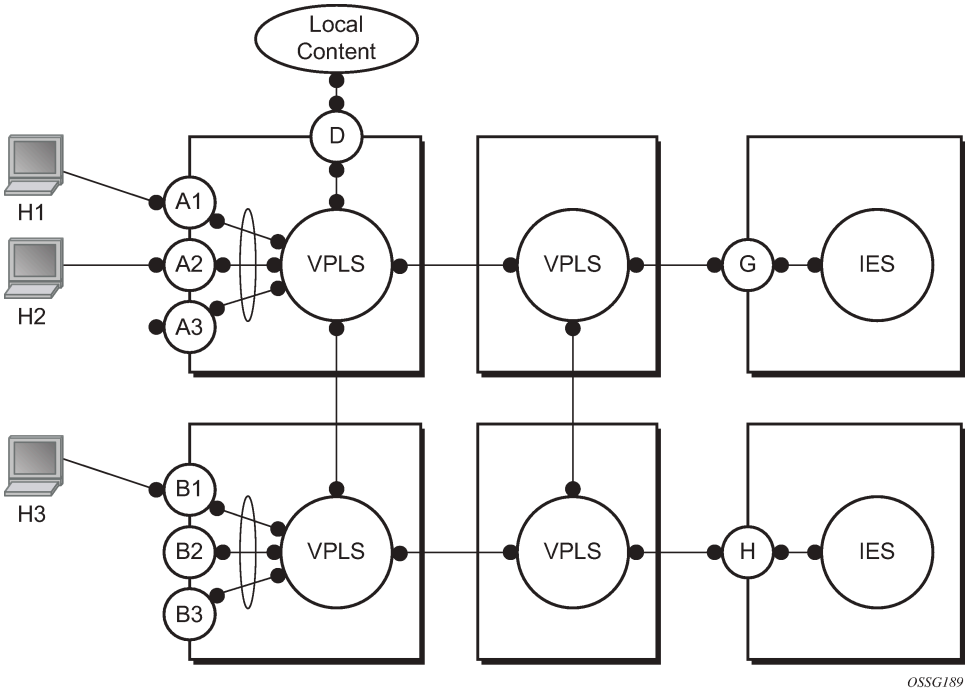
The 7450 ESS, 7750 SR, and 7950 XRS routers learn the source MAC addresses of the traffic arriving on their access and network ports.

Each router maintains an FDB for each VPLS service instance and learned MAC addresses are populated in the FDB table of the service. All traffic is switched based on MAC addresses and forwarded between all objects in the VPLS service. Unknown destination packets (for example, the destination MAC address has not been learned) are forwarded on all objects to all participating nodes for that service until the target station responds and the MAC address is learned by the routers associated with that service.

3.2.4.1 MAC learning protection

In a Layer 2 environment, subscribers or customers connected to SAPs A or B can create a denial of service attack by sending packets sourcing the gateway MAC address. This moves the learned gateway MAC from the uplink SDP/SAP to the subscriber's or customer's SAP causing all communication to the gateway to be disrupted. If local content is attached to the same VPLS (D), a similar attack can be launched against it. Communication between subscribers or customers is also disallowed but split horizon is not sufficient in the topology shown in Figure 57: MAC learning protection.

Figure 57: MAC learning protection



The 7450 ESS, 7750 SR, and 7950 XRS routers enable MAC learning protection capability for SAPs and SDPs. With this mechanism, forwarding and learning rules apply to the non-protected SAPs. Assume hosts H1, H2, and H3 (Figure 57: MAC learning protection) are non-protected while IES interfaces G and H are protected. When a frame arrives at a protected SAP/SDP, the MAC is learned as usual. When a frame arrives from a non-protected SAP or SDP, the frame must be dropped if the source MAC address is protected and the MAC address is not relearned. The system allows only packets with a protected MAC destination address.

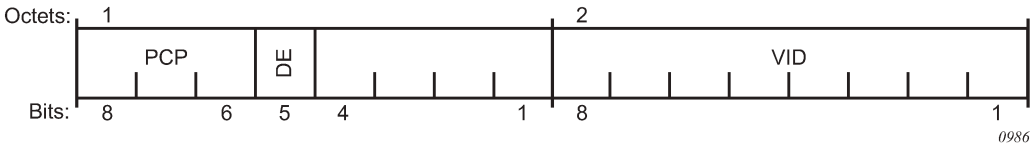
The system can be configured statically. The addresses of all protected MACs are configured. Only the IP address can be included and use a dynamic mechanism to resolve the MAC address (**cpe-ping**). All protected MACs in all VPLS instances in the network must be configured.

To eliminate the ability of a subscriber or customer to cause a DoS attack, the node restricts the learning of protected MAC addresses based on a statically defined list. Also, the destination MAC address is checked against the protected MAC list to verify that a packet entering a restricted SAP has a protected MAC as a destination.

3.2.4.2 DEI in IEEE 802.1ad

The IEEE 802.1ad-2005 standard allows drop eligibility to be conveyed separately from priority in Service VLAN TAGs (S-TAGs) so that all of the previously introduced traffic types can be marked as drop eligible. The S-TAG has a new format where the priority and discard eligibility parameters are conveyed in the 3-bit Priority Code Point (PCP) field and, respectively, in the DE bit (Figure 58: DE bit in the 802.1ad S-TAG).

Figure 58: DE bit in the 802.1ad S-TAG



The DE bit allows the S-TAG to convey eight forwarding classes/distinct emission priorities, each with a drop eligible indication.

When the DE bit is set to 0 (DE=FALSE), the related packet is not discarded eligible. This is the case for the packets that are within the CIR limits and must be prioritized in case of congestion. If the DEI is not used or backwards compliance is required, the DE bit should be set to zero on transmission and ignored on reception.

When the DE bit is set to 1 (DE=TRUE), the related packet is discarded eligible. This is the case for the packets that are sent above the CIR limit (but below the PIR). In case of congestion, these packets are the first ones to be dropped.

3.2.5 VPLS using G.8031 protected Ethernet tunnels

The use of MPLS tunnels provides a way to scale the core while offering fast failover times using MPLS FRR. In environments where Ethernet services are deployed using native Ethernet backbones, Ethernet tunnels are provided to achieve the same fast failover times as in the MPLS FRR case. There are still service provider environments where Ethernet services are deployed using native Ethernet backbones.

The Nokia VPLS implementation offers the capability to use core Ethernet tunnels compliant with ITU-T G.8031 specification to achieve 50 ms resiliency for backbone failures. This is required to comply with the stringent SLAs provided by service providers in the current competitive environment. The implementation also allows a LAG-emulating Ethernet tunnel providing a complimentary native Ethernet E-LAN capability. The LAG-emulating Ethernet tunnels and G.8031 protected Ethernet tunnels operate independently. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*, "*LAG Emulation using Ethernet Tunnels*".

When using Ethernet tunnels, the Ethernet tunnel logical interface is created first. The Ethernet tunnel has member ports that are the physical ports supporting the links. The Ethernet tunnel controls SAPs that carry G.8031 and 802.1ag control traffic and user data traffic. Ethernet Service SAPs are configured on the Ethernet tunnel. Optionally, when tunnels follow the same paths, end-to-end services are configured with same-fate Ethernet tunnel SAPs, which carry only user data traffic, and share the fate of the Ethernet tunnel port (if properly configured).

When configuring VPLS and B-VPLS using Ethernet tunnels, the services are very similar.

For examples, see the *IEEE 802.1ah PBB Guide*.

3.2.6 Pseudowire control word

The **control-word** command enables the use of the control word individually on each mesh SDP or spoke-SDP. By default, the control word is disabled. When the control word is enabled, all VPLS packets, including the BPDU frames, are encapsulated with the control word. The Targeted LDP (T-LDP) control plane behavior is the same as the control word for VLL services. The configuration for the two directions of the Ethernet pseudowire should match.

3.2.7 Table management

The following sections describe VPLS features related to management of the FDB.

3.2.7.1 Selective MAC address learning

Source MAC addresses are learned in a VPLS service by default with an entry allocated in the FDB for each address on all line cards. Therefore, all MAC addresses are considered to be global. This operation can be modified so that the line card allocation of some MAC addresses is selective, based on where the service has a configured object.

An example of the advantage of selective MAC address learning is for services to benefit from the higher MAC address scale of some line cards (particularly for network interfaces used by mesh or spoke-SDPs, EVPN-VXLAN tunnels, and EVPN-MPLS destinations) while using lower MAC address scale cards for the SAPs.

Selective MAC addresses are those learned locally and dynamically in the data path (displayed in the **show** output with type "L") or by EVPN (displayed in the **show** output with type "Evpn", excluding those with the sticky bit set, which are displayed with type "EvpnS"). An exception is when a MAC address configured as a conditional static MAC address is learned dynamically on an object other than its monitored object; this can be displayed with type "L" or "Evpn" but is learned as global because of the conditional static MAC configuration.

Selective MAC addresses have FDB entries allocated on line cards where the service has a configured object. When a MAC address is learned, it is allocated an FDB entry on all line cards on which the service has a SAP configured (for LAG or Ethernet tunnel SAPs, the MAC address is allocated an FDB entry on all line cards on which that LAG or Ethernet tunnel has configured ports) and on all line cards that have a network interface port if the service is configured with VXLAN, EVPN-MPLS, or a mesh or spoke-SDP.

When using selective learning in an I-VPLS service, the learned C-MACs are allocated FDB entries on all the line cards where the I-VPLS service has a configured object and on the line cards on which the associated B-VPLS has a configured object. When using selective learning in a VPLS service with **allow-ip-intf-bind** configured (for it to become an R-VPLS), FDB entries are allocated on all line cards on which there is an IES or VPRN interface.

If a new configured object is added to a service and there are sufficient MAC FDB resources available on the new line cards, the selective MAC addresses present in the service are allocated on the new line cards. Otherwise, if any of the selective MAC addresses currently learned in the service cannot be allocated an FDB entry on the new line cards, those MAC addresses are deleted from all line cards. Such a deletion increments the FailedMacCmplxMapUpdts statistic displayed in the **tools dump service vpls-fdb-stats** output.

When the set of configured objects changes for a service using selective learning, the system must reallocate its FDB entries accordingly, which can cause FDB entry "allocate" or "free" operations to become

pending temporarily. The pending operations can be displayed using the **tools dump service id fdb** command.

When a global MAC address is to be learned, there must be a free FDB entry in the service and system FDBs and on all line cards in the system for it to be accepted. When a selective MAC address is to be learned, there must be a free FDB entry in the service and system FDBs and on all line cards where the service has a configured object for it to be accepted.

To demonstrate the selective MAC address learning logic, consider the following:

- a system has three line cards: 1, 2, and 3
- two VPLS services are configured on the system:
  - VPLS 1 having learned MAC addresses M1, M2, and M3 and has configured SAPs 1/1/1 and 2/1/1
  - VPLS 2 having learned MAC addresses M4, M5, and M6 and has configured SAPs 2/1/2 and 3/1/1

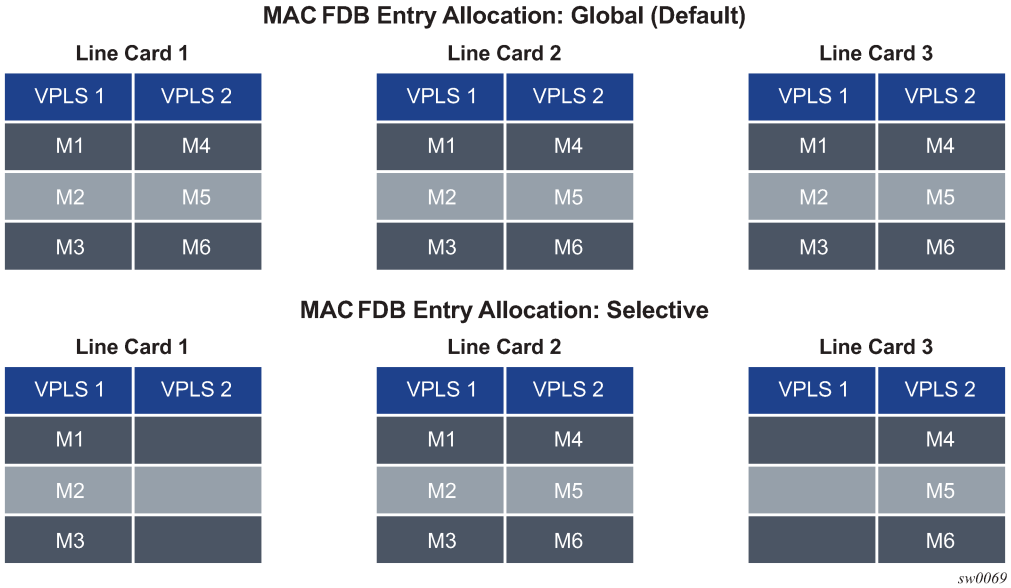
This is shown in Table 11: MAC address learning logic example .

Table 11: MAC address learning logic example

	Learned MAC addresses	Configured SAPs
VPLS1	M1, M2, M3	SAP 1/1/1 SAP 2/1/1
VPLS2	M4, M5, M6	SAP 2/1/2 SAP 3/1/1

Figure 59: MAC FDB entry allocation: global versus selective shows the FDB entry allocation when the MAC addresses are global and when they are selective. Notice that in the selective case, all MAC addresses are allocated FDB entries on line card 2, but line card 1 and 3 only have FDB entries allocated for services VPLS 1 and VPLS 2, respectively.

Figure 59: MAC FDB entry allocation: global versus selective



Selective MAC address learning can be enabled as follows within any VPLS service, except for B-VPLS and R-VPLS services:

```
configure
        service
            vpls <service-id> create
                [no] selective-learned-fdb
```

Enabling selective MAC address learning has no effect on single line card systems.

When selective learning is enabled or disabled in a VPLS service, the system may need to reallocate FDB entries; this can cause temporary pending FDB entry allocate or free operations. The pending operations can be displayed using the **tools dump service id fdb** command.

3.2.7.1.1 Example operational information

The **show** and **tools dump** command output can display the global and selective MAC addresses along with the MAC address limits and the number of allocated and free MAC-address FDB entries. The **show** output displays the system and card FDB usage, while the **tools** output displays the FDB per service with respect to MAC addresses and cards.

The configuration for the following output is similar to the simple example above:

- the system has three line cards: 1, 2, and 5
- the system has two VPLS services:
  - VPLS 1 is an EVPN-MPLS service with a SAP on 5/1/1:1 and uses a network interface on 5/1/5.
  - VPLS 2 has two SAPs on 2/1/1:2 and 2/1/2:2.

The first output shows the default where all MAC addresses are global. The second enables selective learning in the two VPLS services.

3.2.7.1.1.1 Global MAC address learning only (default)

By default, VPLS 1 and 2 are not configured for selective learning, so all MAC addresses are global:

```
*A:PE1# show service id [1,2] fdb | match expression ", Service|Sel Learned FDB"
Forwarding Database, Service 1
Sel Learned FDB   : Disabled
Forwarding Database, Service 2
Sel Learned FDB   : Disabled
*A:PE1#
```

Traffic is sent into the services, resulting in the following MAC addresses being learned:

```
*A:PE1# show service fdb-mac
===============================================================================
Service Forwarding Database
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:00:00:00:01:01 sap:5/1/1:1              L/0      01/31/17 08:44:37
1         00:00:00:00:01:02 sap:5/1/1:1              L/0      01/31/17 08:44:37
1         00:00:00:00:01:03 eMpls:                   EvpnS    01/31/17 08:41:38
                                                     P
                            10.251.72.58:262142
1         00:00:00:00:01:04 eMpls:                   EvpnS    01/31/17 08:41:38
                                                     P
                            10.251.72.58:262142
2         00:00:00:00:02:01 sap:2/1/2:2              L/0      01/31/17 08:44:37
2         00:00:00:00:02:02 sap:2/1/2:2              L/0      01/31/17 08:44:37
2         00:00:00:02:02:03 sap:2/1/1:2              L/0      01/31/17 08:44:37
2         00:00:00:02:02:04 sap:2/1/1:2              L/0      01/31/17 08:44:37
-------------------------------------------------------------------------------
No. of Entries: 8
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static Lf=Leaf
===============================================================================
*A:PE1#
```

A total of eight MAC addresses are learned. There are two MAC addresses learned locally on SAP 5/1/1:1 in service VPLS 1 (type "L"), and another two MAC addresses learned using EVPN with the sticky bit set, also in service VPLS 1 (type "EvpnS"). A further two sets of two MAC addresses are learned on SAP 2/1/1:2 and 2/1/2:2 in service VPLS 2 (type "L").

The system and line card FDB usage is shown as follows:

```
*A:PE1# show service system fdb-usage
===============================================================================
FDB Usage
===============================================================================
System
-------------------------------------------------------------------------------
Limit:     511999
Allocated: 8
Free:      511991
Global:    8
-------------------------------------------------------------------------------
Line Cards
-------------------------------------------------------------------------------
Card        Selective         Allocated         Limit             Free
```

```
-------------------------------------------------------------------------------
1           0                8                511999           511991
2           0                8                511999           511991
5           0                8                511999           511991
-------------------------------------------------------------------------------
===============================================================================
*A:PE1#
```

The system MAC address limit is 511999, of which eight are allocated, and the rest are free. All eight MAC addresses are global and are allocated on cards 1, 2, and 5. There are no selective MAC addresses. This output can be reduced to specific line cards by specifying the card's slot ID as a parameter to the command.

To see the MAC address information per service, **tools dump** commands can be used, as follows for VPLS 1. The following output displays the card status:

```
*A:PE1# tools dump service id 1 fdb card-status
===============================================================================
VPLS FDB Card Status at 01/31/2017 08:44:38
===============================================================================
Card                Allocated           PendAlloc           PendFree
-------------------------------------------------------------------------------
1                   4                   0                   0
2                   4                   0                   0
5                   4                   0                   0
===============================================================================
*A:PE1#
```

All of the line cards have four FDB entries allocated in VPLS 1. The "PendAlloc" and "PendFree" columns show the number of pending MAC address allocate and free operations, which are all zero.

The following output displays the MAC address status for VPLS 1:

```
*A:PE1# tools dump service id 1 fdb mac-status
===============================================================================
VPLS FDB MAC status at 01/31/2017 08:44:38
===============================================================================
MAC Address         Type                Status : Card list
-------------------------------------------------------------------------------
00:00:00:00:01:01   Global              Allocated : All
00:00:00:00:01:02   Global              Allocated : All
00:00:00:00:01:03   Global              Allocated : All
00:00:00:00:01:04   Global              Allocated : All
===============================================================================
*A:PE1#
```

The type and card list for each MAC address in VPLS 1 is displayed. VPLS 1 has learned four MAC addresses: the two local MAC addresses on SAP 5/1/1:1 and the two EvpnS MAC addresses. Each MAC address has an FDB entry allocated on all line cards. This output can be further reduced by optionally including a specified MAC address, a specific card, and the operational pending state.

3.2.7.1.1.2 Selective and global MAC address learning

Selective MAC address learning is now enabled in VPLS 1 and VPLS 2, as follows:

```
*A:PE1# show service id [1,2] fdb | match expression ", Service|Sel Learned FDB"
Forwarding Database, Service 1
Sel Learned FDB   : Enabled
```

```
Forwarding Database, Service 2
Sel Learned FDB   : Enabled
*A:PE1#
```

The MAC addresses learned are the same, with the same traffic being sent; however, there are now selective MAC addresses that are allocated FDB entries on different line cards.

The system and line card FDB usage is as follows:

```
*A:PE1# show service system fdb-usage
===============================================================================
FDB Usage
===============================================================================
System
-------------------------------------------------------------------------------
Limit:     511999
Allocated: 8
Free:      511991
Global:    2
-------------------------------------------------------------------------------
Line Cards
-------------------------------------------------------------------------------
Card        Selective         Allocated         Limit             Free
-------------------------------------------------------------------------------
1           0                 2                 511999            511997
2           4                 6                 511999            511993
5           2                 4                 511999            511995
-------------------------------------------------------------------------------
===============================================================================
*A:PE1#
```

The system MAC address limit and allocated numbers have not changed but now there are only two global MAC addresses; these are the two EvpnS MAC addresses.

There are two FDB entries allocated on card 1, which are the global MAC addresses; there are no services or network interfaces configured on card 1, so the FDB entries allocated are for the global MAC addresses.

Card 2 has six FDB entries allocated in total: two for the global MAC addresses plus four for the selective MAC addresses in VPLS 2 (these are the two sets of two local MAC addresses in VPLS 2 on SAP 2/1/1:2 and 2/1/2:2).

Card 5 has four FDB entries allocated in total: two for the global MAC addresses plus two for the selective MAC addresses in VPLS 1 (these are the two local MAC addresses in VPLS 1 on SAP 5/1/1:1).

This output can be reduced to specific line cards by specifying the card's slot ID as a parameter to the command.

To see the MAC address information per service, **tools dump** commands can be used for VPLS 1.

The following output displays the card status:

```
*A:PE1# tools dump service id 1 fdb card-status
===============================================================================
VPLS FDB Card Status at 01/31/2017 08:44:39
===============================================================================
Card                Allocated           PendAlloc           PendFree
-------------------------------------------------------------------------------
1                   2                   0                   0
2                   2                   0                   0
5                   4                   0                   0
===============================================================================
*A:PE1#
```

There are two FDB entries allocated on line card 1, two on line card 2, and four on line card 5. The "PendAlloc" and "PendFree" columns are all zeros.

The following output displays the MAC address status for VPLS 1:

```
*A:PE1# tools dump service id 1 fdb mac-status
===============================================================================
VPLS FDB MAC status at 01/31/2017 08:44:39
===============================================================================
MAC Address         Type                Status : Card list
-------------------------------------------------------------------------------
00:00:00:00:01:01   Select              Allocated : 5
00:00:00:00:01:02   Select              Allocated : 5
00:00:00:00:01:03   Global              Allocated : All
00:00:00:00:01:04   Global              Allocated : All
===============================================================================
*A:PE1#
```

The type and card list for each MAC address in VPLS 1 is displayed. VPLS 1 has learned four MAC addresses: the two local MAC addresses on SAP 5/1/1:1 and the two EvpnS MAC addresses. The local MAC addresses are selective and have FDB entries allocated only on card 5. The global MAC addresses are allocated on all line cards. This output can be further reduced by optionally including a specified MAC address, a specific card, and the operational pending state.

3.2.7.2 System FDB size

The system FDB table size is configurable as follows:

```
configure
        service
            system
                fdb-table-size table-size
```

where table-size can have values in the range from 255999 to 2047999 (2000k).

The default, minimum, and maximum values for the table size are dependent on the chassis type. To support more than 500k MAC addresses, the CPMs provisioned in the system must have at least 16 GB memory. The maximum system FDB table size also limits the maximum FDB table size of any card within the system.

The actual achievable maximum number of MAC addresses depends on the MAC address scale supported by the active cards and whether selective learning is enabled.

If an attempt is made to configure the system FDB table size such that:

- the new size is greater than or equal to the current number of allocated FDB entries, the command succeeds and the new system FDB table size is used
- the new size is less than the number of allocated FDB entries, the command fails with an error message. In this case, the user is expected to reduce the current FDB usage (for example, by deleting statically configured MAC addresses, shutting down EVPN, clearing learned MACs, and so on) to lower the number of allocated MAC addresses in the FDB so that it does not exceed the system FDB table size being configured.

The logic when attempting a rollback is similar; however, when rolling back to a configuration where the system FDB table size is smaller than the current system FDB table size, the system flushes all learned MAC addresses (by performing a **shutdown** then **no shutdown** in all VPLS services) to allow the rollback to continue.

The system FDB table size can be larger than some of the line card FDB sizes, resulting in the possibility that the current number of allocated global MAC addresses is larger than the maximum FDB size supported on some line cards. When a new line card is provisioned, the system checks whether the line card's FDB can accommodate all of the currently allocated global MAC addresses. If it can, then the provisioning succeeds; if it cannot, then the provisioning fails and an error is reported. If the provisioning fails, the number of global MACs allocated must be reduced in the system to a number that the new line card can accommodate, then the **card-type** must be reprovisioned.

3.2.7.3 Per-VPLS service FDB size

The following MAC table management features are available for each instance of a SAP or spoke-SDP within a particular VPLS service instance.

MAC FDB size limits allow users to specify the maximum number of MAC FDB entries that are learned locally for a SAP or remotely for a spoke-SDP. If the configured limit is reached, no new addresses is learned from the SAP or spoke-SDP until at least one FDB entry is aged out or cleared.

- When the limit is reached on a SAP or spoke-SDP, packets with unknown source MAC addresses are still forwarded (this default behavior can be changed by configuration). By default, if the destination MAC address is known, it is forwarded based on the FDB, and if the destination MAC address is unknown, it is flooded. Alternatively, if discard unknown is enabled at the VPLS service level, any packets from unknown source MAC addresses are discarded at the SAP.
- The log event SAP MAC Limit Reached is generated when the limit is reached. When the condition is cleared, the log event SAP MAC Limit Reached Condition Cleared is generated.
- Disable learning allows users to disable the dynamic learning function on a SAP or a spoke-SDP of a VPLS service instance.
- Disable aging allows users to turn off aging for learned MAC addresses on a SAP or a spoke-SDP of a VPLS service instance.

3.2.7.4 System FDB size alarms

High and low watermark alarms give warning when the system MAC FDB usage is high. An alarm is generated when the number of FDB entries allocated in the system FDB reaches 95% of the total system FDB table size and is cleared when it reduces to 90% of the system FDB table size. These percentages are not configurable.

3.2.7.5 Line card FDB size alarms

High and low watermark alarms give warning when a line card's MAC FDB usage is high. An alarm is generated when the number of FDB entries allocated in a line card FDB reaches 95% of its maximum FDB table size and is cleared when it reduces to 90% of its maximum FDB table size. These percentages are not configurable.

3.2.7.6 Per VPLS FDB size alarms

The size of the VPLS FDB can be configured with a low watermark and a high watermark, expressed as a percentage of the total FDB size limit. If the actual FDB size grows above the configured high watermark

percentage, an alarm is generated. If the FDB size falls below the configured low watermark percentage, the alarm is cleared by the system.

3.2.7.7 Local and remote aging timers

Like a Layer 2 switch, learned MACs within a VPLS instance can be aged out if no packets are sourced from the MAC address for a specified period of time (the aging time). In each VPLS service instance, there are independent aging timers for locally learned MAC and remotely learned MAC entries in the FDB. A local MAC address is a MAC address associated with a SAP because it ingressed on a SAP. A remote MAC address is a MAC address received by an SDP from another router for the VPLS instance. The local-age timer for the VPLS instance specifies the aging time for locally learned MAC addresses, and the remote-age timer specifies the aging time for remotely learned MAC addresses.

In general, the remote-age timer is set to a longer period than the local-age timer to reduce the amount of flooding required for unknown destination MAC addresses. The aging mechanism is considered a low priority process. In most situations, the aging out of MAC addresses happens within tens of seconds beyond the age time. However, it, can take up to two times their respective age timer to be aged out.

3.2.7.8 Disable MAC aging

The MAC aging timers can be disabled, which prevents any learned MAC entries from being aged out of the FDB. When aging is disabled, it is still possible to manually delete or flush learned MAC entries. Aging can be disabled for learned MAC addresses on a SAP or a spoke-SDP of a VPLS service instance.

3.2.7.9 Disable MAC learning

When MAC learning is disabled for a service, new source MAC addresses are not entered in the VPLS FDB, whether the MAC address is local or remote. MAC learning can be disabled for individual SAPs or spoke-SDPs.

3.2.7.10 Unknown MAC discard

Unknown MAC discard is a feature that discards all packets that ingress the service where the destination MAC address is not in the FDB. The normal behavior is to flood these packets to all endpoints in the service.

Unknown MAC discard can be used with the disable MAC learning and disable MAC aging options to create a fixed set of MAC addresses allowed to ingress and traverse the service.

3.2.7.11 VPLS and rate limiting

Traffic that is normally flooded throughout the VPLS can be rate limited on SAP ingress through the use of service ingress QoS policies. In a service ingress QoS policy, individual queues can be defined per forwarding class to provide shaping of broadcast traffic, MAC multicast traffic, and unknown destination MAC traffic.

3.2.7.12 MAC move

The MAC move feature is useful to protect against undetected loops in a VPLS topology as well as the presence of duplicate MACs in a VPLS service.

If two clients in the VPLS have the same MAC address, the VPLS experiences a high relearn rate for the MAC. When MAC move is enabled, the 7450 ESS, 7750 SR, or 7950 XRS shuts down the SAP or spoke-SDP and creates an alarm event when the threshold is exceeded.

MAC move allows sequential order port blocking. By configuration, some VPLS ports can be configured as "non-blockable", which allows a simple level of control of which ports are being blocked during loop occurrence. There are two sophisticated control mechanisms that allow blocking of ports in a sequential order:

1. Configuration capabilities to group VPLS ports and to define the order in which they should be blocked
2. Criteria defining when individual groups should be blocked

For the first control mechanism, configuration CLI is extended by definition of "primary" and "secondary" ports. Per default, all VPLS ports are considered "tertiary" ports unless they are explicitly declared primary or secondary. The order of blocking always follows a strict order starting from tertiary to secondary, and then primary.

The definition of criteria for the second control mechanism is the number of periods during which the specified relearn rate has been exceeded. The mechanism is based on the cumulative factor for every group of ports. Tertiary VPLS ports are blocked if the relearn rate exceeds the configured threshold during one period, while secondary ports are blocked only when relearn rates are exceeded during two consecutive periods, and primary ports when exceeded during three consecutive periods. The retry timeout period must be larger than the period before blocking the highest priority port so that the retry timeout sufficiently spans across the period required to block all ports in sequence. The period before blocking the highest priority port is the cumulative factor of the highest configured port multiplied by 5 seconds (the retry timeout can be configured through the CLI).

3.2.7.13 Auto-learn MAC protect

This section provides information about auto-learn-mac-protect and restrict-protected-src discard-frame features.

VPLS solutions usually involve learning MAC addresses in order for traffic to be forwarded to the correct SAP/SDP. If a MAC address is learned on the wrong SAP/SDP, traffic would be redirected away from its intended destination. This could occur through a misconfiguration, a problem in the network, or by a malicious source creating a DoS attack, and is applicable to any type of VPLS network; for example, mobile backhaul or residential service delivery networks. The auto-learn-mac-protect feature can be used to safeguard against the possibility of MAC addresses being learned on the wrong SAP/SDP.

This feature provides the ability to automatically protect source MAC addresses that have been learned on a SAP or a spoke/mesh SDP and prevent frames with the same protected source MAC address from entering into a different SAP/spoke or mesh SDP.

This is a complementary solution to features such as mac-move and mac-pinning, but has the advantage that MAC moves are not seen and it has a low operational complexity. If a MAC is initially learned on the wrong SAP/SDP, the operator can clear the MAC from the MAC FDB in order for it to be relearned on the correct SAP/SDP.

Two separate commands are used, which provide the configuration flexibility of separating the identification (learning) function from the application of the restriction (discard).

The **auto-learn-mac-protect** and **restrict-protected-src** commands allow the following functions:

- the ability to enable the automatic protection of a learned MAC using the **auto-learn-mac-protect** command under a SAP/spoke or mesh SDP/SHG context
- the ability to discard frames associated with automatically protected MACs instead of shutting down the entire SAP/SDP as with the restrict-protected-src feature. This is enabled using a **restrict-protected-src discard-frame** command in the SAP/spoke or mesh SDP/SHG context. An optimized alarm mechanism is used to generate alarms related to these discards. The frequency of alarm generation is fixed to be, at most, one alarm per MAC address per forwarding complex per 10 minutes in a VPLS service.

If the auto-learn-mac-protect or restrict-protected-src discard-frame feature is configured under an SHG, the operation applies only to SAPs in the SHG, not to spoke-SDPs in the SHG. If required, these parameters can also be enabled explicitly under specific SAPs/spoke-SDPs within the SHG.

Applying or removing auto-learn-mac-protect or restrict-protected-src discard-frame to/from a SAP, spoke or mesh SDP, or SHG, clears the MACs on the related objects (for the SHG, this results in clearing the MACs only on the SAPs within the SHG).

The use of restrict-protected-src discard-frame and both the **restrict-protected-src [alarm-only]** command and with the configuration of manually protected MAC addresses, using the **mac-protect** command, within a specified VPLS are mutually exclusive.

The following rules govern the changes to the state of protected MACs:

- Automatically learned protected MACs are subject to normal removal, aging (unless disabled), and flushing, at which time the associated entries are removed from the FDB.
- Automatically learned protected MACs can only move from their learned SAP/spoke or mesh SDP if they enter a SAP/spoke or mesh SDP without restrict-protected-src enabled.

If a MAC address does legitimately move between SAPs/spoke or mesh SDPs after it has been automatically protected on a specified SAP/spoke or mesh SDP (thereby causing discards when received on the new SAP/spoke or mesh SDP), the operator must manually clear the MAC from the FDB for it to be learned in the new/correct location.

MAC addresses that are manually created (using static-mac, static-host with a MAC address specified, or oam mac-populate) are not protected even if they are configured on a SAP/spoke or mesh SDP that has auto-learn-mac-protect enabled on it. Also, the MAC address associated with an R-VPLS IP interface is protected within its VPLS service such that frames received with this MAC address as the source address are discarded (this is not based on the auto-learn MAC protect function). However, VRRP MAC addresses associated with an R-VPLS IP interface are not protected either in this way or using the auto-learn MAC protect function.

MAC addresses that are dynamically created (learned, using static-host with no MAC address specified, or lease-populate) are protected when the MAC address is learned on a SAP/spoke or mesh SDP that has auto-learn-mac-protect enabled on it.

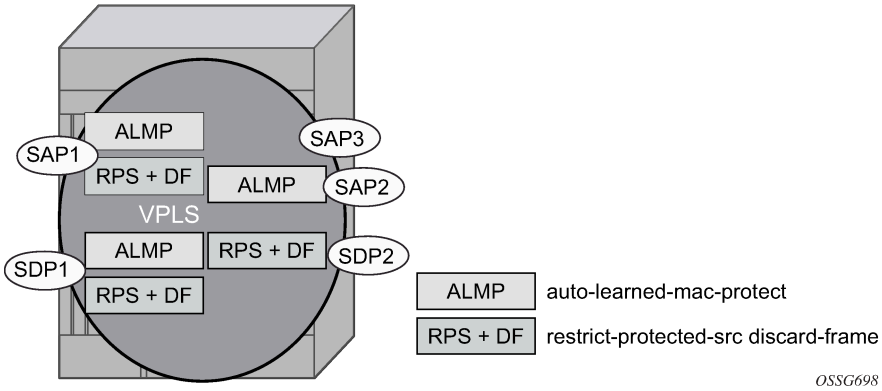
The actions of the following features are performed in the order listed.

1. Restrict-protected-src
2. MAC-pinning
3. MAC-move

3.2.7.13.1 Operation

Figure 60: Auto-learn-mac-protect operation shows a specific configuration using auto-learn-mac-protect and restrict-protected-src discard-frame to describe their operation for the 7750 SR, 7450 ESS, or 7950 XRS.

Figure 60: Auto-learn-mac-protect operation



A VPLS service is configured with SAP1 and SDP1 connecting to access devices and SAP2, SAP3, and SDP2 connecting to the core of the network. The auto-learn-mac-protect feature is enabled on SAP1, SAP3, and SDP1, and restrict-protected-src discard-frame is enabled on SAP1, SDP1, and SDP2. The following series of events describes the details of the functionality:

Assume that the FDB is empty at the start of each sequence.

Sequence 1:

1. A frame with source MAC A enters SAP1, MAC A is learned on SAP1, and MAC-A/SAP1 is protected because of the presence of the auto-learn-mac-protect on SAP1.
2. All subsequent frames with source MAC A entering SAP1 are forwarded into the VPLS.
3. Frames with source MAC A enter either SDP1 or SDP2, these frames are discarded, and an alarm indicating MAC A and SDP1/SDP2 is initiated because of the presence of the restrict-protected-src discard-frame on SDP1/SDP2.
4. The above continues, with MAC-A/SAP1 protected in the FDB until MAC A on SAP1 is removed from the FDB.

Sequence 2:

1. A frame with source MAC A enters SAP1, MAC A is learned on SAP1, and MAC-A/SAP1 is protected because of the presence of the auto-learn-mac-protect on SAP1.
2. A frame with source MAC A enters SAP2. As restrict-protected-src is not enabled on SAP2, MAC A is relearned on SAP2 (but not protected), replacing the MAC-A/SAP1 entry in the FDB.
3. All subsequent frames with source MAC A entering SAP2 are forwarded into the VPLS. This is because restrict-protected-src is not enabled SAP2 and auto-learn-mac-protect is not enabled on SAP2, so the FDB is not changed.
4. A frame with source MAC A enters SAP1, MAC A is relearned on SAP1, and MAC-A/SAP1 is protected because of the presence of the auto-learn-mac-protect on SAP1.

Sequence 3:

1. A frame with source MAC A enters SDP2, MAC A is learned on SDP2, but is not protected as auto-learn-mac-protect is not enabled on SDP2.
2. A frame with source MAC A enters SDP1, and MAC A is relearned on SDP1 because previously it was not protected. Consequently, MAC-A/SDP1 is protected because of the presence of the auto-learn-mac-protect on SDP1.

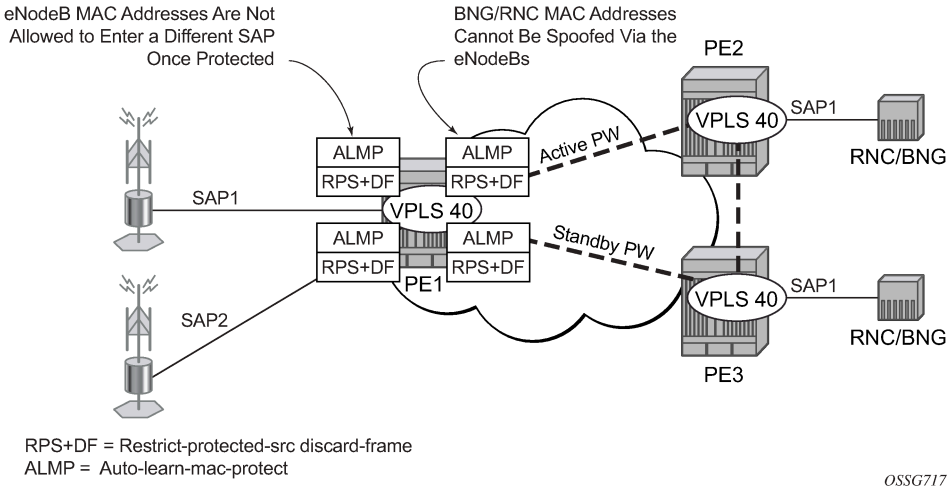
Sequence 4:

1. A frame with source MAC A enters SAP1, MAC A is learned on SAP1, and MAC-A/SAP1 is protected because of the presence of the auto-learn-mac-protect on SAP1.
2. A frame with source MAC A enters SAP3. As restrict-protected-src is not enabled on SAP3, MAC A is relearned on SAP3 and the MAC-A/SAP1 entry is removed from the FDB with MAC-A/SAP3 being added as protected to the FDB (because auto-learn-mac-protect is enabled on SAP3).
3. All subsequent frames with source MAC A entering SAP3 are forwarded into the VPLS.
4. A frame with source MAC A enters SAP1, these frames are discarded, and an alarm indicating MAC A and SAP1 is initiated because of the presence of the restrict-protected-src discard-frame on SAP1.

Example use

Figure 61: Auto-learn-mac-protect example shows a possible configuration using auto-learn-mac-protect and restrict-protected-src discard-frame in a mobile backhaul network, with the focus on PE1 for the 7750 SR or 7950 XRS.

Figure 61: Auto-learn-mac-protect example



To protect the MAC addresses of the BNG/RNCs on PE1, the **auto-learn-mac-protect** command is enabled on the pseudowires connecting PE1 to PE2 and PE3. Enabling the **restrict-protected-src discard-frame** command on the SAPs toward the eNodeBs prevents frames with the source MAC addresses of the BNG/RNCs from entering PE1 from the eNodeBs.

The MAC addresses of the eNodeBs are protected in two ways. In addition to the above commands, enabling the **auto-learn-mac-protect** command on the SAPs toward the eNodeBs prevents the MAC addresses of the eNodeBs being learned on the wrong eNodeB SAP. Enabling the **restrict-protected-src discard-frame** command on the pseudowires connecting PE1 to PE2 and PE3 protects the eNodeB MAC addresses from being learned on the pseudowires. This may happen if their MAC addresses are incorrectly injected into VPLS 40 on PE2/PE3 from another eNodeB aggregation PE.

The above configuration is equally applicable to other Layer 2 VPLS-based aggregation networks; for example, to business or residential service networks.

3.2.8 Split horizon SAP groups and split horizon spoke SDP groups

Within the context of VPLS services, a loop-free topology within a fully meshed VPLS core is achieved by applying a split horizon forwarding concept that packets received from a mesh SDP are never forwarded to other mesh SDPs within the same service. The advantage of this approach is that no protocol is required to detect loops within the VPLS core network.

In applications such as DSL aggregation, it is useful to extend this split horizon concept also to groups of SAPs and spoke-SDPs. This extension is referred to as a split horizon SAP group or residential bridging.

Traffic arriving on an SAP or a spoke-SDP, or both, within a split horizon group is not copied to other SAPs and spoke-SDPs in the same split horizon group (but is copied to SAPs/spoke-SDPs in other split horizon groups if these exist within the same VPLS).

3.2.9 VPLS and spanning tree protocol

Nokia's VPLS service provides a bridged or switched Ethernet Layer 2 network. Equipment connected to SAPs forward Ethernet packets into the VPLS service. The 7450 ESS, 7750 SR, or 7950 XRS participating in the service learns where the customer MAC addresses reside, on ingress SAPs or ingress SDPs.

Unknown destinations, broadcasts, and multicasts are flooded to all other SAPs in the service. If SAPs are connected together, either through misconfiguration or for redundancy purposes, loops can form and flooded packets can keep flowing through the network. The Nokia implementation of the STP is designed to remove these loops from the VPLS topology. This is done by putting one or several SAPs or spoke-SDPs, or both, in the discarding state.

Nokia's implementation of STP incorporates some modifications to make the operational characteristics of VPLS more effective.

The STP instance parameters allow the balancing between resiliency and speed of convergence extremes. Modifying particular parameters can affect the behavior. For information about command usage, descriptions, and CLI syntax, see Configuring a VPLS service with CLI.

3.2.9.1 Spanning tree operating modes

Per VPLS instance, a preferred STP variant can be configured. The STP variants supported are:

rstp	Rapid Spanning Tree Protocol (RSTP) compliant with IEEE 802.1D-2004 - default mode
dot1w	compliant with IEEE 802.1w
comp-dot1w	operation as in RSTP but backwards compatible with IEEE 802.1w (this mode allows interoperability with some MTU types)
mstp	compliant with the Multiple Spanning Tree Protocol specified in IEEE 802.1Q-REV/D5.0-09/2005. This mode of operation is only supported in a Management VPLS (M-VPLS).

While the 7450 ESS, 7750 SR, or 7950 XRS initially use the mode configured for the VPLS, it dynamically falls back (on a per-SAP basis) to STP (IEEE 802.1D-1998) based on the detection of a BPDU of a different format. A trap or log entry is generated for every change in spanning tree variant.

Some older 802.1w compliant RSTP implementations may have problems with some of the features added in the 802.1D-2004 standard. Interworking with these older systems is improved with the comp-dot1w mode. The differences between the RSTP mode and the comp-dot1w mode are:

- The RSTP mode implements the improved convergence over shared media feature; for example, RSTP transitions from discarding to forwarding in 4 seconds when operating over shared media. The comp-dot1w mode does not implement this 802.1D-2004 improvement and transitions conform to 802.1w in 30 seconds (both modes implement fast convergence over point-to-point links).
- In the RSTP mode, the transmitted BPDUs contain the port's designated priority vector (DPV) (conforms to 802.1D-2004). Older implementations may be confused by the DPV in a BPDU and may fail to recognize an agreement BPDU correctly. This would result in a slow transition to a forwarding state (30 seconds). For this reason, in the comp-dot1w mode, these BPDUs contain the port's port priority vector (conforms to 802.1w).

The 7450 ESS, 7750 SR, and 7950 XRS support two BPDU encapsulation formats, and can dynamically switch between the following supported formats (on a per-SAP basis):

- IEEE 802.1D STP
- Cisco PVST

3.2.9.2 Multiple spanning tree

The Multiple Spanning Tree Protocol (MSTP) extends the concept of IEEE 802.1w RSTP by allowing grouping and associating VLANs to Multiple Spanning Tree Instances (MSTI). Each MSTI can have its own topology, which provides architecture enabling load balancing by providing multiple forwarding paths. At the same time, the number of STP instances running in the network is significantly reduced as compared to Per VLAN STP (PVST) mode of operation. Network fault tolerance is also improved because a failure in one instance (forwarding path) does not affect other instances.

The Nokia implementation of M-VPLS is used to group different VPLS instances under a single RSTP instance. Introducing MSTP into the M-VPLS allows interoperating with traditional Layer 2 switches in an access network and provides an effective solution for dual homing of many business Layer 2 VPNs into a provider network.

3.2.9.2.1 Redundancy access to VPLS

The GigE MAN portion of the network is implemented with traditional switches. Using MSTP running on individual switches facilitates redundancy in this part of the network. To provide dual homing of all VPLS services accessing from this part of the network, the VPLS PEs must participate in MSTP.

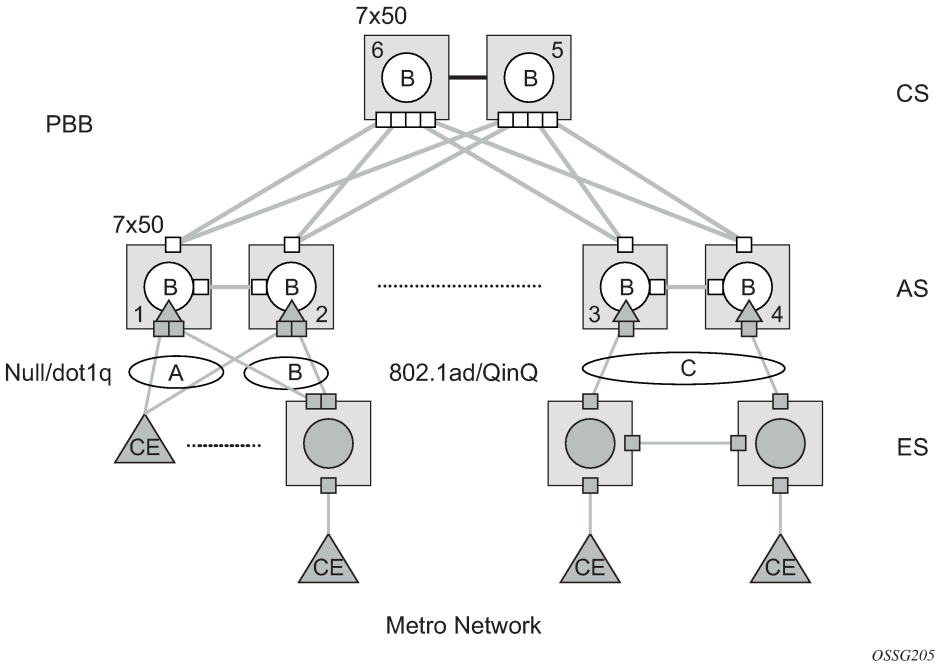
This can be achieved by configuring M-VPLS on VPLS-PEs (only PEs directly connected to the GigE MAN network), then assigning different managed-VLAN ranges to different MSTP instances. Typically, the M-VPLS would have SAPs with null encapsulations (to receive, send, and transmit MSTP BPDUs) and a mesh SDP to interconnect a pair of VPLS PEs.

Different access scenarios are displayed in Figure 62: Access resiliency as an example of network diagrams dually connected to the PBB PEs:

Access Type B	One QinQ switch connected by QinQ/801ad SAPs

Access Type A	Source devices connected by null or dot1q SAPs
Access Type C	Two or more ES devices connected by QinQ/802.1ad SAPs

Figure 62: Access resiliency



The following mechanisms are supported for the I-VPLS:

- STP/RSTP can be used for all access types.
- M-VPLS with MSTP can be used as is just for access type A. MSTP is required for access type B and C.
- LAG and MC-LAG can be used for access type A and B.
- Split-horizon-group does not require residential.

PBB I-VPLS inherits current STP configurations from the regular VPLS and M-VPLS.

3.2.9.3 MSTP for QinQ SAPs

MSTP runs in a M-VPLS context and can control SAPs from source VPLS instances. QinQ SAPs are supported. The outer tag is considered by MSTP as part of VLAN range control.

3.2.9.4 Provider MSTP

Provider MSTP is specified in IEEE-802.1ad-2005. It uses a provider bridge group address instead of a regular bridge group address used by STP, RSTP, and MSTP BPDUs. This allows for implicit separation of source and provider control planes.

The 802.1ad access network sends PBB PE P-MSTP BPDUs using the specified MAC address and also works over QinQ interfaces. P-MSTP mode is used in PBBN for core resiliency and loop avoidance.

Similar to regular MSTP, the STP mode (for example, PMSTP) is only supported in VPLS services where the m-VPLS flag is configured.

3.2.9.4.1 MSTP general principles

MSTP represents a modification of RSTP that allows the grouping of different VLANs into multiple MSTIs. To enable different devices to participate in MSTIs, they must be consistently configured. A collection of interconnected devices that have the same MST configuration (region-name, revision, and VLAN-to-instance assignment) comprises an MST region.

There is no limit to the number of regions in the network, but every region can support a maximum of 16 MSTIs. Instance 0 is a special instance for a region, known as the Internal Spanning Tree (IST) instance. All other instances are numbered from 1 to 4094. IST is the only spanning-tree instance that sends and receives BPDUs (typically, BPDUs are untagged). All other spanning-tree instance information is included in MSTP records (M-records), which are encapsulated within MSTP BPDUs. This means that a single BPDU carries information for multiple MSTIs, which reduces overhead of the protocol.

Any MSTI is local to an MSTP region and completely independent from an MSTI in other MST regions. Two redundantly connected MST regions use only a single path for all traffic flows (no load balancing between MST regions or between MST and SST region).

Traditional Layer 2 switches running MSTP protocol assign all VLANs to the IST instance per default. The operator may then "re-assign" individual VLANs to a specified MSTI by configuring per VLAN assignment. This means that an SR-series PE can be considered as a part of the same MST region only if the VLAN assignment to IST and MSTIs is identical to the one of Layer 2 switches in the access network.

3.2.9.4.2 MSTP in the SR-series platform

The SR-series platform uses a concept of M-VPLS to group different SAPs under a single STP instance. The VLAN range covering SAPs to be managed by a specified M-VPLS is declared under a specific M-VPLS SAP definition. MSTP mode-of-operation is only supported in an M-VPLS.

When running MSTP, by default, all VLANs are mapped to the CIST. At the VPLS level, VLANs can be assigned to specific MSTIs. When running RSTP, the operator must explicitly indicate, per SAP, which VLANs are managed by that SAP.

3.2.9.5 Enhancements to the spanning tree protocol

To interconnect PE devices across the backbone, service tunnels (SDPs) are used. These service tunnels are shared among multiple VPLS instances. The Nokia implementation of the STP incorporates some enhancements to make the operational characteristics of VPLS more effective. The implementation of STP on the router is modified to guarantee that service tunnels are not blocked in any circumstance without imposing artificial restrictions on the placement of the root bridge within the network. The modifications introduced are fully compliant with the 802.1D-2004 STP specification.

When running MSTP, spoke-SDPs cannot be configured. Also, ensure that all bridges connected by mesh SDPs are in the same region. If not, the mesh is prevented from becoming active (trap is generated).

To achieve this, all mesh SDPs are dynamically configured as either root ports or designated ports. The PE devices participating in each VPLS mesh determine (using the root path cost learned as part of the normal protocol exchange) which of the 7450 ESS, 7750 SR, or 7950 XRS devices is closest to the root of the network. This PE device is internally designated as the primary bridge for the VPLS mesh. As a result of

this, all network ports on the primary bridges are assigned the designated port role and therefore remain in the forwarding state.

The second part of the solution ensures that the remaining PE devices participating in the STP instance see the SDP ports as a lower-cost path to the root than a path that is external to the mesh. Internal to the PE nodes participating in the mesh, the SDPs are treated as zero-cost paths toward the primary bridge. As a consequence, the path through the mesh is seen as lower cost than any alternative and the PE node designates the network port as the root port. This approach ensures that network ports always remain in forwarding state.

In combination, these two features ensure that network ports are never blocked and maintain interoperability with bridges external to the mesh that are running STP instances.

3.2.9.5.1 L2PT termination

L2PT is used to transparently transport protocol data units (PDUs) of Layer 2 protocols such as STP, CDP, VTP, PAGP, and UDLD. This allows running these protocols between customer CPEs without involving backbone infrastructure.

The 7450 ESS, 7750 SR, and 7950 XRS routers allow transparent tunneling of PDUs across the VPLS core. However, in some network designs, the VPLS PE is connected to CPEs through a legacy Layer 2 network, and it does not have direct connections. In such environments, termination of tunnels through such infrastructure is required.

L2PT tunnels PDUs by overwriting MAC destination addresses at the ingress of the tunnel to a proprietary MAC address such as 01-00-0c-cd-cd-d0. At the egress of the tunnel, this MAC address is then overwritten back to the MAC address of the respective Layer 2 protocol.

The 7450 ESS, 7750 SR, and 7950 XRS routers support L2PT termination for STP BPDUs. More specifically:

- At ingress of every SAP/spoke-SDP that is configured as L2PT termination, all PDUs with a MAC destination address of 01-00-0c-cd-cd-d0 are intercepted and their MAC destination address is overwritten to the MAC destination address used for the corresponding protocol (PVST, STP, RSTP). The type of the STP protocol can be derived from LLC and SNAP encapsulation.
- In the egress direction, all STP PDUs received on all VPLS ports are intercepted and L2PT encapsulation is performed for SAP/spoke-SDPs configured as L2PT termination points. Because of implementation reasons, PDU interception and redirection to CPM can be performed only at ingress. Therefore, to comply with the above requirement, as soon as at least one port of a specified VPLS service is configured as L2PT termination port, redirection of PDUs to CPM is set on all other ports (SAPs, spoke-SDPs, and mesh SDPs) of the VPLS service.

L2PT termination can be enabled only if STP is disabled in a context of the specified VPLS service.

3.2.9.5.2 BPDU translation

VPLS networks are typically used to interconnect different customer sites using different access technologies such as Ethernet and bridged-encapsulated ATM PVCs. Typically, different Layer 2 devices can support different types of STP, even if they are from the same vendor. In some cases, it is necessary to provide BPDU translation to provide an interoperable e2e solution.

To address these network designs, BPDU format translation is supported on 7450 ESS, 7750 SR, and 7950 XRS devices. If enabled on a specified SAP or spoke-SDP, the system intercepts all BPDUs destined for that interface and perform required format translation such as STP-to-PVST or the other way around.

Similarly, BPDU interception and redirection to the CPM is performed only at ingress, meaning that as soon as at least one port within a specified VPLS service has BPDU translation enabled, all BPDUs received on any of the VPLS ports are redirected to the CPM.

BPDU translation requires all encapsulation actions that the data path would perform for a specified outgoing port (such as adding VLAN tags depending on the outer SAP and the SDP encapsulation type) and adding or removing all the required VLAN information in a BPDU payload.

This feature can be enabled on a SAP only if STP is disabled in the context of the specified VPLS service.

3.2.9.5.3 L2PT and BPDU translation

Cisco Discovery Protocol (CDP), Digital Trunking Protocol (DTP), Port Aggregation Protocol (PAGP), Uni-directional Link Detection (UDLD), and Virtual Trunk Protocol (VTP) are supported. These protocols automatically pass the other protocols tunneled by L2PT toward the CPM and all carry the same specific Cisco MAC.

The existing L2PT limitations apply.

- The protocols apply only to VPLS.
- The protocols and running STP on the same VPLS as soon as one SAP has L2PT enabled are mutually exclusive.
- Forwarding occurs on the CPM.

3.2.10 VPLS redundancy

The VPLS standard (RFC 4762, *Virtual Private LAN Services Using LDP Signaling*) includes provisions for hierarchical VPLS, using point-to-point spoke-SDPs. Two applications have been identified for spoke-SDPs:

- to connect Multi-Tenant Units (MTUs) to PEs in a metro area network
- to interconnect the VPLS nodes of two metro networks

In both applications, the spoke-SDPs serve to improve the scalability of VPLS. While node redundancy is implicit in non-hierarchical VPLS services (using a full mesh of SDPs between PEs), node redundancy for spoke-SDPs needs to be provided separately.

Nokia routers have implemented special features for improving the resilience of hierarchical VPLS instances, in both MTU and inter-metro applications.

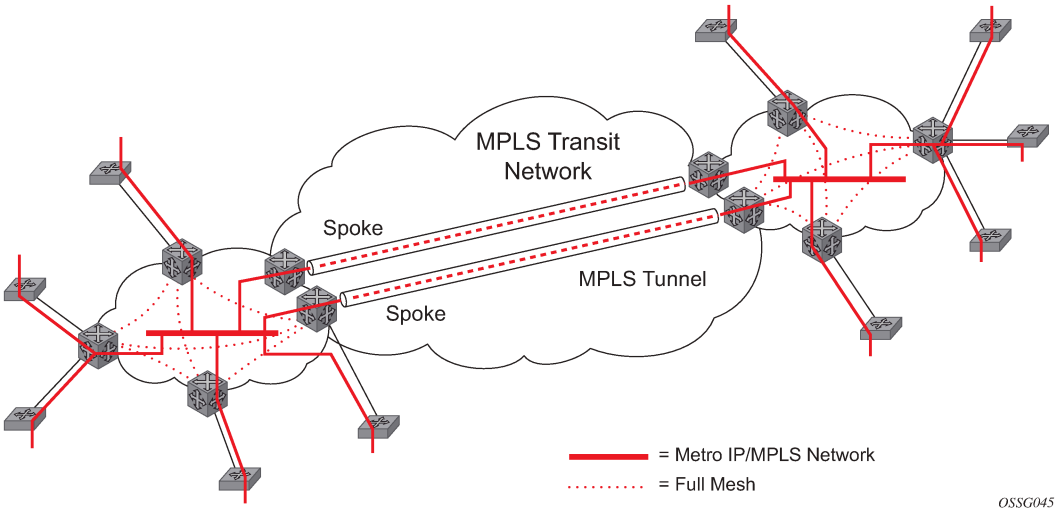
3.2.10.1 Spoke SDP redundancy for metro interconnection

When two or more meshed VPLS instances are interconnected by redundant spoke-SDPs (as shown in Figure 63: HVPLS with spoke redundancy), a loop in the topology results. To remove such a loop from the topology, STP can be run over the SDPs (links) that form the loop, such that one of the SDPs is blocked. As running STP in each and every VPLS in this topology is not efficient, the node includes functionality that can associate a number of VPLSs to a single STP instance running over the redundant SDPs. Therefore, node redundancy is achieved by running STP in one VPLS and applying the conclusions of this STP to the other VPLS services. The VPLS instance running STP is referred to as the "management VPLS" or M-VPLS.

If the active node fails, STP on the management VPLS in the standby node changes the link states from disabled to active. The standby node then broadcasts a MAC flush LDP control message in each of the protected VPLS instances, so that the address of the newly active node can be relearned by all PEs in the VPLS.

It is possible to configure two management VPLS services, where both VPLS services have different active spokes (this is achieved by changing the path cost in STP). By associating different user VPLSs with the two management VPLS services, load balancing across the spokes can be achieved.

Figure 63: HVPLS with spoke redundancy



3.2.10.2 Spoke SDP based redundant access

This feature provides the ability to have a node deployed as MTUs (Multi-Tenant Units) to be multihomed for VPLS to multiple routers deployed as PEs without requiring the use of M-VPLS.

In the configuration example shown in Figure 63: HVPLS with spoke redundancy, the MTUs have spoke-SDPs to two PE devices. One is designated as the primary and one as the secondary spoke-SDP. This is based on a precedence value associated with each spoke.

The secondary spoke is in a blocking state (both on receive and transmit) as long as the primary spoke is available. When the primary spoke becomes unavailable (because of the link failure, PEs failure, and so on), the MTUs immediately switch traffic to the backup spoke and start receiving traffic from the standby spoke. Optional revertive operation (with configurable switch-back delay) is supported. Forced manual switchover is also supported.

To speed up the convergence time during a switchover, MAC flush is configured. The MTUs generate a MAC flush message over the newly unblocked spoke when a spoke change occurs. As a result, the PEs receiving the MAC flush, flush all MACs associated with the impacted VPLS service instance and forward the MAC flush to the other PEs in the VPLS network if **propagate-mac-flush** is enabled.

3.2.10.3 Inter-domain VPLS resiliency using multi-chassis endpoints

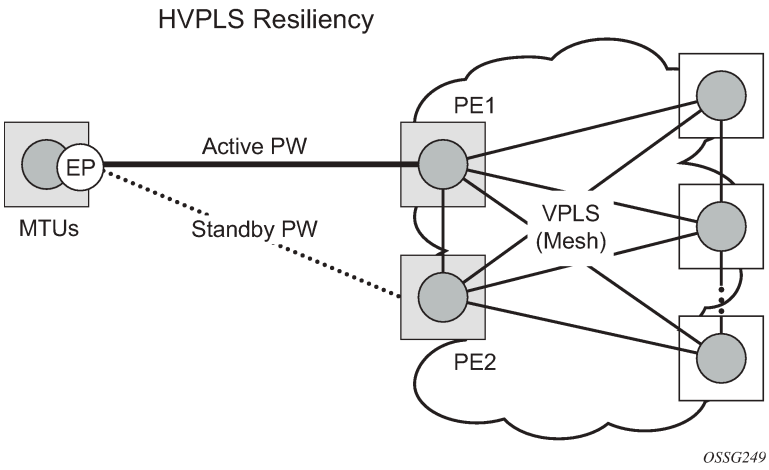
Inter-domain VPLS refers to a VPLS deployment where sites may be located in different domains. An example of inter-domain deployment can be where different metro domains are interconnected over a

Wide Area Network (Metro1-WAN-Metro2) or where sites are located in different autonomous systems (AS1-ASBRs-AS2).

Multi-chassis endpoint (MC-EP) provides an alternate solution that does not require RSTP at the gateway VPLS PEs while still using pseudowires to interconnect the VPLS instances located in the two domains. It is supported in both VPLS and PBB-VPLS on the B-VPLS side.

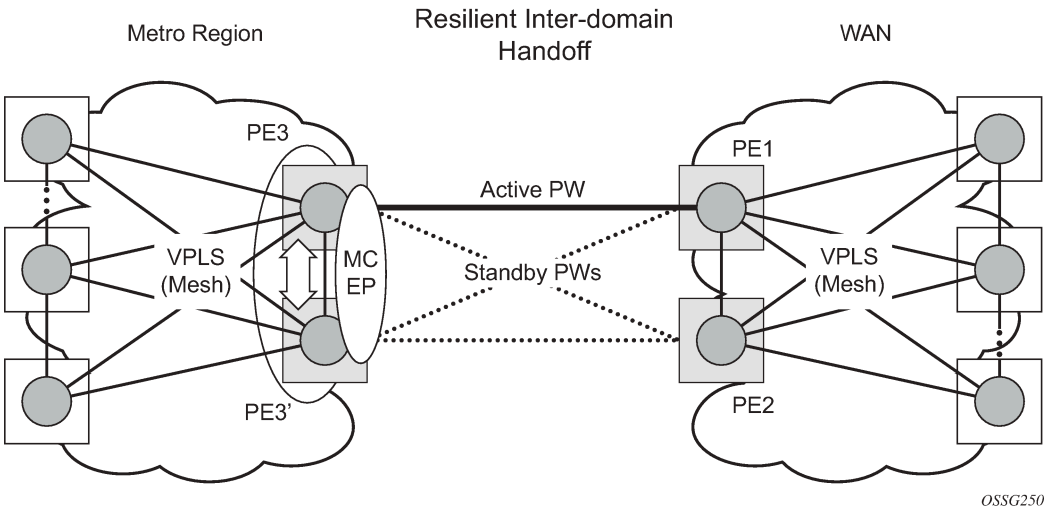
MC-EP expands the single chassis endpoint based on active-standby pseudowires for VPLS, shown in Figure 64: HVPLS resiliency based on AS pseudowires.

Figure 64: HVPLS resiliency based on AS pseudowires



The active-standby pseudowire solution is appropriate for the scenario when only one VPLS PE (MTUs) needs to be dual-homed to two core PEs (PE1 and PE2). When multiple VPLS domains need to be interconnected, the above solution provides a single point of failure at the MTU-s. The example shown in Figure 65: Multi-chassis pseudowire endpoint for VPLS can be used.

Figure 65: Multi-chassis pseudowire endpoint for VPLS



The two gateway pairs, PE3-PE3' and PE1-PE2, are interconnected using a full mesh of four pseudowires out of which only one pseudowire is active at any time.

The concept of pseudowire endpoint for VPLS provides multi-chassis resiliency controlled by the MC-EP pair, PE3-PE3' in this example. This scenario, referred to as multi-chassis pseudowire endpoint for VPLS, provides a way to group pseudowires distributed between PE3 and PE3 chassis in a virtual endpoint that can be mapped to a VPLS instance.

The MC-EP inter-chassis protocol is used to ensure configuration and status synchronization of the pseudowires that belong to the same MC-EP group on PE3 and PE3. Based on the information received from the peer shelf and the local configuration, the master shelf decides on which pseudowire becomes active.

The MC-EP solution is built around the following components:

- Multi-chassis protocol used to perform the following functions:
  - Selection of master chassis.
  - Synchronization of the pseudowire configuration and status.
  - Fast detection of peer failure or communication loss between MC-EP peers using either centralized BFD, if configured, or its own keep-alive mechanism.
- T-LDP signaling of pseudowire status informs the remote PEs about the choices made by the MC-EP pair.
- Pseudowire data plane is represented by the four pseudowires inter-connecting the gateway PEs.
  - Only one of the pseudowires is activated based on the primary/secondary, preference configuration, and pseudowire status. In case of a tie, the pseudowire located on the master chassis is chosen.
  - The rest of the pseudowires are blocked locally on the MC-EP pair and on the remote PEs as long as they implement the pseudowire active/standby status.

3.2.10.3.1 Fast detection of peer failure using BFD

Although the MC-EP protocol has its own keep-alive mechanisms, sharing a common mechanism for failure detection with other protocols (for example, BGP, RSVP-TE) scales better. MC-EP can be configured to use the centralized BFD mechanism.

Similar to other protocols, MC-EP registers with BFD if the **bfd-enable** command is active under the **config**>**redundancy**>**multi-chassis**>**peer**>**mc-ep** context. As soon as the MC-EP application is activated using no shutdown, it tries to open a new BFD session or register automatically with an existing one. The source-ip configuration under redundancy multi-chassis peer-ip is used to determine the local interface while the peer-ip is used as the destination IP for the BFD session. After MC-EP registers with an active BFD session, it uses it for fast detection of MC-EP peer failure. If BFD registration or BFD initialization fails, the MC-EP keeps using its own keep-alive mechanism and it sends a trap to the NMS signaling the failure to register with/open a BFD session.

To minimize operational mistakes and wrong peer interpretation for the loss of BFD session, the following additional rules are enforced when the MC-EP is registering with a BFD session:

- Only the centralized BFD sessions using system or loopback IP interfaces (source-ip parameter) are accepted in order for MC-EP to minimize the false indication of peer loss.
- If the BFD session associated with MC-EP protocol is using a system/loopback interface, the following actions are not allowed under the interface: IP address change, "shutdown", "no bfd" commands. If one of these actions is required under the interface, the operator needs to disable BFD using one the following procedures:
  - The **no bfd-enable** command in the **config**>**redundancy**>**multi-chassis**>**peer**>**mc-ep** context.

Note: This is the recommended procedure.

– The **shutdown** command in the **config**>**redundancy**>**multi-chassis**>**peer**>**mc-ep** or from under **config**>**redundancy**>**multi-chassis**>**peer** contexts.

MC-EP keep-alives are still exchanged for the following reasons:

- As a backup; if the BFD session does not come up or is disabled, the MC-EP protocol uses its own keep-alives for failure detection.
- To ensure the database is cleared if the remote MC-EP peer is shut down or misconfigured (each x seconds; one second suggested as default).

If MC-EP de-registers with BFD using the **no bfd-enable** command, the following processing steps occur:

Note: There should be no pseudowire status change during this process.

1. The local peer indicates to the MC-EP peer that the local BFD is being disabled using the MC-EP peer-config-TLV fields ([BFD local: BFD remote]). This is done to avoid the wrong interpretation of the BFD session loss.
2. The remote peer acknowledges reception indicating through the same peer-config-TLV fields that it is de-registering with the BFD session.
3. Both MC-EP peers de-register and use only keep-alives for failure detection.

Traps are sent when the status of the monitoring of the MC-EP session through BFD changes in the following instances:

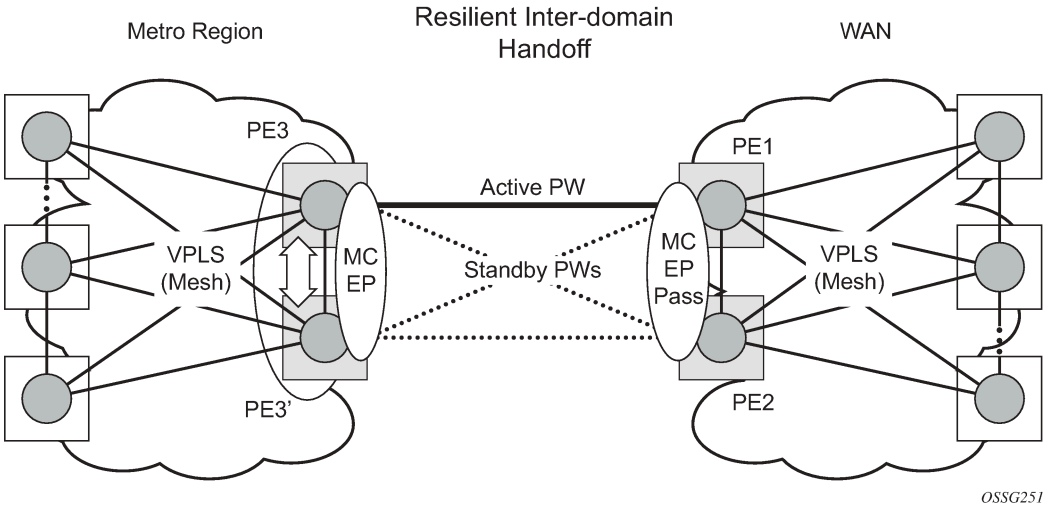
- When red/mc/peer is no shutdown and BFD is not enabled, a notification is sent indicating BFD is not monitoring the MC-EP peering session.
- When BFD changes to open, a notification is sent indicating BFD is monitoring the MC-EP peering session.
- When BFD changes to down/close, a notification is sent indicating BFD is not monitoring the MC-EP peering session.

3.2.10.3.2 MC-EP passive mode

The MC-EP mechanisms are built to minimize the possibility of loops. It is possible that human error could create loops through the VPLS service. One way to prevent loops is to enable the MAC move feature in the gateway PEs (PE3, PE3', PE1, and PE2).

An MC-EP passive mode can also be used on the second PE pair, PE1 and PE2, as a second layer of protection to prevent any loops from occurring if the operator introduces operational errors on the MC-EP PE3, PE3 pair. An example is shown in Figure 66: MC-EP in passive mode.

Figure 66: MC-EP in passive mode



When in passive mode, the MC-EP peers stay dormant as long as one active pseudowire is signaled from the remote end. If more than one pseudowire belonging to the passive MC-EP becomes active, the PE1 and PE2 pair applies the MC-EP selection algorithm to select the best choice and blocks all others. No signaling is sent to the remote pair to avoid flip-flop behavior. A trap is generated each time MC-EP in passive mode activates. Every occurrence of this kind of trap should be analyzed by the operator as it is an indication of possible misconfiguration on the remote (active) MC-EP peering.

For the MC-EP passive mode to work, the pseudowire status signaling for active/standby pseudowires should be enabled. This requires the following CLI configurations:

For the remote MC-EP PE3, PE3 pair:

config>service>vpls>endpoint no suppress-standby-signaling

When MC-EP passive mode is enabled on the PE1 and PE2 pair, the following command is always enabled internally, regardless of the actual configuration:

config>service>vpls>endpoint no ignore-standby-signaling

3.2.10.4 Support for single chassis endpoint mechanisms

In cases of SC-EP, there is a consistency check to ensure that the configuration of the member pseudowires is the same. For example, mac-pining, mac-limit, and ignore standby signaling must be the same. In the MC-EP case, there is no consistency check between the member endpoints located on different chassis. The operator must carefully verify the configuration of the two endpoints to ensure consistency.

The following rules apply for suppress-standby-signaling and ignore-standby parameters:

- Regular MC-EP mode (non-passive) follows the suppress-standby-signaling and ignore-standby settings from the related endpoint configuration.
- For MC-EP configured in passive mode, the following settings are used, regardless of previous configuration: **suppress-standby-sig** and **no ignore-standby-sig**. It is expected that when passive mode is used at one side, the regular MC-EP side activates signaling with **no suppress-stdby-sig**.

- When passive mode is configured in just one of the nodes in the MC-EP peering, the other node is forced to change to passive mode. A trap is sent to the operator to signal the wrong configuration.

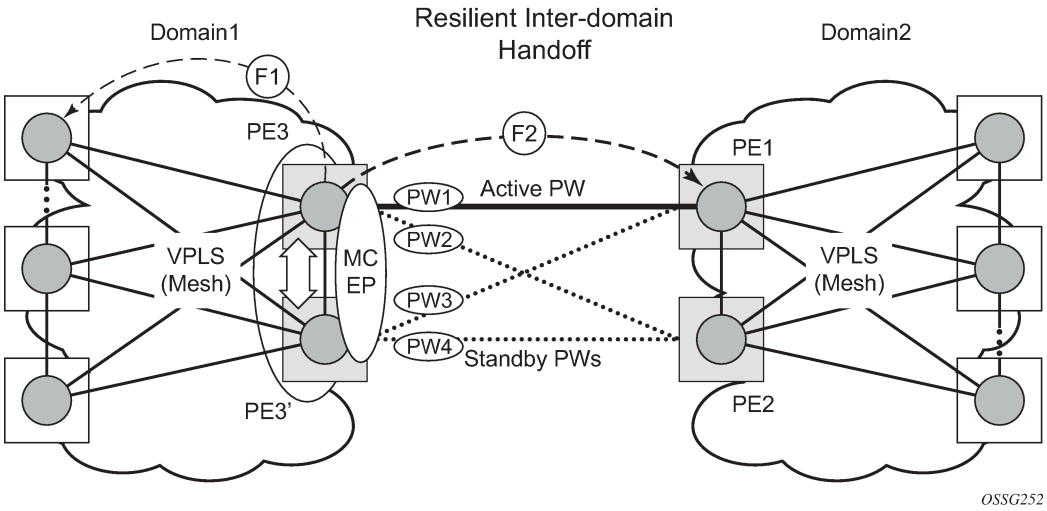
This section also describes how the main mechanisms used for single chassis endpoint are adapted for the MC-EP solution.

3.2.10.4.1 MAC flush support in MC-EP

In an MC-EP scenario, failure of a pseudowire or gateway PE determines activation of one of the next best pseudowires in the MC-EP group. This section describes the MAC flush procedures that can be applied to ensure blackhole avoidance.

Figure 67: MAC flush in the MC-EP solution shows a pair of PE gateways (PE3 and PE3) running MC-EP toward PE1 and PE2 where F1 and F2 are used to indicate the possible direction of the MAC flush, signaled using T-LDP MAC withdraw message. PE1 and PE2 can only use regular VPLS pseudowires and do not have to use an MC-EP or a regular pseudowire endpoint.

Figure 67: MAC flush in the MC-EP solution



Regular MAC flush behavior applies for the LDP MAC withdraw sent over the T-LDP sessions associated with the active pseudowire in the MC-EP; for example, PE3 to PE1. That includes any Topology Change Notification (TCN) events or failures associated with SAPs or pseudowires not associated with the MC-EP.

The following MAC flush behaviors apply to changes in the MC-EP pseudowire selection:

- If the local PW2 becomes active on PE3:
  - On PE3, the MACs mapped to PW1 are moved to PW2.
  - A T-LDP flush-all-but-mine message is sent toward PE2 in the F2 direction and is propagated by PE2 in the local VPLS mesh.
  - No MAC flush is sent in the F1 direction from PE3.
- If one of the pseudowires on the pair PE3 becomes active; for example, PW4:
  - On PE3, the MACs mapped to PW1 are flushed, the same as for a regular endpoint.
  - PE3 must be configured with **send-flush-on-failure** to send a T-LDP flush-all-from-me message toward VPLS mesh in the F1 direction.

– PE3 sends a T-LDP flush-all-but-mine message toward PE2 in the F2 direction, which is propagated by PE2 in the local VPLS mesh. When MC-EP is in passive mode and the first spoke becomes active, a no MAC flush-all-but-mine message is generated.

3.2.10.4.2 Block-on-mesh-failure support in MC-EP scenario

The following rules describe how the block-on-mesh-failure operates with the MC-EP solution (see Figure 67: MAC flush in the MC-EP solution):

- If PE3 does not have any forwarding path toward Domain1 mesh, it should block both PW1 and PW2 and inform PE3 so one of its pseudowires can be activated.
- To allow the use of block-on-mesh-failure for MC-EP, a block-on-mesh-failure parameter can be specified in the **config**>**service**>**vpls**>**endpoint** context with the following rules:
  - The default is **no block-on-mesh-failure** to allow for easy migration.
  - For a spoke-SDP to be added under an endpoint, the setting for its **block-on-mesh-failure** parameter must be in synchronization with the endpoint parameter.
  - After the spoke-SDP is added to an endpoint, the configuration of its **block-on-mesh-failure** parameter is disabled. A change in endpoint configuration for the **block-on-mesh-failure** parameter is propagated to the individual spoke-SDP configuration.
  - When a spoke-SDP is removed from the endpoint group, it inherits the last configuration from the endpoint parameter.
  - Adding an MC-EP under the related endpoint configuration does not affect the above behavior.

  Before Release 7.0, the **block-on-mesh-failure** command could not be enabled under **config**>**service**>**vpls**>**endpoint** context. For a spoke-SDP to be added to an (single-chassis) endpoint, its **block-on-mesh-failure** had to be disabled (**config**>**service**>**vpls**>**spoke-sdp**>**no block-on-mesh-failure**). Then, the configuration of **block-on-mesh-failure** under a spoke-SDP is blocked.
- If **block-on-mesh-failure** is enabled on PE1 and PE2, these PEs signal pseudowire standby status toward the MC-EP PE pair. PE3 and PE3 should consider the pseudowire status signaling from remote PE1 and PE2 when making the selection of the active pseudowire.

3.2.10.4.3 Support for force spoke SDP in MC-EP

In a regular (single chassis) endpoint scenario, the following command can be used to force a specific SDP binding (pseudowire) to become active:

tools perform service id *service-id* **endpoint** *force*

In the MC-EP case, this command has a similar effect when there is a single forced SDP binding in an MC-EP. The forced SDP binding (pseudowire) is selected as active.

However, when the command is run at the same time as both MC-EP PEs, when the endpoints belong to the same MC-EP, the regular MC-EP selection algorithm (for example, the operational status ⇒ precedence value) is applied to determine the winner.

3.2.10.4.4 Revertive behavior for primary pseudowires in an MC-EP

For a single-chassis endpoint, a revert-time command is provided under the VPLS endpoint.

In a regular endpoint, the revert-time setting affects just the pseudowire defined as primary (precedence 0). For a failure of the primary pseudowire followed by restoration, the revert-timer is started. After it expires, the primary pseudowire takes the active role in the endpoint. This behavior does not apply for the case when both pseudowires are defined as secondary; that is, if the active secondary pseudowire fails and is restored, it stays in standby until a configuration change or a force command occurs.

In the MC-EP case, the revertive behavior is supported for pseudowire defined as primary (precedence 0). The following rules apply:

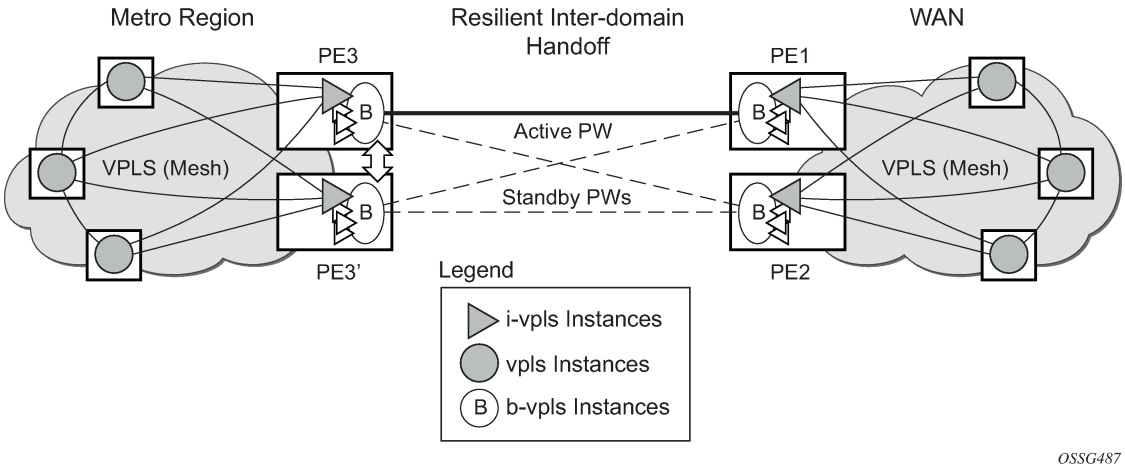
- The revert-time setting under each individual endpoint control the behavior of the local primary pseudowire if one is configured under the local endpoint.
- The secondary pseudowires behave as in the regular endpoint case.

3.2.10.5 Using B-VPLS for increased scalability and reduced convergence times

The PBB-VPLS solution can be used to improve scalability of the solution and to reduce convergence time. If PBB-VPLS is deployed starting at the edge PEs, the gateway PEs contain only B-VPLS instances. The MC-EP procedures described for regular VPLS apply.

PBB-VPLS can also be enabled just on the gateway MC-EP PEs, as shown in Figure 68: MC-EP with B-VPLS.

Figure 68: MC-EP with B-VPLS



Multiple I-VPLS instances may be used to represent in the gateway PEs the customer VPLS instances using the PBB-VPLS M:1 model described in the PBB section. A backbone VPLS (B-VPLS) is used in this example to administer the resiliency for all customer VPLS instances at the domain borders. Just one MC-EP is required to be configured in the B-VPLS to address hundreds or even thousands of customer VPLS instances. If load balancing is required, multiple B-VPLS instances may be used to ensure even distribution of the customers across all the pseudowires interconnecting the two domains. In this example, four B-VPLSs are able to load share the customers across all four possible pseudowire paths.

The use of MC-EP with B-VPLS is strictly limited to cases where VPLS mesh exists on both sides of a B-VPLS. For example, active/standby pseudowires resiliency in the I-VPLS context where PE3 and PE3' are PE-rs cannot be used because there is no way to synchronize the active/standby selection between the two domains.

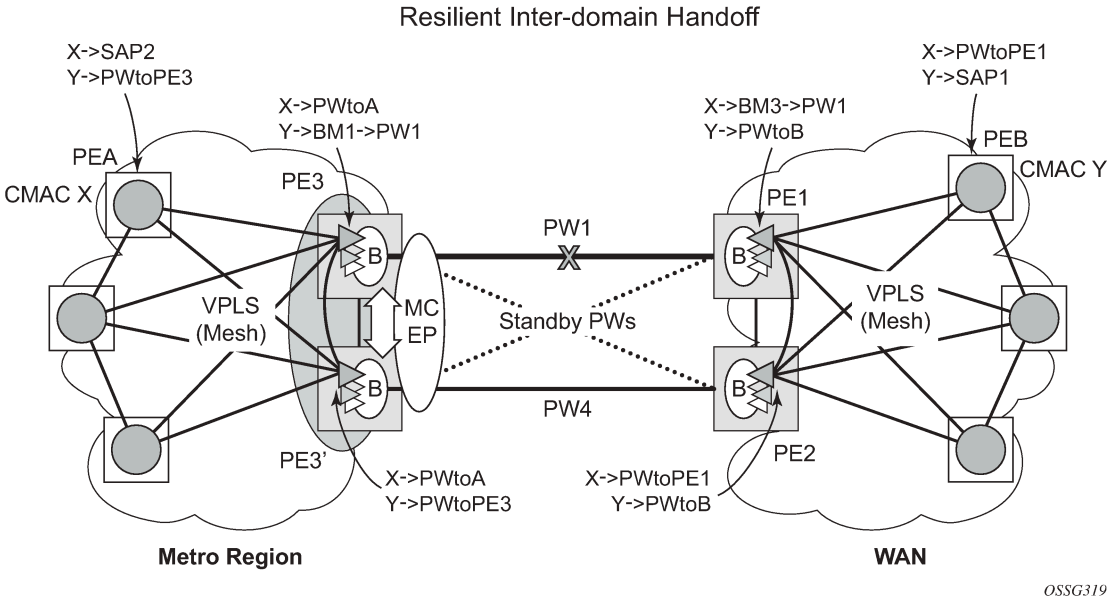
For a similar reason, MC-LAG resiliency in the I-VPLS context on the gateway PEs participating in the MC-EP (PE3 and PE3') should not be used.

For the PBB topology in Figure 68: MC-EP with B-VPLS, block-on-mesh-failure in the I-VPLS domain does not have any effect on the B-VPLS MC-EP side. That is because mesh failure in one I-VPLS should not affect other I-VPLSs sharing the same B-VPLS.

3.2.10.6 MAC flush additions for PBB VPLS

The scenario shown in Figure 69: MC-EP with B-VPLS failure scenario is used to define the blackholing problem in PBB-VPLS using MC-EP.

Figure 69: MC-EP with B-VPLS failure scenario



In the topology shown in Figure 69: MC-EP with B-VPLS failure scenario, PEA and PEB are regular VPLS PEs participating in the VPLS mesh deployed in the metro and WAN region, respectively. As the traffic flows between CEs with C-MAC X and C-MAC Y, the FDB entries in PEA, PE3, PE1 and PEB are installed. An LDP flush-all-but-mine message is sent from PE3 to PE2 to clear the B-VPLS FDBs. The traffic between C-MAC X and C-MAC Y is blackholed as long as the entries from the VPLS and I-VPLS FDBs along the path are not removed. This may take as long as 300 seconds, the usual aging timer used for MAC entries in a VPLS FDB.

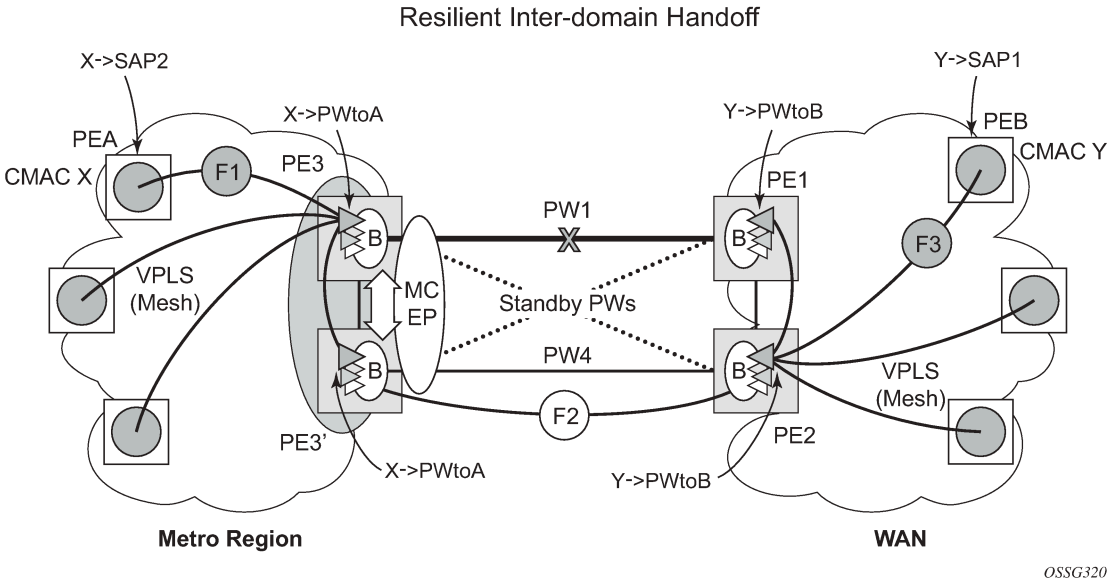
A MAC flush is required in the I-VPLS space from PBB PEs to PEA and PEB to avoid blackholing in the regular VPLS space.

In the case of a regular VPLS, the following procedure is used:

1. PE3 sends a flush-all-from-me message toward its local blue I-VPLS mesh to PE3 and PEA when its MC-EP becomes disabled.
2. PE3 sends a flush-all-but-mine message on the active PW4 to PE2, which is then propagated by PE2 (propagate-mac-flush must be on) to PEB in the WAN I-VPLS mesh.

For consistency, a similar procedure is used for the B-VPLS case as shown in Figure 70: MC-EP with B-VPLS MAC flush solution.

Figure 70: MC-EP with B-VPLS MAC flush solution



In this example, the MC-EP activates B-VPLS PW4 because of either a link/node failure or because of an MC-EP selection re-run that affected the previously active PW1. As a result, the endpoint on PE3 containing PW1 goes down.

The following steps apply:

1. PE3 sends, in the local I-VPLS context, an LDP flush-all-from-me message (marked with F1) to PEA and to the other regular VPLS PEs, including PE3. The following command enables this behavior on a per I-VPLS basis: **config>service>vpls ivpls>send-flush-on-bvpls-failure**.

   As a result, PEA, PE3, and the other local VPLS PEs in the metro clear the VPLS FDB entries associated with PW to PE3.

2. PE3 clears the entries associated with PW1 and sends, in the B-VPLS context, an LDP flush-all-but-mine message (marked with F2) toward PE2 on the active PW4.

   As a result, PE2 clears the B-VPLS FDB entries not associated with PW4.

3. PE2 propagates the MAC flush-all-but-mine (marked with F3) from B-VPLS in the related I-VPLS contexts toward all participating VPLS PEs; for example, in the blue I-VPLS to PEB, PE1. It also clears all the C-MAC entries associated with I-VPLS pseudowires.

   The following command enables this behavior on a per I-VPLS basis:

   config>service>vpls ivpls>propagate-mac-flush-from-bvpls

   As a result, PEB, PE1, and the other local VPLS PEs in the WAN clear the VPLS FDB entries associated with PW to PE2.

Note: This command does not control the propagation in the related I-VPLS of the B-VPLS LDP MAC flush containing a PBB TLV (B-MAC and ISID list).

Similar to regular VPLS, LDP signaling of the MAC flush follows the active topology; for example, no MAC flush is generated on standby pseudowires.

Other failure scenarios are addressed using the same or a subset of the above steps:

- If the pseudowire (PW2) in the same endpoint with PW1 becomes active instead of PW4, there is no MAC flush of F1 type.
- If the pseudowire (PW3) in the same endpoint becomes active instead of PW4, the same procedure applies.

For an SC/MC endpoint configured in a B-VPLS, failure/deactivation of the active pseudowire member always generates a local MAC flush of all the B-MAC associated with the pseudowire. It never generates a MAC move to the newly active pseudowire even if the endpoint stays up. That is because in SC-EP/MC-EP topology, the remote PE may be the terminating PBB PE and may not be able to reach the B-MAC of the other remote PE. Therefore, connectivity between them exists only over the regular VPLS mesh.

For the same reasons, Nokia recommends that static B-MAC not be used on SC/MC endpoints.

3.2.11 VPLS access redundancy

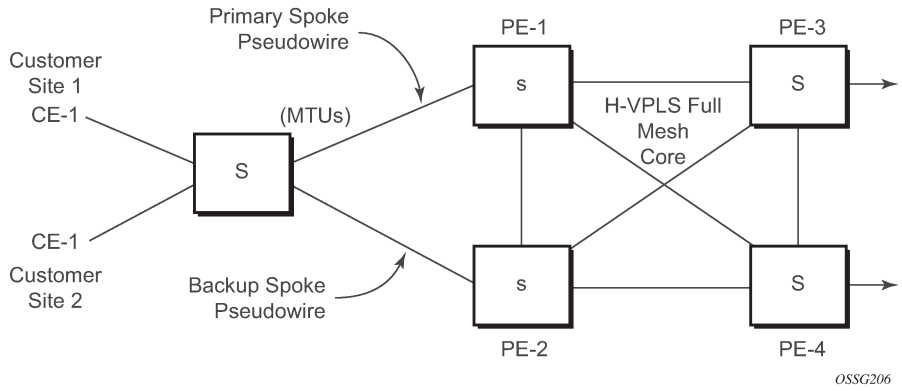
A second application of hierarchical VPLS is using MTUs that are not MPLS-enabled that must have Ethernet links to the closest PE node. To protect against failure of the PE node, an MTU can be dual-homed and have two SAPs on two PE nodes.

There are several mechanisms that can be used to resolve a loop in an access circuit; however, from an operations perspective, they can be subdivided into two groups:

- STP-based access, with or without M-VPLS.
- Non-STP based access using mechanisms such as MC-LAG, MC-APS, MC-Ring.

3.2.11.1 STP-based redundant access to VPLS

Figure 71: Dual-homed MTUs in two-tier hierarchy H-VPLS



In the configuration shown in Figure 71: Dual-homed MTUs in two-tier hierarchy H-VPLS, STP is activated on the MTU and two PEs to resolve a potential loop. STP only needs to run in a single VPLS instance, and the results of the STP calculations are applied to all VPLSs on the link.

In this configuration, the scope of the STP domain is limited to MTU and PEs, while any topology change needs to be propagated in the whole VPLS domain including mesh SDPs. This is done by using so-called MAC-flush messages defined by RFC 4762. In the case of STP as an loop resolution mechanism, every TCN received in the context of an STP instance is translated into an LDP-MAC address withdrawal message (also referred to as a MAC-flush message) requesting to clear all FDB entries except the ones

learned from the originating PE. Such messages are sent to all PE peers connected through SDPs (mesh and spoke) in the context of VPLS services, which are managed by the specified STP instance.

3.2.11.2 Redundant access to VPLS without STP

The Nokia implementation also includes alternative methods for providing a redundant access to Layer 2 services, such as MC-LAG, MC-APS, or MC-Ring. Also in this case, the topology change event needs to be propagated into the VPLS topology to provide fast convergence. The topology change propagation and its corresponding MAC flush processing in a VPLS service without STP is described in Dual homing to a VPLS service.

3.2.12 Object grouping and state monitoring

This feature introduces a generic operational group object that associates different service endpoints (pseudowires, SAPs, IP interfaces) located in the same or in different service instances.

The operational group status is derived from the status of the individual components using specific rules specific to the application using the feature. A number of other service entities, the monitoring objects, can be configured to monitor the operational group status and to perform specific actions as a result of status transitions. For example, if the operational group goes down, the monitoring objects are brought down.

3.2.12.1 VPLS applicability — block on VPLS a failure

This feature is used in VPLS to enhance the existing BGP MH solution by providing a block-on-group failure function similar to the block-on-mesh failure feature implemented for LDP VPLS. On the PE selected as the Designated Forwarder (DF), if the rest of the VPLS endpoints fail (pseudowire spokes/pseudowire mesh or SAPs, or both), there is no path forward for the frames sent to the MH site selected as DF. The status of the VPLS endpoints, other than the MH site, is reflected by bringing down/up the objects associated with the MH site.

Support for the feature is provided initially in VPLS and B-VPLS instance types for LDP VPLS, with or without BGP-AD and for BGP VPLS. The following objects may be placed as components of an operational group: BGP VPLS pseudowires, SAPs, spoke-pseudowire, BGP-AD pseudowires. The following objects are supported as monitoring objects: BGP MH site, individual SAP, spoke-pseudowire.

The following rules apply:

- An object can only belong to one group at a time.
- An object that is part of a group cannot monitor the status of any group.
- An object that monitors the status of a group cannot be part of any group.
- An operational group may contain any combination of member types: SAP, spoke-pseudowire, BGP-AD or BGP VPLS pseudowires.
- An operational group may contain members from different VPLS service instances.
- Objects from different services may monitor the operational group.
- The operational group feature may coexist in parallel with the block-on-mesh failure feature as long as they are running in different VPLS instances.

There are two steps involved in enabling the block-on-mesh failure feature in a VPLS scenario:

1. Identify a set of objects whose forwarding state should be considered as a whole group, then group them under an operational group using the **oper-group** CLI command.
2. Associate other existing objects (clients) with the **oper-group** command using the **monitor-group** CLI command; its forwarding state is derived from the related operational group state.

The status of the operational group (oper-group) is dictated by the status of one or more members according to the following rule:

- The oper-group goes down if all the objects in the oper-group go down; the oper-group comes up if at least one of the components is up.
- An object in the oper-group is considered down if it is not forwarding traffic in at least one direction. That could be because the operational state is down or the direction is blocked through some resiliency mechanisms.
- If an oper-group is configured but no members are specified yet, its status is considered up. As soon as the first object is configured, the status of the oper-group is dictated by the status of the provisioned members.
- For BGP-AD or BGP VPLS pseudowires associated with the oper-group (under the **config**>**service-vpls**>**bgp**>**pw-template-binding** context), the status of the oper-group is down as long as the pseudowire members are not instantiated (auto-discovered and signaled).

A simple configuration example is described for the case of a BGP VPLS mesh used to interconnect different customer locations. If we assume a customer edge (CE) device is dual-homed to two PEs using BGP MH, the following configuration steps apply:

1. The oper-group bgp-vpls-mesh is created.
2. The BGP VPLS mesh is added to the bgp-vpls-mesh group through the pseudowire template used to create the BGP VPLS mesh.
3. The BGP MH site defined for the access endpoint is associated with the bgp-vpls-mesh group; its status from now on is influenced by the status of the BGP VPLS mesh.

Below is a simple configuration example:

```
service>oper-group bgp-vpls-mesh-1 create
service>vpls>bgp>pw-template-binding> oper-group bgp-vpls-mesh-1
service>vpls>site> monitor-group bgp-vpls-mesh-1
```

3.2.13 MAC flush message processing

The previous sections described operating principles of several redundancy mechanisms available in the context of VPLS service. All of them rely on MAC flush messages as a tool to propagate topology change in a context of the specified VPLS. This section summarizes basic rules for generation and processing of these messages.

As described in respective sections, the 7450 ESS, 7750 SR, and 7950 XRS support two types of MAC flush message: flush-all-but-mine and flush-mine. The main difference between these messages is the type of action they signal. Flush-all-but-mine messages request clearing of all FDB entries that were learned from all other LDP peers except the originating PE. This type is also defined by RFC 4762 as an LDP MAC address withdrawal with an empty MAC address list.

Flush-all-mine messages request clearing all FDB entries learned from the originating PE. This means that this message has the opposite effect of the flush-all-but-mine message. This type is not included in the RFC 4762 definition and it is implemented using vendor-specific TLV.

The advantages and disadvantages of the individual types should be apparent from examples in the previous section. The description here summarizes actions taken on reception and the conditions under which individual messages are generated.

Upon reception of MAC flush messages (regardless of the type), an SR-series PE takes the following actions:

1. Clears FDB entries of all indicated VPLS services conforming to the definition.
2. Propagates the message (preserving the type) to all LDP peers, if the propagate-mac-flush flag is enabled at the corresponding VPLS level.

The flush-all-but-mine message is generated under the following conditions:

- The flush-all-but-mine message is received from the LDP peer and the propagate-mac-flush flag is enabled. The message is sent to all LDP peers in the context of the VPLS service it was received.
- The TCN message in a context of STP instance is received. The flush-all-but-mine message is sent to all LDP peers connected with spoke and mesh SDPs in a context of VPLS service controlled by the specified STP instance (based on M-VPLS definition). If all LDP peers are in the STP domain, that is, the M-VPLS and the uVPLS (user VPLS) both have the same topology, the router does not send any flush-all-but-mine message. If the router has uVPLS LDP peers outside the STP domain, the router sends flush-all-but-mine messages to all its uVPLS peers.

  Note: The 7750 SR does not send a withdrawal if the M-VPLS does not contain a mesh SDP. A mesh SDP must be configured in the M-VPLS to send withdrawals.

- The flush-all-but-mine message is generated when switchover between spoke-SDPs of the same endpoint occurs. The message is sent to the LDP peer connected through the newly active spoke-SDP.

The flush-mine message is generated under the following conditions:

- The flush-mine message is received from the LDP peer and the propagate-mac-flush flag is enabled. The message is sent to all LDP peers in the context of the VPLS service it was received.
- The flush-mine message is generated when a SAP or SDP transitions from operationally up to an operationally down state and the send-flush-on-failure flag is enabled in the context of the specified VPLS service. The message is sent to all LDP peers connected in the context of the specified VPLS service. The send-flush-on-failure flag is blocked in M-VPLS and is only allowed to be configured in a VPLS service managed by M-VPLS. This is to prevent both messages being sent at the same time.
- The flush-mine message is generated when an MC-LAG SAP or MC-APS SAP transitions from an operationally up state to an operationally down state. The message is sent to all LDP peers connected in the context of the specified VPLS service.
- The flush-mine message is generated when an MC-Ring SAP transitions from operationally up to an operationally down state or when MC-Ring SAP transitions to slave state. The message is sent to all LDP peers connected in the context of the specified VPLS service.

3.2.13.1 Dual homing to a VPLS service

Figure 72: Dual-homed CE connection to VPLS

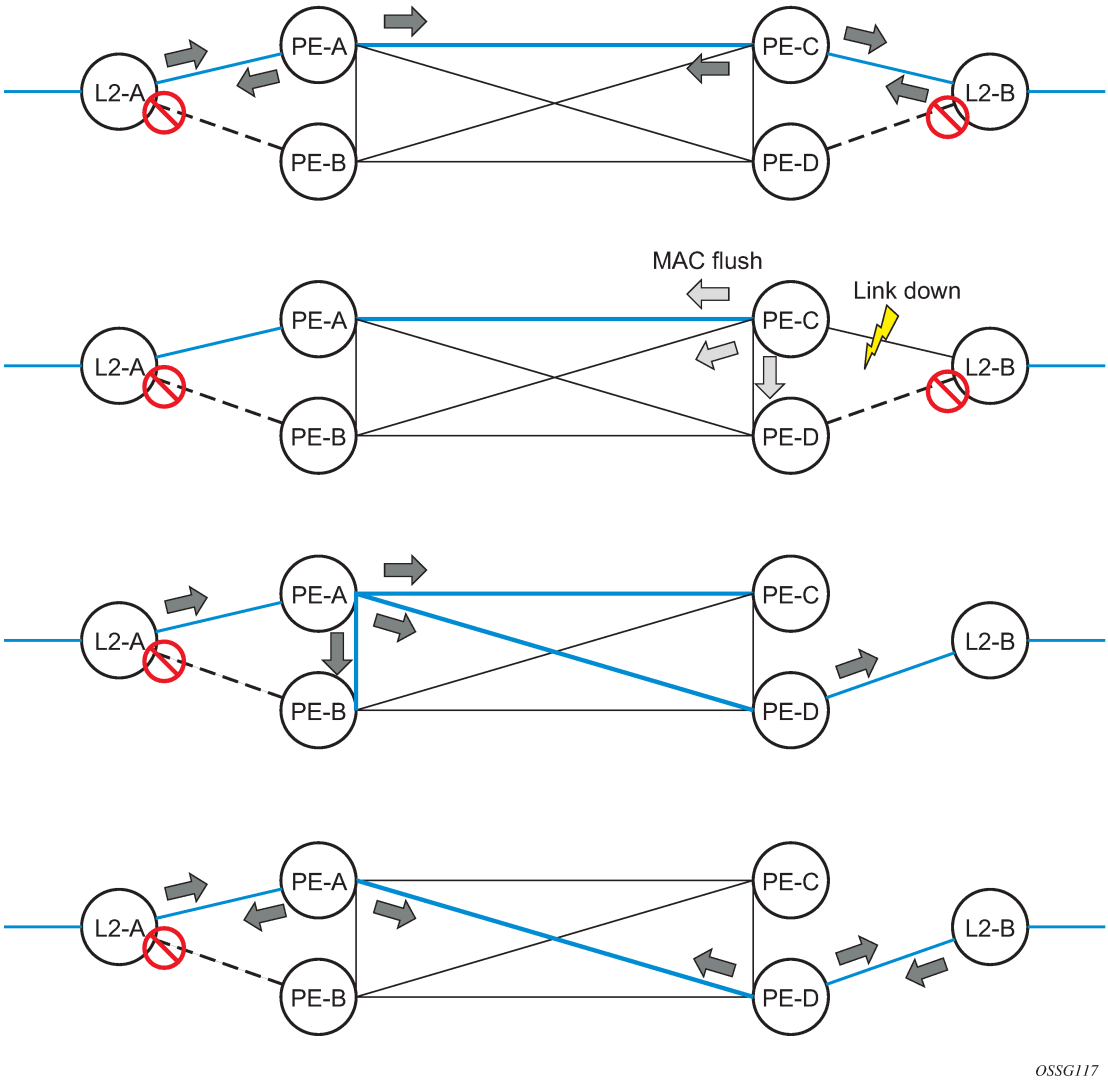


Figure 72: Dual-homed CE connection to VPLS shows a dual-homed connection to VPLS service (PE-A, PE-B, PE-C, PE-D) and operation in case of link failure (between PE-C and L2-B). Upon detection of a link failure, PE-C sends MAC-address-withdraw messages, which indicates to all LDP peers that they should flush all MAC addresses learned from PE-C. This leads to a broadcasting of packets addressing affected hosts and relearning in case an alternative route exists.

The message described here is different than the message described in RFC 4762, *Virtual Private LAN Services Using LDP Signaling*. The difference is in the interpretation and action performed in the receiving PE. According to the standard definition, upon receipt of a MAC withdraw message, all MAC addresses, except the ones learned from the source PE, are flushed. This section specifies that all MAC addresses learned from the source are flushed. This message has been implemented as an LDP address withdraw message with vendor-specific type, length, and value (TLV), and is called the flush-all-from-me message.

The RFC 4762 compliant message is used in VPLS services for recovering from failures in STP (Spanning Tree Protocol) topologies. The mechanism described in this section represents an alternative solution.

The advantage of this approach (as compared to STP-based methods) is that only the affected MAC addresses are flushed and not the full forwarding database. While this method does not provide a mechanism to secure alternative loop-free topology, the convergence time depends on the speed that the specified CE device opens an alternative link (L2-B switch in Figure 72: Dual-homed CE connection to VPLS) as well as on the speed that PE routers flush their FDB.

In addition, this mechanism is effective only if PE and CE are directly connected (no hub or bridge) as the mechanism reacts to the physical failure of the link.

3.2.13.2 MC-Ring and VPLS

The use of multi-chassis ring control in a combination with the plain VPLS SAP is supported by the FDB in individual ring nodes, in case the link (or ring node) failure cannot be cleared on the 7750 SR or 7950 XRS.

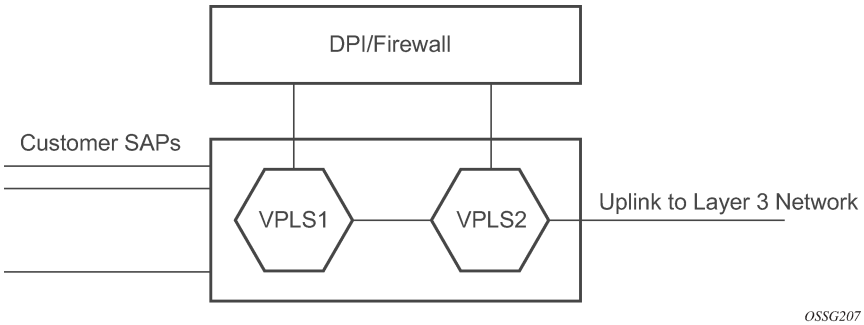
This combination is not easily blocked in the CLI. If configured, the combination may be functional but the switchover times are proportional to MAC aging in individual ring nodes or to the relearning rate, or both, because of the downstream traffic.

Redundant plain VPLS access in ring configurations, therefore, exclude corresponding SAPs from the multi-chassis ring operation. Configurations such as M-VPLS can be applied.

3.2.14 ACL next-hop for VPLS

The ACL next-hop for VPLS feature enables an ACL that has a forward to a SAP or SDP action specified to be used in a VPLS service to direct traffic with specific match criteria to a SAP or SDP. This allows traffic destined for the same gateway to be split and forwarded differently based on the ACL.

Figure 73: Application 1 diagram



Policy routing is a popular tool used to direct traffic in Layer 3 networks. As Layer 2 VPNs become more popular, especially in network aggregation, policy forwarding is required. Many providers are using methods such as DPI servers, transparent firewalls, or Intrusion Detection/Prevention Systems (IDS/IPS). Because these devices are bandwidth limited, providers want to limit traffic forwarded through them. In the setup shown in Figure 73: Application 1 diagram, a mechanism is required to direct some traffic coming from a SAP to the DPI without learning, and other traffic coming from the same SAP directly to the gateway uplink-based learning.

This feature allows the provider to create a filter that forwards packets to a specific SAP or SDP. The packets are then forwarded to the destination SAP regardless of learned destination. The SAP can either

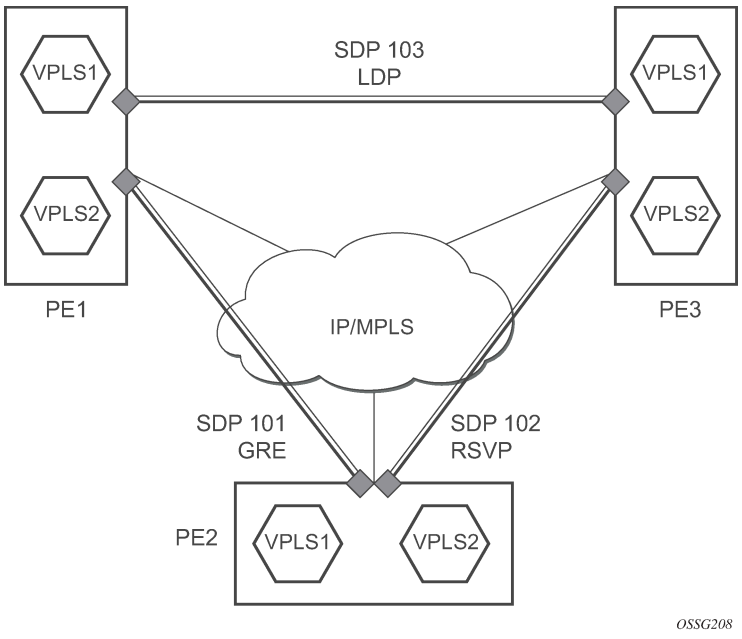
terminate a Layer 2 firewall, perform deep packet inspection (DPI) directly, or may be configured to be part of a cross-connect bridge into another service. This is useful when running the DPI remotely using VLLs. If an SDP is used, the provider can terminate it in a remote VPLS or VLL service where the firewall is connected. The filter can be configured under a SAP or SDP in a VPLS service. All packets (unicast, multicast, broadcast, and unknown) can be delivered to the destination SAP/SDP.

The filter may be associated with SAPs/SDPs belonging to a VPLS service only if all actions in the ACL forward to SAPs/SDPs that are within the context of that VPLS. Other services do not support this feature. An ACL that contains this feature is allowed, but the system drops any packet that matches an entry with this action.

3.2.15 SDP statistics for VPLS and VLL services

The simple three-node network in Figure 74: SDP statistics for VPLS and VLL services shows two MPLS SDPs and one GRE SDP defined between the nodes. These SDPs connect VPLS1 and VPLS2 instances that are defined in the three nodes. With this feature, the operator has local CLI-based as well as SNMP-based statistics collection for each VC used in the SDPs. This allows for traffic management of tunnel usage by the different services and with aggregation of the total tunnel usage.

Figure 74: SDP statistics for VPLS and VLL services



SDP statistics allow providers to bill customers on a per-SDP per-byte basis. This destination-based billing model can be used by providers with a variety of circuit types and have different costs associated with the circuits. An accounting file allows the collection of statistics in bulk.

3.2.16 BGP auto-discovery for LDP VPLS

BGP Auto-Discovery (BGP AD) for LDP VPLS is a framework for automatically discovering the endpoints of a Layer 2 VPN, offering an operational model similar to that of an IP VPN. This allows carriers to

leverage existing network elements and functions, including but not limited to, route reflectors and BGP policies to control the VPLS topology.

BGP AD complements an already established and well-deployed Layer 2 VPN signaling mechanism target LDP, providing one-touch provisioning for LDP VPLS, where all the related PEs are discovered automatically. The service provider may make use of existing BGP policies to regulate the exchanges between PEs in the same, or in different, autonomous system (AS) domains. The addition of BGP AD procedures does not require carriers to uproot their existing VPLS deployments nor to change the signaling protocol.

3.2.16.1 BGP AD overview

The BGP protocol establishes neighbor relationships between configured peers. An open message is sent after the completion of the three-way TCP handshake. This open message contains information about the BGP peer sending the message. This message contains Autonomous System Number (ASN), BGP version, timer information, and operational parameters, including capabilities. The capabilities of a peer are exchanged using two numerical values: the Address Family Identifier (AFI) and Subsequent Address Family Identifier (SAFI). These numbers are allocated by the Internet Assigned Numbers Authority (IANA). BGP AD uses AFI 65 (L2VPN) and SAFI 25 (BGP VPLS). For a complete list of allocations, see *http://www.iana.org/assignments/address-family-numbers* and SAFI *http://www.iana.org/assignments/safi-namespace*.

3.2.16.2 Information model

Following the establishment of the peer relationship, the discovery process begins as soon as a new VPLS service instance is provisioned on the PE.

Two VPLS identifiers are used to indicate the VPLS membership and the individual VPLS instance:

- **VPLS-ID**

  Membership information, and unique network-wide identifier; the same value is assigned for all VPLS switch instances (VSIs) belonging to the same VPLS. VPLS-ID is encodable and carried as a BGP extended community in one of the following formats:

  – A two-octet AS-specific extended community

  – An IPv4 address-specific extended community

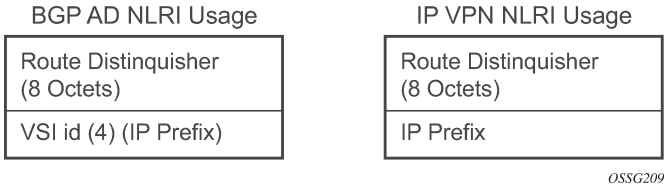
- **VSI-ID**

  This is the unique identifier for each individual VSI, built by concatenating a route distinguisher (RD) with a 4-byte identifier (usually the system IP of the VPLS PE), encoded and carried in the corresponding BGP NLRI.

To advertise this information, BGP AD employs a simplified version of the BGP VPLS NLRI where just the RD and the next four bytes are used to identify the VPLS instance. There is no need for Label Block and Label Size fields as T-LDP signals the service labels later on.

The format of the BGP AD NLRI is very similar to the one used for IP VPN, as shown in Figure 75: BGP AD NLRI versus IP VPN NLRI. The system IP may be used for the last four bytes of the VSI ID, further simplifying the addressing and the provisioning process.

Figure 75: BGP AD NLRI versus IP VPN NLRI



Network Layer Reachability Information (NLRI) is exchanged between BGP peers indicating how to reach prefixes. The NLRI is used in the Layer 2 VPN case to tell PE peers how to reach the VSI, instead of specific prefixes. The advertisement includes the BGP next hop and a route target (RT). The BGP next hop indicates the VSI location and is used in the next step to determine which signaling session is used for pseudowire signaling. The RT, also coded as an extended community, can be used to build a VPLS full mesh or an HVPLS hierarchy through the use of BGP import/export policies.

BGP is only used to discover VPN endpoints and the corresponding far-end PEs. It is not used to signal the pseudowire labels. This task remains the responsibility of targeted-LDP (T-LDP).

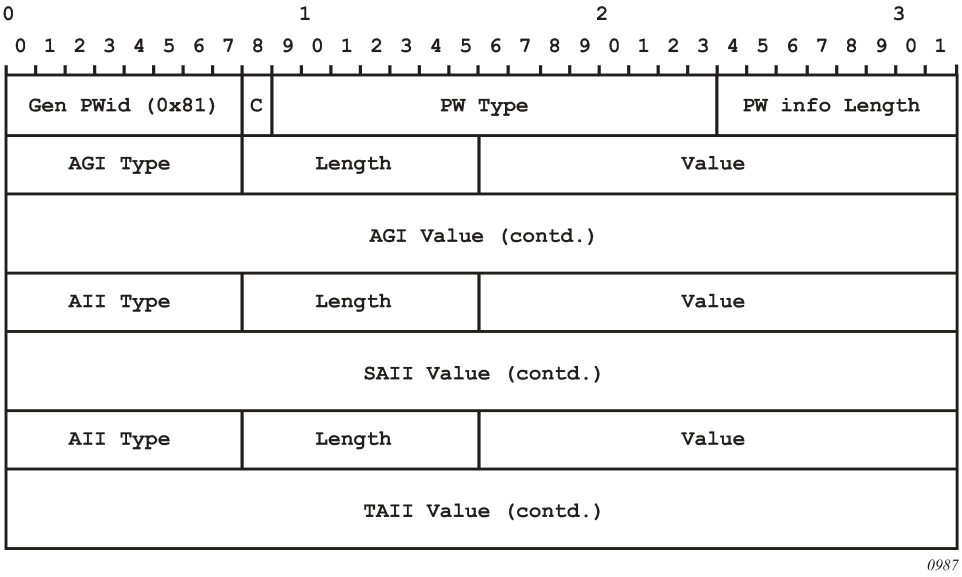
3.2.16.3 FEC element for T-LDP signaling

Two LDP FEC elements are defined in RFC 4447, *PW Setup & Maintenance Using LDP*. The original pseudowire-ID FEC element 128 (0x80) employs a 32-bit field to identify the virtual circuit ID and was used extensively in the initial VPWS and VPLS deployments. The simple format is easy to understand but does not provide the required information model for BGP auto-discovery function. To support BGP AD and other new applications, a new Layer 2 FEC element, the generalized FEC (0x81), is required.

The generalized pseudowire-ID FEC element has been designed for auto-discovery applications. It provides a field, the address group identifier (AGI), that is used to signal the membership information from the VPLS-ID. Separate address fields are provided for the source and target address associated with the VPLS endpoints, called the Source Attachment Individual Identifier (SAII) and Target Attachment Individual Identifier (TAII), respectively. These fields carry the VSI ID values for the two instances that are to be connected through the signaled pseudowire.

The detailed format for FEC 129 is shown in Figure 76: Generalized pseudowire-ID FEC element.

Figure 76: Generalized pseudowire-ID FEC element



Each of the FEC fields are designed as a sub-TLV equipped with its own type and length, providing support for new applications. To accommodate the BGP AD information model, the following FEC formats are used:

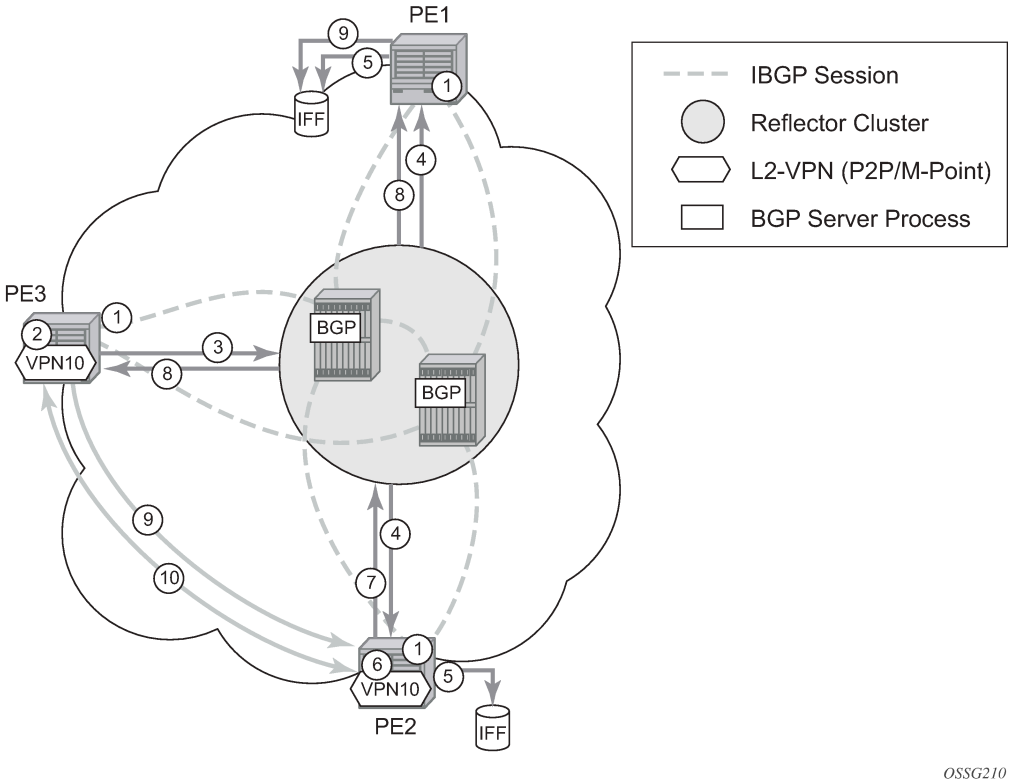
- AGI (type 1) is identical in format and content to the BGP extended community attribute used to carry the VPLS-ID value.
- Source AII (type 1) is a 4-byte value to carry the local VSI-ID (outgoing NLRI minus the RD).
- Target AII (type 1) is a 4-byte value to carry the remote VSI-ID (incoming NLRI minus the RD).

3.2.16.4 BGP-AD and target LDP (T-LDP) interaction

BGP is responsible for discovering the location of VSIs that share the same VPLS membership. LDP protocol is responsible for setting up the pseudowire infrastructure between the related VSIs by exchanging service-specific labels between them.

After the local VPLS information is provisioned in the local PE, the related PEs participating in the same VPLS are identified through BGP AD exchanges. A list of far-end PEs is generated and triggers the creation, if required, of the necessary T-LDP sessions to these PEs and the exchange of the service-specific VPN labels. The steps for the BGP AD discovery process and LDP session establishment and label exchange are shown in Figure 77: BGP-AD and T-LDP interaction.

Figure 77: BGP-AD and T-LDP interaction



The following corresponds with the actions in Figure 77: BGP-AD and T-LDP interaction:

1. Establish IBGP connectivity RR.
2. Configure VPN (10) on edge node (PE3).
3. Announce VPN to RR using BGP-AD.
4. Send membership update to each client of the cluster.
5. LDP exchange or inbound FEC filtering (IFF) of non-match or VPLS down.
6. Configure VPN (10) on edge node (PE2).
7. Announce VPN to RR using BGP-AD.
8. Send membership update to each client of the cluster.
9. LDP exchange or inbound FEC filtering (IFF) of non-match or VPLS down.
10. Complete LDP bidirectional pseudowire establishment FEC 129.

3.2.16.5 SDP usage

Service Access Points (SAPs) are linked to transport tunnels using Service Distribution Points (SDPs). The service architecture allows services to be abstracted from the transport network.

MPLS transport tunnels are signaled using the Resource Reservation Protocol (RSVP-TE) or by the Label Distribution Protocol (LDP). The capability to automatically create an SDP only exists for LDP-based

transport tunnels. Using a manually provisioned SDP is available for both RSVP-TE and LDP transport tunnels. For more information about MPLS, LDP and RSVP, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide*.

GRE transport tunnels use GRE encapsulation and can be used with manually provisioned or auto created SDPs.

3.2.16.6 Automatic creation of SDPs

When BGP AD is used for LDP VPLS, with an LDP or GRE transport tunnel, there is no requirement to manually create an SDP. The LDP or GRE SDP can be automatically instantiated using the information advertised by BGP AD. This simplifies the configuration on the service node.

The use of an automatically created GRE tunnel is enabled by creating the PW template used within the service with the parameter **auto-gre-sdp**. The GRE SDP and SDP binding is created after a matching BGP route has been received.

Enabling LDP on the IP interfaces connecting all nodes between the ingress and the egress builds transport tunnels based on the best IGP path. LDP bindings are automatically built and stored in the hardware. These entries contain an MPLS label pointing to the best next hop along the best path toward the destination.

When two endpoints need to connect and no SDP exists, a new SDP is automatically constructed. New services added between two endpoints that already have an automatically created SDP are immediately used; no new SDP is constructed. The far-end information is learned from the BGP next hop information in the NLRI. When services are withdrawn with a BGP_Unreach-NLRI, the automatically established SDP remains up while at least one service is connected between those endpoints. An automatically created SDP is removed and the resources are released when the only or last service is removed.

The service provider has the option of associating the auto-discovered SDP with a split horizon group using the pw-template-binding option, to control the forwarding between pseudowires and to prevent Layer 2 service loops.

An auto-discovered SDP using a pw-template-binding without a split horizon group configured has similar traffic flooding behavior as a spoke-SDP.

3.2.16.7 Manually provisioned SDP

The carrier is required to manually provision the SDP if they create transport tunnels using RSVP-TE. Operators have the option to choose a manually configured SDP if they use LDP as the tunnel signaling protocol. The functionality is the same regardless of the signaling protocol.

Creating a BGP AD-enabled VPLS service on an ingress node with the manually provisioned SDP option causes the tunnel manager to search for an existing SDP that connects to the far-end PE. The far-end IP information is learned from the BGP next hop information in the NLRI. If a single SDP exists to that PE, it is used. If no SDP is established between the two endpoints, the service remains down until a manually configured SDP becomes active.

When multiple SDPs exist between two endpoints, the tunnel manager selects the appropriate SDP. The algorithm prefers SDPs with the best (lower) metric. If there are multiple SDPs with equal metrics, the operational state of the SDPs with the best metric is considered. If the operational state is the same, the SDP with the higher SDP-ID is used. If an SDP with a preferred metric is found with an operational state that is not active, the tunnel manager flags it as ineligible and restarts the algorithm.

3.2.16.8 Automatic instantiation of pseudowires (SDP bindings)

The choice of manual or auto-provisioned SDPs has limited impact on the amount of required provisioning. Most of the savings are achieved through the automatic instantiation of the pseudowire infrastructure (SDP bindings). This is achieved for every auto-discovered VSI through the use of the pseudowire template concept. Each VPLS service that uses BGP AD contains the pw-template-binding option defining specific Layer 2 VPN parameters. This command references a PW template, which defines the pseudowire parameters. The same PW template may be referenced by multiple VPLS services. As a result, changes to these pseudowire templates have to be treated with caution as they may impact many customers simultaneously.

The Nokia implementation provides for safe handling of pseudowire templates. Changes to the pseudowire templates are not automatically propagated. Tools are provided to evaluate and distribute the changes. The following command is used to distribute changes to a PW template at the service level to one or all services that use that template:

PERs-4# tools perform service id 300 eval-pw-template 1 allow-service-impact

If the service ID is omitted, all services are updated. The type of change made to the PW template influences how the service is impacted:

- Adding or removing a **split-horizon-group** causes the router to destroy the original object and re-create it using the new value.
- Changing parameters in the **vc-type** {**ether | vlan**} command requires LDP to re-signal the labels.

Both of these changes are service affecting. Other changes are not service affecting.

3.2.16.9 Mixing statically configured and auto-discovered pseudowires in a VPLS

The services implementation allows for manually provisioned and auto-discovered pseudowire (SDP bindings) to coexist in the same VPLS instance (for example, both FEC128 and FEC 129 are supported). This allows for gradual introduction of auto-discovery into an existing VPLS deployment.

As FEC 128 and 129 represent different addressing schemes, it is important to ensure that only one is used at any time between the same two VPLS instances. Otherwise, both pseudowires may become active causing a loop that may adversely impact the correct functioning of the service. It is recommended that FEC128 pseudowire be disabled as soon as the FEC129 addressing scheme is introduced in a portion of the network. Alternatively, RSTP may be used during the migration as a safety mechanism to provide additional protection against operational errors.

3.2.16.10 Resiliency schemes

The use of BGP AD on the network side, or in the backbone, does not affect the different resiliency schemes Nokia has developed in the access network. This means that both Multi-Chassis Link Aggregation (MC-LAG) and Management-VPLS (M-VPLS) can still be used.

BGP AD may coexist with Hierarchical-VPLS (H-VPLS) resiliency schemes (for example, dual-homed MTUs devices to different PE-rs nodes) using existing methods (M-VPLS and statically configured active/standby pseudowire endpoint).

If provisioned SDPs are used by BGP AD, M-VPLS may be employed to provide loop avoidance. However, it is currently not possible to auto-discover active/standby pseudowires and to instantiate the related endpoint.

3.2.17 BGP VPLS

The Nokia BGP VPLS solution, compliant with RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*, is described in this section.

Figure 78: BGP VPLS solution

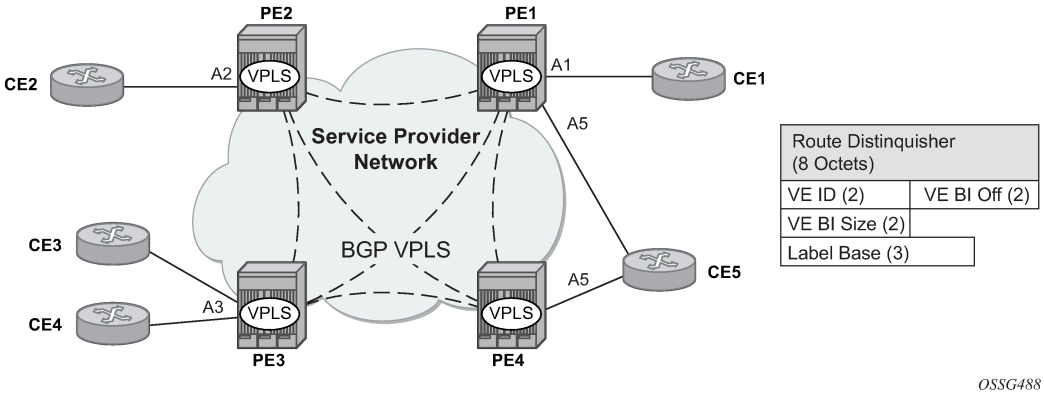


Figure 78: BGP VPLS solution shows the service representation for BGP VPLS mesh. The major BGP VPLS components and the deltas from LDP VPLS with BGP AD are as follows:

- Data plane is identical with the LDP VPLS solution; for example, VPLS instances interconnected by pseudowire mesh. Split horizon groups may be used for loop avoidance between pseudowires.
- Addressing is based on a 2-byte VE-ID assigned to the VPLS instance.

  BGP-AD for LDP VPLS: 4-byte VSI-ID (system IP) identifies the VPLS instance.
- The target VPLS instance is identified by the Route Target (RT) contained in the MP-BGP advertisement (extended community attribute).

  BGP-AD: a new MP-BGP extended community is used to identify the VPLS. RT is used for topology control.
- Auto-discovery is MP-BGP based; the same AFI, SAFI is used as for LDP VPLS BGP-AD.
  - The BGP VPLS updates are distinguished from the BGP-AD updates based on the value of the NLRI prefix length: 17 bytes for BGP VPLS, 12 bytes for BGP-AD.
  - BGP-AD NLRI is shorter because there is no need to carry pseudowire label information as T-LDP does the pseudowire signaling for LDP VPLS.
- Pseudowire label signaling is MP-BGP based. Therefore, the BGP NLRI content also includes label-related information; for example, block offset, block size, and label base.
  - LDP VPLS: target LDP (T-LDP) is used for signaling the pseudowire service label.
  - The Layer 2 extended community proposed in RFC 4761 is used to signal pseudowire characteristics; for example, VPLS status, control word, and sequencing.

3.2.17.1 Pseudowire signaling details

The pseudowire is set up using the following NLRI fields:

- **VE Block offset (VBO)**

  This is used to define each VE-ID set for which the NLRI is targeted.

  – VBO = n*VBS+1; for VBS = 8, this results in 1, 9, 17, 25, …

  – Targeted Remote VE-IDs are from VBO to (VBO + VBS - 1)

- **VE Block size (VBS)**

  Defines how many contiguous pseudowire labels are reserved, starting with the Label Base; Nokia implementation always uses a value of eight (8).

- **Label Base (LB)**

  This is the local allocated label base. The next eight consecutive labels available are allocated for remote PEs.

This BGP update is telling the other PEs that accept the RT: reach me (VE-ID = x), use a pseudowire label of LB + VE-ID – VBO using the BGP NLRI for which VBO =< local VE-ID < VBO + VBS.

Following is an example of how this algorithm works, assuming PE1 has VE-ID 7 configured:

1. PE1 allocates a label block of eight consecutive labels available, starting with LB = 1000.
2. PE1 starts sending a BGP update with pseudowire information of VBO = 1, VBS = 8, LB = 1000 in the NLRI.
3. This pseudowire information is accepted by all participating PEs with VE-IDs from 1 to 8.
4. Each of the receiving PEs uses the pseudowire label = LB + VE-ID - VBO to send traffic back to the originator PE. For example, VE-ID 2 uses pseudowire label 1001.

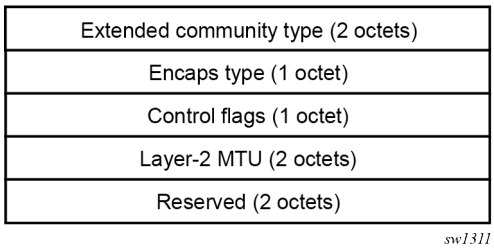
Assuming that VE-ID = 10 is configured in another PE4, the following procedure applies:

1. PE4 sends a BGP update with the new VE-ID in the network that is received by all the other participating PEs, including PE1.
2. Upon reception, PE1 generates another label block of 8 labels for the VBO = 9. For example, the initial PE creates new pseudowire signaling information of VBO = 9, VBS = 8, LB = 3000, and insert it in a new NLRI and BGP update that is sent in the network.
3. This new NLRI is used by the VE-IDs from 9 to 16 to establish pseudowires back to the originator PE1. For example, PE4 with VE-ID 10 uses pseudowire label 3001 to send VPLS traffic back to PE1.
4. The PEs owning the set of VE-IDs from 1 to 8 ignore this NLRI.

In addition to the pseudowire label information, the "Layer2 Info Extended Community" attribute must be included in the BGP update for BGP VPLS to signal the attributes of all the pseudowires that converge toward the originator VPLS PE.

The format is as follows.

Figure 79: Layer-2 information extended community



The meaning of the fields are as follows:

- **extended community type**

  This is the value allocated by IANA for this attribute is 0x800A.

- **encaps type**

  Encapsulation type identifies the type of pseudowire encapsulation. The only value used by BGP VPLS is 19 (13 in HEX). This value identifies the encapsulation to be used for pseudowire instantiated through BGP signaling, which is the same as the one used for Ethernet pseudowire type in regular VPLS. There is no support for an equivalent Ethernet VLAN pseudowire in BGP VPLS in BGP signaling.

- **control flags**

  This field is control information concerning the pseudowires (see Figure 78: BGP VPLS solution).

- **Layer 2 MTU**

  This is the Maximum Transmission Unit to be used on the pseudowires

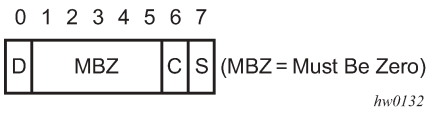
- **reserved**

  This field is reserved and must be set to zero and ignored on reception except where it is used for VPLS preference.

  For inter-AS, the preference information must be propagated between autonomous systems. Consequently, as the VPLS preference in a BGP-VPLS or BGP multihoming update extended community is zero, the local preference is copied by the egress ASBR into the VPLS preference field before sending the update to the EBGP peer. The adjacent ingress ASBR then copies the received VPLS preference into the local preference to prevent the update being considered malformed.

Figure 80: Control flag bit vector format shows the detailed format for the control flags bit vector.

Figure 80: Control flag bit vector format



The following bits in the control flags are defined as follows:

S	sequenced delivery of frames that must or must not be used when sending VPLS packets to this PE, depending on whether S is 1 or 0, respectively
C	a Control word that must or must not be present when sending VPLS packets to this PE, depending on whether C is 1 or 0, respectively. By default, Nokia implementation uses value 0

MBZ	Must Be Zero bits, set to zero when sending and ignored when receiving
D	indicates the status of the whole VPLS instance (VSI); D = 0 if Admin and Operational status are up, D = 1 otherwise

Following are the events that set the D-bit to 1 to indicate VSI down status in the BGP update message sent out from a PE:

- Local VSI is shut down administratively using the **config service vpls shutdown** command.
- All the related endpoints (SAPs or LDP pseudowires) are down.
- There are no related endpoints (SAPs or LDP pseudowires) configured yet in the VSI.

  The intent is to save the core bandwidth by not establishing the BGP pseudowires to an empty VSI.
- Upon reception of a BGP update message with D-bit set to 1, all the receiving VPLS PEs must mark related pseudowires as down.

The following events do not set the D-bit to 1:

- The local VSI is deleted; a BGP update with UNREACH-NLRI is sent out. Upon reception, all remote VPLS PEs must remove the related pseudowires and BGP routes.
- If the local SDP goes down, only the BGP pseudowires mapped to that SDP go down. There is no BGP update sent.

The **adv-service-mtu** command can be used to override the MTU value used in BGP signaling to the far-end of the pseudowire. This value is also used to validate the value signaled by the far-end PE unless **ignore-l2vpn-mtu-mismatch** is also configured.

If the **ignore-l2vpn-mtu-mismatch** command is configured, the router does not check the value of the "Layer 2 MTU" in the "Layer2 Info Extended Community" received in a BGP update message against the local service MTU, or against the MTU value signaled by this router. The router brings up the BGP-VPLS service regardless of any MTU mismatch.

3.2.17.2 Supported VPLS features

BGP VPLS includes support for a new type of pseudowire signaling based on MP-BGP, based on the existing VPLS instance; therefore, it inherits all the existing Ethernet switching functions.

The use of an automatically created GRE tunnel is enabled by creating the PW template used within the service with the parameter **auto-gre-sdp**. The GRE SDP and SDP binding is created after a matching BGP route has been received.

Following are some of the most important VPLS features ported to BGP VPLS:

- VPLS data plane features: for example, FDB management, SAPs, LAG access, and BUM rate limiting
- MPLS tunneling: LDP, LDP over RSVP-TE, RSVP-TE, GRE, and MP-BGP based on RFC 8277 (Inter-AS option C solution)

  Note: Pre-provisioned SDPs must be configured when RSVP-signaled transport tunnels are used.

- HVPLS topologies, hub and spoke traffic distribution
- Coexists with LDP VPLS (with or without BGP-AD) in the same VPLS instance:

  LDP and BGP-signaling should operate in disjoint domains to simplify loop avoidance.

- Coexists with BGP-based multihoming
- BGP VPLS is supported as the control plane for B-VPLS
- Supports IGMP/PIM snooping for IPv4
- Support for High Availability is provided
- Ethernet Service OAM toolset is supported: IEEE 802.1ag, Y.1731.

  Not supported OAM features: CPE Ping, MAC trace/ping/populate/purge
- Support for RSVP and LSP P2MP LSP for VPLS/B-VPLS BUM

3.2.18 VCCV BFD support for VPLS services

The SR OS supports RFC 5885, which specifies a method for carrying BFD in a pseudowire associated channel. For general information about VCCV BFD, limitations, and configuring, see the VLL Services chapter.

VCCV BFD is supported on the following VPLS services:

- T-LDP spoke-SDP termination on VPLS (including I-VPLS, B-VPLS, and R-VPLS)
- H-VPLS spoke-SDP
- BGP VPLS
- VPLS with BGP auto-discovery

To configure VCCV BFD for H-VPLS (where the pseudowire template does not apply), configure the BFD template using the **configure service vpls spoke-sdp bfd-template name** command, then enable it using the **configure service vpls spoke-sdp bfd-enable** command.

For BGP VPLS, a BFD template is referenced from the pseudowire template binding context. To configure VCCV BFD for BGP VPLS, use the **configure service vpls bgp pw-template-binding bfd-template name** command and enable it using the **configure service vpls bgp pw-template-binding bfd-enable** command.

For BGP-AD VPLS, a BFD template is referenced from the pseudowire template context. To configure VCCV BFD for BGP-AD, use the **configure service vpls bgp-ad pw-template-binding bfd-template name** command, and enable it using the **configure service vpls bgp-ad pw-template-binding bfd-enable** command.

3.2.19 BGP multihoming for VPLS

This section describes BGP-based procedures for electing a designated forwarder among the set of PEs that are multihomed to a customer site. Only the local PEs are actively participating in the selection algorithm. The PEs remote from the dual-homed CE are not required to participate in the designated forwarding election for a remote dual-homed CE.

The main components of the BGP-based multihoming solution for VPLS are:

- Provisioning model
- MP-BGP procedures
- Designated Forwarder Election

- Blackhole avoidance, indicating the designated forwarder change toward the core PEs and access PEs or CEs
- The interaction with pseudowire signaling (BGP/LDP)

Figure 81: BGP multihoming for VPLS

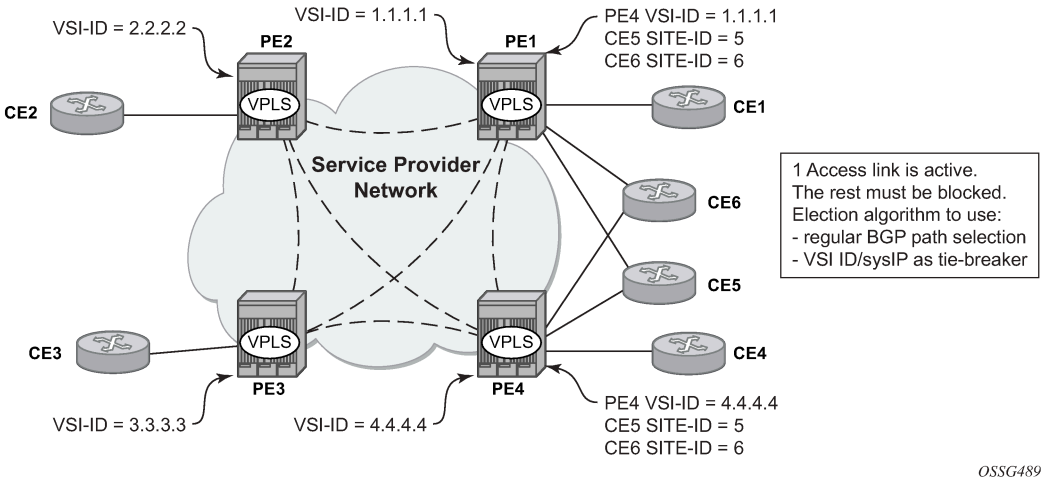


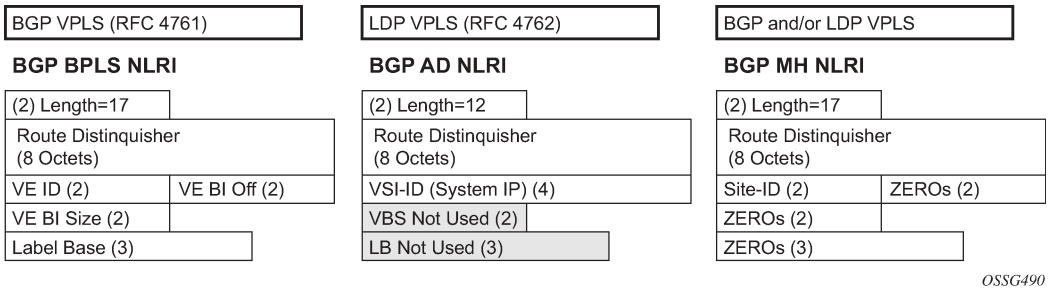
Figure 81: BGP multihoming for VPLS shows the VPLS using BGP multihoming for the case of multihomed CEs. Although the figure shows the case of a pseudowire infrastructure signaled with LDP for an LDP VPLS using BGP-AD for discovery, the procedures are identical for BGP VPLS or for a mix of BGP- and LDP-signaled pseudowires.

3.2.19.1 Information model and required extensions to L2VPN NLRI

VPLS Multihoming using BGP-MP expands on the BGP AD and BGP VPLS provisioning model. The addressing for the multihomed site is still independent from the addressing for the base VSI (VSI-ID or, respectively, VE-ID). Every multihomed CE is represented in the VPLS context through a site-ID, which is the same on the local PEs. The site-ID is unique within the scope of a VPLS. It serves to differentiate between the multihomed CEs connected to the same VPLS Instance (VSI). For example, in Figure 82: BGP MH-NLRI for VPLS multihoming, CE5 is assigned the same site-ID on both PE1 and PE4. For the same VPLS instance, different site-IDs are assigned for multihomed CE5 and CE6; for example, site-ID 5 is assigned for CE5 and site-ID 6 is assigned for CE6. The single-homed CEs (CE1, 2, 3, and 4) do not require allocation of a multihomed site-ID. They are associated with the addressing for the base VSI, either VSI-ID or VE-ID.

The new information model required changes to the BGP usage of the NLRI for VPLS. The extended MH NLRI for Multi-Homed VPLS is compared with the BGP AD and BGP VPLS NLRIs in Figure 82: BGP MH-NLRI for VPLS multihoming.

Figure 82: BGP MH-NLRI for VPLS multihoming



The BGP VPLS NLRI described in RFC 4761 is used to carry a 2-byte site-ID that identifies the MH site. The last seven bytes of the BGP VPLS NLRI used to instantiate the pseudowire are not used for BGP-MH and are zeroed out. This NLRI format translates into the following processing path in the receiving VPLS PE:

- BGP VPLS PE: no label information means there is no need to set up a BGP pseudowire.
- BGP AD for LDP VPLS: length =17 indicates a BGP VPLS NLRI that does not require any pseudowire LDP signaling.

The processing procedures described in this section start from the above identification of the BGP update as not destined for pseudowire signaling.

The RD ensures that the NLRIs associated with a specific site-ID on different PEs are seen as different by any of the intermediate BGP nodes (RRs) on the path between the multihomed PEs. That is, different RDs must be used on the MH PEs every time an RR or an ASBR is involved to guarantee the MH NLRIs reach the PEs involved in VPLS MH.

The L2-Info extended community from RFC 4761 is used in the BGP update for MH NLRI to initiate a MAC flush for blackhole avoidance, to indicate the operational and admin status for the MH site or the DF election status.

After the pseudowire infrastructure between VSIs is built using either RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*, or RFC 4761 procedures, or a mix of pseudowire signaling procedures, on activation of a multihomed site, an election algorithm must be run on the local and remote PEs to determine which site is the designated forwarder (DF). The end result is that all the related MH sites in a VPLS are placed in standby except for the site selected as DF. Nokia BGP-based multihoming solution uses the DF election procedure described in the IETF working group document *draft-ietf-bess-vpls-multihoming-01*. The implementation allows the use of BGP local preference and the received VPLS preference but does not support setting the VPLS preference to a non-zero value.

3.2.19.2 Supported services and multihoming objects

This feature is supported for the following services:

- LDP VPLS with or without BGP-AD
- BGP VPLS (BGP multihoming for inter-AS BGP-VPLS services is not supported)
- mix of the above
- PBB B-VPLS on BCB
- PBB I-VPLS (see IEEE 802.1ah Provider Backbone Bridging for more information)

The following access objects can be associated with MH Site:

- SAPs
- SDP bindings (pseudowire object), both mesh-SDP and spoke-SDP
- Split Horizon Group

  Under the SHG we can associate either one or multiple of the following objects: SAPs, pseudowires (BGP VPLS, BGP-AD, provisioned and LDP-signaled spoke-SDP and mesh-SDP)

3.2.19.3 Blackhole avoidance

Blackholing refers to the forwarding of frames to a PE that is no longer carrying the designated forwarder. This could happen for traffic from:

- Core PE participating in the main VPLS
- Customer Edge devices (CEs)
- Access PEs (pseudowires between them and the MH PEs are associated with MH sites)

Changes in DF election results or MH site status must be detected by all of the above network elements to provide for Blackhole Avoidance.

3.2.19.3.1 MAC flush to the core PEs

Assuming that there is a transition of the existing DF to non-DF status, the PE that owns the MH site experiencing this transition generates a MAC flush-all-from-me (negative MAC flush) toward the related core PEs. Upon reception, the remote PEs flush all the MACs learned from the MH PE.

MAC flush-all-from-me indication message is sent using the following core mechanisms:

- For LDP VPLS running between core PEs, existing LDP MAC flush is used.
- For pseudowire signaled with BGP VPLS, MAC flush is provided implicitly using the L2-Info Extended community to indicate a transition of the active MH site; for example, the attached objects going down or more generically, the entire site going from Designated Forwarder (DF) to non-DF.
- Double flushing does not happen as it is expected that between any pair of PEs, there exists only one type of pseudowires, either BGP or LDP pseudowire, but not both.

3.2.19.3.2 Indicating non-DF status toward the access PE or CE

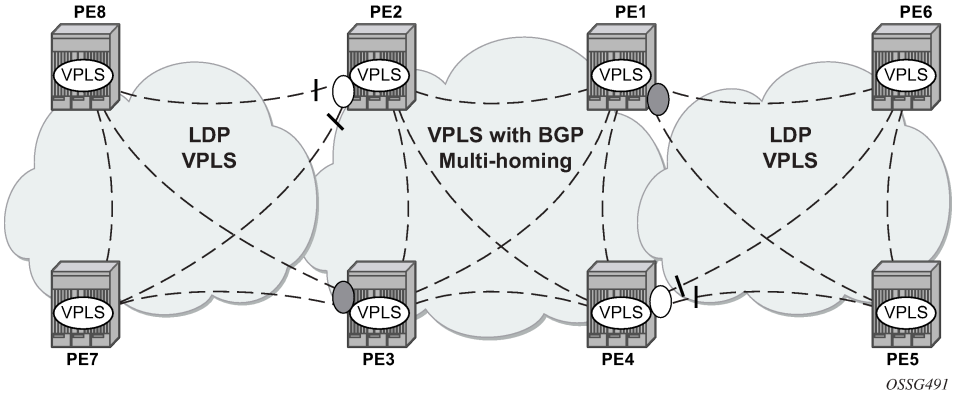
For the CEs or access PEs, support is provided for indicating the blocking of the MH site using the following procedures:

- For MH Access PE running LDP pseudowires, the LDP standby-status is sent to all LDP pseudowires.
- For MH CEs, site deactivation is linked to a CCM failure on a SAP that has a down MEP configured.

3.2.19.4 BGP multihoming for VPLS inter-domain resiliency

BGP MH for VPLS can be used to provide resiliency between different VPLS domains. An example of a multihoming topology is shown in Figure 83: BGP MH used in an HVPLS topology.

Figure 83: BGP MH used in an HVPLS topology



LDP VPLS domains are interconnected using a core VPLS domain, either BGP VPLS or LDP VPLS. The gateway PEs, for example PE2 and PE3, are running BGP multihoming where one MH site is assigned to each of the pseudowires connecting the access PE, PE7, and PE8 in this example.

Alternatively, the MH site can be associated with multiple access pseudowires using an access SHG. The **configure service vpls site failed-threshold** command can be used to indicate the number of pseudowire failures that are required for the MH site to be declared down.

3.2.20 Multicast-aware VPLS

VPLS is a Layer 2 service; therefore, multicast and broadcast frames are normally flooded in a VPLS. Broadcast frames are targeted to all receivers. However, for IP multicast, normally for a multicast group, only some receivers in the VPLS are interested. Flooding to all sites can cause wasted network bandwidth and unnecessary replication on the ingress PE router.

To avoid this condition, VPLS is IP multicast-aware; therefore, it forwards IP multicast traffic based on multicast states to the object on which the IP multicast traffic is requested. This is achieved by enabling the following related IP multicast protocol snooping:

- IGMP snooping
- MLD snooping
- PIM snooping

3.2.20.1 IGMP snooping for VPLS

When IGMP snooping is enabled in a VPLS service, IGMP messages received on SAPs and SDPs are snooped to determine the scope of the flooding for a specified stream or (S,G). IGMP snooping operates in a proxy mode, where the system summarizes upstream IGMP reports and responds to downstream queries. For a description of IGMP snooping, see the *7450 ESS, 7750 SR, and VSR Triple Play Service Delivery Architecture Guide*, "*IGMP Snooping*".

Streams are sent to all SAPs and SDPs on which there is a multicast router (either discovered dynamically from received query messages or configured statically using the **mrouter-port** command) and on which an active join for that stream has been received. The Mrouter port configuration adds a (*,*) entry into the MFIB, which causes all groups (and IGMP messages) to be sent out of the respective object and causes IGMP messages received on that object to be discarded.

Directly-connected multicast sources are supported when IGMP snooping is enabled.

IGMP snooping is enabled at the service level.

IGMP is not supported in the following:

- B-VPLS, routed I-VPLS, PBB-VPLS services
- a router configured with **enable-inter-as-vpn** or **enable-rr-vpn-forwarding**
- the following forms of default SAP:
  - *
  - *.null
  - *.*
- a VPLS service configured with a connection profile VLAN SAP

3.2.20.2 MLD snooping for VPLS

MLD snooping is an IPv6 version of IGMP snooping. The guidelines and procedures are similar to IGMP snooping as previously described. However, MLD snooping uses MAC-based forwarding. See MAC-based IPv6 multicast forwarding for more information. Directly connected multicast sources are supported when MLD snooping is enabled.

MLD snooping is enabled at the service level and is not supported in the following services:

- B-VPLS
- Routed I-VPLS
- EVPN-MPLS services
- PBB-EVPN services

MLD snooping is not supported under the following forms of default SAP:

- *
- *.null
- *.*

MLD snooping is not supported in a VPLS service configured with a connection profile VLAN SAP.

3.2.20.3 PIM snooping for VPLS

PIM snooping for VPLS allows a VPLS PE router to build multicast states by snooping PIM protocol packets that are sent over the VPLS. The VPLS PE then forwards multicast traffic based on the multicast states. When all receivers in a VPLS are IP multicast routers running PIM, multicast forwarding in the VPLS is efficient when PIM snooping for VPLS is enabled.

Because of PIM join/prune suppression, to make PIM snooping operate over VPLS pseudowires, two options are available: plain PIM snooping and PIM proxy. PIM proxy is the default behavior when PIM snooping is enabled for a VPLS.

PIM snooping is supported for both IPv4 and IPv6 multicast by default and can be configured to use SG-based forwarding (see IPv6 multicast forwarding for more information).

Directly connected multicast sources are supported when PIM snooping is enabled.

The following restrictions apply to PIM snooping:

- PIM snooping for IPv4 and IPv6 is not supported:
  - in the following services:
    - PBB B-VPLS
    - R-VPLS (including I-VPLS and BGP EVPN)
    - PBB-EVPN B-VPLS
    - EVPN-VXLAN R-VPLS
  - on a router configured with **enable-inter-as-vpn** or **enable-rr-vpn-forwarding**
  - under the following forms of default SAP:
    - *
    - *.null
    - *.*
  - in a VPLS service configured with a connection profile VLAN SAP
  - with connected SR OSs configured with **improved-assert**
  - with subscriber management in the VPLS service
  - as a mechanism to drive MCAC
- PIM snooping for IPv6 is not supported:
  - in the following services:
    - PBB I-VPLS
    - BGP-VPLS
    - BGP EVPN (including PBB-EVPN)
    - VPLS E-Tree
    - Management VPLS
  - with the configuration of MLD snooping

3.2.20.3.1 Plain PIM snooping

In a plain PIM snooping configuration, VPLS PE routers only snoop; PIM messages are generated on their own. Join/prune suppression must be disabled on CE routers.

When plain PIM snooping is configured, if a VPLS PE router detects a condition where join/prune suppression is not disabled on one or more CE routers, the PE router puts PIM snooping into the PIM proxy state. A trap is generated that reports the condition to the operator and is logged to the syslog. If the condition changes, for example, join/prune suppression is disabled on CE routers, the PE reverts to the plain PIM snooping state. A trap is generated and is logged to the syslog.

3.2.20.3.2 PIM proxy

For PIM proxy configurations, VPLS PE routers perform the following:

- snoop hellos and flood hellos in the fast data path

- consume join/prune messages from CE routers
- generate join/prune messages upstream using the IP address of one of the downstream CE routers
- run an upstream PIM state machine to determine whether a join/prune message should be sent upstream

Join/prune suppression is not required to be disabled on CE routers, but it requires all PEs in the VPLS to have PIM proxy enabled. Otherwise, CEs behind the PEs that do not have PIM proxy enabled may not be able to get multicast traffic that they are interested in if they have join/prune suppression enabled.

When PIM proxy is enabled, if a VPLS PE router detects a condition where join/prune suppression is disabled on all CE routers, the PE router puts PIM proxy into a plain PIM snooping state to improve efficiency. A trap is generated to report the scenario to the operator and is logged to the syslog. If the condition changes, for example, join/prune suppression is enabled on a CE router, PIM proxy is placed back into the operational state. Again, a trap is generated to report the condition to the operator and is logged to the syslog.

3.2.20.4 IPv6 multicast forwarding

When MLD snooping or PIM snooping for IPv6 is enabled, the forwarding of IPv6 multicast traffic is MAC-based; see MAC-based IPv6 multicast forwarding for more information.

The operation with PIM snooping for IPv6 can be changed to SG-based forwarding; see SG-based IPv6 multicast forwarding for more information.

The following command configures the IPv6 multicast forwarding mode with the default being **mac-based**:

configure service vpls mcast-ipv6-snooping-scope {sg-based | mac-based}

The forwarding mode can only be changed when PIM snooping for IPv6 is disabled.

3.2.20.4.1 MAC-based IPv6 multicast forwarding

This section describes IPv6 multicast address to MAC address mapping and IPv6 multicast forwarding entries.

For IPv6 multicast address to MAC address mapping, Ethernet MAC addresses in the range of 33-33-00-00-00-00 to 33-33-FF-FF-FF-FF are reserved for IPv6 multicast. To map an IPv6 multicast address to a MAC-layer multicast address, the low-order 32 bits of the IPv6 multicast address are mapped directly to the low-order 32 bits in the MAC-layer multicast address.

For IPv6 multicast forwarding entries, IPv6 multicast snooping forwarding entries are based on MAC addresses, while native IPv6 multicast forwarding entries are based on IPv6 addresses. When both MLD snooping or PIM snooping for IPv6 and native IPv6 multicast are enabled on the same device, both types of forwarding entries are supported on the same forward plane, although they are used for different services.

The following output shows a service with PIM snooping for IPv6 that has received joins for two multicast groups from different sources. As the forwarding mode is MAC-based, there is a single MFIB entry created to forward these two groups.

```
*A:PE# show service id 1 pim-snooping group ipv6
===============================================================================
PIM Snooping Groups ipv6
===============================================================================
Group Address           Source Address          Type     Incoming          Num
```

```
                                                        Intf              Oifs
------------------------------------------------------------------------------
ff0e:db8:1000::1        2001:db8:1000::1        (S,G)   SAP:1/1/1         2
ff0e:db8:1001::1        2001:db8:1001::1        (S,G)   SAP:1/1/1         2
------------------------------------------------------------------------------
Groups : 2
==============================================================================
*A:PE#

*A:PE# show service id 1 all | match "Mcast IPv6 scope"
Mcast IPv6 scope  : mac-based
*A:PE#

*A:PE# show service id 1 mfib
==============================================================================
Multicast FIB, Service 1
==============================================================================
Source Address  Group Address         Port Id                     Svc Id   Fwd
                                                                           Blk
------------------------------------------------------------------------------
*               33:33:00:00:00:01     sap:1/1/1                   Local    Fwd
                                      sap:1/1/2                   Local    Fwd
------------------------------------------------------------------------------
Number of entries: 1
==============================================================================
*A:PE#
```

3.2.20.4.2 SG-based IPv6 multicast forwarding

When PIM snooping for IPv6 is configured, SG-based forwarding can be enabled, which causes the IPv6 multicast forwarding to be based on both the source (if specified) and destination IPv6 address in the received join.

Enabling SG-based forwarding increases the MFIB usage if the source IPv6 address or higher 96 bits of the destination IPv6 address varies in the received joins compared to using MAC-based forwarding.

The following output shows a service with PIM snooping for IPv6 that has received joins for two multicast groups from different sources. As the forwarding mode is SG-based, there are two MFIB entries, one for each of the two groups.

```
*A:PE# show service id 1 pim-snooping group ipv6
==============================================================================
PIM Snooping Groups ipv6
==============================================================================
Group Address           Source Address          Type     Incoming          Num
                                                         Intf              Oifs
------------------------------------------------------------------------------
ff0e:db8:1000::1        2001:db8:1000::1        (S,G)    SAP:1/1/1         2
ff0e:db8:1001::1        2001:db8:1001::1        (S,G)    SAP:1/1/1         2
------------------------------------------------------------------------------
Groups : 2
==============================================================================
*A:PE#

*A:PE# show service id 1 all | match "Mcast IPv6 scope"
Mcast IPv6 scope  : sg-based
*A:PE#

*A:PE# show service id 1 mfib
==============================================================================
```

```
Multicast FIB, Service 1
===============================================================================
Source Address  Group Address         Port Id                     Svc Id   Fwd
                                                                        Blk
-------------------------------------------------------------------------------
2001:db8:1000:* ff0e:db8:1000::1      sap:1/1/1                   Local    Fwd
                                      sap:1/1/2                   Local    Fwd
2001:db8:1001:* ff0e:db8:1001::1      sap:1/1/1                   Local    Fwd
                                      sap:1/1/2                   Local    Fwd
-------------------------------------------------------------------------------
Number of entries: 2
===============================================================================
*A:PE#
```

SG-based IPv6 multicast forwarding is supported when both plain PIM snooping and PIM proxy are supported.

SG-based forwarding is only supported on FP3- or higher-based line cards. It is supported in all services in which PIM snooping for IPv6 is supported, with the same restrictions.

It is not supported in the following services:

- PBB B-VPLS
- PBB I-VPLS
- Routed-VPLS (including with I-VPLS and BGP-EVPN)
- BGP-EVPN-MPLS (including PBB-EVPN)
- VPLS E-Tree
- Management VPLS

In any specific service, SG-based forwarding and MLD snooping are mutually exclusive. Consequently, MLD snooping uses MAC-based forwarding.

It is not supported in services with:

- subscriber management
- multicast VLAN Registration
- video interface

It is not supported on connected SR OS routers configured with **improved-assert**.

It is not supported with the following forms of default SAP:

- *
- *.null
- *.*

3.2.20.5 PIM and IGMP/MLD snooping interaction

When both PIM snooping for IPv4 and IGMP snooping are enabled in the same VPLS service, multicast traffic is forwarded based on the combined multicast forwarding table. When PIM snooping is enabled, IGMP queries are forwarded but not snooped, consequently the IGMP querier needs to be seen either as a PIM neighbor in the VPLS service or the SAP toward it configured as an IGMP Mrouter port.

There is no interaction between PIM snooping for IPv6 and PIM snooping for IPv4/IGMP snooping when all are enabled within the same VPLS service. The configurations of PIM snooping for IPv6 and MLD snooping are mutually exclusive.

When PIM snooping is enabled within a VPLS service, all IP multicast traffic and flooded PIM messages (these include all PIM snooped messages when not in PIM proxy mode and PIM hellos when in PIM proxy mode) are sent to any SAP or SDP binding configured with an IGMP-snooping Mrouter port. This occurs even without IGMP-snooping enabled but is not supported in a BGP-VPLS or M-VPLS service.

3.2.20.6 Multi-chassis synchronization for Layer 2 snooping states

To achieve a faster failover in scenarios with redundant active/standby routers performing Layer 2 multicast snooping, it is possible to synchronize the snooping state from the active router to the standby router, so that if a failure occurs the standby router has the Layer 2 multicast snooped states and is able to forward the multicast traffic immediately. Without this capability, there would be a longer delay in re-establishing the multicast traffic path because it would wait for the Layer 2 states to be snooped.

Multi-chassis synchronization (MCS) is enabled per peer router and uses a **sync-tag**, which is configured on the objects requiring synchronization on both of the routers. This allows MCS to map the state of a set of objects on one router to a set of objects on the other router. Specifically, objects relating to a **sync-tag** on one router are backed up by, or are backing up, the objects using the same **sync-tag** on the other router (the state is synchronized from the active object on one router to its backup objects on the standby router).

The object type must be the same on both routers; otherwise, a mismatch error is reported. The same **sync-tag** value can be reused for multiple peer/object combinations, where each combination represents a different set of synchronized objects; however, a **sync-tag** cannot be configured on the same object to more than one peer.

The **sync-tag** is configured per port and can relate to a specific set of dot1q or QinQ VLANs on that port, as follows.

CLI syntax:

```
configure
  redundancy
    multi-chassis
      peer ip-address [create]
        sync
          port port-id [sync-tag sync-tag] [create]
            range encap-range sync-tag sync-tag
```

For IGMP snooping and PIM snooping for IPv4 to work correctly with MCS on QinQ ports using x.* SAPs, one of the following must be true:

- MCS is configured with a **sync-tag** for the entire port.
- The IGMP snooping SAP and the MCS **sync-tag** must be provisioned with the same Q-tag values when using the range parameter.

3.2.20.6.1 IGMP snooping synchronization

MCS for IGMP snooping synchronizes the join/prune state information from IGMP messages received on the related port/VLANs corresponding to their associated **sync-tag**. It is enabled as follows.

CLI syntax:

```
configure
  redundancy
    multi-chassis
      peer ip-address [create]
        sync
          igmp-snooping
```

IGMP snooping synchronization is supported wherever IGMP snooping is supported (except in EVPN for VXLAN). See IGMP snooping for VPLS for more information. IGMP snooping synchronization is also only supported for the following active/standby redundancy mechanisms:

- MC-LAG
- MC-Ring
- Single-Active Multihoming (EVPN-MPLS and PBB-EVPN I-VPLS)
- Single-Active Multihoming (EVPN-MPLS VPRN and IES routed VPLS)

Configuring an Mrouter port under an object that has the synchronization of IGMP snooping states enabled is not recommended. The Mrouter port configuration adds a (*,*) entry into the MFIB, which causes all groups (and IGMP messages) to be sent out of the respective object. In addition, the **mrouter-port** command causes all IGMP messages on that object to be discarded. However, the (*,*) entry is not synchronized by MCS. Consequently, the Mrouter port could cause the two MCS peers to be forwarding different sets of multicast streams out of the related object when each is active.

3.2.20.6.2 MLD snooping synchronization

MCS for MLD snooping is not supported. The command is not blocked for backward-compatibility reasons but has no effect on the system if configured.

3.2.20.6.3 PIM snooping for IPv4 synchronization

MCS for PIM snooping for IPv4 synchronizes the neighbor information from PIM hellos and join/prune state information from PIM for IPv4 messages received on the related SAPs and spoke-SDPs corresponding to the **sync-tag** associated with the related ports and SDPs, respectively. Use the following CLI syntax to enable MCS for PIM snooping for IPv4 synchronization.

CLI syntax:

```
configure
   redundancy
     multi-chassis
       peer ip-address [create]
         sync
           pim-snooping [saps] [spoke-sdps]
```

Any PIM hello state information received over the MCS connection from the peer router takes precedence over locally snooped hello information. This ensures that any PIM hello messages received on the active router that are then flooded, for example through the network backbone, and received over a local SAP or SDP on the standby router are not inadvertently used in the standby router's VPLS service.

The synchronization of PIM snooping state is only supported for manually configured spoke-SDPs. It is not supported for spoke-SDPs configured within an endpoint.

When synchronizing the PIM state between two spoke-SDPs, if both spoke-SDPs go down, the PIM state is maintained on both until one becomes active to ensure that the PIM state is preserved when a spoke-SDP recovers.

Appropriate actions based on the expiration of PIM-related timers on the standby router are only taken after it has become the active peer for the related object (after a failover).

PIM snooping for IPv4 synchronization is supported wherever PIM snooping for IPv4 is supported, excluding the following services:

- BGP-VPLS
- VPLS E-Tree
- management VPLS

See PIM snooping for VPLS for more details.

PIM snooping for IPv4 synchronization is also only supported for the following active/standby redundancy mechanisms on dual-homed systems:

- MC-LAG
- BGP multihoming
- active/standby pseudowires
- single-active multihoming (EVPN-MPLS and PBB-EVPN I-VPLS)

Configuring an Mrouter port under an object that has the synchronization of PIM snooping for IPv4 states enabled is not recommended. The Mrouter port configuration adds a (*,*) entry into the MFIB, which causes all groups (and PIM messages) to be sent out of the respective object. In addition, the **mrouter-port** command causes all PIM messages on that object to be discarded. However, the (*,*) entry is not synchronized by MCS. Consequently, the Mrouter port could cause the two MCS peers to be forwarding different sets of multicast streams out of the related object when each is active.

3.2.20.7 VPLS multicast-aware high availability features

The following features are High Availability capable:

- Configuration redundancy (all the VPLS multicast-aware configurations can be synchronized to the standby CPM)
- Local snooping states as well as states distributed by LDP can be synchronized to the standby CPM.
- Operational states can also be synchronized; for example, the operational state of PIM proxy.

3.2.21 RSVP and LDP P2MP LSP for forwarding VPLS/B-VPLS BUM and IP multicast packets

This feature enables the use of a P2MP LSP as the default tree for forwarding Broadcast, Unicast unknown, and Multicast (BUM) packets of a VPLS or B-VPLS instance. The P2MP LSP is referred to in this case as the Inclusive Provider Multicast Service Interface (I-PMSI).

When enabled, this feature relies on BGP Auto-Discovery (BGP-AD) or BGP-VPLS to discover the PE nodes participating in a specified VPLS/B-VPLS instance. The BGP route contains the information required to signal both the point-to-point (P2P) PWs used for forwarding unicast known Ethernet frames and the RSVP P2MP LSP used to forward the BUM frames. The root node signals the P2MP LSP based on an

LSP template associated with the I-PMSI at configuration time. The leaf node automatically joins the P2MP LSP that matches the I-PMSI tunnel information discovered via BGP.

If IGMP or PIM snooping are configured on the VPLS instance, multicast packets matching an L2 multicast Forwarding Information Base (FIB) record are also forwarded over the P2MP LSP.

The user enables the use of an RSVP P2MP LSP as the I-PMSI for forwarding Ethernet BUM and IP multicast packets in a VPLS/B-VPLS instance using the following context:

config>service>vpls [b-vpls]>provider-tunnel>inclusive>rsvp>lsp-template p2mp-lsp-template-name

The user enables the use of an LDP P2MP LSP as the I-PMSI for forwarding Ethernet BUM and IP multicast packets in a VPLS instance using the following context:

config>service>vpls [b-vpls]>provider-tunnel>inclusive>mldp

After the user performs a **no shutdown** under the context of the inclusive node and the expiration of a delay timer, BUM packets are forwarded over an automatically signaled mLDP P2MP LSP or over an automatically signaled instance of the RSVP P2MP LSP specified in the LSP template.

The user can specify that the node is both root and leaf in the VPLS instance:

config>service>vpls [b-vpls]>provider-tunnel>inclusive>root-and-leaf

The **root-and-leaf** command is required; otherwise, this node behaves as a leaf-only node by default. When the node is leaf only for the I-PMSI of type P2MP RSVP LSP, no PMSI Tunnel Attribute is included in BGP-AD route update messages and, therefore, no RSVP P2MP LSP is signaled, but the node can join an RSVP P2MP LSP rooted at other PE nodes participating in this VPLS/B-VPLS service. The user must still configure an LSP template even if the node is a leaf only. For the I-PMSI of type mLDP, the leaf-only node joins I-PMSI rooted at other nodes it discovered but does not include a PMSI Tunnel Attribute in BGP route update messages. This way, a leaf-only node forwards packets to other nodes in the VPLS/B-VPLS using the point-to-point spoke-SDPs.

BGP-AD (or BGP-VPLS) must have been enabled in this VPLS/B-VPLS instance or the execution of the **no shutdown** command under the context of the inclusive node is failed and the I-PMSI does not come up.

Any change to the parameters of the I-PMSI, such as disabling the P2MP LSP type or changing the LSP template, requires that the inclusive node be first shut down. The LSP template is configured in MPLS.

If the P2MP LSP instance goes down, VPLS/B-VPLS immediately reverts the forwarding of BUM packets to the P2P PWs. However, the user can restore at any time the forwarding of BUM packets over the P2P PWs by performing a **shutdown** under the context of the inclusive node.

This feature is supported with VPLS, H-VPLS, B-VPLS, and BGP-VPLS. It is not supported with I-VPLS and R-VPLS.

3.2.22 MPLS entropy label and hash label

The router supports the MPLS entropy label (RFC 6790) and the Flow Aware Transport label (known as the hash label) (RFC 6391). These labels allow LSR nodes in a network to load-balance labeled packets in a much more granular fashion than allowed by simply hashing on the standard label stack. See the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide* for more information.

The entropy label is supported for LDP VPLS and BGP-AD VPLS, as well as Epipe and Ipipe spoke-SDP termination on VPLS services. To configure insertion of the entropy label on a spoke-SDP or mesh-SDP of a specific service, use the **entropy-label** command in the **spoke-sdp**, **mesh-sdp**, or **pw-template**

contexts. Note that the entropy label is only inserted if the far end of the MPLS tunnel is also entropy-label-capable.

The hash label is supported for LDP VPLS, BGP-AD, and BGP-VPLS VPLS as well as Epipe and Ipipe spoke-SDP termination on VPLS services. Configure it using the following commands:

```
configure service epipe spoke-sdp hash-label
```

```
configure service ipipe spoke-sdp hash-label
```

```
configure service pw-template hash-label
```

```
configure service vpls mesh-sdp hash-label
```

```
configure service vpls spoke-sdp hash-label
```

Optionally, the **hash-label signal-capability** command can be configured. If the user only configures **hash-label** command, the hash label is sent (and it is expected to be received) in all the packets. However, if the **hash-label signal-capability** command is configured, the use of the hash label is signaled and only used in case the peer PE signals support for hash label in its TLDP signaling or BGP-VPLS route (RFC8395).

Either the hash label or the entropy label can be configured on one object, but not both.

3.3 Routed VPLS and I-VPLS

This section provides information about Routed VPLS (R-VPLS) and I-VPLS. R-VPLS and I-VPLS apply to the 7450 ESS and 7750 SR.

3.3.1 IES or VPRN IP interface binding

For the remainder of this section, R-VPLS and Routed I-VPLS both re described as a VPLS service, and differences are pointed out where applicable.

A standard IP interface within an existing IES or VPRN service context may be bound to a service name. Subscriber and group IP interfaces are not allowed to bind to a VPLS or I-VPLS service context or I-VPLS. A VPLS service only supports binding for a single IP interface.

While an IP interface may only be bound to a single VPLS service, the routing context containing the IP interface (IES or VPRN) may have other IP interfaces bound to other VPLS service contexts of the same type (all VPLS or all I-VPLS). That is, R-VPLS allows the binding of IP interfaces in IES or VPRN services to be bound to VPLS services and Routed I-VPLS allows of IP interfaces in IES or VPRN services to be bound to I-VPLS services.

3.3.1.1 Assigning a service name to a VPLS service

When a service name is applied to any service context, the name and service ID association is registered with the system. A service name cannot be assigned to more than one service ID.

Special consideration is provided to a service name that is assigned to a VPLS service that has the **configure service vpls allow-ip-int-bind** command enabled. If a name is applied to the VPLS service while the flag is set, the system scans the existing IES and VPRN services for an IP interface that is bound to the specified service name. If an IP interface is found, the IP interface is attached to the VPLS service associated with the name. Only one interface can be bound to the specified name.

If the **allow-ip-int-bind** command is not enabled on the VPLS service, the system does not attempt to resolve the VPLS service name to an IP interface. The corresponding IP interface is bound and becomes operational up as soon as the **allow-ip-int-bind** flag is configured on the VPLS. There is no need to toggle the **shutdown/no shutdown** command.

If an IP interface is not currently bound to the service name used by the VPLS service, no action is taken at the time of the service name assignment.

3.3.1.2 Service binding requirements

If the defined service ID is created on the system, the system checks to ensure that the service type is VPLS. If the service type is not VPLS or I-VPLS, service creation is not allowed and the service ID remains undefined within the system.

If the created service type is VPLS, the IP interface is eligible to enter the operationally up state.

3.3.1.3 Bound service name assignment

If a bound service name is assigned to a service within the system, the system first checks to ensure the service type is VPLS or I-VPLS. Secondly, the system ensures that the service is not already bound to another IP interface through the service ID. If the service type is not VPLS or I-VPLS or the service is already bound to another IP interface through the service ID, the service name assignment fails.

If a single VPLS service ID and service name is assigned to two separate IP interfaces, the VPLS service is not allowed to enter the operationally up state.

3.3.1.4 Binding a service name to an IP interface

An IP interface within an IES or VPRN service context may be bound to a service name at any time. Only one interface can be bound to a service.

When an IP interface is bound to a service name and the IP interface is administratively up, the system scans for a VPLS service context using the name and takes one of the following actions:

- If the name is not currently in use by a service, the IP interface is placed in an operationally down: non-existent service name or inappropriate service type state.
- If the name is currently in use by a non-VPLS service or the wrong type of VPLS service, the IP interface is placed in the operationally down: non-existent service name or inappropriate service type state.
- If the name is currently in use by a VPLS service without the **allow-ip-int-bind** flag set, the IP interface is placed in the operationally down: VPLS service **allow-ip-int-bind** flag not set state. There is no need to toggle the **shutdown/no shutdown** command.
- If the name is currently in use by a valid VPLS service and the **allow-ip-int-bind** flag is set, the IP interface is eligible to be placed in the operationally up state depending on other operational criteria being met.

3.3.1.5 Bound service deletion or service name removal

If a VPLS service is deleted while bound to an IP interface, the IP interface enters the Down: non-existent svc-ID operational state. If the IP interface was bound to the VPLS service name, the IP interface enters the Down: non-existent svc-name operational state. No console warning is generated.

If the created service type is VPLS, the IP interface is eligible to enter the operationally up state.

3.3.1.6 IP interface attached VPLS service constraints

When a VPLS service has been bound to an IP interface through its service name, the service name assigned to the service cannot be removed or changed unless the IP interface is first unbound from the VPLS service name.

A VPLS service that is currently attached to an IP interface cannot be deleted from the system unless the IP interface is unbound from the VPLS service name.

The **allow-ip-int-bind** flag within an IP interface attached VPLS service cannot be reset. The IP interface must first be unbound from the VPLS service name to reset the flag.

3.3.1.7 IP interface and VPLS operational state coordination

When the IP interface is successfully attached to a VPLS service, the operational state of the IP interface is dependent upon the operational state of the VPLS service.

The VPLS service remains down until at least one virtual port (SAP, spoke-SDP, or mesh SDP) is operational.

3.3.2 IP interface MTU and fragmentation

The VPLS service is affected by two MTU values: port MTUs and the VPLS service MTU. The MTU on each physical port defines the largest Layer 2 packet (including all DLC headers) that may be transmitted out a port. The VPLS has a service level MTU that defines the largest packet supported by the service. This MTU does not include the local encapsulation overhead for each port (QinQ, Dot1Q, TopQ, or SDP service delineation fields and headers) but does include the remainder of the packet.

As virtual ports are created in the system, the virtual port cannot become operational unless the configured port MTU minus the virtual port service delineation overhead is greater than or equal to the configured VPLS service MTU. Therefore, an operational virtual port is ensured to support the largest packet traversing the VPLS service. The service delineation overhead on each Layer 2 packet is removed before forwarding into a VPLS service. VPLS services do not support fragmentation and must discard any Layer 2 packet larger than the service MTU after the service delineation overhead is removed.

When an IP interface is associated with a VPLS service, the IP-MTU is based on either the administrative value configured for the IP interface or an operational value derived from VPLS service MTU. The operational IP-MTU cannot be greater than the VPLS service MTU minus 14 bytes.

- If the configured (administrative) IP-MTU is configured for a value greater than the normalized IP-MTU, based on the VPLS service-MTU, then the operational IP-MTU is reset to equal the normalized IP-MTU value (VPLS service MTU – 14 bytes).

- If the configured (administrative) IP-MTU is configured for a value less than or equal to the normalized IP-MTU, based on the VPLS service-MTU, then the operational IP-MTU is set to equal the configured (administrative) IP-MTU value.

3.3.2.1 Unicast IP routing into a VPLS service

The VPLS service MTU and the IP interface MTU parameters may be changed at any time.

3.3.3 ARP and VPLS FDB interactions

Two address-oriented table entries are used when routing into a VPLS service. An ARP entry is used on the routing side to determine the destination MAC address used by an IP next-hop. In the case where the destination IP address in the routed packet is a host on the local subnet represented by the VPLS instance, the destination IP address is used as the next-hop IP address in the ARP cache lookup. If the destination IP address is in a remote subnet that is reached by another router attached to the VPLS service, the routing lookup returns the local IP address on the VPLS service of the remote router. If the next-hop is not currently in the ARP cache, the system generates an ARP request to determine the destination MAC address associated with the next-hop IP address.

IP routing to all destination hosts associated with the next-hop IP address stops until the ARP cache is populated with an entry for the next-hop. The ARP cache may be populated with a static ARP entry for the next-hop IP address. While dynamically populated ARP entries age out according to the ARP aging timer, static ARP entries never age out.

The second address table entry that affects VPLS routed packets is the MAC destination lookup in the VPLS service context. The MAC associated with the ARP table entry for the IP next-hop may or may not currently be populated in the VPLS Layer 2 FDB table. While the destination MAC is unknown (not populated in the VPLS FDB), the system floods all packets destined for that MAC (routed or bridged) to all virtual ports within the VPLS service context. When the MAC is known (populated in the VPLS FDB), all packets destined for the MAC (routed or bridged) are targeted to the specific virtual port where the MAC has been learned.

As with ARP entries, static MAC entries may be created in the VPLS FDB. Dynamically learned MAC addresses are allowed to age out or be flushed from the VPLS FDB, while static MAC entries always remain associated with a specific virtual port. Dynamic MACs may also be relearned on another VPLS virtual port than the current virtual port in the FDB. In this case, the system automatically moves the MAC FDB entry to the new VPLS virtual port.

The MAC address associated with the R-VPLS IP interface is protected within its VPLS service such that frames received with this MAC address as the source address are discarded. VRRP MAC addresses are not protected in this way.

3.3.3.1 R-VPLS specific ARP cache behavior

In typical routing behavior, the system uses the IP route table to select the egress interface, and then at the egress forwarding engine, an ARP entry is used to forward the packet to the appropriate Ethernet MAC. With R-VPLS, the egress IP interface may be represented by a multiple egress forwarding engine (wherever the VPLS service virtual ports exist).

To optimize routing performance, the ingress forwarding engine processing has been augmented to perform an ingress ARP lookup to resolve which VPLS MAC address the IP frame must be routed toward.

This MAC address may be currently known or unknown within the VPLS FDB. If the MAC is unknown, the packet is flooded by the ingress forwarding engine to all egress forwarding engines where the VPLS service exists. When the MAC is known on a virtual port, the ingress forwarding engine forwards the packet to the correct egress forwarding engine. Table 12: Ingress routed to VPLS next-hop behavior describes how the ARP cache and MAC FDB entry states interact at ingress and Table 13: Egress R-VPLS next-hop behavior describes the corresponding egress behavior.

Table 12: Ingress routed to VPLS next-hop behavior

Next-hop ARP cache entry	Next-hop MAC FDB entry	Ingress behavior
ARP Cache Miss (No Entry)	Known or Unknown	Flood to all egress forwarding engines associated with the VPLS or I-VPLS context.
	Unknown	Flood to all egress forwarding engines associated with the VPLS or I-VPLS context.
	Unknown	Flood to all egress forwarding engines associated with the VPLS for forwarding to all VPLS or I-VPLS virtual ports.

Table 13: Egress R-VPLS next-hop behavior

Next-hop ARP Cache entry	Next-hop MAC FDB entry	Egress behavior
ARP Cache Miss (No Entry)	Known	No ARP entry. The MAC address is unknown and the ARP request is flooded out of all virtual ports of the VPLS or I-VPLS instance.
	Unknown	Request control engine processing the ARP request to transmit out of all virtual ports associated with the VPLS or I-VPLS service. Only the first egress forwarding engine ARP processing request triggers an egress ARP request.
ARP Cache Hit	Known	Forward out of specific egress VPLS or I-VPLS virtual ports where MAC has been learned.
	Unknown	Flood to all egress VPLS or I-VPLS virtual ports on forwarding engine.

3.3.4 The allow-ip-int-bind VPLS flag

The **allow-ip-int-bind** flag on a VPLS service context is used to inform the system that the VPLS service is enabled for routing support. The system uses the setting of the flag as a key to determine the types of ports and forwarding planes the VPLS service may span.

The system also uses the flag state to define which VPLS features are configurable on the VPLS service to prevent enabling a feature that is not supported when routing support is enabled.

3.3.4.1 R-VPLS SAPs only supported on standard Ethernet ports

The **allow-ip-int-bind** flag is set (routing support enabled) on a VPLS/I-VPLS service. SAPs within the service can be created on standard Ethernet, and CCAG ports. POS is not supported.

3.3.4.2 LAG port membership constraints

If a LAG has a non-supported port type as a member, a SAP for the routing-enabled VPLS service cannot be created on the LAG. When one or more routing enabled VPLS SAPs are associated with a LAG, a non-supported Ethernet port type cannot be added to the LAG membership.

3.3.4.3 R-VPLS feature restrictions

When the **allow-ip-int-bind** flag is set on a VPLS service, the following restrictions apply. The flag also cannot be enabled while any of these features are applied to the VPLS service:

- SDPs used in spoke or mesh SDP bindings cannot be configured as GRE.
- The VPLS service type cannot be B-VPLS or M-VPLS.
- MVR from R-VPLS and to another SAP is not supported.
- Enhanced and Basic Subscriber Management (BSM) features cannot be enabled.
- Network domain on SDP bindings cannot be enabled.
- Per-service hashing is not supported.
- BGP-VPLS is not supported.
- Ingress queuing for split horizon groups is not supported.
- Multiple virtual routers are not supported.

3.3.4.4 Routed I-VPLS feature restrictions

The following restrictions apply to routed I-VPLS.

- Multicast is not supported.
- The VC-VLANs are not supported on SDPs.
- The **force-qtag-forwarding** command is not supported.
- Control words are not supported on B-VPLS SDPs.
- The hash label is not supported on B-VPLS SDPs.
- The **provider-tunnel** is not supported on routed I-VPLS services.

3.3.5 IPv4 and IPv6 multicast routing support

IPv4 and IPv6 multicast routing is supported in a R-VPLS service through its IP interface when the source of the multicast stream is on one side of its IP interface and the receivers are on either side of the IP interface. For example, the source for multicast stream G1 could be on the IP side sending to receivers on

both other regular IP interfaces and the VPLS of the R-VPLS service, while the source for group G2 could be on the VPLS side sending to receivers on both the VPLS and IP side of the R-VPLS service.

IPv4 and IPv6 multicast routing is not supported with Multicast VLAN Registration functions or the configuration of a video interface within the associated VPLS service. It is also not supported in a routed I-VPLS service, or for IPv6 multicast in BGP EVPN-MPLS routed VPLS services. Forwarding IPv4 or IPv6 multicast traffic from the R-VPLS IP interface into its VPLS service on a P2MP LSP is not supported.

The IP interface of a R-VPLS supports the configuration of both PIM and IGMP for IPv4 multicast and for both PIM and MLD for IPv6 multicast.

To forward IPv4/IPv6 multicast traffic from the VPLS side of the R-VPLS service to the IP side, the **forward-ipv4-multicast-to-ip-int** and/or **forward-ipv6-multicast-to-ip-int** commands must be configured as follows:

```
configure
    service
        vpls <service-id>
            allow-ip-int-bind
                forward-ipv4-multicast-to-ip-int
                forward-ipv6-multicast-to-ip-int
            exit
        exit
    exit
exit
```

Enabling IGMP snooping or MLD snooping in the VPLS service is optional, where supported. If IGMP/MLD snooping is enabled, IGMP/MLD must be enabled on the R-VPLS IP interface in order for multicast traffic to be sent into, or received from, the VPLS service. IPv6 multicast uses MAC-based forwarding; see MAC-based IPv6 multicast forwarding for more information.

If both IGMP/MLD and PIM for IPv4/IPv6 are configured on the R-VPLS IP interface in a redundant PE topology, the associated IP interface on one of the PEs must be configured as both the PIM designated router and the IGMP/MLD querier. This ensures that the multicast traffic is sent into the VPLS service, as IGMP/MLD joins are only propagated to the IP interface if it is the IGMP/MLD querier. An alternative to this is to configure the R-VPLS IP interface in the VPLS service as an Mrouter port, as follows:

```
configure
    service
        vpls <service-id>
            allow-ip-int-bind
                igmp-snooping
                    mrouter-port
                mld-snooping
                    mrouter-port
            exit
        exit
    exit
exit
```

This configuration achieves a faster failover in scenarios with redundant routers where multicast traffic is sent to systems on the VPLS side of their R-VPLS services and IGMP/MLD snooping is enabled in the VPLS service. If the active router fails, the remaining router does not have to wait until it sends an IGMP/MLD query into the VPLS service before it starts receiving IGMP/MLD joins and starts sending the multicast traffic into the VPLS service. When the Mrouter port is configured as above, all IGMP/MLD joins (and multicast traffic) are sent to the VPLS service IP interface.

IGMP/MLD snooping should only be enabled when systems, as opposed to PIM routers, are connected to the VPLS service. If IGMP/MLD snooping is enabled when the VPLS service is used for transit traffic for

connected PIM routers, the IGMP/MLD snooping would prevent multicast traffic being forwarded between the PIM routers (as PIM snooping is not supported). A workaround would be to configure the VPLS SAPs and spoke-SDPs (and the R-VPLS IP interface) to which the PIM routers are connected as Mrouter ports.

If IMPM is enabled on an FP on which there is a R-VPLS service with **forward-ipv4-multicast-to-ip-int** or **forward-ipv6-multicast-to-ip-int** configured, the IPv4/IPv6 multicast traffic received in the VPLS service that is forwarded through the IP interface is IMPM-managed even without IGMP/MLD snooping being enabled. This does not apply to traffic that is only flooded within the VPLS service.

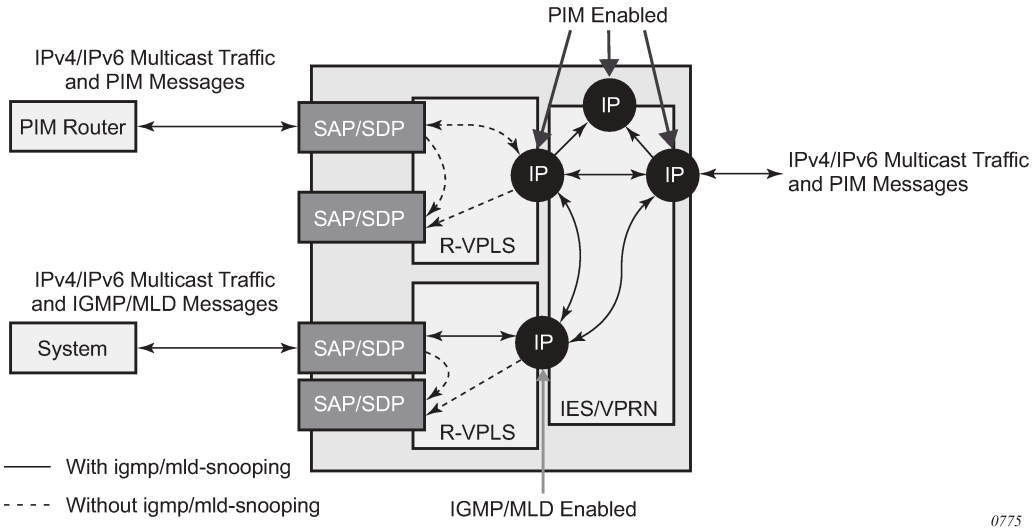
When IPv4/IPv6 multicast traffic is forwarded from a VPLS SAP through the R-VPLS IP interface, the packet count is doubled in the following statistics to represent both the VPLS and IP replication (this reflects the capacity used for this traffic on the ingress queues, which is subject to any configured rates and IMPM capacity management):

- Offered queue statistics
- IMPM managed statistics
- IMPM unmanaged statistics for policed traffic

IPv4 or IPv6 multicast traffic entering the IP side of the R-VPLS service and exiting over a multi-port LAG on the VPLS side of the service is sent on a single link of that egress LAG, specifically the link used for all broadcast, unknown, and multicast traffic.

An example of IPv4/IPv6 multicast in a R-VPLS service is shown in Figure 84: IPv4/IPv6 multicast with a router VPLS service. There are two R-VPLS IP interfaces connected to an IES service with the upper interface connected to a VPLS service in which there is a PIM router and the lower interface connected to a VPLS service in which there is a system using IGMP/MLD.

Figure 84: IPv4/IPv6 multicast with a router VPLS service



The IPv4/IPv6 multicast traffic entering the IES/VPRN service through the regular IP interface is replicated to both the other regular IP interface and the two R-VPLS interfaces if PIM/IGMP/MLD joins have been received on the respective IP interfaces. This traffic is flooded into both VPLS services unless IGMP/MLD snooping is enabled in the lower VPLS service, in which case it is only sent to the system originating the IGMP/MLD join.

The IPv4/IPv6 multicast traffic entering the upper VPLS service from the connected PIM router is flooded in that VPLS service and, if related joins have been received, forwarded to the regular IP interfaces in the

IES/VPRN. It is also be forwarded to the lower VPLS service if an IGMP/MLD join is received on its IP interface, and is flooded in that VPLS service unless IGMP/MLD snooping is enabled.

The IPv4/IPv6 multicast traffic entering the lower VPLS service from the connected system is flooded in that VPLS service, unless IGMP/MLD snooping is enabled, in which case it is only forwarded to SAPs, spoke-SDPs, or the R-VPLS IP interface if joins have been received on them. It is forwarded to the regular IP interfaces in the IES/VPRN service if related joins have been received on those interfaces, and it is also forwarded to the upper VPLS service if a PIM IPv4/IPv6 join is received on its IP interface, this being flooded in that VPLS service.

3.3.6 BGP-AD for R-VPLS support

BGP Auto-Discovery (BGP-AD) for R-VPLS is supported. BGP-AD for LDP VPLS is an already supported framework for automatically discovering the endpoints of a Layer 2 VPN offering an operational model similar to that of an IP VPN.

3.3.7 R-VPLS restrictions

3.3.7.1 VPLS SAP ingress IP filter override

When an IP Interface is attached to a VPLS or an I-VPLS service context, the VPLS SAP provisioned IP filter for ingress routed packets may be optionally overridden to provide special ingress filtering for routed packets. This allows different filtering for routed packets and non-routed packets. The filter override is defined on the IP interface bound to the VPLS service name. A separate override filter may be specified for IPv4 and IPv6 packet types.

If a filter for a specified packet type (IPv4 or IPv6) is not overridden, the SAP specified filter is applied to the packet (if defined).

3.3.7.2 IP interface defined egress QoS reclassification

The SAP egress QoS policy defined forwarding class and profile reclassification rules are not applied to egress routed packets. To allow for egress reclassification, a SAP egress QoS policy ID may be optionally defined on the IP interface that is applied to routed packets that egress the SAPs on the VPLS or I-VPLS service associated with the IP interface. Both unicast directed and MAC unknown flooded traffic apply to this rule. Only the reclassification portion of the QoS policy is applied, which includes IP precedence or DSCP classification rules and any defined IP match criteria and their associated actions.

The policers and queues defined within the QoS policy applied to the IP interface are not created on the egress SAPs of the VPLS service. Instead, the QoS policy applied to the egress SAPs defines the egress policers and queues used by both routed and non-routed egress packets. The forwarding class mappings defined in the egress SAP’s QoS policy also defines which policer or queue handles each forwarding class for both routed and non-routed packets.

3.3.7.3 Remarking for VPLS and routed packets

The remarking of packets to and from an IP interface in an R-VPLS service corresponds to that supported on IP interface, even though the packets ingress or egress a SAP in the VPLS service bound to the IP service. Specifically, this results in the ability to remark the DSCP/prec for these packets.

Packets that ingress and egress SAPs in the VPLS service (not routed through the IP interface) support the regular VPLS QoS and, therefore, the DSCP/prec cannot be remarked.

3.3.7.4 IPv4 multicast routing

When using IPv4 multicast routing, the following are not supported:

- The multicast VLAN registration functions within the associated VPLS service.
- The configuration of a video ISA within the associated VPLS service.
- The configuration of MFIB-allowed MDA destinations under spoke/mesh SDPs within the associated VPLS service.
- The IPv4 multicast routing is not supported in Routed I-VPLS.
- The RFC 6037 multicast tunnel termination (including when the system is a bud node) is not supported on the R-VPLS IP interface for multicast traffic received in the VPLS service.
- Forwarding of multicast traffic from the VPLS side of the service to the IP interface side of the service is not supported for R-VPLS services that have egress VXLAN VTEPs configured.

3.3.7.5 R-VPLS supported routing-related protocols

The following protocols are supported on IP interfaces bound to a VPLS service:

- BGP
- OSPF
- ISIS
- PIM
- IGMP
- BFD
- VRRP
- ARP
- DHCP Relay

3.3.7.6 Spanning tree and split horizon

A R-VPLS context supports all spanning tree and split horizon capabilities that a non-R-VPLS service supports.

3.4 VPLS service considerations

This section describes the 7450 ESS, 7750 SR, and 7950 XRS service features and any special capabilities or considerations as they relate to VPLS services.

3.4.1 SAP encapsulations

VPLS services are designed to carry Ethernet frame payloads, so the services can provide connectivity between any SAPs and SDPs that pass Ethernet frames. The following SAP encapsulations are supported on the 7450 ESS, 7750 SR, and 7950 XRS VPLS services:

- Ethernet null
- Ethernet dot1q
- Ethernet QinQ

3.4.2 VLAN processing

The SAP encapsulation definition on Ethernet ingress ports defines which VLAN tags are used to determine the service to which the packet belongs:

- **null encapsulation defined on ingress**

  Any VLAN tags are ignored and the packet goes to a default service for the SAP.

- **dot1q encapsulation defined on ingress**

  Only the first label is considered.

- **QinQ encapsulation defined on ingress**

  Both labels are considered. The SAP can be defined with a wildcard for the inner label (for example, "100:100.*"). In this situation, all packets with an outer label of 100 are treated as belonging to the SAP. If on the same physical link, there is also an SAP defined with a QinQ encapsulation of 100:100.1, then traffic with 100:1 goes to that SAP and all other traffic with 100 as the first label goes to the SAP with the 100:100.* definition.

In the last two situations above, traffic encapsulated with tags for which there is no definition are discarded.

3.4.3 Ingress VLAN swapping

This feature is supported on VPLS and VLL service where the end-to-end solution is built using two node solutions (requiring SDP connections between the nodes).

In VLAN swapping, only the VLAN ID value is copied to the inner VLAN position. Ethertype of the inner tag is preserved and all consecutive nodes work with that value. Similarly, the dot1p bits value of the outer tag is not preserved.

Figure 85: Ingress VLAN swapping

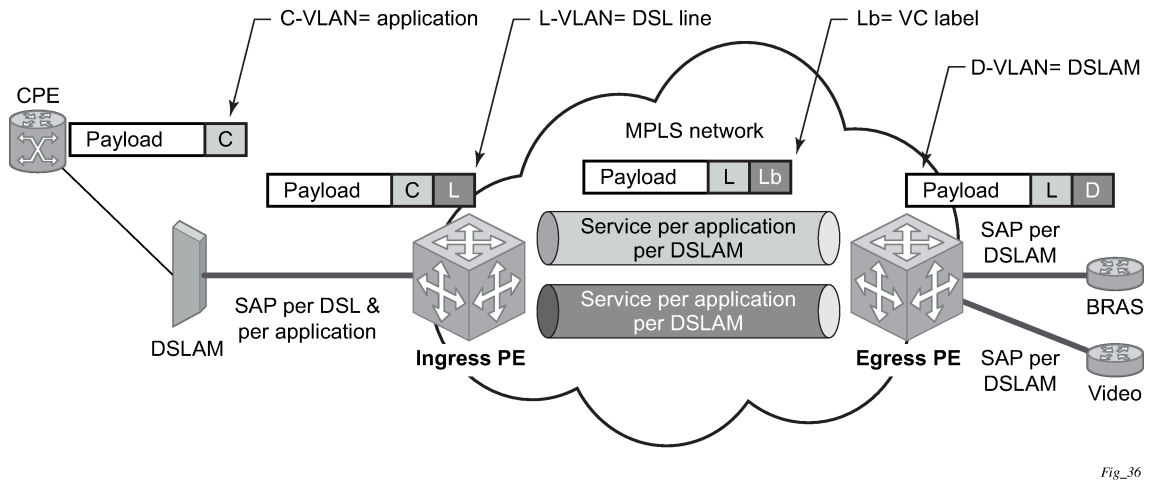


Figure 85: Ingress VLAN swapping describes the network where, at user access side (DSLAM facing SAPs), every subscriber is represented by several QinQ SAPs with inner-tag encoding service and outer-tag encoding subscriber (DSL line). The aggregation side (BRAS or PE-facing SAPs) is represented by a DSL line number (inner VLAN tag) and DSLAM (outer VLAN tag). The effective operation on the VLAN tag is to drop the inner tag at the access side and push another tag at the aggregation side.

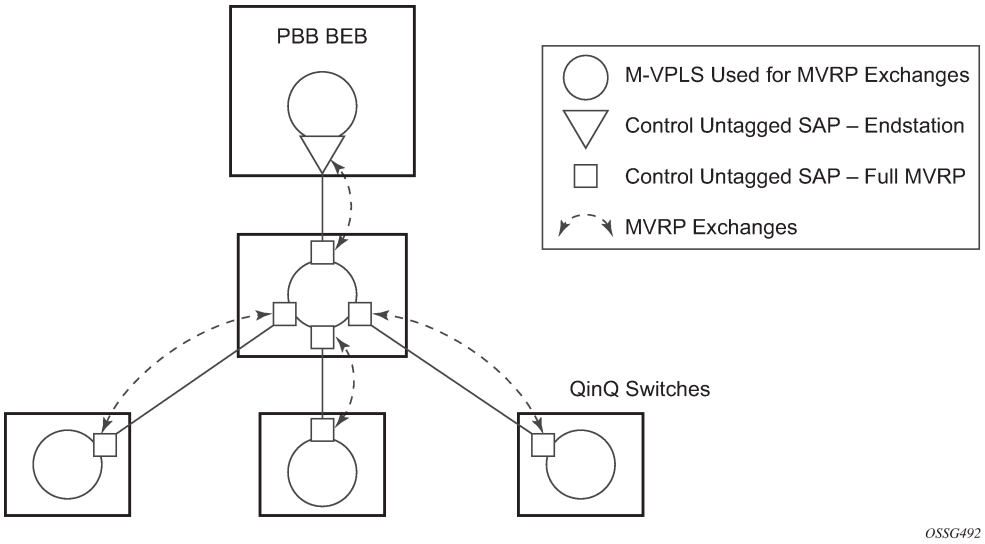
3.4.4 Service auto-discovery using MVRP

IEEE 802.1ak Multiple VLAN Registration Protocol (MVRP) is used to advertise throughout a native Ethernet switching domain one or multiple VLAN IDs to automatically build native Ethernet connectivity for multiple services. These VLAN IDs can be either Customer VLAN IDs (CVID) in an enterprise switching environment, Stacked VLAN IDs (SVID) in a Provider Bridging, QinQ Domain (see the *IEEE 802.1ad*), or Backbone VLAN IDs (BVID) in a Provider Backbone Bridging (PBB) domain (see the *IEEE 802.1ah*).

The initial focus of Nokia MVRP implementation is a Service Provider QinQ domain with or without a PBB core. The QinQ access into a PBB core example is used throughout this section to describe the MVRP implementation. With the exception of end-station components, a similar solution can be used to address a QinQ only or enterprise environments.

The components involved in the MVRP control plane are shown in Figure 86: Infrastructure for MVRP exchanges.

Figure 86: Infrastructure for MVRP exchanges



All the devices involved are QinQ switches with the exception of the PBB BEB which delimits the QinQ domain and ensures the transition to the PBB core. The circles represent Management VPLS instances interconnected by SAPs to build a native Ethernet switching domain used for MVRP control plane exchanges.

The following high-level steps are involved in auto-discovery of VLAN connectivity in a native Ethernet domain using MVRP:

1. Configure the MVRP infrastructure
   - This requires the configuration of a Management VPLS (M-VPLS) context.
   - MSTP may be used in M-VPLS to provide the loop-free topology over which the MVRP exchanges take place.
2. Instantiate related VLAN FDB, trunks in the MVRP, M-VPLS scope
   - The VLAN FDBs (VPLS instances) and associated trunks (SAPs) are instantiated in the same Ethernet switches and on the same "trunk ports" as the M-VPLS.
   - There is no need to instantiate data VPLS instances in the BEB. I-VPLS instances and related downward-facing SAPs are provisioned manually because the ISID-to-VLAN association must be configured.
3. MVRP activation of service connectivity

   When the first two customer UNI or PBB end-station SAPs, or both, are configured on different Ethernet switches in a specific service context, the MVRP exchanges activate service connectivity.

3.4.4.1 Configure the MVRP infrastructure using an M-VPLS context

The following provisioning steps apply.

1. Configure the M-VPLS instances in the switches that participate in MVRP control plane.
2. Configure under the M-VPLS the untagged SAPs to be used for MVRP exchanges; only dot1q or qinq ports are accepted for MVRP enabled M-VPLS.

3. Configure the MVRP parameters at M-VPLS instance or SAP level.

3.4.4.2 Instantiate related VLAN FDBs and trunks in MVRP scope

This requires the configuration in the M-VPLS, under vpls-group, of the following attributes: VLAN ranges, vpls-template and vpls-sap-template bindings. As soon as the VPLS group is enabled, the configured attributes are used to auto-instantiate, on a per-VLAN basis, a VPLS FDB and related SAPs in the switches and on the "trunk ports" specified in the M-VPLS context. The trunk ports are ports associated with an M-VPLS SAP not configured as an end-station.

The following procedure is used:

- The vpls-template binding is used to instantiate the VPLS instance where the service ID is derived from the VLAN value as per service-range configuration.
- The vpls-sap-template binding is used to create dot1q SAPs by deriving from the VLAN value the service delimiter as per service-range configuration.

The above procedure may be used outside of the MVRP context to pre-provision a large number of VPLS contexts that share the same infrastructure and attributes.

The MVRP control of the auto-instantiated services can be enabled using the **mvrp-control** command under the **vpls-group**.

- If **mvrp-control** is disabled, the auto-created VPLS instances and related SAPs are ready to forward.
- If **mvrp-control** is enabled, the auto-created VPLS instances are instantiated initially with an empty flooding domain. According to the operator configuration the MVRP exchanges gradually enable service connectivity – between configured SAPs in the data VPLS context.

  This also provides protection against operational mistakes that may generate flooding throughout the auto-instantiated VLAN FDBs.

From an MVRP perspective, these SAPs can be either "full MVRP" or "end-station" interfaces.

A full MVRP interface is a full participant in the local M-VPLS scope as described below.

- VLAN attributes received in an MVRP registration on this MVRP interface are declared on all the other full MVRP SAPs in the control VPLS.
- VLAN attributes received in an MVRP registration on other full MVRP interfaces in the local M-VPLS context are declared on this MVRP interface.

In an MVRP end-station interface, the attributes registered on that interface have local significance, as described below.

- VLAN attributes received in an MVRP registration on this interface are not declared on any other MVRP SAPs in the control VPLS. The attributes are registered only on the local port.
- Only locally active VLAN attributes are declared on the end-station interface; VLAN attributes registered on any other MVRP interfaces are not declared on end-station interfaces.
- Also defining an M-VPLS SAP as an end-station does not instantiate any objects on the local switch; the command is used just to define which SAP needs to be monitored by MVRP to declare the related VLAN value.

The following example describes the M-VPLS configuration required to auto-instantiate the VLAN FDBs and related trunks in non-PBB switches.

```
mrp
        – no shutdown
```

```
        – mmrp
            – shutdown
        – mvrp
            – no shutdown
    sap 1/1/1:0
        – mrp mvrp
            – no shutdown
    sap 2/1/2:0
        – mrp mvrp
            – no shutdown
    sap 3/1/10:0
        – mrp mvrp
            – no shutdown
    vpls-group 1
        – service-range 100-2000
        – vpls-template-binding Autovpls1
        – sap-template-binding Autosap1
            – mvrp-control
        – no shutdown
```

A similar M-VPLS configuration may be used to auto-instantiate the VLAN FDBs and related trunks in PBB switches. The vpls-group command is replaced by the end-station command under the downward-facing SAPs as in the following example.

```
config>service>vpls control-mvrp m-vpls create customer 1
        – [..]
        – sap 1/1/1:0
            – mvrp mvrp
                – endstation-vid-group 1 vlan-id 100-2000
                – no shutdown
```

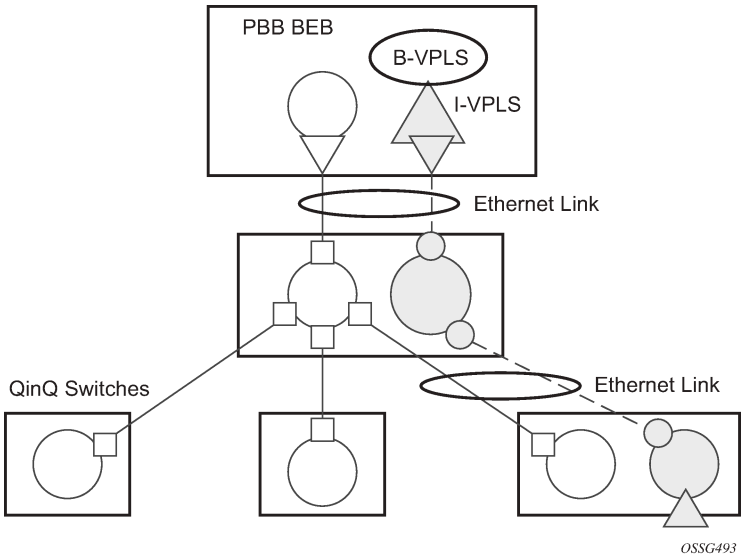
3.4.4.3 MVRP activation of service connectivity

As new Ethernet services are activated, UNI SAPs need to be configured and associated with the VLAN IDs (VPLS instances) auto-created using the procedures described in the previous sections. These UNI SAPs may be located in the same VLAN domain or over a PBB backbone. When UNI SAPs are located in different VLAN domains, an intermediate service translation point must be used at the PBB BEB, which maps the local VLAN ID through an I-VPLS SAP to a PBB ISID. This BEB SAP is playing the role of an end-station from an MVRP perspective for the local VLAN domain.

This section discusses how MVRP is used to activate service connectivity between a BEB SAP and a UNI SAP located on one of the switches in the local domain. A similar procedure is used in the case of UNI SAPs configured on two switches located in the same access domain. No end-station configuration is required on the PBB BEB if all the UNI SAPs in a service are located in the same VLAN domain.

The service connectivity instantiation through MVRP is shown in Figure 87: Service instantiation with MVRP - QinQ to PBB example.

Figure 87: Service instantiation with MVRP - QinQ to PBB example



In this example, the UNI and service translation SAPs are configured in the data VPLS represented by the gray circles. This instance and associated trunk SAPs were instantiated using the procedures described in the previous sections. The following are configuration rules:

- on the BEB, an I-VPLS SAP must be configured toward the local switching domain (see yellow triangle facing downward in Figure 87: Service instantiation with MVRP - QinQ to PBB example).
- on the UNI facing the customer, a "customer" SAP is configured on the lower left switch (see yellow triangle facing upward in Figure 87: Service instantiation with MVRP - QinQ to PBB example).

As soon as the first UNI SAP becomes active in the data VPLS on the ES, the associated VLAN value is advertised by MVRP throughout the related M-VPLS context. As soon as the second UNI SAP becomes available on a different switch, or in our example on the PBB BEB, the MVRP proceeds to advertise the associated VLAN value throughout the same M-VPLS. The trunks that experience MVRP declaration and registration in both directions become active, instantiating service connectivity as represented by the big and small yellow circles shown in the figure.

A hold-time parameter (**config>service>vpls>mrp>mvrp>hold-time**) is provided in the M-VPLS configuration to control when the end-station or last UNI SAP is considered active from an MVRP perspective. The hold-time controls the amount of MVRP advertisements generated on fast transitions of the end-station or UNI SAPs.

If the **no hold-time** setting is used, the following rules apply:

- MVRP stops declaring the VLAN only when the last provisioned UNI SAP associated locally with the service is deleted.
- MVRP starts declaring the VLAN as soon as the first provisioned SAP is created in the associated VPLS instance, regardless of the operational state of the SAP.

If a non-zero "hold-time" setting is used, the following rules apply:

- When a SAP in down state is added, MVRP does not declare the associated VLAN attribute. The attribute is declared immediately when the SAP comes up.
- When the SAP goes down, the MVRP waits until "hold-time" expiry before withdrawing the declaration.

For QinQ end-station SAPs, only **no hold-time** setting is allowed.

Only the following PBB Epipe and I-VPLS SAP types are eligible to activate MVRP declarations:

- dot1q: for example, 1/1/2:100
- qinq or qinq default: for example, 1/1/1:100.1 and respectively 1/1/1:100.*, respectively; the outer VLAN 100 is used as MVRP attribute as long as it belongs to the MVRP range configured for the port
- null port and dot1q default cannot be used

Examples of steps required to activate service connectivity for VLAN 100 using MVRP follows.

In the data VPLS instance (VLAN 100) controlled by MVRP, on the QinQ switch, example:

```
config>service>vpls 100
        – sap 9/1/1:10 //UNI sap using CVID 10 as service delimiter
            – no shutdown
```

In I-VPLS on PBB BEB, example:

```
config>service>vpls 1000 i-vpls
        – sap 8/1/2:100 //sap (using MVRP VLAN 100 on endstation port in M-VPLS
            – no shutdown
```

3.4.4.4 MVRP control plane

MVRP is based on the IEEE 802.1ak MRP specification where STP is the supported method to be used for loop avoidance in a native Ethernet environment. M-VPLS and the associated MSTP (or P-MSTP) control plane provides the loop avoidance component in the Nokia implementation. Nokia MVRP may also be used in a non-MSTP, loop-free topology.

3.4.4.5 STP-MVRP interaction

Table 14: MSTP and MVRP interaction table shows the expected interaction between STP (MSTP or P-MSTP) and MVRP.

Table 14: MSTP and MVRP interaction table

Item	M-VPLS service xSTP	M-VPLS SAP STP	Register/declare data VPLS VLAN on M-VPLS SAP	DSFS (Data SAP Forwarding State) controlled by	Data path forwarding with MVRP enabled controlled by
1	(p)MSTP	Enabled	Based on M-VPLS SAP's MSTP forwarding state	MSTP only	DSFS and MVRP
2	(p)MSTP	Disabled	Based on M-VPLS SAP's operating state	—	MVRP
3	Disabled	Enabled or Disabled	Based on M-VPLS SAP's operating state	—	MVRP

Note:

- Running STP in data VPLS instances controlled by MVRP is not allowed.
- Running STP on MVRP-controlled end-station SAPs is not allowed.

3.4.4.5.1 Interaction between MVRP and instantiated SAP status

This section describes how MVRP reacts to changes in the instantiated SAP status.

There are a number of mechanisms that may generate operational or admin down status for the SAPs and VPLS instances controlled by MVRP:

1. Port down
2. MAC move
3. Port MTU too small
4. Service MTU too small

The shutdown of the whole instantiated VPLS or instantiated SAPs is disabled in both VPLS and VPLS SAP templates. The **no shutdown** option is automatically configured.

In the **port down** case, the MVRP is also operationally down on the port so no VLAN declaration occurs.

When MAC move is enabled in a data VPLS controlled by MVRP, in case a MAC move happens, one of the instantiated SAPs controlled by MVRP may be blocked. The SAP blocking by MAC move is not reported though to the MVRP control plane. As a result, MVRP keeps declaring and registering the related VLAN value on the control SAPs, including the one that shares the same port with the instantiate SAP blocked by MAC move, as long as MVRP conditions are met. For MVRP, an active control SAP is one that has MVRP enabled and MSTP is not blocking it for the VLAN value on the port. Also in the related data VPLS, one of the two conditions must be met for the declaration of the VLAN value: there must be either a local user SAP or at least one MVRP registration received on one of the control SAPs for that VLAN.

In the last two cases, VLAN attributes get declared or registered even when the instantiated SAP is operationally down, also with the MAC move case.

3.4.4.5.2 Using temporary flooding to optimize failover times

MVRP advertisements use the active topology, which may be controlled through loop avoidance mechanisms like MSTP. When the active topology changes as a result of network failures, the time it takes for MVRP to bring up the optimal service connectivity may be added on top of the regular MSTP convergence time. Full connectivity also depends on the time it takes for the system to complete flushing of bad MAC entries.

To minimize the effects of MAC flushing and MVRP convergence, a temporary flooding behavior is implemented. When enabled, the temporary flooding eliminates the time it takes to flush the MAC tables. In the initial implementation, the temporary flooding is initiated only on reception of an STP TCN.

While temporary flooding is active, all the frames received in the extended data VPLS context are flooded while the MAC flush and MVRP convergence take place. The extended data VPLS context comprises all instantiated trunk SAPs regardless of the MVRP activation status. A timer option is also available to configure a fixed period of time, in seconds, during which all traffic is flooded (BUM or known unicast). When the flood-time expires, traffic is delivered according to the regular FDB content. The timer value should be configured to allow auxiliary processes like MAC flush and MVRP to converge. The temporary flooding behavior applies to all VPLS types. MAC learning continues during temporary flooding.

Temporary flooding behavior is enabled using the **temp-flooding** command under **config**>**service**>**vpls** or **config**>**service**>**template**>**vpls-template** contexts and is supported in VPLS regardless of whether MVRP is enabled.

For temporary flooding in VPLS, the following rules apply:

- If discard-unknown is enabled, there is no temporary flooding.
- Temporary flooding while active applies also to static MAC entries; after the MAC FDB is flushed it reverts back to the static MAC entries.
- If MAC learning is disabled, fast or temporary flooding is still enabled.
- Temporary flooding is not supported in B-VPLS context when MMRP is enabled. The use of a flood-time procedure provides a better procedure for this kind of environment.

3.4.5 VPLS E-Tree services

This section describes VPLS E-Tree services.

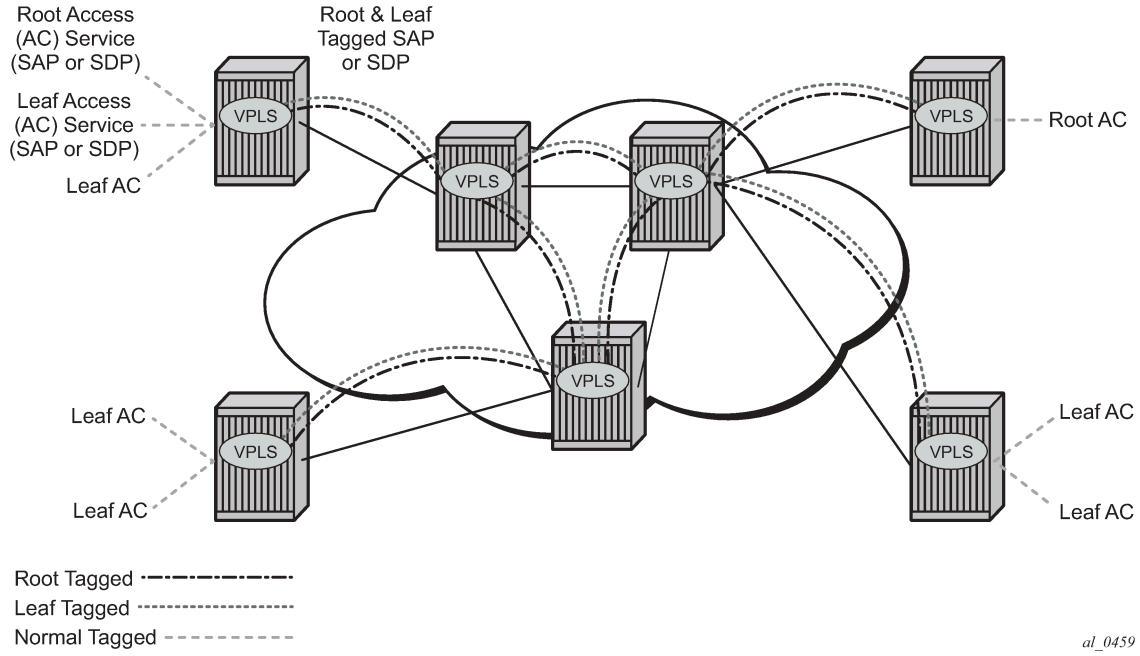
3.4.5.1 VPLS E-Tree services overview

The VPLS E-Tree service offers a VPLS service with Root and Leaf designated access SAPs and SDP bindings, which prevent any traffic flow from leaf to leaf directly. With a VPLS E-Tree, the split horizon group capability is inherent for leaf SAPs (or SDP bindings) and extends to all the remote PEs that are part of the same VPLS E-Tree service. This feature is based on IETF Draft *draft-ietf-l2vpn-vpls-pe-etree*.

A VPLS E-Tree service may support an arbitrary number of leaf access (leaf-ac) interfaces, root access (root-ac) interfaces, and root-leaf tagged (root-leaf-tag) interfaces. Leaf-ac interfaces are supported on SAPs and SDP binds and can only communicate with root-ac interfaces (also supported on SAPs and SDP binds). Leaf-ac to leaf-ac communication is not allowed. Root-leaf-tag interfaces (supported on SAPs and SDP bindings) are tagged with root and leaf VIDs to allow remote VPLS instances to enforce the E-Tree forwarding.

Figure 88: E-Tree service shows a network with two root-ac interfaces and several leaf-ac SAPs (also could be SDPs). The figure indicates two VIDs in use to each service within the service with no restrictions on the AC interfaces. The service guarantees no leaf-ac to leaf-ac traffic.

Figure 88: E-Tree service



3.4.5.2 Leaf-ac and root-ac SAPs

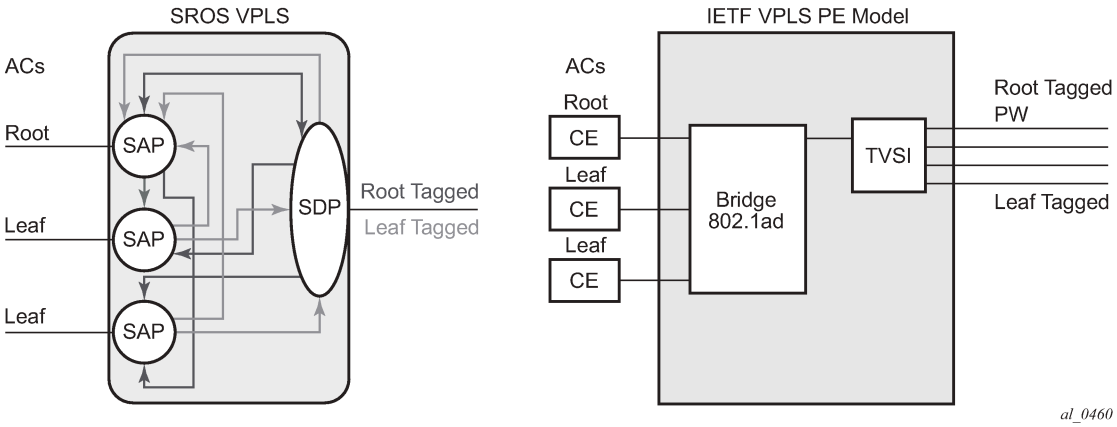
Figure 89: Mapping PE model to VPLS service shows the terminology used for E-Tree in IETF Draft *draft-ietf-l2vpn-vpls-pe-etree* and a mapping to SR OS terms.

An Ethernet service access SAP is characterized as either a leaf-ac or a root-ac for a VPLS E-Tree service. As far as SR OS is concerned, these are normal SAPs with either no tag (Null), priority tag, or dot1q or QinQ encapsulation on the frame. Functionally, a root-ac is a normal SAP and does not need to be differentiated from the regular SAPs except that it is associated with a root behavior in a VPLS E-Tree.

Leaf-ac SAPs have restrictions; for example, a SAP configured for a leaf-ac can never send frames to another leaf-ac directly (local) or through a remote node. Leaf-ac SAPs on the same VPLS instance behave as if they are part of a split horizon group (SHG) locally. Leaf-ac SAPs that are on other nodes need to have the traffic marked as originating "from a Leaf" in the context of the VPLS service when carried on PWs and SAPs with tags (VLANs).

Root-ac SAPs on the same VPLS can talk to any root-ac or leaf-ac.

Figure 89: Mapping PE model to VPLS service



3.4.5.3 Leaf-ac and root-ac SDP binds

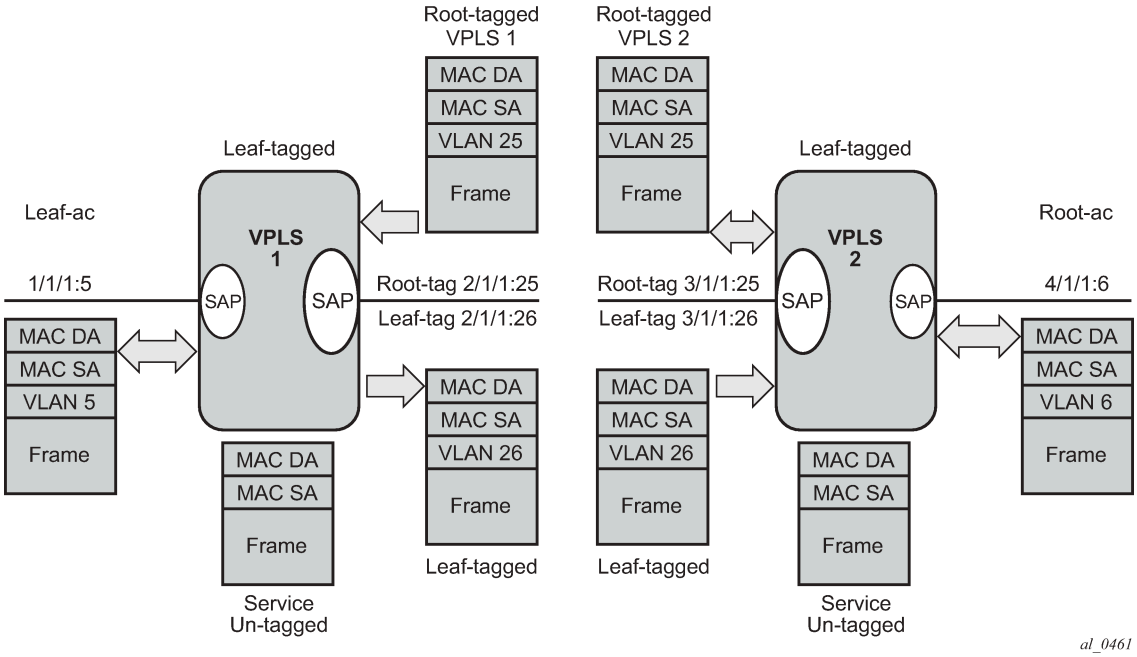
Untagged SDP binds for access can also be designated as root-ac or leaf-ac. This type of E-Tree interface is required for devices that do not support E-Tree, such as the 7210 SAS, to enable them to be connected with pseudowires. Such devices are root or leaf only and do not require having a tagged frame with a root or leaf indication.

3.4.5.4 Root-leaf-tag SAPs

Support on root-leaf-tag SAPs requires that the outer VID is overloaded to indicate root and leaf. To support the SR service model for a SAP, the ability to send and receive two different tags on a single SAP has been added. Figure 90: Leaf and root tagging dot1q shows the behavior when a root-ac and leaf-ac exchange traffic over a root-leaf-tag SAP. Although the figure shows two SAPs connecting VPLS instances 1 and 2, the CLI shows a single SAP with the format:

```
sap 2/1/1:25 root-leaf-tag leaf-tag 26 create
```

Figure 90: Leaf and root tagging dot1q



The root-leaf-tag SAP performs all of the operations for egress and ingress traffic for both tags (root and leaf):

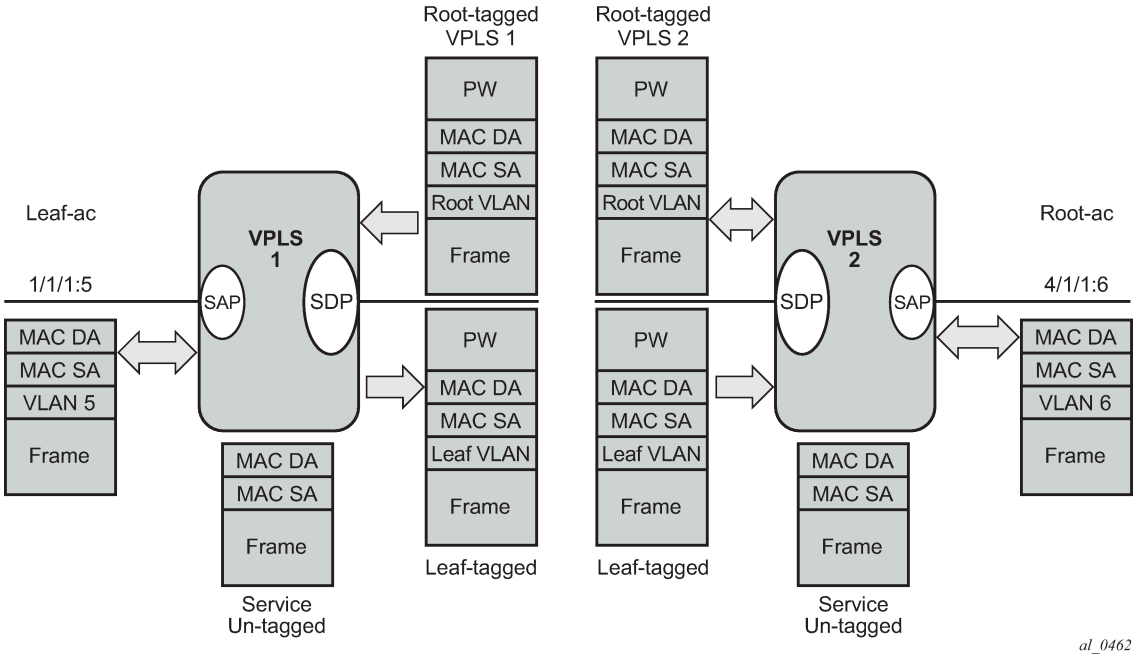
- When receiving a frame, the outer tag VID is compared against the configured root or leaf VIDs and the frame forwarded accordingly.
- When transmitting, the system adds a root VLAN (in the outer tag) on frames with an internal indication of Root, and a leaf VLAN on frames with an internal indication of Leaf.

3.4.5.5 Root-leaf-tag SDP binds

Typically, in a VPLS environment over MPLS, mesh and spoke-SDP binds interconnect the local VPLS instances to remote PEs. To support VPLS E-Tree, the root and leaf traffic is sent over the SDP bind using a fixed VLAN tag value. The SR OS implementation uses a fixed VLAN ID 1 for root and fixed VLAN ID 2 for leaf. The root and leaf tags are considered a global value and signaling is not supported. The vc-type on root-leaf-tag SDP binds must be VLAN. The vlan-vc-tag command is blocked in root-leaf-tag SDP-binds.

Figure 91: Leaf and root tagging PW shows the behavior when leaf-ac or root-ac interfaces exchange traffic over a root-leaf-tag SDP-binding.

Figure 91: Leaf and root tagging PW



3.4.5.6 Interaction between VPLS E-Tree services and other features

As a general rule, any CPM-generated traffic is always root traffic (STP, OAM, and so on) and any received control plane frame is marked with a root/leaf indication based on which E-Tree interface it arrived at. Some other particular feature interactions are as follows:

- ETH-CFM and E-Tree have limited conjunctive uses. ETH-CFM allows the operator to verify connectivity between the various endpoints of the service as well as execute troubleshooting and performance gathering functions. Continuity Checking, ETH-CC, is a method by which endpoints are configured and messages are passed between them at regular configured intervals. When CCM-enabled MEPs are configured, all MEPs in the same maintenance association, the grouping typically along the service lines, must know about every other endpoint in the service. This is the main principle behind continuity verification (all endpoints in communication).

  Although the maintenance points configured within the E-Tree service adhere to the forwarding rules of the Leaf and the Root, local population of the MEP database used by the ETH-CFM function may make it appear that the forwarding plane is broken when it is not. All MEPs that are locally configured within a service are automatically added to the local MEP database. However, because of the Leaf and Root forwarding rules, not all of these MEPs can receive the required peer CCM-message to avoid CCM Defect conditions. It is suggested, when deploying CCM enabled MEPs in an E-Tree configuration, these CCM-enabled MEPs are configured on Root entities. If Leaf access requires CCM verification, then down MEPs in separate maintenance associations should be configured. This consideration is only for operators who need to deploy CCM in E-Tree environments. No other ETH-CFM tools query or use this database.

- Legacy OAM commands (cpe-ping, mac-ping, mac-trace, mac-populate, and mac-purge) are not supported in E-Tree service contexts. Although some configuration may result in normal behavior for

some commands, not all commands or configurations yield the expected results. Standards-based ETH-CFM tools should be used in place of the proprietary legacy OAM command set.

- IGMP and PIM snooping for IPv4 work on VPLS E-Tree services. Routers should use root-ac interfaces so the multicast traffic can be delivered properly.
- xSTP is supported in VPLS E-Tree services; however, when configuring STP in VPLS E-Tree services, the following considerations apply:
  - STP must be carefully used so that STP does not block needless objects.
  - xSTP is not aware of the leaf-to-leaf topology; for example, for leaf-to-leaf traffic, even if there is no loop in the forwarding plane, xSTP may block leaf-ac SAPs or SDP binds.
  - Because xSTP is not aware of the root-leaf topology either, root ports may end up blocked before leaf interfaces.
  - When xSTP is used as an access redundancy mechanism, Nokia recommends connecting the dual-homed device to the same type of E-Tree AC, to avoid unexpected forwarding behaviors when xSTP converges.
- Redundancy mechanisms such as MC-LAG, SDP bind end-points, or BGP-MH are fully supported on VPLS E-Tree services. However, eth-tunnel SAPs or eth-ring control SAPs are not supported on VPLS E-Tree services.

3.5 Configuring a VPLS service with CLI

This section provides information to configure a VPLS service using the command line interface.

3.5.1 Basic configuration

The following fields require specific input (there are no defaults) to configure a basic VPLS service:

- Customer ID (for more information see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*)
- For a local service, configure two SAPs, specifying local access ports and encapsulation values.
- For a distributed service, configure a SAP and an SDP for each far-end node.

The following example shows a configuration of a local VPLS service on ALA-1.

```
*A:ALA-1>config>service>vpls# info
----------------------------------------------
...
        vpls 9001 customer 6 create
            description "Local VPLS"
            stp
                shutdown
            exit
            sap 1/2/2:0 create
                description "SAP for local service"
            exit
            sap 1/1/5:0 create
                description "SAP for local service"
            exit
            no shutdown
----------------------------------------------
```

```
*A:ALA-1>config>service>vpls#
```

The following example shows a configuration of a distributed VPLS service between ALA-1, ALA-2, and ALA-3.

```
*A:ALA-1>config>service# info
----------------------------------------------
...
        vpls 9000 customer 6 create
            shutdown
            description "This is a distributed VPLS."
        exit
...
----------------------------------------------
*A:ALA-1>config>service#

*A:ALA-2>config>service# info
----------------------------------------------
...
        vpls 9000 customer 6 create
            description "This is a distributed VPLS."
            stp
                shutdown
            exit
            sap 1/1/5:16 create
                description "VPLS SAP"
            exit
            spoke-sdp 2:22 create
            exit
            mesh-sdp 8:750 create
            exit
            no shutdown
        exit
...
----------------------------------------------
*A:ALA-2>config>service#

*A:ALA-3>config>service# info
----------------------------------------------
...
        vpls 9000 customer 6 create
            description "This is a distributed VPLS."
            stp
                shutdown
            exit
            sap 1/1/3:33 create
                description "VPLS SAP"
            exit
            spoke-sdp 2:22 create
            exit
            mesh-sdp 8:750 create
            exit
            no shutdown
        exit
...
----------------------------------------------
*A:ALA-3>config>service#
```

3.5.2 Common configuration tasks

About this task

This task provides a brief overview of the actions that must be performed to configure both local and distributed VPLS services and provides the CLI commands.

For VPLS services the procedure to be followed is presented below:

Procedure

Step 1. Associate VPLS service with a customer ID.

Step 2. Define SAPs:

- Select nodes and ports
- Optionally, select the following:
  - QoS policies other than the default (configured in config>qos context)
  - filter policies (configured in config>filter context)
  - accounting policy (configured in config>log context)

Step 3. Associate SDPs for (distributed services).

Step 4. Modify STP default parameters (optional) (see VPLS and spanning tree protocol)

Step 5. Enable the service.

3.5.3 Configuring VPLS components

Use the CLI syntax displayed in the following sections to configure VPLS components.

3.5.3.1 Creating a VPLS service

Use the following CLI syntax to create a VPLS service.

CLI syntax:

```
config>service# vpls service-id [customer customer-id] [vpn vpn-id] [m-vpls] [b-vpls | i-vpls]
 [create]
      description description-string
      no shutdown
```

The following example shows a VPLS configuration:

```
*A:ALA-1>config>service>vpls# info
----------------------------------------------
...
        vpls 9000 customer 6 create
            description "This is a distributed VPLS."
            stp
                shutdown
            exit
        exit
...
----------------------------------------------
*A:ALA-1>config>service>vpls#
```

3.5.3.2 Enabling MMRP

When the Multiple MAC Registration Protocol (MMRP) is enabled in the B-VPLS, it advertises the presence of the I-VPLS instances associated with this B-VPLS.

The following example shows a configuration with MMRP enabled.

```
*A:PE-B>config>service# info
----------------------------------------------
        vpls 11 customer 1 vpn 11 i-vpls create
            backbone-vpls 100:11
            exit
            stp
                shutdown
            exit
            sap 1/5/1:11 create
            exit
            sap 1/5/1:12 create
            exit
            no shutdown
        exit
        vpls 100 customer 1 vpn 100 b-vpls create
            service-mtu 2000
            stp
                shutdown
            exit
            mrp
                flood-time 10
                no shutdown
            exit
            sap 1/5/1:100 create
            exit
            spoke-sdp 3101:100 create
            exit
            spoke-sdp 3201:100 create
            exit
            no shutdown
        exit
----------------------------------------------
*A:PE-B>config>service#
```

Because I-VPLS 11 is associated with B-VPLS 100, MMRP advertises the group B-MAC 01:1e:83:00:00:0b) associated with I-VPLS 11 through a declaration on all the B-SAPs and B-SDPs. If the remote node also declares an I-VPLS 11 associated with its B-VPLS 10, then this results in a registration for the group B-MAC. This also creates the MMRP multicast tree (MFIB entries). In this case, sdp 3201:100 is connected to a remote node that declares the group B-MAC.

The following show commands display the current MMRP information for this scenario:

```
*A:PE-C# show service id 100 mrp
-------------------------------------------------------------------------------
MRP Information
-------------------------------------------------------------------------------
Admin State        : Up                Failed Register Cnt: 0
Max Attributes     : 1023              Attribute Count    : 1
Attr High Watermark: 95%               Attr Low Watermark : 90%
Flood Time         : 10
-------------------------------------------------------------------------------
*A:PE-C# show service id 100 mmrp mac
-------------------------------------------------------------------------------
```

```
SAP/SDP                                 MAC Address       Registered  Declared
-------------------------------------------------------------------------------
sap:1/5/1:100                           01:1e:83:00:00:0b No          Yes
sdp:3101:100                            01:1e:83:00:00:0b No          Yes
sdp:3201:100                            01:1e:83:00:00:0b Yes         Yes
-------------------------------------------------------------------------------


*A:PE-C# show service id 100 sdp 3201:100 mrp
-------------------------------------------------------------------------------
Sdp Id 3201:100 MRP Information
-------------------------------------------------------------------------------
Join Time          : 0.2 secs                Leave Time        : 3.0 secs
Leave All Time     : 10.0 secs               Periodic Time     : 1.0 secs
Periodic Enabled   : false
Rx Pdus            : 7                       Tx Pdus           : 23
Dropped Pdus       : 0
Rx New Event       : 0                       Rx Join-In Event  : 6
Rx In Event        : 0                       Rx Join Empty Evt : 1
Rx Empty Event     : 0                       Rx Leave Event    : 0
Tx New Event       : 0                       Tx Join-In Event  : 4
Tx In Event        : 0                       Tx Join Empty Evt : 19
Tx Empty Event     : 0                       Tx Leave Event    : 0
-------------------------------------------------------------------------------
SDP MMRP Information
-------------------------------------------------------------------------------
MAC Address       Registered        Declared
-------------------------------------------------------------------------------
01:1e:83:00:00:0b Yes               Yes
-------------------------------------------------------------------------------
Number of MACs=1 Registered=1 Declared=1
-------------------------------------------------------------------------------
*A:PE-C#


*A:PE-C#  show service id 100 mfib
===============================================================================
Multicast FIB, Service 100
===============================================================================
Source Address  Group Address         Sap/Sdp Id              Svc Id   Fwd/Blk
-------------------------------------------------------------------------------
*               01:1E:83:00:00:0B     sdp:3201:100            Local    Fwd
-------------------------------------------------------------------------------
Number of entries: 1
===============================================================================
*A:PE-C#
```

3.5.3.2.1 Enabling MAC move

The **mac-move** feature is useful to protect against undetected loops in your VPLS topology as well as the presence of duplicate MACs in a VPLS service. For example, if two clients in the VPLS have the same MAC address, the VPLS experiences a high re-learn rate for the MAC and shuts down the SAP or spoke-SDP when the threshold is exceeded.

Use the following CLI syntax to configure **mac-move** parameters.

CLI syntax:

```
config>service# vpls service-id [customer customer-id] [vpn vpn-id] [m-vpls]
    – mac-move
        – primary-ports
```

```
            – spoke-sdp
            – cumulative-factor
        – exit
        – secondary-ports
            – spoke-sdp
            – sap
        – exit
        – move-frequency frequency
        – retry-timeout timeout
        – no shutdown
```

The following example shows a **mac-move** configuration:

```
*A:ALA-2009>config>service>vpls>mac-move# show service id 500 mac-move
===============================================================================
Service Mac Move Information
===============================================================================
Service Id        : 500                 Mac Move          : Enabled
Primary Factor    : 4                   Secondary Factor  : 2
Mac Move Rate     : 2                   Mac Move Timeout  : 10
Mac Move Retries  : 3
-------------------------------------------------------------------------------
SAP Mac Move Information: 2/1/3:501
-------------------------------------------------------------------------------
Admin State       : Up                  Oper State        : Down
Flags              : RelearnLimitExceeded
Time to come up   : 1 seconds           Retries Left      : 1
Mac Move          : Blockable           Blockable Level   : Tertiary
-------------------------------------------------------------------------------
SAP Mac Move Information: 2/1/3:502
-------------------------------------------------------------------------------
Admin State       : Up                  Oper State        : Up
Flags              : None
Time to RetryReset: 267 seconds         Retries Left      : none
Mac Move          : Blockable           Blockable Level   : Tertiary
-------------------------------------------------------------------------------
SDP Mac Move Information: 21:501
-------------------------------------------------------------------------------
Admin State       : Up                  Oper State        : Up
Flags              : None
Time to RetryReset: never               Retries Left      : 3
Mac Move          : Blockable           Blockable Level   : Secondary
-------------------------------------------------------------------------------
SDP Mac Move Information: 21:502
-------------------------------------------------------------------------------
Admin State       : Up                  Oper State        : Down
Flags              : RelearnLimitExceeded
Time to come up   : never               Retries Left      : none
Mac Move          : Blockable           Blockable Level   : Tertiary
===============================================================================
*A:*A:ALA-2009>config>service>vpls>mac-move#
```

3.5.3.2.2 Configuring STP bridge parameters in a VPLS

Modifying some of the Spanning Tree Protocol parameters allows the operator to balance STP between resiliency and speed of convergence extremes. Modifying particular parameters, as follows, must be done in the constraints of the following two formulas:

2 x (Bridge_Forward_Delay - 1.0 seconds) ≥ Bridge_Max_Age Bridge_Max_Age ≥ 2 x (Bridge_Hello0_Time + 1.0 seconds)

The following STP parameters can be modified at VPLS level:

- Bridge STP admin state
- Mode
- Bridge priority
- Max age
- Forward delay
- Hello time
- MST instances
- MST max hops
- MST name
- MST revision

STP always uses the locally configured values for the first three parameters (Admin State, Mode, and Priority).

For the parameters Max Age, Forward Delay, Hello Time, and Hold Count, the locally configured values are only used when this bridge has been elected root bridge in the STP domain; otherwise, the values received from the root bridge are used. The exception to this rule is: when STP is running in RSTP mode, the Hello Time is always taken from the locally configured parameter. The other parameters are only used when running mode MSTP.

3.5.3.2.2.1 Bridge STP admin state

The administrative state of STP at the VPLS level is controlled by the **shutdown** command.

When STP on the VPLS is administratively disabled, any BPDUs are forwarded transparently through the 7450 ESS, 7750 SR, or 7950 XRS. When STP on the VPLS is administratively enabled, but the administrative state of a SAP or spoke-SDP is down, BPDUs received on such a SAP or spoke-SDP are discarded.

CLI syntax:

```
config>service>vpls service-id# stp
no shutdown
```

3.5.3.2.2.2 Mode

To be compatible with the different iterations of the IEEE 802.1D standard, the 7450 ESS, 7750 SR, and 7950 XRS support several variants of the Spanning Tree protocol:

rstp	Rapid Spanning Tree Protocol (RSTP) compliant with IEEE 802.1D-2004 - default mode.
dot1w	compliant with IEEE 802.1w
comp-dot1w	operation as in RSTP but backwards compatible with IEEE 802.1w (this mode was introduced for interoperability with some MTU types)

mstp	compliant with the Multiple Spanning Tree Protocol specified in IEEE 802.1Q REV/D5.0-09/2005. This mode of operation is only supported in an M-VPLS
pmstp	compliant with the Multiple Spanning Tree Protocol specified in IEEE 802.1Q REV/D3.0-04/2005 but with some changes to make it backwards compatible to 802.1Q 2003 edition and IEEE 802.1w

See section Spanning tree operating modes for more information about these modes.

CLI syntax:

```
config>service>vpls service-id# stp
mode {rstp | comp-dot1w | dot1w | mstp}
```

The default variant of the Spanning Tree protocol is rstp.

3.5.3.2.2.3 Bridge priority

The **bridge-priority** command is used to populate the priority portion of the bridge ID field within outbound BPDUs (the most significant 4 bits of the bridge ID). It is also used as part of the decision process when determining the best BPDU between messages received and sent. When running MSTP, this is the bridge priority used for the CIST.

All values are truncated to multiples of 4096, conforming with IEEE 802.1t and 802.1D-2004.

CLI syntax:

```
config>service>vpls service-id# stp
priority bridge-priority
```

Range

1 to 65535

Default

32768

Restore Default

no priority

3.5.3.2.2.4 Max age

The **max-age** command indicates how many hops a BPDU can traverse the network starting from the root bridge. The message age field in a BPDU transmitted by the root bridge is initialized to 0. Each other bridge takes the message_age value from BPDUs received on their root port and increment this value by 1. Therefore, the message_age reflects the distance from the root bridge. BPDUs with a message age exceeding max-age are ignored.

STP uses the max-age value configured in the root bridge. This value is propagated to the other bridges by the BPDUs. The default value of **max-age** is 20. This parameter can be modified within a range of 6 to 40, limited by the standard STP parameter interaction formulas.

CLI syntax:

```
config>service>vpls service-id# stp
```

```
max-age max-info-age
```

Range

6 to 40 seconds

Default

20 seconds

Restore Default

no max-age

3.5.3.2.2.5 Forward delay

RSTP, as defined in the IEEE 802.1D-2004 standards, normally transitions to the forwarding state by a handshaking mechanism (rapid transition), without any waiting times. If handshaking fails (for example, on shared links, as follows), the system falls back to the timer-based mechanism defined in the original STP (802.1D-1998) standard.

A shared link is a link with more than two Ethernet bridges (for example, a shared 10/100BaseT segment). The port-type command is used to configure a link as point-to-point or shared (see section SAP link type).

For timer-based transitions, the 802.1D-2004 standard defines an internal variable forward-delay, which is used in calculating the default number of seconds that a SAP or spoke-SDP spends in the discarding and learning states when transitioning to the forwarding state. The value of the forward-delay variable depends on the STP operating mode of the VPLS instance:

- In RSTP mode, but only when the SAP or spoke-SDP has not fallen back to legacy STP operation, the value configured by the **hello-time** command is used.
- In all other situations, the value configured by the **forward-delay** command is used.

CLI syntax:

```
config>service>vpls service-id# stp
forward-delay seconds
```

Range

4 to 30 seconds

Default

15 seconds

Restore Default

no forward-delay

3.5.3.2.2.6 Hello time

The **hello-time** command configures the Spanning Tree Protocol (STP) hello time for the Virtual Private LAN Service (VPLS) STP instance.

The **seconds** parameter defines the default timer value that controls the sending interval between BPDU configuration messages by this bridge, on ports where this bridge assumes the designated role.

The active hello time for the spanning tree is determined by the root bridge (except when the STP is running in RSTP mode, then the hello time is always taken from the locally configured parameter).

The configured hello-time value can also be used to calculate the bridge forward delay; see Forward delay.

CLI syntax:

```
config>service>vpls service-id# stp
hello-time hello-time
```

Range

1 to 10 seconds

Default

2 seconds

Restore Default

no hello-time

3.5.3.2.2.7 Hold count

The **hold-count** command configures the peak number of BPDUs that can be transmitted in a period of one second.

CLI syntax:

```
config>service>vpls service-id# stp
hold-count count-value
```

Range

1 to 10

Default

6

Restore Default

no hold-count

3.5.3.2.2.8 MST instances

You can create up to 15 mst-instances. They can range from 1 to 4094. By changing path-cost and priorities, you can ensure that each instance forms its own tree within the region, therefore ensure that different VLANs follow different paths.

You can assign non-overlapping VLAN ranges to each instance. VLANs that are not assigned to an instance are implicitly assumed to be in instance 0, which is also called the CIST. This CIST cannot be deleted or created.

The parameters that can be defined per instance are **mst-priority** and **vlan-range**.

mst-priority

the bridge-priority for this specific mst-instance. It follows the same rules as bridge-priority. For the CIST, the bridge-priority is used.

vlan-range

the VLANs are mapped to this specific mst-instance. If no VLAN-ranges are defined in any mst-instances, then all VLANs are mapped to the CIST.

3.5.3.2.2.9 MST max hops

The mst-max-hops command defines the maximum number of hops the BPDU can traverse inside the region. Outside the region, max-age is used.

3.5.3.2.2.10 MST name

The MST name defines the name that the operator gives to a region. Together with MST revision and the VLAN to mst-instance mapping, it forms the MST configuration identifier. Two bridges that have the same MST configuration identifier form a region if they exchange BPDUs.

3.5.3.2.2.11 MST revision

The MST revision together with MST-name and VLAN to MST-instance mapping define the MST configuration identifier. Two bridges that have the same MST configuration identifier form a region if they exchange BPDUs.

3.5.3.3 Configuring GSMP parameters

The following parameters must be configured in order for GSMP to function:

- One or more GSMP sessions
- One or more ANCP policies
- For basic subscriber management only, ANCP static maps
- For enhanced subscriber management only, associate subscriber profiles with ANCP policies

Use the following CLI syntax to configure GSMP parameters.

CLI syntax:

```
config>service>vpls# gsmp
    – group name [create]
        – ancp
            – dynamic-topology-discover
            – oam
        – description description-string
        – hold-multiplier multiplier
        – keepalive seconds
        – neighbor ip-address [create]
            – description v
            – local-address ip-address
            – priority-marking dscp dscp-name
            – priority-marking prec ip-prec-value
            – [no] shutdown
        – [no] shutdown
    – [no] shutdown
```

This example shows a GSMP group configuration.

```
A:ALA-48>config>service>vpls>gsmp# info
----------------------------------------------
                group "group1" create
```

```
                    description "test group config"
                    neighbor 10.10.10.104 create
                        description "neighbor1 config"
                        local-address 10.10.10.103
                        no shutdown
                    exit
                    no shutdown
                exit
                no shutdown
-------------------------------------------
A:ALA-48>config>service>vpls>gsmp#
```

3.5.3.4 Configuring a VPLS SAP

A default QoS policy is applied to each ingress and egress SAP. Additional QoS policies can be configured in the **config**>**qos** context. There are no default filter policies. Filter policies are configured in the **config**>**filter** context and must be explicitly applied to a SAP. Use the following CLI syntax to create:

- Local VPLS SAPs
- Distributed VPLS SAPs

3.5.3.4.1 Local VPLS SAPs

To configure a local VPLS service, enter the **sap sap-id** command twice with different port IDs in the same service configuration.

The following example shows a local VPLS configuration:

```
*A:ALA-1>config>service# info
-------------------------------------------
...
        vpls 90001 customer 6 create
            description "Local VPLS"
            stp
                shutdown
            exit
            sap 1/2/2:0 create
                description "SAP for local service"
            exit
            sap 1/1/5:0 create
                description "SAP for local service"
            exit
            no shutdown
        exit
-------------------------------------------
*A:ALA-1>config>service#
*A:ALA-1>config>service# info
-------------------------------------------
       vpls 1150 customer 1 create
            fdb-table-size 1000
            fdb-table-low-wmark 5
            fdb-table-high-wmark 80
            local-age 60
            stp
                shutdown
            exit
            sap 1/1/1:1155 create
            exit
```

```
            sap 1/1/2:1150 create
            exit
            no shutdown
        exit
-------------------------------------------
*A:ALA-1>config>service#
```

3.5.3.4.2 Distributed VPLS SAPs

To configure a distributed VPLS service, you must configure service entities on originating and far-end nodes. You must use the same service ID on all ends (for example, create a VPLS service ID 9000 on ALA-1, ALA-2, and ALA-3). A distributed VPLS consists of a SAP on each participating node and an SDP bound to each participating node.

For SDP configuration information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*. For SDP binding information, see Configuring SDP bindings.

The following example shows a configuration of VPLS SAPs configured for ALA-1, ALA-2, and ALA-3.

```
*A:ALA-1>config>service# info
----------------------------------------------
...
        vpls 9000 customer 6 vpn 750 create
            description "Distributed VPLS services."
            stp
                shutdown
            exit
            sap 1/2/5:0 create
                description "VPLS SAP"
                multi-service-site "West"
            exit
        exit
...
----------------------------------------------
*A:ALA-1>config>service#


*A:ALA-2>config>service# info
----------------------------------------------
...
        vpls 9000 customer 6 vpn 750 create
            description "Distributed VPLS services."
            stp
                shutdown
            exit
            sap 1/1/2:22 create
                description "VPLS SAP"
                multi-service-site "West"
            exit
        exit
...
----------------------------------------------
*A:ALA-2>config>service#


*A:ALA-3>config>service# info
----------------------------------------------
...
        vpls 9000 customer 6 vpn 750 create
            description "Distributed VPLS services."
            stp
```

```
                shutdown
            exit
            sap 1/1/3:33 create
                description "VPLS SAP"
                multi-service-site "West"
            exit
        exit
...
----------------------------------------------
*A:ALA-3>config>service#
```

3.5.3.4.3 Configuring SAP-specific STP parameters

When a VPLS has STP enabled, each SAP within the VPLS has STP enabled by default. Subsequent sections describe SAP-specific STP parameters in detail.

3.5.3.4.3.1 SAP STP administrative state

The administrative state of STP within a SAP controls how BPDUs are transmitted and handled when received. The allowable states are:

- **SAP admin up**

  The default administrative state is up for STP on a SAP. BPDUs are handled in the normal STP manner on a SAP that is administratively up.

- **SAP admin down**

  An administratively down state allows a service provider to prevent a SAP from becoming operationally blocked. BPDUs do not originate out the SAP toward the customer.

  If STP is enabled on VPLS level, but disabled on the SAP, received BPDUs are discarded. Discarding the incoming BPDUs allows STP to continue to operate normally within the VPLS service while ignoring the down SAP. The specified SAP is always in an operationally forwarding state.

Note: The administratively down state allows a loop to form within the VPLS.

CLI syntax:

```
config>service>vpls>sap>stp#
[no] shutdown
```

Range	shutdown or no shutdown
Default	no shutdown (SAP admin up)

3.5.3.4.3.2 SAP virtual port number

The virtual port number uniquely identifies a SAP within configuration BPDUs. The internal representation of a SAP is unique to a system and has a reference space much bigger than the 12 bits definable in a configuration BPDU. STP takes the internal representation value of a SAP and identifies it with its own

virtual port number that is unique to every other SAP defined on the VPLS. The virtual port number is assigned at the time that the SAP is added to the VPLS.

Because the order in which SAPs are added to the VPLS is not preserved between reboots of the system, the virtual port number may change between restarts of the STP instance. To achieve consistency after a reboot, the virtual port number can be specified explicitly.

CLI syntax:

```
config>service>vpls>sap# stp
port-num number
```

Range

1 to 2047

Default

(automatically generated)

Restore Default

no port-num

3.5.3.4.3.3 SAP priority

SAP priority allows a configurable tie-breaking parameter to be associated with a SAP. When configuration BPDUs are being received, the configured SAP priority is used in some circumstances to determine whether a SAP is designated or blocked. These are the values used for CIST when running MSTP for the 7450 ESS or 7750 SR.

In traditional STP implementations (802.1D-1998), this field is called the port priority and has a value of 0 to 255. This field is coupled with the port number (0 to 255 also) to create a 16-bit value. In the latest STP standard (802.1D-2004), only the upper 4 bits of the port priority field are used to encode the SAP priority. The remaining 4 bits are used to extend the port ID field into a 12-bit virtual port number field. The virtual port number uniquely references a SAP within the STP instance. See SAP virtual port number for more information about the virtual port number.

STP computes the actual SAP priority by taking the configured priority value and masking out the lower four bits. The result is the value that is stored in the SAP priority parameter. For example, if a value of 0 was entered, masking out the lower 4 bits would result in a parameter value of 0. If a value of 255 was entered, the result would be 240.

The default value for SAP priority is 128. This parameter can be modified within a range of 0 to 255; 0 being the highest priority. Masking causes the values actually stored and displayed to be 0 to 240, in increments of 16.

CLI syntax:

```
config>service>vpls>sap>stp#
priority stp-priority
```

Range

0 to 255 (240 largest value, in increments of 16)

Default

128

Restore Default

no priority

3.5.3.4.3.4 SAP path cost

The SAP path cost is used by STP to calculate the path cost to the root bridge. The path cost in BPDUs received on the root port is incremented with the configured path cost for that SAP. When BPDUs are sent out of other egress SAPs, the newly calculated root path cost is used. These are the values used for CIST when running MSTP.

STP suggests that the path cost is defined as a function of the link bandwidth. Because SAPs are controlled by complex queuing dynamics, in the 7450 ESS, 7750 SR, and 7950 XRS the STP path cost is a purely static configuration.

The default value for SAP path cost is 10. This parameter can be modified within a range of 1 to 65535; 1 being the lowest cost.

CLI syntax:

```
config>service>vpls>sap>stp#
path-cost sap-path-cost
```

Range

1 to 200000000

Default

10

Restore Default

no path-cost

3.5.3.4.3.5 SAP edge port

The SAP edge-port command is used to reduce the time it takes a SAP to reach the forwarding state when the SAP is on the edge of the network, and therefore has no further STP bridge to handshake with.

The **edge-port** command is used to initialize the internal OPER_EDGE variable. At any time, when OPER_EDGE is false on a SAP, the normal mechanisms are used to transition to the forwarding state (see Forward delay). When OPER_EDGE is true, STP assumes that the remote end agrees to transition to the forwarding state without actually receiving a BPDU with an agreement flag set.

The OPER_EDGE variable is dynamically set to false if the SAP receives BPDUs (the configured edge-port value does not change). The OPER_EDGE variable is dynamically set to true if auto-edge is enabled and STP concludes there is no bridge behind the SAP.

When STP on the SAP is administratively disabled, and re-enabled, the OPER_EDGE is re-initialized to the value configured for edge-port.

Valid values for SAP edge-port are enabled and disabled, with disabled being the default.

CLI syntax:

```
config>service>vpls>sap>stp#
[no] edge-port
```

Default

no edge-port

3.5.3.4.3.6 SAP auto edge

The SAP **edge-port** command is used to instruct STP to dynamically decide whether the SAP is connected to another bridge.

If auto-edge is enabled, and STP concludes there is no bridge behind the SAP, the OPER_EDGE variable is dynamically set to true. If auto-edge is enabled, and a BPDU is received, the OPER_EDGE variable is dynamically set to true (see SAP edge port).

Valid values for SAP auto-edge are enabled and disabled with enabled being the default.

CLI syntax:

```
config>service>vpls>sap>stp#
[no] auto-edge
```

Default

auto-edge

3.5.3.4.3.7 SAP link type

The SAP **link-type** parameter instructs STP on the maximum number of bridges behind this SAP. If there is only a single bridge, transitioning to forwarding state is based on handshaking (fast transitions). If more than two bridges are connected by a shared media, their SAPs should all be configured as shared, and timer-based transitions are used.

Valid values for SAP link-type are shared and pt-pt with pt-pt, being the default.

CLI syntax:

```
config>service>vpls>sap>stp#
link-type {pt-pt | shared}
```

Default

link-type pt-pt

Restore Default

no link-type

3.5.3.4.4 STP SAP operational states

The operational state of STP within a SAP controls how BPDUs are transmitted and handled when received. Subsequent sections describe STP SAP operational states.

3.5.3.4.4.1 Operationally disabled

Operationally disabled is the normal operational state for STP on a SAP in a VPLS that has any of the following conditions:

- VPLS state administratively down

- SAP state administratively down
- SAP state operationally down

If the SAP enters the operationally up state with the STP administratively up and the SAP STP state is up, the SAP transitions to the STP SAP discarding state.

When, during normal operation, the router detects a downstream loop behind a SAP or spoke-SDP, BPDUs can be received at a very high rate. To recover from this situation, STP transitions the SAP to disabled state for the configured forward-delay duration.

3.5.3.4.4.2 Operationally discarding

A SAP in the discarding state only receives and sends BPDUs, building the local correct STP state for each SAP while not forwarding actual user traffic. The duration of the discarding state is described in section Forward delay.

Note: In previous versions of the STP standard, the discarding state was called a blocked state.

3.5.3.4.4.3 Operationally learning

The learning state allows population of the MAC forwarding table before entering the forwarding state. In this state, no user traffic is forwarded.

3.5.3.4.4.4 Operationally forwarding

Configuration BPDUs are sent out of a SAP in the forwarding state. Layer 2 frames received on the SAP are source learned and destination forwarded according to the FDB. Layer 2 frames received on other forwarding interfaces and destined for the SAP are also forwarded.

3.5.3.4.4.5 SAP BPDU encapsulation state

IEEE 802.1d (referred as Dot1d) and Cisco's per VLAN Spanning Tree (PVST) BPDU encapsulations are supported on a per-SAP basis for the 7450 ESS and 7750 SR. STP is associated with a VPLS service like PVST is associated per VLAN. The main difference resides in the Ethernet and LLC framing and a type-length-value (TLV) field trailing the BPDU.

Table 15: Spoke SDP BPDU encapsulation states shows differences between Dot1d and PVST Ethernet BPDU encapsulations based on the interface encap-type field.

Each SAP has a Read-Only operational state that shows which BPDU encapsulation is currently active on the SAP. The states are:

- **dot1d**

  This state specifies that the switch is currently sending IEEE 802.1d standard BPDUs. The BPDUs are tagged or non-tagged based on the encapsulation type of the egress interface and the encapsulation value defined in the SAP. A SAP defined on an interface with encapsulation type dot1q continues in the dot1d BPDU encapsulation state until a PVST encapsulated BPDU is received, in which case, the SAP converts to the PVST encapsulation state. Each received BPDU must be properly IEEE 802.1q tagged

if the interface encapsulation type is defined as dot1q. PVST BPDUs is silently discarded if received when the SAP is on an interface defined with a null encapsulation type.

- **PVST**

  This state specifies that the switch is currently sending proprietary encapsulated BPDUs. PVST BPDUs are only supported on Ethernet interfaces with the encapsulation type set to dot1q. The SAP continues in the PVST BPDU encapsulation state until a dot1d encapsulated BPDU is received, in which case, the SAP reverts to the dot1d encapsulation state. Each received BPDU must be properly IEEE 802.1q tagged with the encapsulation value defined for the SAP. PVST BPDUs are silently discarded if received when the SAP is on an interface defined with a null encapsulation type.

Dot1d is the initial and only SAP BPDU encapsulation state for SAPs defined on Ethernet interface with encapsulation type set to null.

Each transition between encapsulation types optionally generates an alarm that can be logged and optionally transmitted as an SNMP trap on the 7450 ESS or 7750 SR.

3.5.3.4.5 Configuring VPLS SAPs with split horizon

To configure a VPLS service with a split horizon group, add the **split-horizon-group** parameter when creating the SAP. Traffic arriving on a SAP within a split horizon group is not copied to other SAPs in the same split horizon group.

The following example shows a VPLS configuration with split horizon enabled:

```
*A:ALA-1>config>service# info
----------------------------------------------
...
    vpls 800 customer 6001 vpn 700 create
        description "VPLS with split horizon for DSL"
        stp
            shutdown
        exit
        sap 1/1/3:100 split-horizon-group DSL-group1 create
            description "SAP for residential bridging"
        exit
        sap 1/1/3:200 split-horizon-group DSL-group1 create
            description "SAP for residential bridging"
        exit
        split-horizon-group DSL-group1
            description "Split horizon group for DSL"
        exit
        no shutdown
    exit
...
----------------------------------------------
*A:ALA-1>config>service#
```

3.5.3.4.6 Configuring MAC learning protection

To configure MAC learning protection, configure split horizon, MAC protection, and SAP parameters on the 7450 ESS or 7750 SR.

The following example shows a VPLS configuration with split horizon enabled:

```
A:ALA-48>config>service>vpls# info
```

```
------------------------------------------
            description "local VPLS"
            split-horizon-group "DSL-group1" create
                restrict-protected-src
                restrict-unprotected-dst
            exit
            mac-protect
                 mac ff:ff:ff:ff:ff:ff
            exit
            sap 1/1/9:0 create
                ingress
                    scheduler-policy "SLA1"
                    qos 100 shared-queuing
                exit
                egress
                    scheduler-policy "SLA1"
                    filter ip 10
                exit
                restrict-protected-src
                arp-reply-agent
                host-connectivity-verify source-ip 10.144.145.1
            exit
...
------------------------------------------
A:ALA-48>config>service>vpls#
```

3.5.3.5 Configuring SAP subscriber management parameters

Use the following CLI syntax to configure subscriber management parameters on a VPLS service SAP on the 7450 ESS and 7750 SR. The policies and profiles that are referenced in the **def-sla-profile**, **def-sub-profile**, **non-sub-traffic**, and **sub-ident-policy** commands must already be configured in the **config**>**subscr-mgmt** context.

CLI syntax:

```
config>service>vpls service-id
    – sap sap-id [split-horizon-group group-name]
        – sub-sla-mgmt
            – def-sla-profile default-sla-profile-name
            – def-sub-profile default-subscriber-profile-name
            – mac-da-hashing
            – multi-sub-sap [number-of-sub]
            – no shutdown
            – single-sub-parameters
                – non-sub-traffic sub-profile sub-profile-name sla-profile sla-profile-name
 [subscriber sub-ident-string]
                – profiled-traffic-only
            – sub-ident-policy sub-ident-policy-name
```

The following example shows a subscriber management configuration:

```
A:ALA-48>config>service>vpls#
------------------------------------------
            description "Local VPLS"
            stp
                shutdown
            exit
            sap 1/2/2:0 create
                description "SAP for local service"
                sub-sla-mgmt
```

```
                    def-sla-profile "sla-profile1"
                    sub-ident-policy "SubIdent1"
                exit
            exit
            sap 1/1/5:0 create
                description "SAP for local service"
            exit
            no shutdown
----------------------------------------------
A:ALA-48>config>service>vpls#
```

3.5.3.6 MSTP control over Ethernet tunnels

When MSTP is used to control VLANs, a range of VLAN IDs is normally used to specify the VLANs to be controlled on the 7450 ESS and 7750 SR.

If an Ethernet tunnel SAP is to be controlled by MSTP, the Ethernet tunnel SAP ID needs to be within the VLAN range specified under the mst-instance.

```
vpls 400 customer 1 m-vpls create
            stp
                mode mstp
                mst-instance 111 create
                    vlan-range 1-100
                exit
                mst-name "abc"
                mst-revision 1
                no shutdown
            exit
            sap 1/1/1:0 create // untagged
            exit
            sap eth-tunnel-1 create
            exit
            no shutdown
        exit
        vpls 401 customer 1 create
            stp
                shutdown
            exit
            sap 1/1/1:12 create
            exit
            sap eth-tunnel-1:12 create
                // Ethernet tunnel SAP ID 12 falls within the VLAN
                // range for mst-instance 111
                eth-tunnel
                    path 1 tag 1000
                    path 8 tag 2000
                exit
            exit
            no shutdown
        exit
```

3.5.3.7 Configuring SDP bindings

VPLS provides scaling and operational advantages. A hierarchical configuration eliminates the need for a full mesh of VCs between participating devices. Hierarchy is achieved by enhancing the base VPLS core mesh of VCs with access VCs (spoke) to form two tiers. Spoke SDPs are generally created between Layer

2 switches and placed at the Multi-Tenant Unit (MTU). The PE routers are placed at the service provider's Point of Presence (POP). Signaling and replication overhead on all devices is considerably reduced.

A spoke SDP is treated like the equivalent of a traditional bridge port where flooded traffic received on the spoke-SDP is replicated on all other "ports" (other spoke and mesh SDPs or SAPs) and not transmitted on the port it was received (unless a split horizon group was defined on the spoke-SDP; see section Configuring VPLS spoke SDPs with split horizon).

A spoke SDP connects a VPLS service between two sites and, in its simplest form, could be a single tunnel LSP. A set of ingress and egress VC labels are exchanged for each VPLS service instance to be transported over this LSP. The PE routers at each end treat this as a virtual spoke connection for the VPLS service in the same way as the PE-MTU connections. This architecture minimizes the signaling overhead and avoids a full mesh of VCs and LSPs between the two metro networks.

A mesh SDP bound to a service is logically treated like a single bridge "port" for flooded traffic where flooded traffic received on any mesh SDP on the service is replicated to other "ports" (spoke SDPs and SAPs) and not transmitted on any mesh SDPs.

A VC-ID can be specified with the SDP-ID. The VC-ID is used instead of a label to identify a virtual circuit. The VC-ID is significant between peer SRs on the same hierarchical level. The value of a VC-ID is conceptually independent from the value of the label or any other datalink specific information of the VC.

Figure 92: SDPs — unidirectional tunnels shows an example of a distributed VPLS service configuration of spoke and mesh SDPs (unidirectional tunnels) between routers and MTUs.

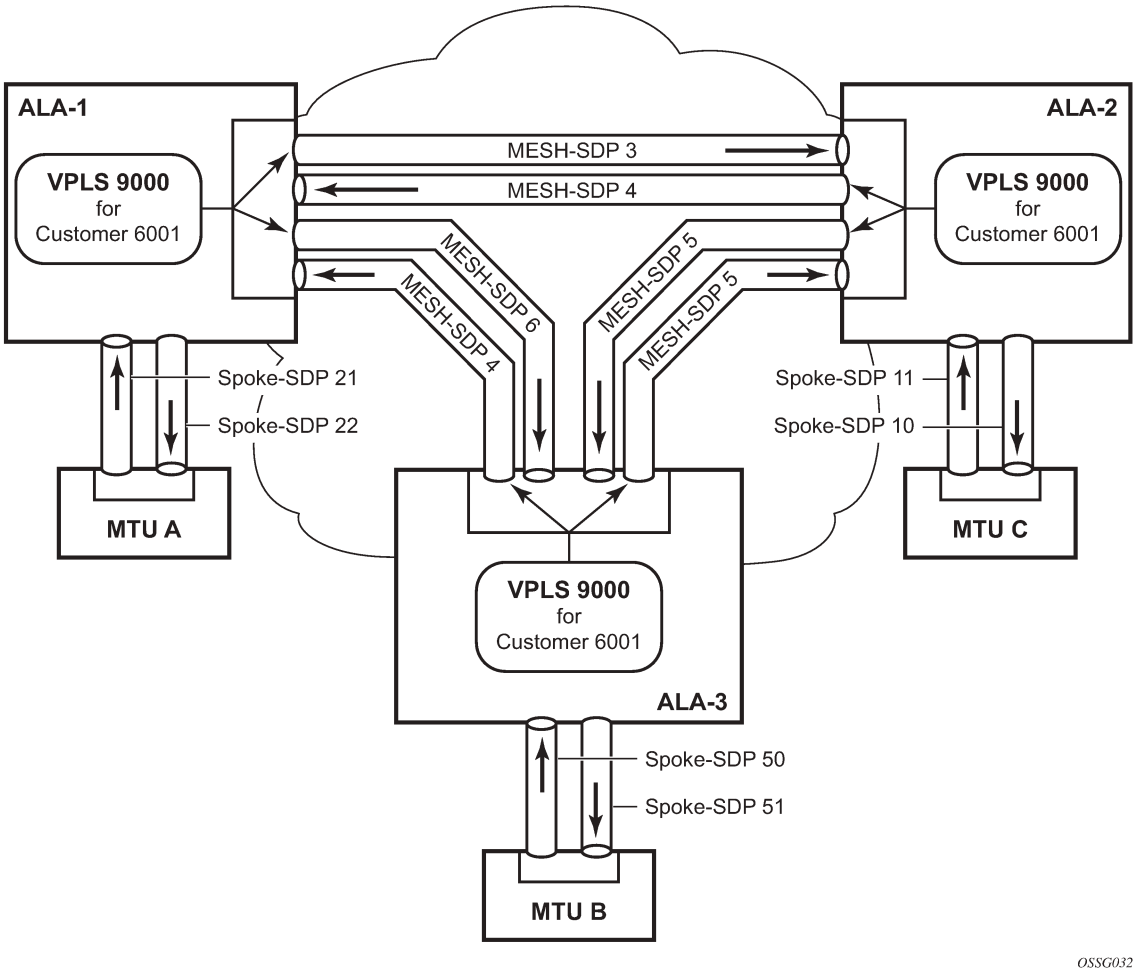
3.5.3.8 Configuring overrides on service SAPs

The following output shows a service SAP queue override configuration example:

```
*A:ALA-48>config>service>vpls>sap# info
----------------------------------------------
...
exit
ingress
    scheduler-policy "SLA1"
    scheduler-override
        scheduler "sched1" create
            parent weight 3 cir-weight 3
        exit
    exit
    policer-control-policy "SLA1-p"
    policer-control-override create
        max-rate 50000
    exit
    qos 100 multipoint-shared
    queue-override
        queue 1 create
            rate 1500000 cir 2000
        exit
    exit
    policer-override
        policer 1 create
            rate 10000
        exit
    exit
exit
egress
    scheduler-policy "SLA1"
    policer-control-policy "SLA1-p"
    policer-control-override create
```

```
        max-rate 60000
    exit
    qos 100
    queue-override
        queue 1 create
            adaptation-rule pir max cir max
        exit
    exit
    policer-override
        policer 1 create
            mbs 2000 kilobytes
        exit
    exit
    filter ip 10
exit
-----------------------------------------------
*A:ALA-48>config>service>vpls>sap#
```

Figure 92: SDPs — unidirectional tunnels



Use the following CLI syntax to create mesh or spoke-SDP bindings with a distributed VPLS service. SDPs must be configured before binding. For information about creating SDPs, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*.

Use the following CLI syntax to configure mesh SDP bindings.

CLI syntax:

```
config>service# vpls service-id
    – mesh-sdp sdp-id[:vc-id] [vc-type {ether | vlan}]
        – egress
            – filter {ip ip-filter-id|mac mac-filter-id}
            – mfib-allowed-mda-destinations
                    – mda mda-id
            – vc-label egress-vc-label
        – ingress
            – filter {ip ip-filter-id|mac mac-filter-id}
            – vc-label ingress-vc-label
        – no shutdown
        – static-mac ieee-address
        – vlan-vc-tag 0..4094
```

Use the following CLI syntax to configure spoke-SDP bindings.

CLI syntax:

```
config>service# vpls service-id
    – spoke-sdp sdp-id:vc-id [vc-type {ether | vlan}] [split-horizon-group group-name]
        – egress
            – filter {ip ip-filter-id|mac mac-filter-id}
            – vc-label egress-vc-label
        – ingress
            – filter {ip ip-filter-id|mac mac-filter-id}
            – vc-label ingress-vc-label
        – limit-mac-move[non-blockable]
        – vlan-vc-tag 0..4094
        – no shutdown
        – static-mac ieee-address
        – stp
            – path-cost stp-path-cost
            – priority stp-priority
            – no shutdown
        – vlan-vc-tag [0..4094]
```

The following examples show SDP binding configurations for ALA-1, ALA-2, and ALA-3 for VPLS service ID 9000 for customer 6:

```
*A:ALA-1>config>service# info
----------------------------------------------
...
        vpls 9000 customer 6 create
            description "This is a distributed VPLS."
            stp
                shutdown
            exit
            sap 1/2/5:0 create
            exit
            spoke-sdp 2:22 create
            exit
            mesh-sdp 5:750 create
            exit
            mesh-sdp 7:750 create
            exit
            no shutdown
        exit
```

```
-----------------------------------------------
*A:ALA-1>config>service#


*A:ALA-2>config>service# info
-----------------------------------------------
...
        vpls 9000 customer 6 create
            description "This is a distributed VPLS."
            stp
                shutdown
            exit
            sap 1/1/2:22 create
            exit
            spoke-sdp 2:22 create
            exit
            mesh-sdp 5:750 create
            exit
            mesh-sdp 7:750 create
            exit
            no shutdown
        exit
-----------------------------------------------


*A:ALA-3>config>service# info
-----------------------------------------------
...
        vpls 9000 customer 6 create
            description "This is a distributed VPLS."
            stp
                shutdown
            exit
            sap 1/1/3:33 create
            exit
            spoke-sdp 2:22 create
            exit
            mesh-sdp 5:750 create
            exit
            mesh-sdp 7:750 create
            exit
            no shutdown
        exit
-----------------------------------------------
*A:ALA-3>config>service#
```

3.5.3.8.1 Configuring spoke-SDP specific STP parameters

When a VPLS has STP enabled, each spoke-SDP within the VPLS has STP enabled by default. Subsequent sections describe spoke-SDP specific STP parameters in detail.

3.5.3.8.1.1 Spoke SDP STP administrative state

The administrative state of STP within a spoke SDP controls how BPDUs are transmitted and handled when received. The allowable states are:

- **spoke-sdp admin up**

The default administrative state is up for STP on a spoke SDP. BPDUs are handled in the normal STP manner on a spoke SDP that is administratively up.

- **spoke-sdp admin down**

  An administratively down state allows a service provider to prevent a spoke SDP from becoming operationally blocked. BPDUs do not originate out the spoke SDP toward the customer.

  If STP is enabled on VPLS level, but disabled on the spoke SDP, received BPDUs are discarded. Discarding the incoming BPDUs allows STP to continue to operate normally within the VPLS service while ignoring the down spoke SDP. The specified spoke SDP is always in an operationally forwarding state.

  Note: The administratively down state allows a loop to form within the VPLS.

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#
[no] shutdown
```

Range	shutdown or no shutdown
Default	no shutdown (spoke-SDP admin up)

3.5.3.8.1.2 Spoke SDP virtual port number

The virtual port number uniquely identifies a spoke SDP within configuration BPDUs. The internal representation of a spoke SDP is unique to a system and has a reference space much bigger than the 12 bits definable in a configuration BPDU. STP takes the internal representation value of a spoke SDP and identifies it with its own virtual port number that is unique to every other spoke-SDP defined on the VPLS. The virtual port number is assigned at the time that the spoke SDP is added to the VPLS.

Because the order in which spoke SDPs are added to the VPLS is not preserved between reboots of the system, the virtual port number may change between restarts of the STP instance. To achieve consistency after a reboot, the virtual port number can be specified explicitly.

CLI syntax:

```
config>service>vpls>spoke-sdp# stp
port-num number
```

Range	1 to 2047
Default	automatically generated
Restore Default	no port-num

3.5.3.8.1.3 Spoke SDP priority

Spoke SDP priority allows a configurable tiebreaking parameter to be associated with a spoke SDP. When configuration BPDUs are being received, the configured spoke-SDP priority is used in some circumstances to determine whether a spoke SDP is designated or blocked.

In traditional STP implementations (802.1D-1998), this field is called the port priority and has a value of 0 to 255. This field is coupled with the port number (0 to 255 also) to create a 16-bit value. In the latest STP standard (802.1D-2004), only the upper 4 bits of the port priority field are used to encode the spoke SDP priority. The remaining 4 bits are used to extend the port ID field into a 12-bit virtual port number field. The virtual port number uniquely references a spoke SDP within the STP instance. See Spoke SDP virtual port number for more information about the virtual port number.

STP computes the actual spoke SDP priority by taking the configured priority value and masking out the lower four bits. The result is the value that is stored in the spoke SDP priority parameter. For instance, if a value of 0 was entered, masking out the lower 4 bits would result in a parameter value of 0. If a value of 255 was entered, the result would be 240.

The default value for spoke SDP priority is 128. This parameter can be modified within a range of 0 to 255; 0 being the highest priority. Masking causes the values actually stored and displayed to be 0 to 240, in increments of 16.

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#
priority stp-priority
```

Range	0 to 255 (240 largest value, in increments of 16)
Default	128
Restore Default	no priority

3.5.3.8.1.4 Spoke SDP path cost

The spoke SDP path cost is used by STP to calculate the path cost to the root bridge. The path cost in BPDUs received on the root port is incremented with the configured path cost for that spoke-SDP. When BPDUs are sent out of other egress spoke SDPs, the newly calculated root path cost is used.

STP suggests that the path cost is defined as a function of the link bandwidth. Because spoke SDPs are controlled by complex queuing dynamics, the STP path cost is a purely static configuration.

The default value for spoke SDP path cost is 10. This parameter can be modified within a range of 1 to 200000000 (1 is the lowest cost).

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#
path-cost stp-path-cost
```

Range	1 to 200000000
Default	10
Restore Default	no path-cost

3.5.3.8.1.5 Spoke SDP edge port

The spoke SDP edge-port command is used to reduce the time it takes a spoke-SDP to reach the forwarding state when the spoke-SDP is on the edge of the network, and therefore has no further STP bridge to handshake with.

The edge-port command is used to initialize the internal OPER_EDGE variable. At any time, when OPER_EDGE is false on a spoke-SDP, the normal mechanisms are used to transition to the forwarding state (see Forward delay). When OPER_EDGE is true, STP assumes that the remote end agrees to transition to the forwarding state without actually receiving a BPDU with an agreement flag set.

The OPER_EDGE variable is dynamically set to false if the spoke SDP receives BPDUs (the configured edge-port value does not change). The OPER_EDGE variable is dynamically set to true if auto-edge is enabled and STP concludes there is no bridge behind the spoke SDP.

When STP on the spoke SDP is administratively disabled and re-enabled, the OPER_EDGE is re-initialized to the spoke-SDP configured for edge-port.

Valid values for spoke SDP edge-port are enabled and disabled, with disabled being the default.

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#
[no] edge-port
```

Default no edge-port

3.5.3.8.1.6 Spoke SDP auto edge

The spoke SDP edge-port command is used to instruct STP to dynamically decide whether the spoke SDP is connected to another bridge.

If auto-edge is enabled, and STP concludes there is no bridge behind the spoke SDP, the OPER_EDGE variable is dynamically set to true. If auto-edge is enabled, and a BPDU is received, the OPER_EDGE variable is dynamically set to true (see Spoke SDP edge port).

Valid values for spoke SDP auto-edge are enabled and disabled, with enabled being the default.

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#
[no] auto-edge
```

Default auto-edge

3.5.3.8.1.7 Spoke SDP link type

The spoke SDP link-type command instructs STP on the maximum number of bridges behind this spoke SDP. If there is only a single bridge, transitioning to forwarding state is based on handshaking (fast transitions). If more than two bridges are connected by a shared media, their spoke SDPs should all be configured as shared, and timer-based transitions are used.

Valid values for spoke SDP link-type are shared and pt-pt, with pt-pt being the default.

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#
link-type {pt-pt|shared}
```

Default link-type pt-pt

Restore Default no link-type

3.5.3.8.2 Spoke SDP STP operational states

The operational state of STP within a spoke SDP controls how BPDUs are transmitted and handled when received. Subsequent sections describe spoke SDP operational states.

3.5.3.8.2.1 Operationally disabled

Operationally disabled is the normal operational state for STP on a spoke SDP in a VPLS that has any of the following conditions:

- VPLS state administratively down
- Spoke SDP state administratively down
- Spoke SDP state operationally down

If the spoke SDP enters the operationally up state with the STP administratively up and the spoke SDP STP state is up, the spoke-SDP transitions to the STP spoke SDP discarding state.

When, during normal operation, the router detects a downstream loop behind a spoke SDP, BPDUs can be received at a very high rate. To recover from this situation, STP transitions the spoke SDP to a disabled state for the configured forward-delay duration.

3.5.3.8.2.2 Operationally discarding

A spoke-SDP in the discarding state only receives and sends BPDUs, building the local correct STP state for each spoke-SDP while not forwarding actual user traffic. The duration of the discarding state is described in section Forward delay.

Note: In previous versions of the STP standard, the discarding state was called a blocked state.

3.5.3.8.2.3 Operationally learning

The learning state allows population of the MAC forwarding table before entering the forwarding state. In this state, no user traffic is forwarded.

3.5.3.8.2.4 Operationally forwarding

Configuration BPDUs are sent out of a spoke-SDP in the forwarding state. Layer 2 frames received on the spoke-SDP are source learned and destination forwarded according to the FDB. Layer 2 frames received on other forwarding interfaces and destined for the spoke-SDP are also forwarded.

3.5.3.8.2.5 Spoke SDP BPDU encapsulation states

IEEE 802.1D (referred as dot1d) and Cisco's per VLAN Spanning Tree (PVST) BPDU encapsulations are supported on a per spoke SDP basis. STP is associated with a VPLS service like PVST is per VLAN. The

main difference resides in the Ethernet and LLC framing and a type-length-value (TLV) field trailing the BPDU.

Table 15: Spoke SDP BPDU encapsulation states shows differences between dot1D and PVST Ethernet BPDU encapsulations based on the interface encap-type field.

Table 15: Spoke SDP BPDU encapsulation states

Field	dot1d encap-type null	dot1d encap-type dot1q	PVST encap-type null	PVST encap-type dot1q
Destination MAC	01:80:c2:00:00:00	01:80:c2:00:00:00	N/A	01:00:0c:cc:cc:cd
Source MAC	Sending Port MAC	Sending Port MAC	N/A	Sending Port MAC
EtherType	N/A	0x81 00	N/A	0x81 00
Dot1p and DEI	N/A	0xe	N/A	0xe
Dot1q	N/A	VPLS spoke-SDP ID	N/A	VPLS spoke-SDP encap value
Length	LLC Length	LLC Length	N/A	LLC Length
LLC DSAP SSAP	0x4242	0x4242	N/A	0xaaaa (SNAP)
LLC CNTL	0x03	0x03	N/A	0x03
SNAP OUI	N/A	N/A	N/A	00 00 0c (Cisco OUI)
SNAP PID	N/A	N/A	N/A	01 0b
CONFIG or TCN BPDU	Standard 802.1d	Standard 802.1d	N/A	Standard 802.1d
TLV: Type and Len	N/A	N/A	N/A	58 00 00 00 02
TLV: VLAN	N/A	N/A	N/A	VPLS spoke-SDP encap value
Padding	As Required	As Required	N/A	As Required

Each spoke SDP has a Read Only operational state that shows which BPDU encapsulation is currently active on the spoke SDP. The following states apply:

- **dot1d**

  Specifies that the switch is currently sending IEEE 802.1D standard BPDUs. The BPDUs are tagged or non-tagged based on the encapsulation type of the egress interface and the encapsulation value defined in the spoke-SDP. A spoke SDP defined on an interface with encapsulation type dot1q continues in the dot1d BPDU encapsulation state until a PVST encapsulated BPDU is received, after which the spoke-SDP converts to the PVST encapsulation state. Each received BPDU must be properly IEEE 802.1q tagged if the interface encapsulation type is defined as dot1q.

- **PVST**

Specifies that the switch is currently sending proprietary encapsulated BPDUs. PVST BPDUs are only supported on Ethernet interfaces with the encapsulation type set to dot1q. The spoke SDP continues in the PVST BPDU encapsulation state until a dot1d encapsulated BPDU is received, in which case the spoke SDP reverts to the dot1d encapsulation state. Each received BPDU must be properly IEEE 802.1q tagged with the encapsulation value defined for the spoke SDP.

Dot1d is the initial and only spoke-SDP BPDU encapsulation state for spoke SDPs defined on an Ethernet interface with encapsulation type set to null.

Each transition between encapsulation types optionally generates an alarm that can be logged and optionally transmitted as an SNMP trap.

3.5.3.8.3 Configuring VPLS spoke SDPs with split horizon

To configure spoke SDPs with a split horizon group, add the **split-horizon-group** parameter when creating the spoke SDP. Traffic arriving on an SAP or spoke-SDP within a split horizon group is not copied to other SAPs or spoke SDPs in the same split horizon group.

The following example shows a VPLS configuration with split horizon enabled:

```
-----------------------------------------------
*A:ALA-1>config>service# *A:ALA-1>config>service# info
-----------------------------------------------
...
vpls 800 customer 6001 vpn 700 create
     description "VPLS with split horizon for DSL"
     stp
         shutdown
     exit
     spoke-sdp 51:15 split-horizon-group DSL-group1 create
     exit
     split-horizon-group DSL-group1
         description "Split horizon group for DSL"
     exit
     no shutdown
exit
...
-----------------------------------------------
*A:ALA-1>config>service#
```

3.5.4 Configuring VPLS redundancy

This section discusses VPLS redundancy service management tasks.

3.5.4.1 Creating a management VPLS for SAP protection

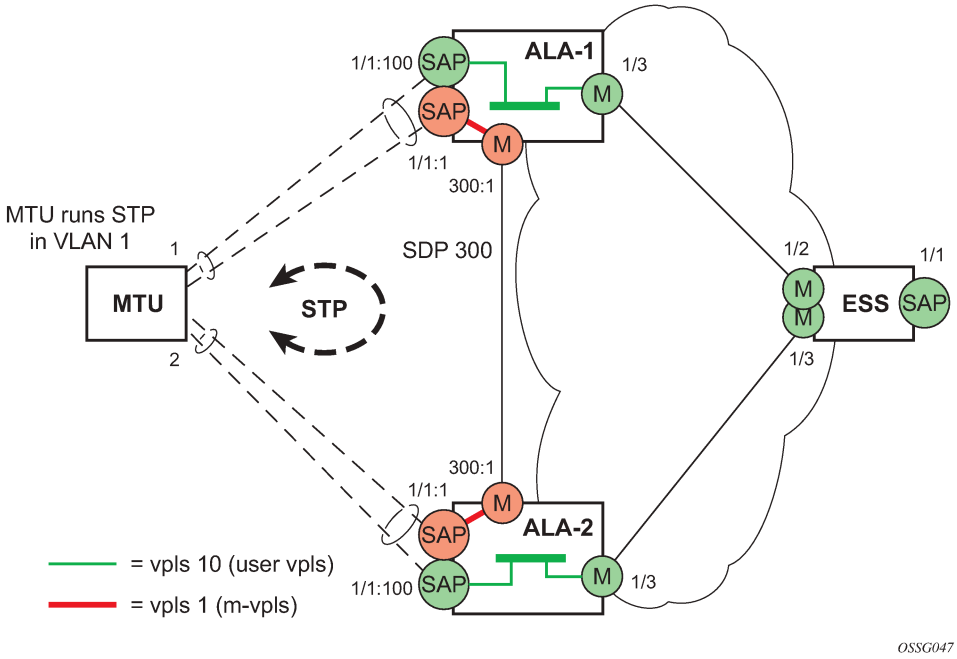
This section provides a brief overview of the tasks that must be performed to configure a management VPLS for SAP protection and provides the CLI commands; see Figure 93: Example configuration for protected VPLS SAP. The following tasks should be performed on both nodes providing the protected VPLS service.

Before configuring a management VPLS, see VPLS redundancy for an introduction to the concept of management VPLS and SAP redundancy.

1. Create an SDP to the peer node.
2. Create a management VPLS.
3. Define a SAP in the M-VPLS on the port toward the MTU. The port must be dot1q or qinq tagged. The SAP corresponds to the (stacked) VLAN on the MTU in which STP is active.
4. Optionally, modify STP parameters for load balancing.
5. Create a mesh SDP in the M-VPLS using the SDP defined in Step 1. Ensure that this mesh SDP runs over a protected LSP (see the following note).
6. Enable the management VPLS service and verify that it is operationally up.
7. Create a list of VLANs on the port that are to be managed by this management VPLS.
8. Create one or more user VPLS services with SAPs on VLANs in the range defined by Step 6.

Note: The mesh SDP should be protected by a backup LSP or Fast Reroute. If the mesh SDP went down, STP on both nodes would go to forwarding state and a loop would occur.

Figure 93: Example configuration for protected VPLS SAP



Use the following CLI syntax to create a management VPLS on the 7450 ESS or 7750 SR.

CLI syntax:

```
config>service# sdp sdp-id mpls create
    – far-end ip-address
    – lsp lsp-name
    – no shutdown
```

CLI syntax:

```
vpls service-id customer customer-id [m-vpls] create
```

```
    – description description-string
    – sap sap-id create
        – managed-vlan-list
            – range vlan-range
    – mesh-sdp sdp-id:vc-id create
    – stp
    – no shutdown
```

The following example shows a VPLS configuration:

```
*A:ALA-1>config>service# info
----------------------------------------------
...
       sdp 300 mpls create
            far-end 10.0.0.20
            lsp "toALA-A2"
            no shutdown
       exit
       vpls 1 customer 1 m-vpls create
            sap 1/1/1:1 create
                managed-vlan-list
                    range 100-1000
                exit
            exit
            mesh-sdp 300:1 create
            exit
            stp
            exit
            no shutdown
        exit
...
----------------------------------------------
*A:ALA-1>config>service#
```

3.5.4.2 Creating a management VPLS for spoke-SDP protection

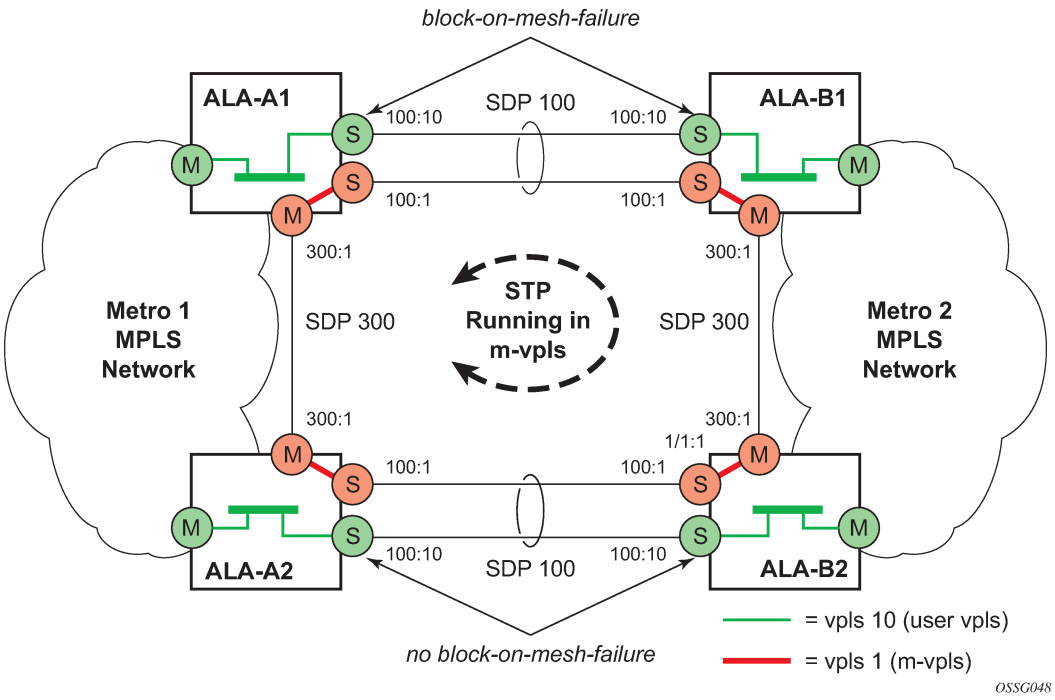
This section provides a brief overview of the tasks that must be performed to configure a management VPLS for spoke-SDP protection and provides the CLI commands; see Figure 94: Example configuration for protected VPLS spoke-SDP. The following tasks should be performed on all four nodes providing the protected VPLS service. Before configuring a management VPLS, see Configuring a VPLS SAP for an introduction to the concept of management VPLS and spoke-SDP redundancy.

1. Create an SDP to the local peer node (node ALA-A2 in the following example).
2. Create an SDP to the remote peer node (node ALA-B1 in the following example).
3. Create a management VPLS.
4. Create a spoke-SDP in the M-VPLS using the SDP defined in Step 1. Ensure that this mesh SDP runs over a protected LSP (see note below).
5. Enable the management VPLS service and verify that it is operationally up.
6. Create a spoke-SDP in the M-VPLS using the SDP defined in Step 2. Optionally, modify STP parameters for load balancing (see Configuring load balancing with management VPLS).
7. Create one or more user VPLS services with spoke-SDPs on the tunnel SDP defined by Step 2.

As long as the user spoke-SDPs created in step 7 are in this same tunnel SDP with the management spoke-SDP created in step 6, the management VPLS protects them.

Note: The SDP should be protected by, for example, a backup LSP or Fast Reroute. If the SDP went down, STP on both nodes would go to forwarding state and a loop would occur.

Figure 94: Example configuration for protected VPLS spoke-SDP



Use the following CLI syntax to create a management VPLS for spoke-SDP protection.

CLI syntax:

```
config>service# sdp sdp-id mpls create
    – far-end ip-address
    – lsp lsp-name
    – no shutdown
```

CLI syntax:

```
vpls service-id customer customer-id [m-vpls] create
    – description description-string
    – mesh-sdp sdp-id:vc-id create
    – spoke-sdp sdp-id:vc-id create
    – stp
    – no shutdown
```

The following example shows a VPLS configuration:

```
*A:ALA-A1>config>service# info
----------------------------------------------
...
       sdp 100 mpls create
            far-end 10.0.0.30
            lsp "toALA-B1"
            no shutdown
       exit
```

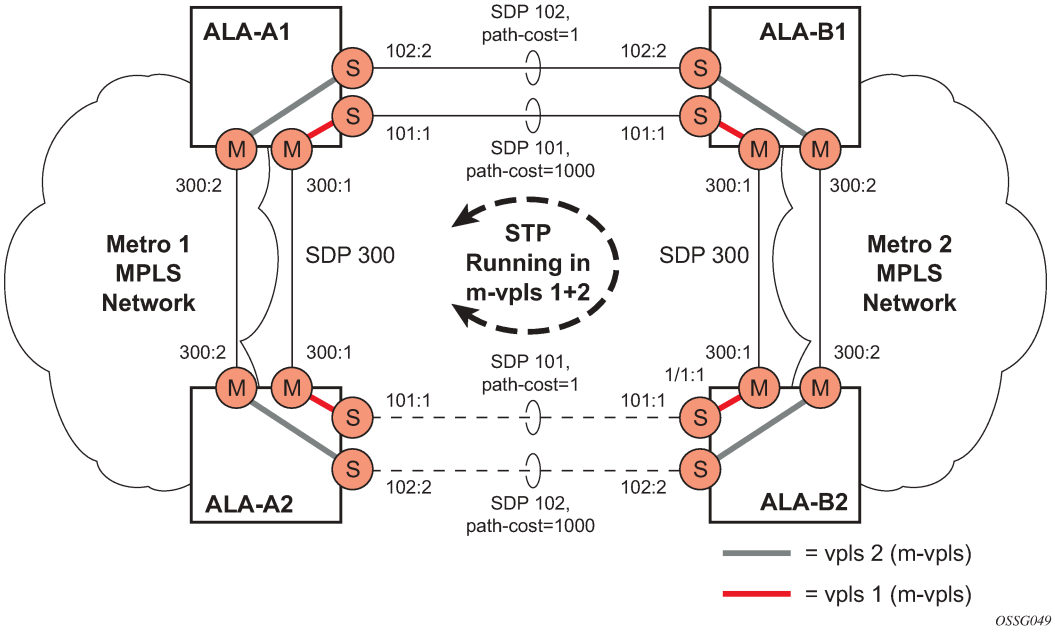
```
        sdp 300 mpls create
             far-end 10.0.0.20
             lsp "toALA-A2"
             no shutdown
        exit
        vpls 101 customer 1 m-vpls create
             spoke-sdp 100:1 create
             exit
             meshspoke-sdp 300:1 create
             exit
             stp
             exit
             no shutdown
         exit
...
----------------------------------------------
*A:ALA-A1>config>service#
```

3.5.4.3 Configuring load balancing with management VPLS

With the concept of management VPLS, it is possible to load balance the user VPLS services across the two protecting nodes. This is done by creating two management VPLS instances, where both instances have different active QinQ spokes (by changing the STP path-cost). When user VPLS services are associated with either of the two management VPLS services, the traffic is split across the two QinQ spokes. Load balancing can be achieved in both the SAP protection and spoke-SDP protection scenarios.

Figure 95: Example configuration for load balancing across two protected VPLS spoke-SDPs shows an example configuration for load balancing across two protected VPLS spoke-SDPs.

Figure 95: Example configuration for load balancing across two protected VPLS spoke-SDPs



Use the following CLI syntax to create load balancing across two management VPLS instances.

CLI syntax:

```
config>service# sdp sdp-id mpls create
far-end ip-address
lsp lsp-name
no shutdown
```

CLI syntax:

```
vpls service-id customer customer-id [m-vpls] create
    – description description-string
    – mesh-sdp sdp-id:vc-id create
    – spoke-sdp sdp-id:vc-id create
        – stp
            – path-cost
    – stp
    – no shutdown
```

Note: The STP path costs in each peer node should be reversed.

The following example shows the VPLS configuration on ALA-A1 (upper left, IP address 10.0.0.10):

```
*A:ALA-A1>config>service# info
----------------------------------------------
...
      sdp 101 mpls create
            far-end 10.0.0.30
            lsp "1toALA-B1"
            no shutdown
      exit
      sdp 102 mpls create
            far-end 10.0.0.30
            lsp "2toALA-B1"
            no shutdown
      exit
 ...
     vpls 101 customer 1 m-vpls create
            spoke-sdp 101:1 create
               stp
                  path-cost 1
               exit
            exit
            mesh-sdp 300:1 create
            exit
            stp
            exit
           no shutdown
      exit
      vpls 102 customer 1 m-vpls create
            spoke-sdp 102:2 create
               stp
                  path-cost 1000
               exit
            exit
            mesh-sdp 300:2 create
            exit
            stp
            exit
            no shutdown
      exit
```

```
...
-----------------------------------------------
*A:ALA-A1>config>service#
```

The following example shows the VPLS configuration on ALA-A2 (lower left, IP address 10.0.0.20):

```
*A:ALA-A2>config>service# info
-----------------------------------------------
...
      sdp 101 mpls create
            far-end 10.0.0.40
            lsp "1toALA-B2"
            no shutdown
      exit
      sdp 102 mpls create
            far-end 10.0.0.40
            lsp "2toALA-B2"
            no shutdown
      exit
 ...
     vpls 101 customer 1 m-vpls create
            spoke-sdp 101:1 create
               stp
                  path-cost 1000
               exit
            exit
            mesh-sdp 300:1 create
            exit
            stp
            exit
           no shutdown
      exit
      vpls 102 customer 1 m-vpls create
            spoke-sdp 102:2 create
               stp
                  path-cost 1
               exit
            exit
            mesh-sdp 300:2 create
            exit
            stp
            exit
            no shutdown
      exit
...
-----------------------------------------------
*A:ALA-A2>config>service#
```

The following example shows the VPLS configuration on ALA-A3 (upper right, IP address 10.0.0.30):

```
*A:ALA-A1>config>service# info
-----------------------------------------------
...
      sdp 101 mpls create
            far-end 10.0.0.10
            lsp "1toALA-A1"
            no shutdown
      exit
      sdp 102 mpls create
            far-end 10.0.0.10
            lsp "2toALA-A1"
            no shutdown
      exit
```

```
 ...
     vpls 101 customer 1 m-vpls create
            spoke-sdp 101:1 create
               stp
                  path-cost 1
               exit
            exit
            mesh-sdp 300:1 create
            exit
            stp
            exit
            no shutdown
      exit
      vpls 102 customer 1 m-vpls create
            spoke-sdp 102:2 create
               stp
                  path-cost 1000
               exit
            exit
            mesh-sdp 300:2 create
            exit
            stp
            exit
            no shutdown
      exit
...
-------------------------------------------
*A:ALA-A1>config>service#
```

The following example shows the VPLS configuration on ALA-A4 (lower right, IP address 10.0.0.40):

```
*A:ALA-A2>config>service# info
-------------------------------------------
...
      sdp 101 mpls create
            far-end 10.0.0.20
            lsp "1toALA-B2"
            no shutdown
      exit
      sdp 102 mpls create
            far-end 10.0.0.20
            lsp "2toALA-B2"
            no shutdown
      exit
 ...
     vpls 101 customer 1 m-vpls create
            spoke-sdp 101:1 create
               stp
                  path-cost 1000
               exit
            exit
            mesh-sdp 300:1 create
            exit
            stp
            exit
            no shutdown
      exit
      vpls 102 customer 1 m-vpls create
            spoke-sdp 102:2 create
               stp
                  path-cost 1
               exit
            exit
```

```
            mesh-sdp 300:2 create
            exit
            stp
            exit
            no shutdown
      exit
...
----------------------------------------------
*A:ALA-A2>config>service#
```

3.5.4.4 Configuring selective MAC flush

Use the following CLI syntax to enable selective MAC flush in a VPLS.

CLI syntax:

```
config>service# vpls service-id
  send-flush-on-failure
```

Use the following CLI syntax to disable selective MAC flush in a VPLS.

CLI syntax:

```
config>service# vpls service-id
  no send-flush-on-failure
```

3.5.4.5 Configuring multi-chassis endpoints

The following output shows configuration examples of multi-chassis redundancy and the VPLS configuration. The configurations in the graphics depicted in Inter-domain VPLS resiliency using multi-chassis endpoints are represented in this output.

Node mapping to the following examples in this section:

- PE3 = Dut-B
- PE3' = Dut-C
- PE1 = Dut-D
- PE2 = Dut-E

PE3 Dut-B

```
*A:Dut-B>config>redundancy>multi-chassis# info
----------------------------------------------
            peer 10.1.1.3 create
                peer-name "Dut-C"
                description "mcep-basic-tests"
                source-address 10.1.1.2
                mc-endpoint
                    no shutdown
                    bfd-enable
                    system-priority 50
                exit
                no shutdown
            exit
----------------------------------------------
*A:Dut-B>config>redundancy>multi-chassis#
```

```
*A:Dut-B>config>service>vpls# info
----------------------------------------------
            fdb-table-size 20000
            send-flush-on-failure
            stp
                shutdown
            exit
            endpoint "mcep-t1" create
                no suppress-standby-signaling
                block-on-mesh-failure
                mc-endpoint 1
                    mc-ep-peer Dut-C
                exit
            exit
            mesh-sdp 201:1 vc-type vlan create
            exit
            mesh-sdp 211:1 vc-type vlan create
            exit
            spoke-sdp 221:1 vc-type vlan endpoint "mcep-t1" create
                stp
                    shutdown
                exit
                block-on-mesh-failure
                precedence 1
            exit
            spoke-sdp 231:1 vc-type vlan endpoint "mcep-t1" create
                stp
                    shutdown
                exit
                block-on-mesh-failure
                precedence 2
            exit
            no shutdown
----------------------------------------------
*A:Dut-B>config>service>vpls#
```

PE3' Dut-C

```
:Dut-C>config>redundancy>multi-chassis# info
----------------------------------------------
            peer 10.1.1.2 create
                peer-name "Dut-B"
                description "mcep-basic-tests"
                source-address 10.1.1.3
                mc-endpoint
                    no shutdown
                    bfd-enable
                    system-priority 21
                exit
                no shutdown
            exit
----------------------------------------------
*A:Dut-C>config>redundancy>multi-chassis#

*A:Dut-C>config>service>vpls# info
----------------------------------------------
            fdb-table-size 20000
            send-flush-on-failure
            stp
                shutdown
            exit
```

```
            endpoint "mcep-t1" create
                no suppress-standby-signaling
                block-on-mesh-failure
                mc-endpoint 1
                    mc-ep-peer Dut-B
                exit
            exit
            mesh-sdp 301:1 vc-type vlan create
            exit
            mesh-sdp 311:1 vc-type vlan create
            exit
            spoke-sdp 321:1 vc-type vlan endpoint "mcep-t1" create
                stp
                    shutdown
                exit
                block-on-mesh-failure
                precedence 3
            exit
            spoke-sdp 331:1 vc-type vlan endpoint "mcep-t1" create
                stp
                    shutdown
                exit
                block-on-mesh-failure
            exit
            no shutdown
-----------------------------------------
*A:Dut-C>config>service>vpls#
```

PE1 Dut-D

```
*A:Dut-D>config>redundancy>multi-chassis# info
-----------------------------------------------
            peer 10.1.1.5 create
                peer-name "Dut-E"
                description "mcep-basic-tests"
                source-address 10.1.1.4
                mc-endpoint
                    no shutdown
                    bfd-enable
                    system-priority 50
                    passive-mode
                exit
                no shutdown
            exit
-----------------------------------------------
*A:Dut-D>config>redundancy>multi-chassis#

*A:Dut-D>config>service>vpls# info
-----------------------------------------------
            fdb-table-size 20000
            propagate-mac-flush
            stp
                shutdown
            exit
            endpoint "mcep-t1" create
                block-on-mesh-failure
                mc-endpoint 1
                    mc-ep-peer Dut-E
                exit
            exit
            mesh-sdp 401:1 vc-type vlan create
            exit
            spoke-sdp 411:1 vc-type vlan endpoint "mcep-t1" create
```

```
                stp
                    shutdown
                exit
                block-on-mesh-failure
                precedence 2
            exit
            spoke-sdp 421:1 vc-type vlan endpoint "mcep-t1" create
                stp
                    shutdown
                exit
                block-on-mesh-failure
                precedence 1
            exit
            mesh-sdp 431:1 vc-type vlan create
            exit
            no shutdown
----------------------------------------------
*A:Dut-D>config>service>vpls#
```

PE2 Dut-E

```
*A:Dut-E>config>redundancy>multi-chassis# info
----------------------------------------------
            peer 10.1.1.4 create
                peer-name "Dut-D"
                description "mcep-basic-tests"
                source-address 10.1.1.5
                mc-endpoint
                    no shutdown
                    bfd-enable
                    system-priority 22
                    passive-mode
                exit
                no shutdown
            exit
----------------------------------------------
*A:Dut-E>config>redundancy>multi-chassis#

*A:Dut-E>config>service>vpls# info
----------------------------------------------
            fdb-table-size 20000
            propagate-mac-flush
            stp
                shutdown
            exit
            endpoint "mcep-t1" create
                block-on-mesh-failure
                mc-endpoint 1
                    mc-ep-peer Dut-D
                exit
            exit
            spoke-sdp 501:1 vc-type vlan endpoint "mcep-t1" create
                stp
                    shutdown
                exit
                block-on-mesh-failure
                precedence 3
            exit
            spoke-sdp 511:1 vc-type vlan endpoint "mcep-t1" create
                stp
                    shutdown
                exit
```

```
                block-on-mesh-failure
            exit
            mesh-sdp 521:1 vc-type vlan create
            exit
            mesh-sdp 531:1 vc-type vlan create
            exit
            no shutdown
-----------------------------------------------
*A:Dut-E>config>service>vpls#
```

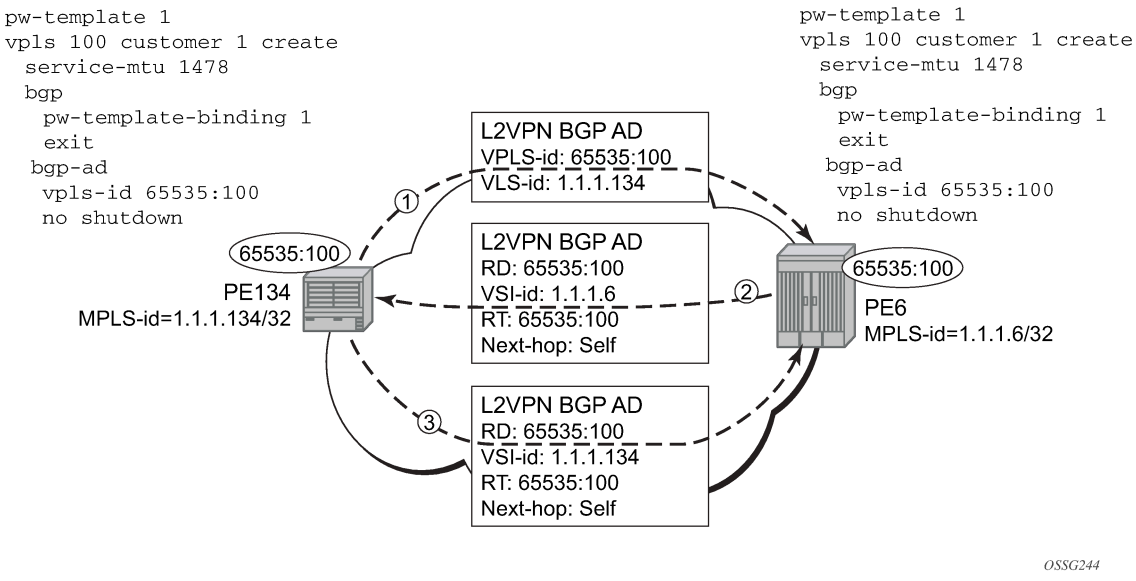
3.5.5 Configuring BGP auto-discovery

This section provides important information to describe the different configuration options used to populate the required BGP AD and generate the LDP generalized pseudowire-ID FEC fields. There are a large number of configuration options that are available with this feature. Not all these configuration options are required to start using BGP AD. At the end of this section, it will be apparent that a simple configuration will automatically generate the required values used by BGP and LDP. In most cases, deployments provide full mesh connectivity between all nodes across a VPLS instance. However, capabilities are available to influence the topology and build hierarchies or hub and spoke models.

3.5.5.1 Configuration steps

Using Figure 96: BGP AD configuration example, assume PE6 was previously configured with VPLS 100 as indicated by the configurations code in the upper right. The BGP AD process commences after PE134 is configured with the VPLS 100 instance, as shown in the upper left. This shows a basic BGP AD configuration. The minimum requirement for enabling BGP AD on a VPLS instance is configuring the VPLS-ID and pointing to a pseudowire template.

Figure 96: BGP AD configuration example



In many cases, VPLS connectivity is based on a pseudowire mesh. To reduce the configuration requirement, the BGP values can be automatically generated using the VPLS-ID and the MPLS router-ID.

By default, the lower six bytes of the VPLS-ID are used to generate the RD and the RT values. The VSI-ID value is generated from the MPLS router-ID. All of these parameters are configurable and can be coded to suit requirements and build different topologies.

```
PE134>config>service>vpls>bgp-ad#
[no] shutdown - Administratively enable/disable BGP auto-discovery
vpls-id - Configure VPLS-ID
vsi-id + Configure VSI-id
```

The **show service** command shows the service information, the BGP parameters, and the SDP bindings in use. When the discovery process is completed successfully, each endpoint has an entry for the service.

```
PE134>#  show service l2-route-table
===============================================================================
Services: L2 Route Information - Summary Service
===============================================================================
Svc Id     L2-Routes (RD-Prefix)                Next Hop        Origin
           Sdp Bind Id
-------------------------------------------------------------------------------
100        65535:100-1.1.1.6                    1.1.1.6         BGP-L2
           17406:4294967295
-------------------------------------------------------------------------------
No. of L2 Route Entries: 1
===============================================================================
PERs6>#

PERs6>#  show service l2-route-table
===============================================================================
Services: L2 Route Information - Summary Service
===============================================================================
Svc Id     L2-Routes (RD-Prefix)                Next Hop        Origin
           Sdp Bind Id
-------------------------------------------------------------------------------
100        65535:100-1.1.1.134                  1.1.1.134       BGP-L2
           17406:4294967295
-------------------------------------------------------------------------------
No. of L2 Route Entries: 1
===============================================================================
PERs6>#
```

When only one of the endpoints has an entry for the service in the l2-routing-table, it is most likely a problem with the RT values used for import and export. This would most likely happen when different import and export RT values are configured using a router policy or the route-target command.

Service-specific commands continue to be available to show service-specific information, including status:

```
PERs6# show service sdp-using
===============================================================================
SDP Using
===============================================================================
SvcId       SdpId               Type    Far End        Opr S* I.Label  E.Label
-------------------------------------------------------------------------------
100         17406:4294967295    BgpAd   10.1.1.134      Up     131063   131067
-------------------------------------------------------------------------------
Number of SDPs : 1
===============================================================================
* indicates that the corresponding row element may have been truncated.
```

BGP AD advertises the VPLS-ID in the extended community attribute, VSI-ID in the NLRI, and the local PE ID in the BGP next hop. At the receiving PE, the VPLS-ID is compared against locally provisioned

information to determine whether the two PEs share a common VPLS. If they do, the BGP information is used in the signaling phase (see Configuring BGP VPLS).

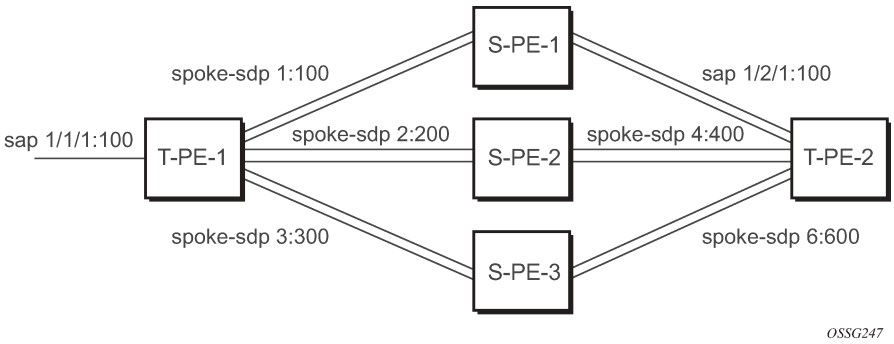
3.5.5.2 LDP signaling

T-LDP is triggered when the VPN endpoints have been discovered using BGP. The T-LDP session between the PEs is established when a session does not exist. The far-end IP address required for the T-LDP identification is learned from the BGP AD next hop information. The pw-template and pw-template-binding configuration statements are used to establish the automatic SDP or to map to the appropriate SDP. The FEC129 content is built using the following values:

- AGI from the locally configured VPLS-ID
- SAII from the locally configured VSI-ID
- TAII from the VSI-ID contained in the last 4 bytes of the received BGP NLRI

Figure 97: BGP AD triggering LDP functions shows the different detailed phases of the LDP signaling path, post BGP AD completion. It also indicates how some fields can be auto-generated when they are not specified in the configuration.

Figure 97: BGP AD triggering LDP functions



The following command shows the LDP peering relationships that have been established (see Figure 98: Show router LDP session output). The type of adjacency is displayed in the "Adj Type" column. In this case, the type is "Both" meaning link and targeted sessions have been successfully established.

Figure 98: Show router LDP session output

```
PERs6# show router ldp session
LDP Sessions
```

Peer LDP Id	Adj Type	State	Msg Sent	Msg Recv	Up Time
1.1.1.134:0	Both	Established	21482	21482	0d 15:38:44

No. of Sessions: 1

0988

The following command shows the specific LDP service label information broken up per FEC element type: 128 or 129, basis (see Figure 99: Show router LDP bindings FEC-type services). The information for FEC element 129 includes the AGI, SAII, and the TAII.

Figure 99: Show router LDP bindings FEC-type services

```
PERs6# show router ldp bindings fec-type services
LDP LSR ID: 1.1.1.6
=============================================================================
Legend: U - Label In Use, N - Label Not In Use, W - Label Withdrawn
        S - Status Signaled Up,  D - Status Signaled Down
        E - Epipe Service, V - VPLS Service, M - Mirror Service
        A - Apipe Service, F - Fpipe Service, I - IES Service, R - VPRN service
        P - Ipipe Service, C - Cpipe Service
        TLV - (Type, Length: Value)
LDP Service FEC 128 Bindings
=============================================================================
Type   VCId       SvcId      SDPId  Peer            IngLbl  EgrLbl  LMTU  RMTU
-----------------------------------------------------------------------------
No Matching Entries Found

LDP Service FEC 129 Bindings
=============================================================================
AGI                                SAII                    TAII
Type               SvcId      SDPId  Peer            IngLbl  EgrLbl  LMTU  RMTU
-----------------------------------------------------------------------------
65535:100                          1.1.1.6                 1.1.1.134
V-Eth              100        17406  1.1.1.134       131063U 131067S 1464  1464
-----------------------------------------------------------------------------
No. of FEC 129s: 1
=============================================================================
```

0989

3.5.5.3 Pseudowire template

The pseudowire template is defined under the top-level **service** command (**config**>**service**>**pw-template**) and specifies whether to use an automatically generated SDP or manually configured SDP. It also provides the set of parameters required for establishing the pseudowire (SDP binding) as follows:

```
PERs6>config>service# pw-template 1 create
 -[no] pw-template <policy-id> [use-provisioned-sdp | prefer-provisioned-sdp]
<policy-id> : [1..2147483647]
<use-provisioned-s*> : keyword
<prefer-provisioned*> : keyword


[no] accounting-pol*    - Configure accounting-policy to be used
[no] auto-learn-mac*    - Enable/Disable automatic update of MAC protect list
[no] block-on-peer-*    - Enable/Disable block traffic on peer fault
[no] collect-stats      - Enable/disable statistics collection
[no] control word       - Enable/Disable the use of Control Word
[no] disable-aging      - Enable/disable aging of MAC addresses
[no] disable-learni*    - Enable/disable learning of new MAC addresses
[no] discard-unknow*    - Enable/disable discarding of frames with unknown source
                          MAC address
     egress             + Spoke SDP binding egress configuration
[no] force-qinq-vc-*    - Forces qinq-vc-type forwarding in the data-path
[no] force-vlan-vc-*    - Forces vlan-vc-type forwarding in the data-path
[no] hash-label         - Enable/disable use of hash-label
     igmp-snooping      + Configure IGMP snooping parameters
     in gr ess          + Spoke SDP binding ingress configuration
[no] l2pt-terminati*    - Configure L2PT termination on this spoke SDP
[no] limit-mac-move     - Configure mac move
[no] mac-pinning        - Enable/disable MAC address pinning on this spoke SDP
[no] max-nbr-mac-ad*    - Configure the maximum number of MAC entries in the FDB
                          from this SDP
[no] restrict-prote*    - Enable/disable protected src MAC restriction
[no] sdp-exclude        - Configure excluded SDP group
```

```
[no] sdp-include        - Configure included SDP group
[no] split-horizon-*    + Configure a split horizon group
     stp                + Configure STP parameters
     vc-type            - Configure VC type
[no] vlan-vc-tag        - Configure VLAN VC tag
```

A **pw-template-binding** command configured within the VPLS service under the **bgp-ad** sub-command is a pointer to the pw-template that should be used. If a VPLS service does not specify an import-rt list, then that binding applies to all route targets accepted by that VPLS. The **pw-template-bind** command can select a different template on a per import-rt basis. It is also possible to specify specific pw-templates for some route targets with a VPLS service and use the single **pw-template-binding** command to address all unspecified but accepted imported targets.

Figure 100: PW-template-binding CLI syntax

```
PERs6>config>service>vpls>bgp-ad# pw-template-binding
  - pw-template-binding <policy-id> [split-hozion-group <group-name>] [import-
rt
    {ext-community, ...(upto 5 max)}]
  - no pw-template-binding <policy-id>

 <policy-id>           : [1..2147483647]
 <group-name>          : [32 chars max]
 <ext-community>       : target:{<ip-addr:comm-val>|<as-number:ext-comm-val>}
                         ip-addr      - a.b.c.d
                         comm-val     - [0..65535]
                         as-number    - [1..65535]
                         ext-comm-val - [0..4294967295]
```

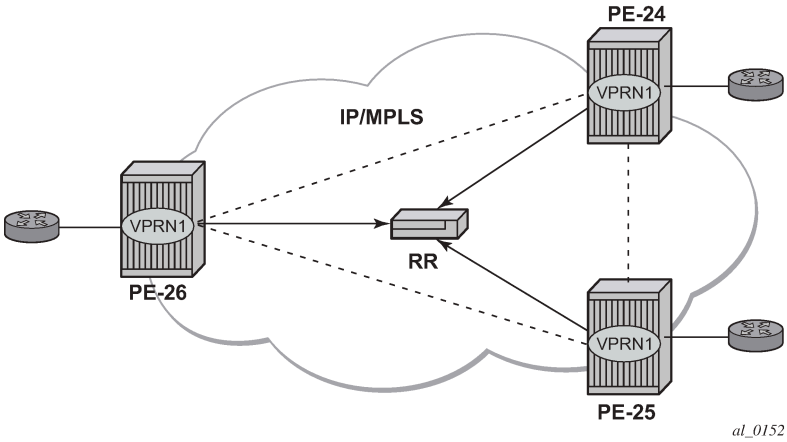
0990

It is important to understand the significance of the split horizon group used by the pw-template. Traditionally, when a VPLS instance was manually created using mesh-SDP bindings, these were automatically placed in a common split horizon group to prevent forwarding between the pseudowire in the VPLS instances. This prevents loops that would have otherwise occurred in the Layer 2 service. When automatically discovering VPLS service using BGP AD, the service provider has the option of associating the auto-discovered pseudowire with a split horizon group to control the forwarding between pseudowires.

3.5.6 Configuring BGP VPLS

This section provides a configuration example required to bring up BGP VPLS in the VPLS PEs depicted in Figure 101: BGP VPLS example.

Figure 101: BGP VPLS example



The red BGP VPLS is configured in the PE24, PE25, and PE26 using the commands shown in the following CLI examples:

```
*A:PE24>config>service>vpls# info
----------------------------------------------
            bgp
                route-distinguisher 65024:600
                route-target export target:65019:600 import target:65019:600
                pw-template-binding 1
            exit
            bgp-vpls
                max-ve-id 100
                ve-name 24
                    ve-id 24
                exit
                no shutdown
            exit
            sap 1/1/20:600.* create
            exit
            no shutdown
----------------------------------------------
*A:PE24>config>service>vpls#


*A:PE25>config>service>vpls# info
----------------------------------------------
            bgp
                route-distinguisher 65025:600
                route-target export target:65019:600 import target:65019:600
                pw-template-binding 1
            exit
            bgp-vpls
                max-ve-id 100
                ve-name 25
                    ve-id 25
                exit
                no shutdown
            exit
            sap 1/1/19:600.* create
            exit
            no shutdown
----------------------------------------------
*A:PE25>config>service>vpls#
```

```
*A:PE26>config>service>vpls# info
----------------------------------------------
            bgp
                route-distinguisher 65026:600
                route-target export target:65019:600 import target:65019:600
                pw-template-binding 1
            exit
            bgp-vpls
                max-ve-id 100
                ve-name 26
                    ve-id 26
                exit
                no shutdown
            exit
            sap 5/2/20:600.* create
            exit
            no shutdown
----------------------------------------------
*A:PE26>config>service>vpls#
```

3.5.6.1 Configuring a VPLS management interface

Use the following CLI syntax to create a VPLS management interface:

CLI syntax:

```
    config>service>vpls# interface ip-int-name
    address ip-address[/mask] [netmask]
    arp-timeout seconds
    description description-string
    mac ieee-address
    no shutdown
    static-arp ip-address ieee-address
```

The following example shows the configuration.

```
A:ALA-49>config>service>vpls>interface# info detail
----------------------------------------------
               no description
               mac 14:31:ff:00:00:00
               address 10.231.10.10/24
               no arp-timeout
               no shutdown
----------------------------------------------
A:ALA-49>config>service>vpls>interface#
```

3.5.7 Configuring policy-based forwarding for DPI in VPLS

The purpose of policy-based forwarding is to capture traffic from a customer and perform a deep packet inspection (DPI) and forward traffic, if allowed, by the DPI on the 7450 ESS or 7750 SR.

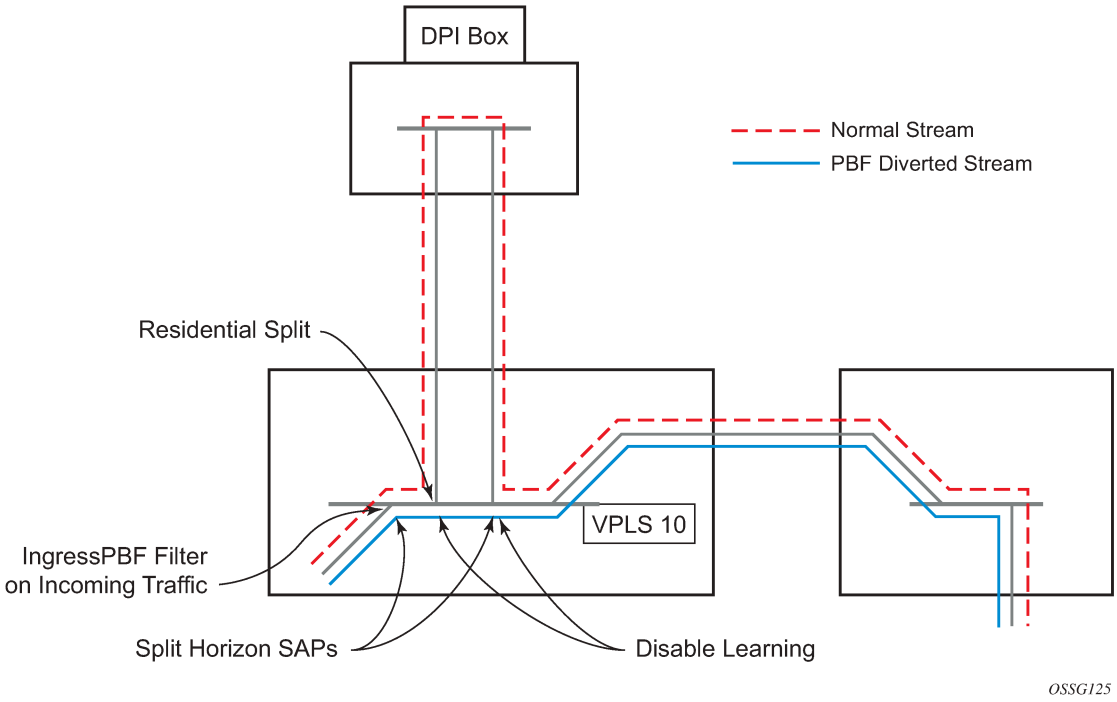
In the following example, the split horizon groups are used to prevent flooding of traffic. Traffic from customers enter at SAP 1/1/5:5. Because of the mac-filter 100 that is applied on ingress, all traffic with dot1p 07 marking is forwarded to SAP 1/1/22:1, which is the DPI.

DPI performs packet inspection/modification and either drops the traffic or forwards the traffic back into the box through SAP 1/1/21:1. Traffic is then sent to spoke-SDP 3:5.

SAP 1/1/23:5 is configured to determine whether the VPLS service is flooding all the traffic. If flooding is performed by the router, traffic would also be sent to SAP 1/1/23:5 (which it should not).

Figure 102: Policy-based forwarding for deep packet inspection shows an example to configure policy-based forwarding for deep packet inspection on a VPLS service. For information about configuring filter policies, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Router Configuration Guide*.

Figure 102: Policy-based forwarding for deep packet inspection



The following example shows the service configuration:

```
*A:ALA-48>config>service# info
----------------------------------------------
...
        vpls 10 customer 1 create
            service-mtu 1400
            split-horizon-group "dpi" residential-group create
            exit
            split-horizon-group "split" create
            exit
            stp
                shutdown
            exit
            igmp-host-tracking
                expiry-time 65535
                no shutdown
            exit
            sap 1/1/21:1 split-horizon-group "split" create
                disable-learning
                static-mac 00:00:00:31:11:01 create
            exit
```

```
            sap 1/1/22:1 split-horizon-group "dpi" create
                disable-learning
                static-mac 00:00:00:31:12:01 create
            exit
            sap 1/1/23:5 create
                static-mac 00:00:00:31:13:05 create
            exit
            no shutdown
        exit
...
-----------------------------------------------
*A:ALA-48>config>service#
```

The following example shows the MAC filter configuration:

```
*A:ALA-48>config>filter# info
-----------------------------------------------
...
        mac-filter 100 create
            default-action forward
            entry 10 create
                match
                    dot1p 7 7
                exit
                log 101
                action forward sap 1/1/22:1
            exit
        exit
...
-----------------------------------------------
*A:ALA-48>config>filter#
```

The following example shows the service configuration with a MAC filter:

```
*A:ALA-48>config>service# info
-----------------------------------------------
...
        vpls 10 customer 1 create
            service-mtu 1400
            split-horizon-group "dpi" residential-group create
            exit
            split-horizon-group "split" create
            exit
            stp
                shutdown
            exit
            igmp-host-tracking
                expiry-time 65535
                no shutdown
            exit
            sap 1/1/5:5 split-horizon-group "split" create
                ingress
                    filter mac 100
                exit
                static-mac 00:00:00:31:15:05 create
            exit
            sap 1/1/21:1 split-horizon-group "split" create
                disable-learning
                static-mac 00:00:00:31:11:01 create
            exit
            sap 1/1/22:1 split-horizon-group "dpi" create
                disable-learning
                static-mac 00:00:00:31:12:01 create
```

```
            exit
            sap 1/1/23:5 create
                static-mac 00:00:00:31:13:05 create
            exit
            spoke-sdp 3:5 create
            exit
            no shutdown
        exit
....
-------------------------------------------
*A:ALA-48>config>service#
```

3.5.8 Configuring VPLS E-Tree services

When configuring a VPLS E-Tree service, the **etree** keyword must be specified when the VPLS service is created. This is the first operation required before any SAPs or SDPs are added to the service, because the E-Tree service type affects the operations of the SAPs and SDP bindings.

When configuring AC SAPs, the configuration model is very similar to normal SAPs. Because the VPLS service must be designated as an E-Tree, the default AC SAP is a root-ac SAP. An E-Tree service with all root-ac behaves just as a regular VPLS service. A leaf-ac SAP must be configured for leaf behavior.

For root-leaf-tag SAPs, the SAP is created with both root and leaf VIDs. The 1/1/1:x.* or 1/1/1:x would be the typical format, where x designates the root tag. A leaf-tag is configured at SAP creation and replaces the x with a leaf-tag VID. Combined statistics for root and leaf SAPs are reported under the SAP. There are no individual statistics shown for root and leaf.

The following example illustrates the configuration of a VPLS E-Tree service with root-ac (default configuration for SAPs and SDP binds) and leaf-ac interfaces, as well as a root leaf tag SAP and SDP bind.

In the example, the SAP 1/1/7:2006.200 is configured using the root-leaf-tag parameter, where the outer VID 2006 is used for root traffic and the outer VID 2007 is used for leaf traffic.

```
*A:ALA-48>config>service# info
-----------------------------------------------
...
        service vpls 2005 etree customer 1 create
            sap 1/1/1:2005 leaf-ac create
            exit
            sap 1/1/7:2006.200 root-leaf-tag leaf-tag 2007 create
            exit
            sap 1/1/7:0.* create
            exit
            spoke-sdp 12:2005 vc-type vlan root-leaf-tag create
                no shutdown
            exit
            spoke-sdp 12:2006 leaf-ac create
                no shutdown
            exit
            no shutdown
        exit
....
*A:ALA-48>config>service# info
-----------------------------------------------
```

3.6 Service management tasks

This section describes VPLS service management tasks.

3.6.1 Modifying VPLS service parameters

You can change existing service parameters. The changes are applied immediately. To display a list of services, use the **show service service-using vpls** command. Enter the parameter such as description, SAP, SDP, or service-MTU command syntax, then enter the new information.

The following shows a modified VPLS configuration:

```
*A:ALA-1>config>service>vpls# info
----------------------------------------------
            description "This is a different description."
            disable-learning
            disable-aging
            discard-unknown
            local-age 500
            remote-age 1000
            stp
                shutdown
            exit
            sap 1/1/5:22 create
                description "VPLS SAP"
            exit
            spoke-sdp 2:22 create
            exit
            no shutdown
----------------------------------------------
*A:ALA-1>config>service>vpls#
```

3.6.2 Modifying management VPLS parameters

To modify the range of VLANs on an access port that are to be managed by an existing management VPLS, the new range should be defined, then the old range is removed. If the old range is removed before a new range is defined, all customer VPLS services in the old range become unprotected and may be disabled.

CLI syntax:

```
config>service# vpls service-id
  sap sap-id
    managed-vlan-list
       [no] range vlan-range
```

3.6.3 Deleting a management VPLS

As with normal VPLS service, a management VPLS cannot be deleted until SAPs and SDPs are unbound (deleted), interfaces are shut down, and the service is shut down on the service level.

Use the following CLI syntax to delete a management VPLS service.

CLI syntax:

```
config>service
[no] vpls service-id
  shutdown
  [no] spoke-sdp sdp-id
  [no] mesh-sdp sdp-id
    shutdown
  [no] sap sap-id
    shutdown
```

3.6.4 Disabling a management VPLS

You can shut down a management VPLS without deleting the service parameters.

When a management VPLS is disabled, all associated user VPLS services are also disabled (to prevent loops). If this is not needed, un-manage the user's VPLS service by removing them from the managed-vlan-list or moving the spoke-SDPs to another tunnel SDP.

CLI syntax:

```
config>service
vpls service-id
   shutdown
```

Example:

```
config>service# vpls 1
config>service>vpls# shutdown
config>service>vpls# exit
```

3.6.5 Deleting a VPLS service

A VPLS service cannot be deleted until SAPs and SDPs are unbound (deleted), interfaces are shut down, and the service is shut down on the service level.

Use the following CLI syntax to delete a VPLS service.

CLI syntax:

```
config>service
[no] vpls service-id
  shutdown
  [no] mesh-sdp sdp-id
    shutdown
  sap sap-id [split-horizon-group group-name]
  no sap sap-id
    shutdown
```

3.6.6 Disabling a VPLS service

You can shut down a VPLS service without deleting the service parameters.

CLI syntax:

```
config>service> vpls service-id
[no] shutdown
```

Example:

```
config>service# vpls 1
config>service>vpls# shutdown
config>service>vpls# exit
```

3.6.7 Re-enabling a VPLS service

Use the following CLI syntax to re-enable a VPLS service that was shut down.

CLI syntax:

```
config>service> vpls service-id
[no] shutdown
```

Example:

```
config>service# vpls 1
config>service>vpls# no shutdown
config>service>vpls# exit
```

4 Layer 2 control protocols

SR OS has awareness of multiple Layer 2 Control Protocols (L2CP). In some configurations, these L2CP frames are extracted and processed by the receiving SR. But there are some deployments where it is desirable to have the SR transparently forward the L2CP frames through a Layer 2 VLL or VPLS service. The SR OS support for this transparent tunnelling is configured on a protocol basis as described in this section.

L2CP frames are processed as follows:

- **tunneled**

  SR OS passes the frames through any associated service.

- **peered**

  SR OS extracts and processes the frame.

- **discarded**

  SR OS extracts the frame and then discards it.

The following L2CP frames are always tunneled:

- STP/RSTP/MSTP (assuming Spanning Tree is not enabled if the ingress SAP is attached to a VPLS service)
- LAMP
- MAC Specific Control Protocols
- Provider Bridge Group Address
- Provider Bridge MVRP Address

Other frame types are processed according to CLI configuration, as follows:

- **PAUSE frames**

  PAUSE frames are transmitted to request backward direction flow control. By default, SR OS peers these frames on reception and pauses the transmit side of the port. On some ports, this behavior can be changed to discard these frames by using the **config**>**port**>**ethernet discard-rx-pause-frames** CLI command.

  PAUSE frames are never tunneled.

  PAUSE frames are identified as frames with Ethertype (0x8808).

- **LACP frames**

  If the port is part of a LAG, the LACP frames are peered. If the port is not part of a LAG, the LACP frames are discarded or tunneled according to the configuration of the **config**>**port**>**ethernet lacp-tunnel** CLI command.

  LACP frames are identified as frames with Ethertype (0x8809) and the slow-protocol subtype (0x01)

- **EFM-OAM frames**

  If the port has EFM-OAM processing enabled, the EFM-OAM frames are peered. If EFM-OAM processing is not enabled on the port, the EFM-OAM frames are discarded or tunneled according to the configuration of the **config**>**port**>**ethernet lacp-tunnel** CLI command.

EFM-OAM frames are identified as frames with Ethertype (0x8809) and the slow-protocol sub-type (0x03).

- **ESMC frames**

  If the port is an input reference to the central frequency clock, the ESMC frames are peered. If the port is not an input reference to the central frequency clock, the ESMC frames are discarded. To override the preceding scenarios and tunnel the ESMC frames, configure the **config>port>ethernet>ssm esmc-tunnel** CLI command.

  ESMC frames are identified as frames with Ethertype (0x8809) and the slow-protocol sub-type (0x0A).

- **802.1x frames**

  By default, the 802.1x frames are extracted when they are received. For extracted frames, if a RADIUS server is configured, the frames are peered; otherwise they are discarded. The extraction can be overridden and the 802.1x frames can be tunneled by using the **config>port>ethernet>dot1x tunneling** CLI command.

  802.1x frames are identified as frames with Ethertype (0x888E).

- **E-LMI frames**

  If Ethernet Local Management Interface (E-LMI) processing is enabled on the port, E-LMI frames are peered. Otherwise, the E-LMI frames are dropped from VPLS and tunneled for Epipe.

  E-LMI frames are identified as frames with Ethertype (0x88EE) and a destination MAC address of 01:80:C2:00:00:07.

- **LLDP frames**

  If LLDP processing is enabled on the port, LLDP frames are peered. Otherwise, LLDP frames are discarded or tunneled according to the configuration of the **config>port>ethernet>lldp>dst-mac tunnel-nearest-bridge** CLI command.

  LLDP frames are identified as frames with Ethertype (0x88CC) and destination MAC address of 01-80-C2-00-00-0E.

- **PTP peer delay frames**

  If the port is configured in the router as an active port within the PTP process, the frames are peered; otherwise, the frames are tunneled.

  PTP message frames are identified as frames with Ethertype (0x88F7).

The maximum transparency of L2CP frames is achieved by configuring the following CLI commands on the port:

```
config port ethernet
  discard-rx-pause-frames
  lacp-tunnel
  efm-oam tunnelling
  ssm esmc-tunnel
  dot1x tunneling
  lldp dst-mac tunnel-nearest-bridge
```

Note: E-LMI must be disabled and the port must not be configured as a PTP port.

5 IEEE 802.1ah Provider Backbone Bridging

5.1 PBB overview

IEEE 802.1ah draft standard (IEEE802.1ah), also known as Provider Backbone Bridges (PBB), defines an architecture and bridge protocols for interconnection of multiple Provider Bridge Networks (PBNs - IEEE802.1ad QinQ networks). PBB is defined in IEEE as a connectionless technology based on multipoint VLAN tunnels. IEEE 802.1ah employs Provider MSTP as the core control plane for loop avoidance and load balancing. As a result, the coverage of the solution is limited by STP scale in the core of large service provider networks.

Virtual Private LAN Service (VPLS), RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*, provides a solution for extending Ethernet LAN services using MPLS tunneling capabilities through a routed, traffic-engineered MPLS backbone without running (M)STP across the backbone. As a result, VPLS has been deployed on a large scale in service provider networks.

The Nokia implementation fully supports a native PBB deployment and an integrated PBB-VPLS model where desirable PBB features such as MAC hiding, service aggregation and the service provider fit of the initial VPLS model are combined to provide the best of both worlds.

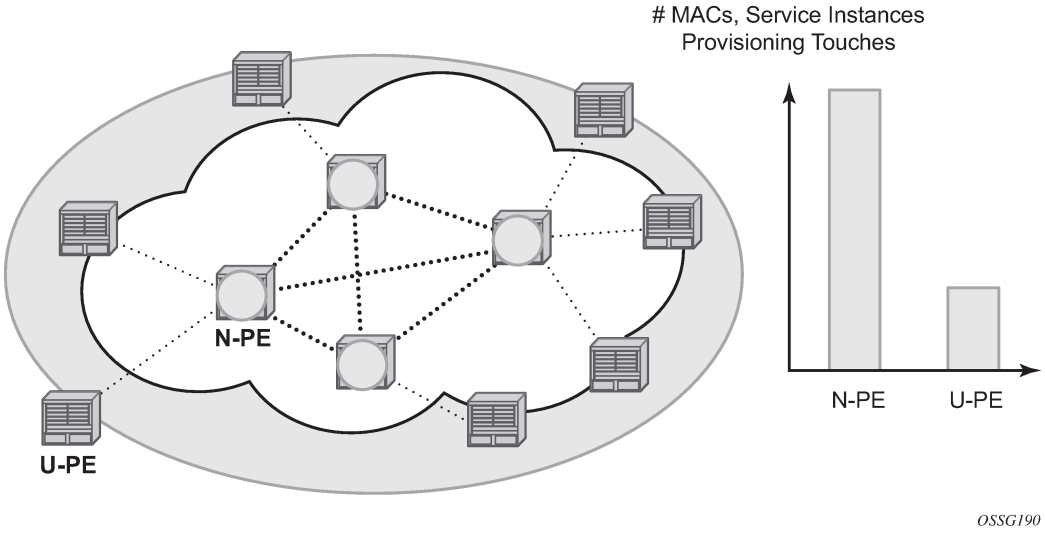
5.2 PBB features

This section provides information about PBB features.

5.2.1 Integrated PBB-VPLS solution

HVPLS introduced a service-aware device in a central core location to provide efficient replication and controlled interaction at domain boundaries. The core network facing provider edge (N-PE) devices have knowledge of all VPLS services and customer MAC addresses for local and related remote regions resulting in potential scalability issues as depicted in Figure 103: Large HVPLS deployment.

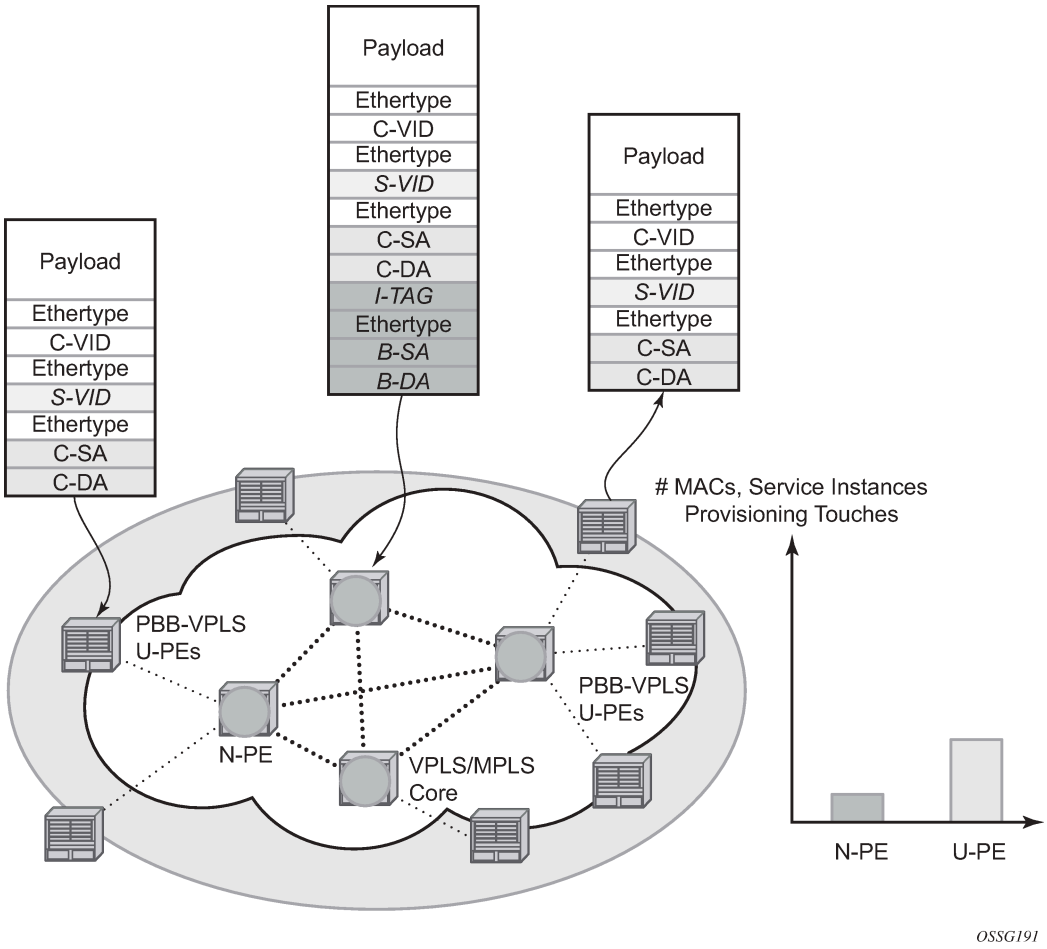
Figure 103: Large HVPLS deployment



In a large VPLS deployment, it is important to improve the stability of the overall solution and to speed up service delivery. These goals are achieved by reducing the load on the N-PEs and respectively minimizing the number of provisioning touches on the N-PEs.

The integrated PBB-VPLS model introduces an additional PBB hierarchy in the VPLS network to address these goals as depicted in Figure 104: Large PBB-VPLS deployment.

Figure 104: Large PBB-VPLS deployment



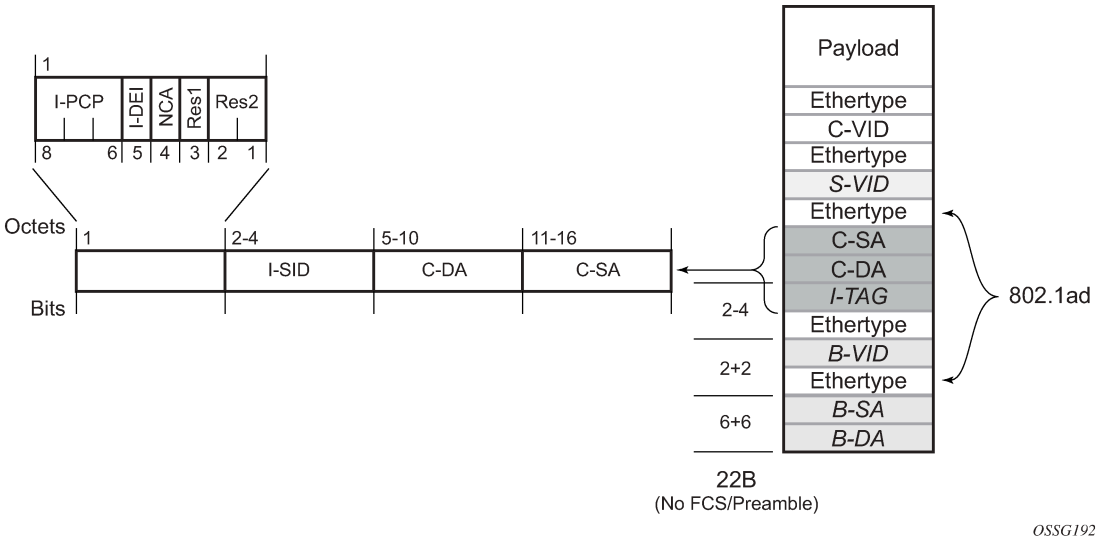
PBB encapsulation is added at the user facing PE (U-PE) to hide the customer MAC addressing and topology from the N-PE devices. The core N-PEs need to only handle backbone MAC addressing and do not need to have visibility of each customer VPN. As a result, the integrated PBB-VPLS solution decreases the load in the N-PEs and improves the overall stability of the backbone.

The Nokia PBB-VPLS solution also provides automatic discovery of the customer VPNs through the implementation of IEEE 802.1ak MMRP minimizing the number of provisioning touches required at the N-PEs.

5.2.2 PBB technology

IEEE 802.1ah specification encapsulates the customer or QinQ payload in a provider header as shown in Figure 105: QinQ payload in provider header example.

Figure 105: QinQ payload in provider header example



PBB adds a regular Ethernet header where the B-DA and B-SA are the backbone destination and respectively, source MACs of the edge U-PEs. The backbone MACs (B-MACs) are used by the core N-PE devices to switch the frame through the backbone.

A special group MAC is used for the backbone destination MAC (B-DA) when handling an unknown unicast, multicast or broadcast frame. This backbone group MAC is derived from the I-service instance identifier (ISID) using the rule: a standard group OUI (01-1E-83) followed by the 24 bit ISID coded in the last three bytes of the MAC address.

The BVID (backbone VLAN ID) field is a regular DOT1Q tag and controls the size of the backbone broadcast domain. When the PBB frame is sent over a VPLS pseudowire, this field may be omitted depending on the type of pseudowire used.

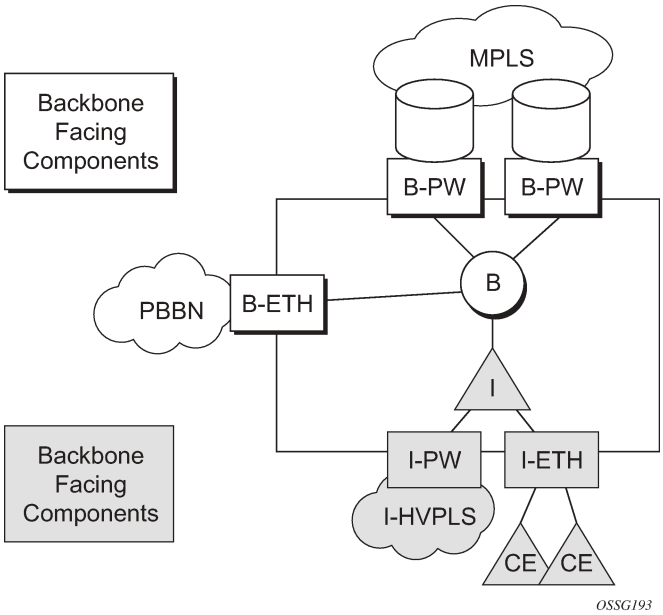
The following ITAG (standard Ether-type value of 0x88E7) has the role of identifying the customer VPN to which the frame is addressed through the 24 bit ISID. Support for service QoS is provided through the priority (3 bit I-PCP) and the DEI (1 bit) fields.

5.2.3 PBB mapping to existing VPLS configurations

The IEEE model for PBB is organized around a B-component handling the provider backbone layer and an I-component concerned with the mapping of the customer/provider bridge (QinQ) domain (MACs, VLANs) to the provider backbone (B-MACs, B-VLANs): for example, the I-component contains the boundary between the customer and backbone MAC domains.

The Nokia implementation is extending the IEEE model for PBB to allow support for MPLS pseudowires using a chain of two VPLS context linked together as depicted in Figure 106: PBB mapping to VPLS configurations.

Figure 106: PBB mapping to VPLS configurations



A VPLS context is used to provide the backbone switching component. The white circle marked B, referred to as backbone-VPLS (B-VPLS), operates on backbone MAC addresses providing a core multipoint infrastructure that may be used for one or multiple customer VPNs. The Nokia B-VPLS implementation allows the use of both native PBB and MPLS infrastructures.

Another VPLS context (I-VPLS) can be used to provide the multipoint I-component functionality emulating the E-LAN service (see the triangle marked "I" in Figure 106: PBB mapping to VPLS configurations). Similar to B-VPLS, I-VPLS inherits from the regular VPLS the pseudowire (SDP bindings) and native Ethernet (SAPs) handoffs accommodating this way different types of access: for example, direct customer link, QinQ or HVPLS.

To support PBB E-Line (point-to-point service), the use of an Epipe as I-component is allowed. All Ethernet SAPs supported by a regular Epipe are also supported in the PBB Epipe.

5.2.4 SAP and SDP support

This section provides information about SAP and SDP support.

5.2.4.1 PBB B-VPLS

- The following describes SAP support for PBB B-VPLS:
  - Ethernet DOT1Q and QinQ are supported. This is applicable to most PBB use cases, for example, one backbone VLAN ID used for native Ethernet tunneling. In the case of QinQ, a single tag x is supported on a QinQ encapsulation port for example (1/1/1:x.* or 1/1/1:x.0).
  - Ethernet null is supported. This is supported for a direct connection between PBB PEs, for example, no BVID is required.
  - Default SAP types are blocked in the CLI for the B-VPLS SAP.

- The following rules apply to the SAP processing of PBB frames:
  - For "transit frames" (not destined for a local B-MAC), there is no need to process the ITAG component of the PBB frames. Regular Ethernet SAP processing is applied to the backbone header (B-MACs and BVID).
  - If a local I-VPLS instance is associated with the B-VPLS, "local frames" originated/terminated on local I-VPLSs are PBB encapsulated/de-encapsulated using the **pbb-etype** provisioned under the related port or SDP component.
- The following describes SDP support for PBB B-VPLS:
  - For MPLS, both mesh and spoke-SDPs with split horizon groups are supported.
  - Similar to regular pseudowire, the outgoing PBB frame on an SDP (for example, B-pseudowire) contains a BVID qtag only if the pseudowire type is Ethernet VLAN. If the pseudowire type is 'Ethernet', the BVID qtag is stripped before the frame goes out.

5.2.4.2 PBB I-VPLS

- **port level**

  All existing Ethernet encapsulation types are supported (for example, null, dot1q, qinq).
- **SAPs**

  The following describes SAP support for PBB I-VPLS:
  - The I-VPLS SAPs can coexist on the same port with SAPs for other business services, for example, VLL, VPLS SAPs.
  - All existing Ethernet encapsulation are supported: null, dot1q, qinq.
- **SDPs**

  GRE and MPLS SDP are spoke-sdp only. Mesh SDPs can just be emulated by using the same split horizon group everywhere.

Existing SAP processing rules still apply for the I-VPLS case; the SAP encapsulation definition on Ethernet ingress ports defines which VLAN tags are used to determine the service that the packet belongs to:

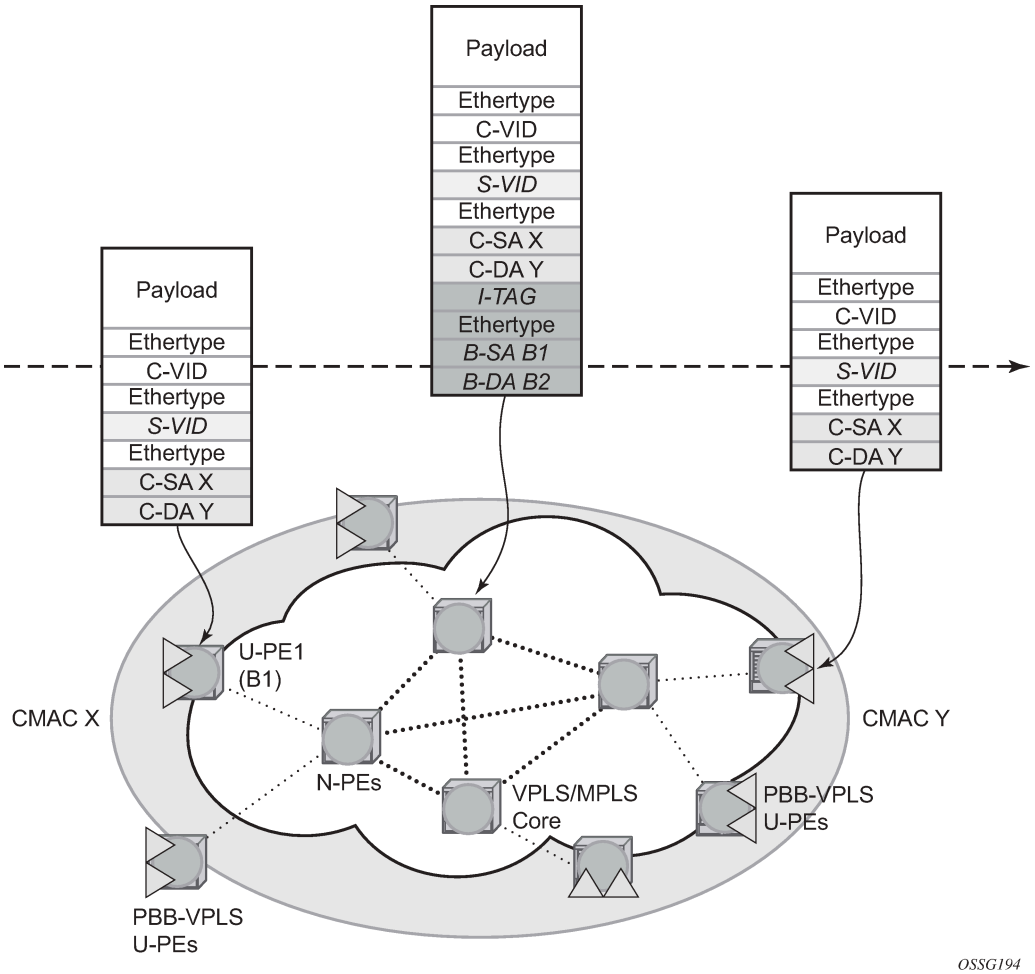
- For null encapsulations defined on ingress, any VLAN tags are ignored and the packet goes to a default service for the SAP
- For dot1q encapsulations defined on ingress, only the first VLAN tag is considered.
- For qinq encapsulations defined on ingress, both VLAN tags are considered; wildcard support is for the inner VLAN tag.
- For dot1q/qinq encapsulations, traffic encapsulated with VLAN tags for which there is no definition is discarded.
- Any VLAN tag used for service selection on the I-SAP is stripped before the PBB encapsulation is added. Appropriate VLAN tags are added at the remote PBB PE when sending the packet out on the egress SAP.

I-VPLS services do not support the forwarding of PBB encapsulated frames received on SAPs or spoke SDPs through their associated B-VPLS service. PBB frames are identified based on the configured PBB Ethertype (0x88e7 by default).

5.2.5 PBB packet walkthrough

This section describes the walkthrough for a packet that traverses the B-VPLS and I-VPLS instances using the example of a unicast frame between two customer stations as depicted in the following network diagram Figure 107: PBB packet walkthrough.

Figure 107: PBB packet walkthrough



The station with C-MAC (customer MAC) X wants to send a unicast frame to C-MAC Y through the PBB-VPLS network. A customer frame arriving at PBB-VPLS U-PE1 is encapsulated with the PBB header. The local I-VPLS FDB on U-PE1 is consulted to determine the destination B-MAC of the egress U-PE for C-MAC Y. In our example, B2 is assumed to be known as the B-DA for Y. If C-MAC Y is not present in the U-PE1 forwarding database, the PBB packet is sent in the B-VPLS using the standard group MAC address for the ISID associated with the customer VPN. If the up link to the N-PE is a spoke pseudowire, the related PWE3 encapsulation is added in front of the B-DA.

Next, only the Backbone Header in green is used to switch the frame through the green B-VPLS/VPLS instances in the N-PEs. At the receiving U-PE2, the C-MAC X is learned as being behind B-MAC B1; then the PBB encapsulation is removed and the lookup for C-MAC Y is performed. In the case where a pseudowire is used between N-PE and U-PE2, the pseudowire encapsulation is removed first.

5.2.5.1 PBB control planes

PBB technology can be deployed in a number of environments. Natively, PBB is an Ethernet data plane technology that offers service scalability and multicast efficiency.

Environment:

- MPLS (mesh and spoke-SDPs)
- Ethernet SAPs

Within these environments, SR OS offers a number of optional control planes:

- Shortest Path Bridging MAC (SPBM) (SAPs and spoke-SDPs); see SPBM
- Rapid Spanning Tree Protocol (RSTP) optionally with MMRP (SAPs and spoke-SDPs); see MMRP support over B-VPLS SAPs and SDPs.
- MSTP optionally with MMRP (SAPs and spoke-SDPs); see Multiple spanning tree.
- Multiple MAC registration Protocol (MMRP) alone (SAPs, spoke and mesh SDPs); see IEEE 802.1ak MMRP for service aggregation and zero touch provisioning.

In general a control plane is required on Ethernet SAPs, or SDPs where there could be physical loops. Some network configurations of Mesh and Spoke SDPs can avoid physical loops and no control plane is required.

The choice of control plane is based on the requirement of the networks. SPBM for PBB offers a scalable link state control plane without B-MAC flooding and learning or MMRP. RSTP and MSTP offer Spanning tree options based on B-MAC flooding and learning. MMRP is used with flooding and learning to improve multicast.

5.2.6 SPBM

Shortest Path Bridging (SPB) enables a next generation control plane for PBB based on IS-IS that adds the stability and efficiency of link state to unicast and multicast services. Specifically this is an implementation of SPBM (SPB MAC mode). Current SR OS PBB B-VPLS offers point-to-point and multipoint to multipoint services with large scale. PBB B-VPLS is deployed in both Ethernet and MPLS networks supporting Ethernet VLL and VPLS services. SPB removes the flooding and learning mode from the PBB Backbone network and replaces MMRP for ISID Group MAC Registration providing flood containment. SR OS SPB provides true shortest path forwarding for unicast and efficient forwarding on a single tree for multicast. It supports selection of shortest path equal cost tie-breaking algorithms to enable diverse forwarding in an SPB network.

5.2.6.1 Flooding and learning versus link state

SPB brings a link state capability that improves the scalability and performance for large networks over the xSTP flooding and learning models. Flooding and learning has two consequences. First, a message invoking a flush must be propagated, second the data plane is allowed to flood and relearn while flushing is happening. Message based operation over these data planes may experience congestion and packet loss.

Table 16: B-VPLS control planes

PBB B-VPLS Control plane	Flooding and learning	Multipath	Convergence time
xSTP	Yes	MSTP	xSTP + MMRP
G.8032	Yes	Multiple Ring instances Ring topologies only	Eth-OAM based + MMRP
SPB-M	No	Yes –ECT based	IS-IS link state (incremental)

Link state operates differently in that only the information that truly changes needs to be updated. Traffic that is not affected by a topology change does not have to be disturbed and does not experience congestion because there is no flooding. SPB is a link state mechanism that uses restoration to reestablish the paths affected by topology change. It is more deterministic and reliable than RSTP and MMRP mechanisms. SPB can handle any number of topology changes and as long as the network has some connectivity, SPB does not isolate any traffic.

5.2.6.2 SPB for B-VPLS

The SR OS model supports PBB Epipes and I-VPLS services on the B-VPLS. SPB is added to B-VPLS in place of other control planes (see Table 16: B-VPLS control planes). SPB runs in a separate instance of IS-IS. SPB is configured in a single service instance of B-VPLS that controls the SPB behavior (via IS-IS parameters) for the SPB IS-IS session between nodes. Up to four independent instances of SPB can be configured. Each SPB instance requires a separate control B-VPLS service. A typical SPB deployment uses a single control VPLS with zero, one or more user B-VPLS instances. SPB is multi-topology (MT) capable at the IS-IS LSP TLV definitions however logical instances offer the nearly the same capability as MT. The SR OS SPB implementation always uses MT topology instance zero. Area addresses are not used and SPB is assumed to be a single area. SPB must be consistently configured on nodes in the system. SPB Regions information and IS-IS hello logic that detect mismatched configuration are not supported.

SPB Link State PDUs (LSPs) contains B-MACs, I-SIDs (for multicast services) and link and metric information for an IS-IS database. Epipe I-SIDs are not distributed in SR OS SPB allowing high scalability of PBB Epipes. I-VPLS I-SIDs are distributed in SR OS SPB and the respective multicast group addresses are automatically populated in forwarding in a manner that provides automatic pruning of multicast to the subset of the multicast tree that supports I-VPLS with a common I-SID. This replaces the function of MMRP and is more efficient than MMRP so that in the future, SPB scales to a greater number of I-SIDs.

SPB on SR OS can leverage MPLS networks or Ethernet networks or combinations of both. SPB allows PBB to take advantage of multicast efficiency and at the same time leverage MPLS features such as resiliency.

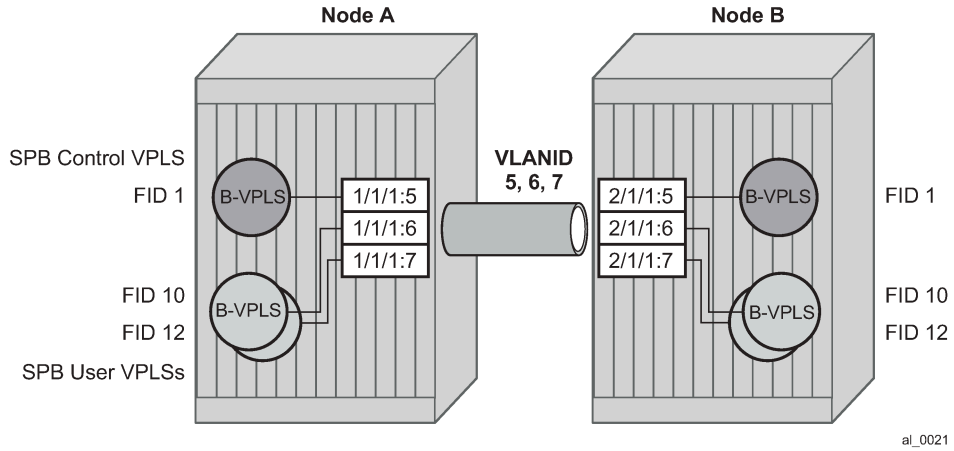
5.2.6.3 Control B-VPLS and user B-VPLS

Control B-VPLS are required for the configuration of the SPB parameters and as a service to enable SPB. Control B-VPLS therefore must be configured everywhere SPB forwarding is expected to be active even if there are no terminating services. SPB uses the logical instance and a Forwarding ID (FID) to identify SPB locally on the node. The FID is used in place of the SPB VLAN identifier (Base VID) in IS-IS LSPs enabling

a reference to exchange SPB topology and addresses. More specifically, SPB advertises B-MACs and I-SIDs in a B-VLAN context. Because the service model in SR OS separates the VLAN Tag used on the port for encapsulation from the VLAN ID used in SPB the SPB VLAN is a logical concept and is represented by configuring a FID. B-VPLS SAPs use VLAN Tags (SAPs with Ethernet encapsulation) that are independent of the FID value. The encapsulation is local to the link in SR/ESS so the SAP encapsulation has been configured the same between neighboring switches. The FID for a specified instance of SPB between two neighbor switches must be the same. The independence of VID encapsulation is inherent to SR OS PBB B-VPLS. This also allows spoke-SDP bindings to be used between neighboring SPB instances without any VID tags. The one exception is mesh SDPs are not supported but arbitrary mesh topologies are supported by SR OS SPB.

Figure 108: Control and user B-VPLS with FIDs shows two switches where an SPB control B-VPLS configured with FID 1 and uses a SAP with 1/1/1:5 therefore using a VLAN Tag 5 on the link. The SAP 1/1/1:1 could also have been be used but in SR OS the VID does not have to equal FID. Alternatively an MPLS PW (spoke-SDP binding) could be for some interfaces in place of the SAP. Figure 108: Control and user B-VPLS with FIDs shows a control VPLS and two user B-VPLS. The User B-VPLS must share the same topology and are required to have interfaces on SAPs/Spoke SDPs on the same links or LAG groups as the B-VPLS. To allow services on different B-VPLS to use a path when there are multiple paths a different ECT algorithm can be configured on a B-VPLS instance. In this case, the user B-VPLS still fate shared the same topology but they may use different paths for data traffic; see Shortest path and single tree.

Figure 108: Control and user B-VPLS with FIDs



Each user BVPLS offers the same service capability as a control B-VPLS and are configured to "follow" or fate share with a control B-VPLS. User B-VPLS must be configured as active on the whole topology where control B-VPLS is configured and active. If there is a mismatch between the topology of a user B-VPLS and the control B-VPLS, only the user B-VPLS links and nodes that are in common with the control B-VPLS function. The services on any B-VPLS are independent of a particular user B-VPLS so a misconfiguration of one of the user B-VPLS does not affect other B-VPLS. For example if a SAP or spoke-SDP is missing in the user B-VPLS any traffic from that user B-VPLS that would use that interface, is missing forwarding information and traffic is dropped only for that B-VPLS. The computation of paths is based only on the control B-VPLS topology.

User B-VPLS instances supporting only unicast services (PBB-Epipes) may share the FID with the other B-VPLS (control or user). This is a configuration short cut that reduces the LSP advertisement size for B-VPLS services but results in the same separation for forwarding between the B-VPLS services. In the case

of PBB-Epipes only B-MACs are advertised per FID but B-MACs are populated per B-VPLS in the FDB. If I-VPLS services are to be supported on a B-VPLS that B-VPLS must have an independent FID.

5.2.6.4 Shortest path and single tree

IEEE 802.1aq standard SPB uses a source specific tree model. The standard model is more computationally intensive for multicast traffic because in addition to the SPF algorithm for unicast and multicast from a single node, an all pairs shorted path needs to be computed for other nodes in the network. In addition, the computation must be repeated for each ECT algorithm. While the standard yields efficient shortest paths, this computation is overhead for systems where multicast traffic volume is low. Ethernet VLL and VPLS unicast services are popular in PBB networks and the SR OS SPB design is optimized for unicast delivery using shortest paths. Ethernet supporting unicast and multicast services are commonly deployed in Ethernet transport networks. SR OS SPB Single tree multicast (also called shared tree or *,G) operates similarly today. The difference is that SPB multicast never floods unknown traffic.

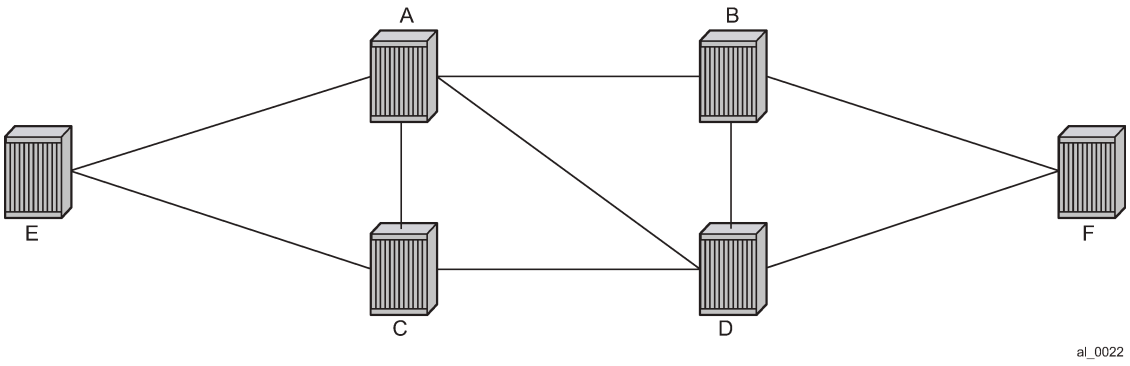
The SR OS implementation of SPB with shortest path unicast and single tree multicast, requires only two SPF computations per topology change reducing the computation requirements. One computation is for unicast forwarding and the other computation is for multicast forwarding.

A single tree multicast requires selecting a root node much like RSTP. Bridge priority controls the choice of root node and alternate root nodes. The numerically lowest Bridge Priority is the criteria for choosing a root node. If multiple nodes have the same Bridge Priority, then the lowest Bridge Identifier (System Identifier) is the root.

In SPB the source-bmac can override the chassis-mac allowing independent control of tie breaking, The shortest path unicast forwarding does not require any special configuration other than selecting the ECT algorithm by configuring a B-VPLS use a FID with low-path-id algorithm or high-path-id algorithm to be the tiebreaker between equal cost paths. Bridge priority allows some adjustment of paths. Configuring link metrics adjusts the number of equal paths.

To illustrate the behavior of the path algorithms an example network is shown in Figure 109: Example partial mesh network.

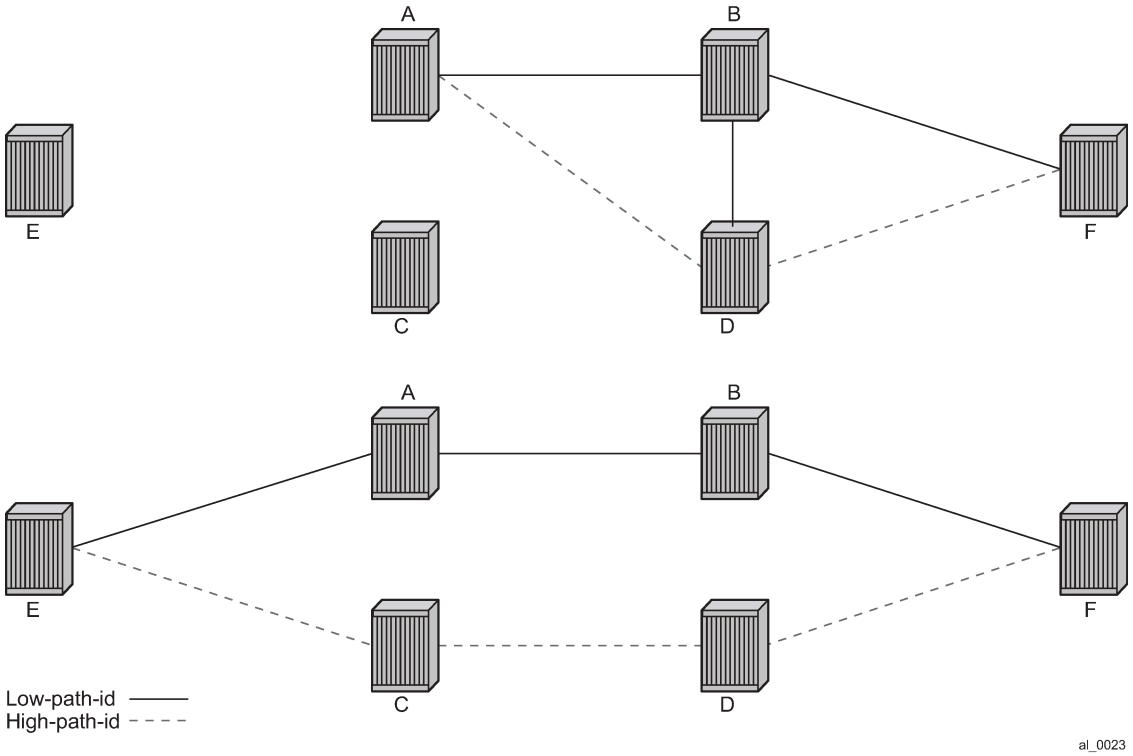
Figure 109: Example partial mesh network



Assume that Node A is the lowest Bridge Identifier and the Multicast root node and all links have equal metrics. Also, assume that Bridge Identifiers are ordered such that Node A has a numerically lower Bridge identifier than Node B, and Node B has lower Bridge Identifier than Node C, and so on, Unicast paths are configured to use shortest path tree (SPT). Figure 110: Unicast paths for low-path-id and high-path-id shows the shortest paths computed from Node A and Node E to Node F. There are only two shortest paths from A to F. A choice of low-path-id algorithm uses Node B as transit node and a path using high-

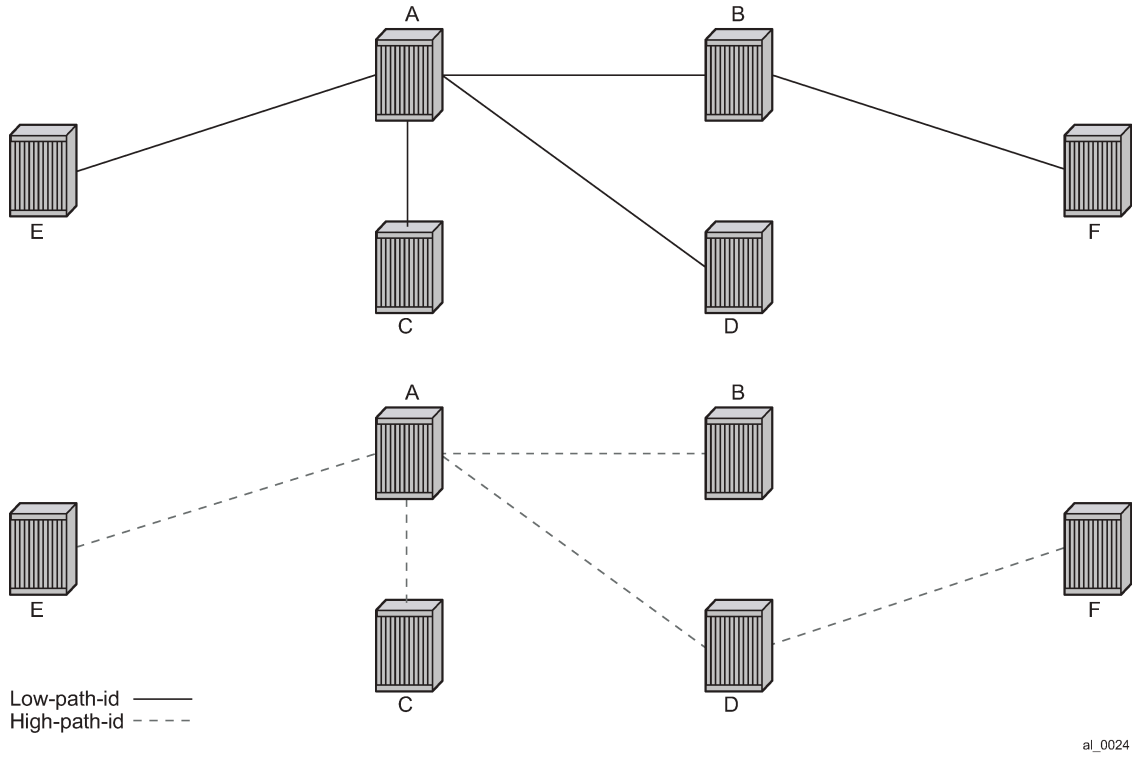
path-id algorithm uses Node D as transit node. The reverse paths from Node F to A are the same (all unicast paths are reverse path congruent). For Node E to Node F there are three paths E-A-B-F, E-A-D-F, and E-C-D-F. The low-path-id algorithm uses path E-A-B-F and the high-path-id algorithm uses E-C-D-F. These paths are also disjoint and are reverse path congruent. Any nodes that are directly connected in this network have only one path between them (not shown for simplicity).

Figure 110: Unicast paths for low-path-id and high-path-id



For Multicast paths the algorithms used are the same low-path-id or high-path-id but the tree is always a single tree using the root selected as described earlier (in this case Node A). Figure 111: Multicast paths for low-path-id and high-path-id shows the multicast paths for low-path-id and high-path-id algorithm.

Figure 111: Multicast paths for low-path-id and high-path-id



All nodes in this network use one of these trees. The path for multicast to/from Node A is the same as unicast traffic to/from Node A for both low-path-id and high-path-id. However, the multicast path for other nodes is now different from the unicast paths for some destinations. For example, Node E to Node F is now different for high-path-id because the path must transit the root Node A. In addition, the Node E multicast path to C is E-A-C even though E has a direct path to Node C. A rule of thumb is that the node chosen to be root should be a well-connected node and have available resources. In this example, Node A and Node D are the best choices for root nodes.

The distribution of I-SIDs allows efficient pruning of the multicast single tree on a per I-SID basis because only MFIB entries between nodes on the single tree are populated. For example, if Nodes A, B and F share an I-SID and they use the low-path–id algorithm only those three nodes would have multicast traffic for that I-SID. If the high-path-id algorithm is used traffic from Nodes A and B must go through D to get to Node F.

5.2.6.5 Data path and forwarding

The implementation of SPB on SR OS uses the PBB data plane. There is no flooding of B-MAC based traffic. If a B-MAC is not found in the FDB, traffic is dropped until the control plane populates that B-MAC. Unicast B-MAC addresses are populated in all FDBs regardless of I-SID membership. There is a unicast FDB per B-VPLS both control B-VPLS and user BVPLS. B-VPLS instances that do not have any I-VPLS, have only a default multicast tree and do not have any multicast MFIB entries.

The data plane supports an ingress check (reverse path forwarding check) for unicast and multicast frames on the respective trees. Ingress check is performed automatically. For unicast or multicast frames the B-MAC of the source must be in the FDB and the interface must be valid for that B-MAC or traffic is dropped. The PBB encapsulation (See PBB technology) is unchanged from current SR OS. Multicast frames

use the PBB Multicast Frame format and SPBM distributes I-VPLS I-SIDs which allows SPB to populate forwarding only to the relevant branches of the multicast tree. Therefore, SPB replaces both spanning tree control and MMRP functionality in one protocol.

By using a single tree for multicast the amount of MFIB space used for multicast is reduced (per source shortest path trees for multicast are not currently offered on SR OS). In addition, a single tree reduces the amount of computation required when there is topology change.

5.2.6.6 SPB Ethernet OAM

Ethernet OAM works on Ethernet services and use a combination of unicast with learning and multicast addresses. SPB on SR OS supports both unicast and multicast forwarding, but with no learning and unicast and multicast may take different paths. In addition, SR OS SPB control plane offers a wide variety of show commands. The SPB IS-IS control plane takes the place of many Ethernet OAM functions. SPB IS-IS frames (Hello and PDU and so on) are multicast but they are per SPB interface on the control B-VPLS interfaces and are not PBB encapsulated.

All Client Ethernet OAM is supported from I-VPLS interfaces and PBB Epipe interfaces across the SPB domain. Client OAM is the only true test of the PBB data plane. The only forms of Eth-OAM supported directly on SPB B-VPLS are Virtual MEPS (vMEPs). Only CCM is supported on these vMEPs; vMEPs use a S-TAG encapsulation and follow the SPB multicast tree for the specified B-VPLS. Each MEP has a unicast associated MAC to terminate various ETH-CFM tools. However, CCM messages always use a destination Layer 2 multicast using 01:80:C2:00:00:3x (where x = 0 to 7). vMEPs terminate CCM with the multicast address. Unicast CCM can be configured for point to point associations or hub and spoke configuration but this would not be typical (when unicast addresses are configured on vMEPs they are automatically distributed by SPB in IS-IS).

Up MEPs on services (I-VPLS and PBB Epipes) are also supported and these behave as any service OAM. These OAM use the PBB encapsulation and follow the PBB path to the destination.

Link OAM or 802.1ah EFM is supported below SPB as standard. This strategy of SPB IS-IS and OAM gives coverage.

Table 17: SPB Ethernet OAM operation summary

OAM origination	Data plane support	Comments
PBB-Epipe or Customer CFM on PBB Epipe. Up MEPs on PBB Epipe.	Fully Supported. Unicast PBB frames encapsulating unicast/multicast.	Transparent operation. Uses Encapsulated PBB with Unicast B-MAC address.
I-VPLS or Customer CFM on I-VPLS. Up MEPs on I-VPLS.	Fully Supported. Unicast/Multicast PBB frames determined by OAM type.	Transparent operation. Uses Encapsulated PBB frames with Multicast/Unicast B-MAC address.
vMEP on B-VPLS Service.	CCM only. S-Tagged Multicast Frames.	Ethernet CCM only. Follows the Multicast tree. Unicast addresses may be configured for peer operation.

In summary SPB offers an automated control plane and optional Eth-CFM/Eth-EFM to allow monitoring of Ethernet Services using SPB. B-VPLS services PBB Epipes and I-VPLS services support the existing set of Ethernet capabilities.

5.2.6.7 SPB levels

Levels are part of IS-IS. SPB supports Level 1 within a control B-VPLS. Future enhancements may make use of levels.

5.2.7 SPBM to non-SPBM interworking

By using static definitions of B-MACs and ISIDs interworking of PBB Epipes and I-VPLS between SPBM networks and non-SPBM PBB networks can be achieved.

5.2.7.1 Static MACs and static ISIDs

To extend SPBM networks to other PBB networks, static MACs and ISIDs can be defined under SPBM SAPs/SDPs. The declaration of a static MAC in an SPBM context allows a non-SPBM PBB system to receive frames from an SPBM system. These static MACs are conditional on the SAP/SDP operational state. Currently this is only supported for SPBM because SPBM can advertise these B-MACs and ISIDs without any requirement for flushing. The B-MAC (and B-MAC to ISID) must remain consistent when advertised in the IS-IS database.

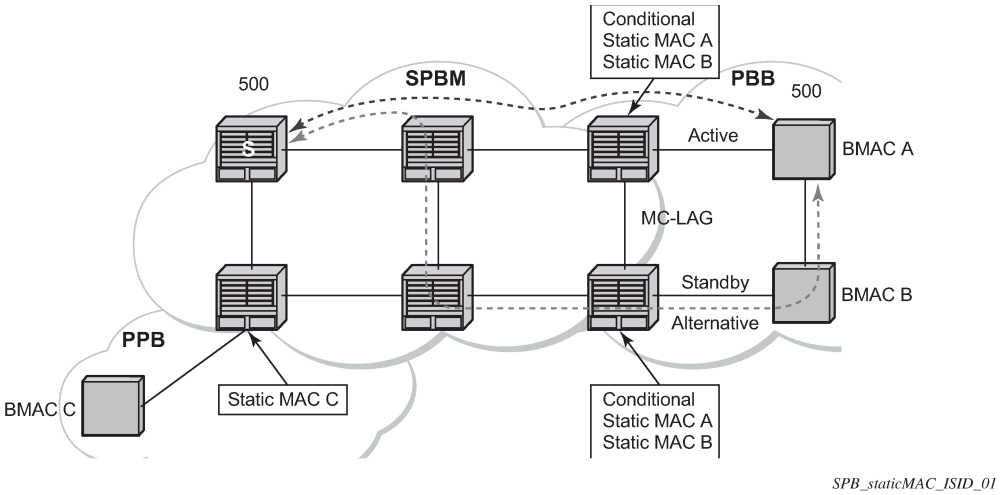
The declaration of static-isids allows an efficient connection of ISID based services. The ISID is advertised as supported on the local nodal B-MAC and the static B-MACs which are the true destinations for the ISIDs are also advertised. When the I-VPLS learn the remote B-MAC they associate the ISID with the true destination B-MAC. Therefore if redundancy is used the B-MACs and ISIDs that are advertised must be the same on any redundant interfaces.

If the interface is an MC-LAG interface the static MAC and ISIDs on the SAPs/SDPs using that interface are only active when the associated MC-LAG interface is active. If the interface is a spoke-SDP on an active/ standby pseudo wire (PW) the ISIDs and B-MACs are only active when the PW is active.

5.2.7.2 Epipe static configuration

For Epipe only, the B-MACs need to be advertised. There is no multicast for PBB Epipes. Unicast traffic follows the unicast path shortest path or single tree. By configuring remote B-MACs Epipes can be setup to non-SPBM systems. A special conditional static-mac is used for SPBM PBB B-VPLS SAPs/SDPs that are connected to a remote system. In the diagram ISID 500 is used for the PBB Epipe but only conditional MACs A and B are configured on the MC-LAG ports. The B-VPLS advertises the static MAC either always or optionally based on a condition of the port forwarding.

Figure 112: Static MACs example



5.2.7.2.1 I-VPLS static config

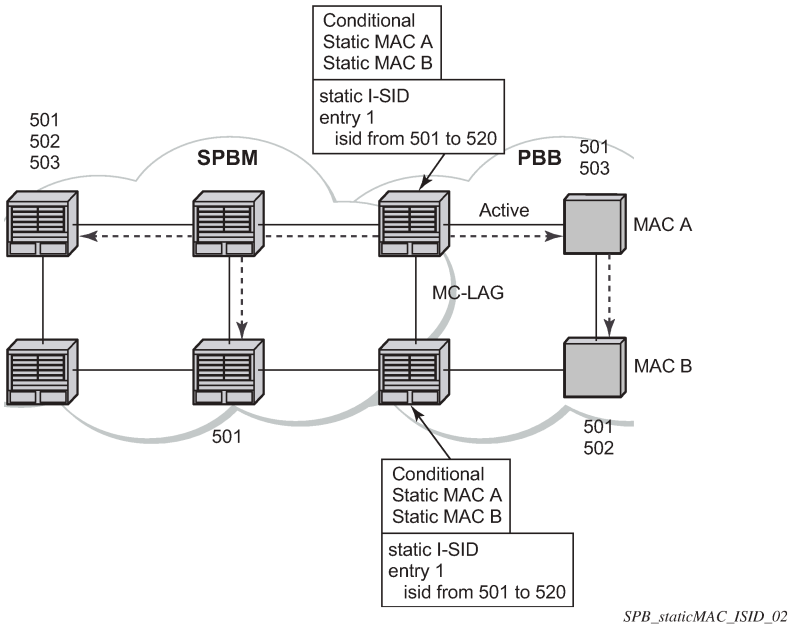
I-VPLS static config consists of two components: static-mac and static ISIDs that represent a remote B-MAC-ISID combination.

The static-MACs are configured as with Epipe, the special conditional static-mac is used for SPBM PBB B-VPLS SAPs/SDPs that are connected to a remote system. The B-VPLS advertises the static MAC either always or optionally based on a condition of the port forwarding.

The static-isids are created under the B-VPLS SAP/SDPs that are connected to a non-SPBM system. These ISIDs are typically advertised but may be controlled by ISID policy.

For I-VPLS ISIDs the ISIDs are advertised and multicast MAC are automatically created using PBB-OUI and the ISID. SPBM supports the pruned multicast single tree. Unicast traffic follows the unicast path shortest path or single tree. Multicast/and unknown Unicast follow the pruned single tree for that ISID.

Figure 113: Static ISIDs example



5.2.7.3 SPBM ISID policies

ISID policies are an optional aspect of SPBM which allow additional control of ISIDs for I-VPLS. PBB services using SPBM automatically populate multicast for I-VPLS and static-isids. Incorrect use of isid-policy can create black holes or additional flooding of multicast.

To enable more flexible multicast, ISID policies control the amount of MFIB space used by ISIDs by trading off the default Multicast tree and the per ISID multicast tree. Occasionally customers want services that use I-VPLS that have multiple sites but use primarily unicast. The ISID policy can be used on any node where an I-VPLS is defined or static ISIDs are defined.

The typical use is to suppress the installation of the ISID in the MFIB using use-def-mcast and the distribution of the ISID in SPBM by using no advertise-local.

The use-def-mcast policy instructs SPBM to use the default B-VPLS multicast forwarding for the ISID range. The ISID multicast frame remains unchanged by the policy (the standard format with the PBB OUI and the ISID as the multicast destination address) but no MFIB entry is allocated. This causes the forwarding to use the default BVID multicast tree which is not pruned. When this policy is in place it only governs the forwarding locally on the current B-VPLS.

The advertise local policy ISID policies are applied to both static ISIDs and I-VPLS ISIDs. The policies define whether the ISIDs are advertised in SPBM and whether the use the local MFIB. When ISIDs are advertised they use the MFIB in the remote nodes. Locally the use of the MFIB is controlled by the **use-def-mcast** policy.

The types of interfaces are summarized in Table 18: SPBM ISID policies table.

Table 18: SPBM ISID policies table

Service type	ISID policy on B-VPLS	Notes
Epipe	No effect	PBB Epipe ISIDs are not advertised or in MFIB.
I-VPLS	None: Uses ISID Multicast tree. Advertised ISIDs of I-VPLS.	I-VPLS uses dedicated (pruned) multicast tree. ISIDs are advertised.
I-VPLS (for Unicast)	use-def-mcast no advertise-local	I-VPLS uses default Multicast. Policy only required where ISIDs are defined. ISIDs not advertised. must be consistently defined on all nodes with same ISIDs.
I-VPLS (for Unicast)	use-def-mcast advertise-local	I-VPLS uses default Multicast. Policy only required where ISIDs are defined. ISIDs advertised and pruned tree used elsewhere. May be inconsistent for an ISID.
Static ISIDs for I-VPLS interworking	None: (recommended) Uses ISID Multicast tree	I-VPLS uses dedicated (pruned) multicast tree. ISIDs are advertised.
Static ISIDs for I-VPLS interworking (defined locally)	use-def-mcast	I-VPLS uses default Multicast. Policy only required where ISIDs are configured or where I-VPLS is located.
No MFIB for any ISIDs Policy defined on all nodes	use-def-mcast no advertise-local	Each B-VPLS with the policy does not install MFIB. Policy defined on all switches ISIDs are defined. ISIDs advertised and pruned tree used elsewhere. May be inconsistent for an ISID.

5.2.8 ISID policy control

5.2.8.1 Static ISID advertisement

Static ISIDs are advertised between using the SPBM Service Identifier and Unicast Address sub-TLV in IS-IS when there is no ISID policy. This TLV advertises the local B-MAC and one or more ISIDs. The B-MAC used is the source-bmac of the Control/User VPLS. Typically remote B-MACs (the ultimate source-bmac) and the associated ISIDs are configured as static under the SPBM interface. This allows all remote B-MACs and all remote ISIDs to be configured one time per interface.

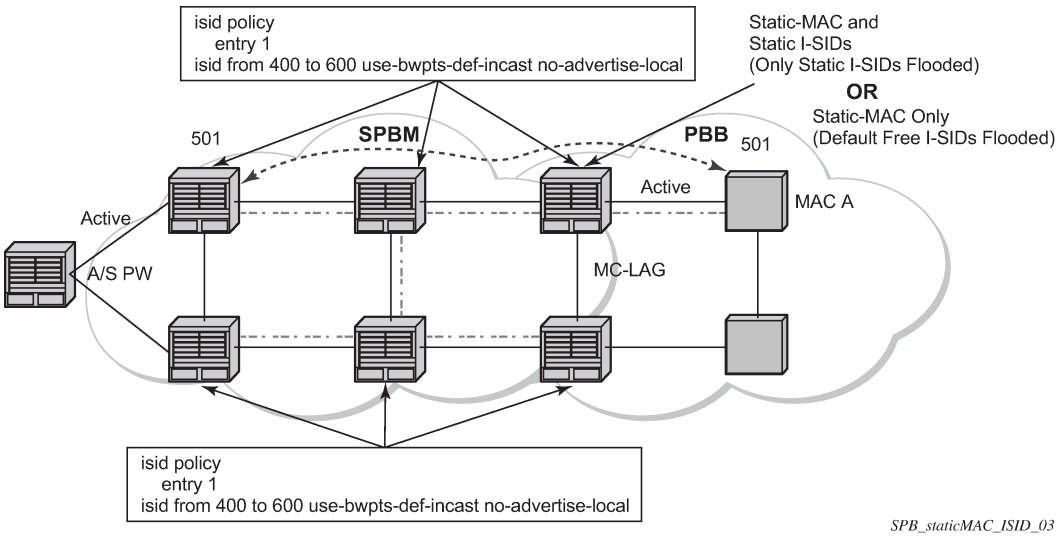
5.2.8.2 I-VPLS for unicast service

If the service is using unicast only an I-VPLS still uses MFIB space and SPBM advertises the ISID. By using the default multicast tree locally, a node saves MFIB space. By using the no advertise-local SPBM does not advertise the ISIDs covered by the policy. Note the actual PBB multicast frames are the same regardless of policy. Unicast traffic is the not changed for the ISID policies.

The Static B-MAC configuration is allowed under Multi-Chassis LAG (MC-LAG) based SAPs and active/standby PW SDPs.

Unicast traffic follows the unicast path shortest path or single tree. By using the ISID policy Multicast/and unknown Unicast traffic (BUM) follows the default B-VPLS tree in the SPBM domain. This should be used sparingly for any high volume of multicast services.

Figure 114: ISID policy example



5.2.9 Default behaviors

When static ISIDs are defined the default is to advertise the static ISIDs when the interface parent (SAP or SDP) is up.

If the advertisement is not needed, an ISID policy can be created to prevent advertising the ISID.

- **use-def-mcast**

  If a policy is defined with use-def-mcast the local MFIB does not contain an Multicast MAC based on the PBB OUI+ ISID and the frame is flooded out the local tree. This applies to any node where the policy is defined. On other nodes if the ISID is advertised the ISID uses the MFIB for that ISID.

- **no advertise-local**

  If a policy of no advertise-local is defined in the ISIDs, the policy is not advertised. This combination should be used everywhere there is an I-VPLS with the ISID or where the Static ISID is defined to prevent black holes. If an ISID is to be moved from advertising to no advertising it is advisable to use **use-def-mcast** on all the nodes for that ISID which allows the MFIB to not be installed and starts using the default multicast tree at each node with that policy. Then the no advertise-local option can be used.

Each Policy may be used alone or in combination.

5.2.10 Example network configuration

Figure 115: Example network

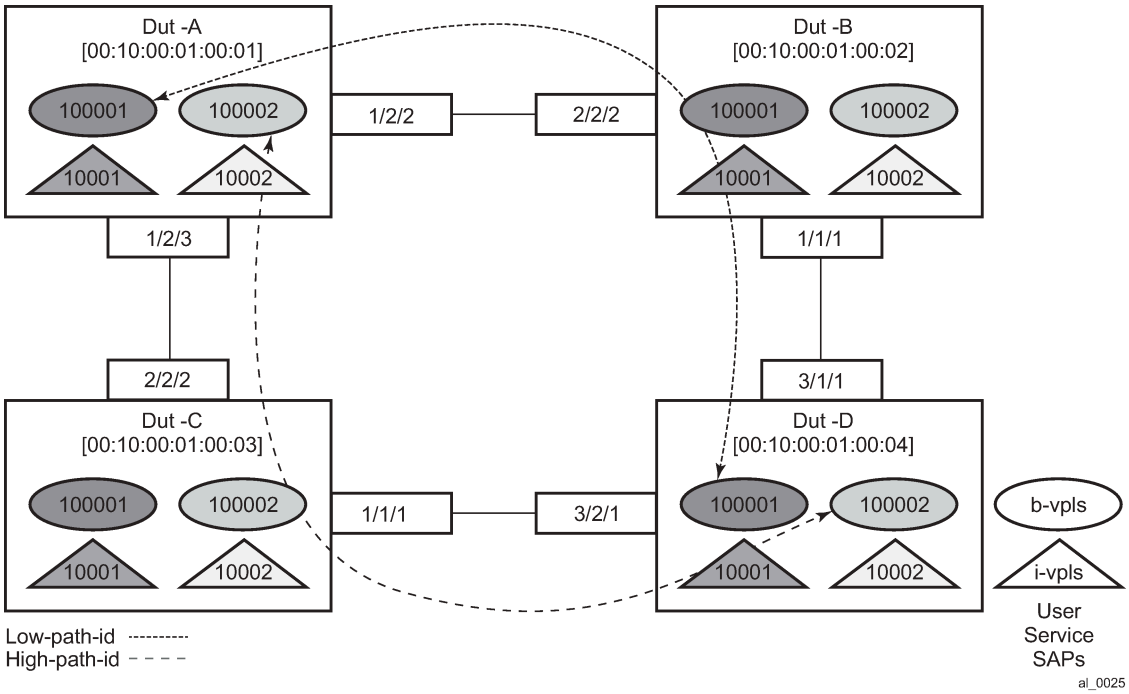


Figure 115: Example network shows an example network showing four nodes with SPB B-VPLS. The SPB instance is configured on the B-VPLS 100001. B-VPLS 100001 uses FID 1 for SPB instance 1024. All B-MACs and I-SIDs are learned in the context of B-VPLS 100001. B-VPLS 100001 has an i-vpls 10001 service, which also uses the I-SID 10001. B-VPLS 100001 is configured to use VID 1 on SAPs 1/2/2 and 1/2/3 and while the VID does not need to be the same as the FID the VID does however need to be the same on the other side (Dut-B and Dut-C).

A user B-VPLS service 100002 is configured and it uses B-VPLS 10001 to provide forwarding. It fate shares the control topology. In Figure 115: Example network, the control B-VPLS uses the low-path-id algorithm and the user B-VPLS uses high-path-id algorithm. Any B-VPLS can use any algorithm. The difference is illustrated in the path between Dut A and Dut D. The short dashed line through Dut-B is the low-path-id algorithm and the long dashed line thought Dut C is the high-path-id algorithm.

5.2.10.1 Example configuration for Dut-A

```
Dut-A:
Control B-VPLS:*A:Dut-A>config>service>vpls# pwc
------------------------------------------------------------------------------
Present Working Context :
------------------------------------------------------------------------------
 <root>
  configure
  service
  vpls "100001"
------------------------------------------------------------------------------
```

```
*A:Dut-A>config>service>vpls# info
----------------------------------------------
            pbb
                source-bmac 00:10:00:01:00:01
            exit
            stp
                shutdown
            exit
            spb 1024 fid 1 create
                level 1
                    ect-algorithm fid-range 100-100 high-path-id
                exit
                no shutdown
            exit
            sap 1/2/2:1.1 create
                spb create
                    no shutdown
                exit
            exit
            sap 1/2/3:1.1 create
                spb create
                    no shutdown
                exit
            exit
            no shutdown
----------------------------------------------
User B-VPLS:
*A:Dut-A>config>service>vpls# pwc
------------------------------------------------------------------------------
Present Working Context :
------------------------------------------------------------------------------
 <root>
  configure
  service
  vpls "100002"
------------------------------------------------------------------------------
*A:Dut-A>config>service>vpls# info
----------------------------------------------
            pbb
                source-bmac 00:10:00:02:00:01
            exit
            stp
                shutdown
            exit
            spbm-control-vpls 100001 fid 100
            sap 1/2/2:1.2 create
            exit
            sap 1/2/3:1.2 create
            exit
            no shutdown
----------------------------------------------

I-VPLS:
configure service
        vpls 10001 customer 1 i-vpls create
            service-mtu 1492
            pbb
                backbone-vpls 100001
                exit
            exit
            stp
                shutdown
            exit
            sap 1/2/1:1000.1 create
```

```
            exit
            no shutdown
        exit
        vpls 10002 customer 1 i-vpls create
            service-mtu 1492
            pbb
                backbone-vpls 100002
                exit
            exit
            stp
                shutdown
            exit
            sap 1/2/1:1000.2 create
            exit
            no shutdown
        exit
exit
```

5.2.10.1.1 Show commands outputs

The **show base** command outputs a summary of the instance parameters under a control B-VPLS. The **show** command for a user B-VPLS indicates the control B-VPLS. The base parameters except for Bridge Priority and Bridge ID must match on neighbor nodes.

```
*A:Dut-A# show service id 100001 spb base
===============================================================================
Service SPB Information
===============================================================================
Admin State        : Up                 Oper State         : Up
ISIS Instance      : 1024               FID                : 1
Bridge Priority    : 8                  Fwd Tree Top Ucast : spf
Fwd Tree Top Mcast : st
Bridge Id          : 80:00.00:10:00:01:00:01
Mcast Desig Bridge : 80:00.00:10:00:01:00:01
===============================================================================
ISIS Interfaces
===============================================================================
Interface                       Level CircID  Oper State   L1/L2 Metric
-------------------------------------------------------------------------------
sap:1/2/2:1.1                   L1    65536   Up           10/-
sap:1/2/3:1.1                   L1    65537   Up           10/-
-------------------------------------------------------------------------------
Interfaces : 2
===============================================================================
FID ranges using ECT Algorithm
-------------------------------------------------------------------------------
1-99      low-path-id
100-100   high-path-id
101-4095  low-path-id
===============================================================================
```

The **show adjacency** command displays the system ID of the connected SPB B-VPLS neighbors and the associated interfaces to connect those neighbors.

```
*A:Dut-A# show service id 100001 spb adjacency
===============================================================================
ISIS Adjacency
===============================================================================
System ID               Usage State Hold Interface                    MT Enab
-------------------------------------------------------------------------------
```

```
Dut-B                   L1    Up    19   sap:1/2/2:1.1                No
Dut-C                   L1    Up    21   sap:1/2/3:1.1                No
-------------------------------------------------------------------------------
Adjacencies : 2
===============================================================================
```

Details about the topology can be displayed with the **database** command. There is a detail option that displays the contents of the LSPs.

```
*A:Dut-A# show service id 100001 spb database
===============================================================================
ISIS Database
===============================================================================
LSP ID                                Sequence  Checksum Lifetime Attributes
-------------------------------------------------------------------------------
Displaying Level 1 database
-------------------------------------------------------------------------------
Dut-A.00-00                           0xc       0xbaba   1103     L1
Dut-B.00-00                           0x13      0xe780   1117     L1
Dut-C.00-00                           0x13      0x85a    1117     L1
Dut-D.00-00                           0xe       0x174a   1119     L1
Level (1) LSP Count : 4
===============================================================================
```

The **show routes** command shows the next hop if for the MAC addresses both unicast and multicast. The path to 00:10:00:01:00:04 (Dut-D) shows the low-path-id algorithm ID. For FID one the neighbor is Dut-B and for FID 100 the neighbor is Dut-C. Because Dut-A is the root of the multicast single tree the multicast forwarding is the same for Dut-A. However, unicast and multicast routes differ on most other nodes. Also the I-SIDs exist on all of the nodes so I-SID base multicast follows the multicast tree exactly. If the I-SID had not existed on Dut-B or Dut-D then for FID 1 there would be no entry. Note only designated nodes (root nodes) show metrics. Non-designated nodes do not show metrics.

```
*A:Dut-A# show service id 100001 spb routes
===============================================================
MAC Route Table
===============================================================
Fid  MAC                                       Ver.   Metric
      NextHop If                  SysID
---------------------------------------------------------------
Fwd Tree: unicast
---------------------------------------------------------------
1    00:10:00:01:00:02                        10     10
      sap:1/2/2:1.1               Dut-B
1    00:10:00:01:00:03                        10     10
      sap:1/2/3:1.1               Dut-C
1    00:10:00:01:00:04                        10     20
      sap:1/2/2:1.1               Dut-B
100  00:10:00:02:00:02                        10     10
      sap:1/2/2:1.1               Dut-B
100  00:10:00:02:00:03                        10     10
      sap:1/2/3:1.1               Dut-C
100  00:10:00:02:00:04                        10     20
      sap:1/2/3:1.1               Dut-C

Fwd Tree: multicast
---------------------------------------------------------------
1    00:10:00:01:00:02                        10     10
      sap:1/2/2:1.1               Dut-B
1    00:10:00:01:00:03                        10     10
      sap:1/2/3:1.1               Dut-C
1    00:10:00:01:00:04                        10     20
```

```
      sap:1/2/2:1.1                Dut-B
100  00:10:00:02:00:02                          10     10
      sap:1/2/2:1.1                Dut-B
100  00:10:00:02:00:03                          10     10
      sap:1/2/3:1.1                Dut-C
100  00:10:00:02:00:04                          10     20
      sap:1/2/3:1.1                Dut-C
-------------------------------------------------------------
No. of MAC Routes: 12
=============================================================

=============================================================
ISID Route Table
=============================================================
Fid  ISID                                       Ver.
      NextHop If                   SysID
-------------------------------------------------------------
1    10001                                      10
      sap:1/2/2:1.1                Dut-B
      sap:1/2/3:1.1                Dut-C
100  10002                                      10
      sap:1/2/2:1.1                Dut-B
      sap:1/2/3:1.1                Dut-C
-------------------------------------------------------------
No. of ISID Routes: 2
=============================================================
```

The **show service spb fdb** command shows the programmed unicast and multicast source MACs in SPB-managed B-VPLS service.

```
*A:Dut-A# show service id 100001 spb fdb

===============================================================================
User service FDB information
===============================================================================
MacAddr           UCast Source          State   MCast Source          State
-------------------------------------------------------------------------------
00:10:00:01:00:02 1/2/2:1.1             ok      1/2/2:1.1             ok
00:10:00:01:00:03 1/2/3:1.1             ok      1/2/3:1.1             ok
00:10:00:01:00:04 1/2/2:1.1             ok      1/2/2:1.1             ok
-------------------------------------------------------------------------------
Entries found: 3
===============================================================================

*A:Dut-A# show service id 100002 spb fdb
===============================================================================
User service FDB information
===============================================================================
MacAddr           UCast Source          State   MCast Source          State
-------------------------------------------------------------------------------
00:10:00:02:00:02 1/2/2:1.2             ok      1/2/2:1.2             ok
00:10:00:02:00:03 1/2/3:1.2             ok      1/2/3:1.2             ok
00:10:00:02:00:04 1/2/3:1.2             ok      1/2/3:1.2             ok
-------------------------------------------------------------------------------
Entries found: 3
===============================================================================
```

The **show service spb mfib** command shows the programmed multicast ISID MAC addresses in SPB-managed B-VPLS service shows the multicast ISID pbb group mac addresses in SPB-managed B-VPLS.

Other types of *,G multicast traffic is sent over the multicast tree and these MACs are not shown. OAM traffic that uses multicast (for example vMEP CCM) takes this path for example.

```
*A:Dut-A# show service id 100001 spb mfib
===============================================================================
User service MFIB information
===============================================================================
MacAddr           ISID     Status
-------------------------------------------------------------------------------
01:1E:83:00:27:11 10001    Ok
-------------------------------------------------------------------------------
Entries found: 1
===============================================================================
*A:Dut-A# show service id 100002 spb mfib
===============================================================================
User service MFIB information
===============================================================================
MacAddr           ISID     Status
-------------------------------------------------------------------------------
01:1E:83:00:27:12 10002    Ok
-------------------------------------------------------------------------------
Entries found: 1
===============================================================================
```

5.2.10.1.2 Debug commands

Use the following commands to debug an SPB-managed B-VPLS service:

- **debug service id svc-id spb**
- **debug service id svc-id spb adjacency [{sap sap-id | spoke-sdp sdp-id:vc-id | nbr-system-id}]**
- **debug service id svc-id spb interface [{sap sap-id | spoke-sdp sdp-id:vc-id}]**
- **debug service id svc-id spb l2db**
- **debug service id svc-id spb lsdb [{system-id | lsp-id}]**
- **debug service id svc-id spb packet [packet-type] [{sap sap-id | spoke-sdp sdp-id:vc-id}] [detail]**
- **debug service id svc-id spb spf system-id**

5.2.10.1.3 Tools commands

Use the following commands to troubleshoot an SPB-managed B-VPLS service:

- **tools perform service id svc-id spb run-manual-spf**
- **tools dump service id svc-id spb**
- **tools dump service id svc-id spb default-multicast-list**
- **tools dump service id svc-id spb fid fid default-multicast-list**
- **tools dump service id svc-id spb fid fid forwarding-path destination isis-system-id forwarding-tree {unicast | multicast}**

5.2.10.1.4 Clear commands

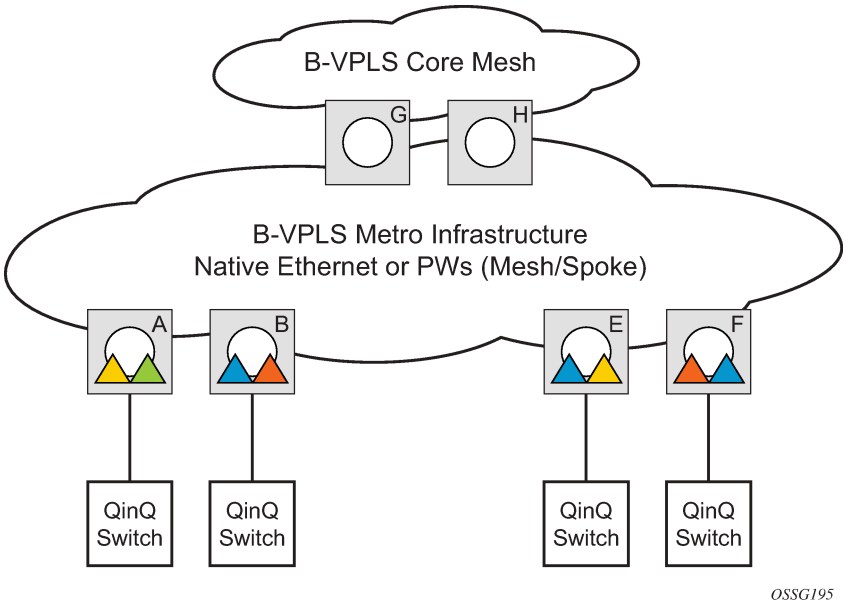
Use the following commands to clear SPB-related data:

- **clear service id svc-id spb**
- **clear service id svc-id spb adjacency system-id**
- **clear service id svc-id spb database system-id**
- **clear service id svc-id spb spf-log**
- **clear service id svc-id spb statistics**

5.2.11 IEEE 802.1ak MMRP for service aggregation and zero touch provisioning

IEEE 802.1ah supports an M:1 model where multiple customer services, represented by ISIDs, are transported through a common infrastructure (B-component). The Nokia PBB implementation supports the M:1 model allowing for a service architecture where multiple customer services (I-VPLS or Epipe) can be transported through a common B-VPLS infrastructure as depicted in Figure 116: Customer services transported in 1 B-VPLS (M:1 model).

Figure 116: Customer services transported in 1 B-VPLS (M:1 model)



The B-VPLS infrastructure represented by the white circles is used to transport multiple customer services represented by the triangles of different colors. This service architecture minimizes the number of provisioning touches and reduces the load in the core PEs: for example, G and H use less VPLS instances and pseudowire.

In a real life deployment, different customer VPNs do not share the same community of interest – for example, VPN instances may be located on different PBB PEs. The M:1 model depicted in Figure 117: Flood containment requirement in M:1 model requires a per VPN flood containment mechanism so that VPN traffic is distributed just to the B-VPLS locations that have customer VPN sites: for example, flooded traffic originated in the blue I-VPLS should be distributed just to the PBB PEs where blue I-VPLS instances are present – PBB PE B, E and F.

Per customer VPN distribution trees need to be created dynamically throughout the BVPLS as new customer I-VPLS instances are added in the PBB PEs.

The Nokia PBB implementation employs the IEEE 802.1ak Multiple MAC Registration Protocol (MMRP) to dynamically build per I-VPLS distribution trees inside a specific B-VPLS infrastructure.

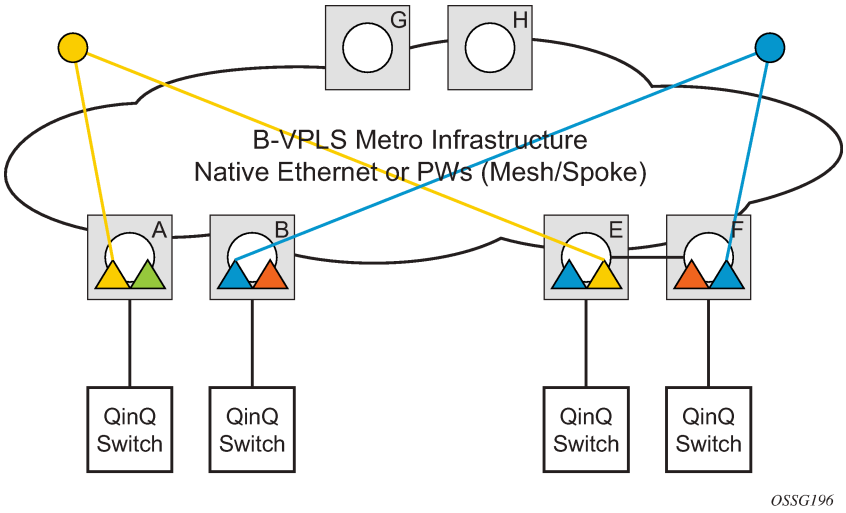
IEEE 802.1ak Multiple Registration Protocol (MRP) – Specifies changes to IEEE Std 802.1Q that provide a replacement for the GARP, GMRP and GVRP protocols. MMRP application of IEEE 802.1ak specifies the procedures that allow the registration/de-registration of MAC addresses over an Ethernet switched infrastructure.

In the PBB case, as I-VPLS instances are enabled in a specific PE, a group B-MAC address is by default instantiated using the standard based PBB Group OUI and the ISID value associated with the I-VPLS.

When a new I-VPLS instance is configured in a PE, the IEEE 802.1ak MMRP application is automatically invoked to advertise the presence of the related group B-MAC on all active B-VPLS SAPs and SDP bindings.

When at least two I-VPLS instances with the same ISID value are present in a B-VPLS, an optimal distribution tree is built by MMRP in the related B-VPLS infrastructure as depicted in Figure 117: Flood containment requirement in M:1 model.

Figure 117: Flood containment requirement in M:1 model



5.2.12 MMRP support over B-VPLS SAPs and SDPs

MMRP is supported in B-VPLS instances over all the supported BVPLS SAPs and SDPs, including the primary and standby pseudowire scheme implemented for VPLS resiliency.

When a B-VPLS with MMRP enabled receives a packet destined for a specific group B-MAC, it checks its own MFIB entries and if the group B-MAC does not exist, it floods it everywhere. This should never happen as this kind of packet is generated at the I-VPLS/PBB PE when a registration was received for a local I-VPLS group B-MAC.

5.2.12.1 I-VPLS changes and related MMRP behavior

This section describes the MMRP behavior for different changes in IVPLS.

- When an ISID is set for a specific I-VPLS and a link to a related B-VPLS is activated (for example, through the **configure service vpls backbone-vpls vpls id:isid** command), the group B-MAC address is declared on all B-VPLS virtual ports (SAPs or SDPs).
- When the ISID is changed from one value to a new one, the old group B-MAC address is undeclared on all ports and the new group B-MAC address is declared on all ports in the B-VPLS.
- When the I-VPLS is disassociated with the B-VPLS, the old group B-MAC is no longer advertised as a local attribute in the B-VPLS if no other peer B-VPLS PEs have it declared.
- When an I-VPLS goes operationally down (either all SAPs/SDPs are down) or the I-VPLS is shutdown, the associated group B-MAC is undeclared on all ports in the B-VPLS.
- When the I-VPLS is deleted, the group B-MAC should already be undeclared on all ports in the B-VPLS because the I-VPLS has to be shutdown to delete it.

5.2.12.2 Limiting the number of MMRP entries on a per B-VPLS basis

The MMRP exchanges create one entry per attribute (group B-MAC) in the B-VPLS where MMRP protocol is running. When the first registration is received for an attribute, an MFIB entry is created for it.

The Nokia implementation allows the user to control the number of MMRP attributes (group B-MACs) created on a per B-VPLS basis. Control over the number of related MFIB entries in the B-VPLS FDB is inherited from previous releases through the use of the **configure service vpls mfib-table-size table-size** command. This ensures that no B-VPLS takes up all the resources from the total pool.

5.2.12.3 Optimization for improved convergence time

Assuming that MMRP is used in a specific B-VPLS, under failure conditions the time it takes for the B-VPLS forwarding to resume may depend on the data plane and control plane convergence plus the time it takes for MMRP exchanges to settle down the flooding trees on a per ISID basis.

To minimize the convergence time, the Nokia PBB implementation offers the selection of a mode where B-VPLS forwarding reverts for a short time to flooding so that MMRP has enough time to converge. This mode can be selected through configuration using the **configure service vpls b-vpls mrp flood-time value** command where **value** represents the amount of time in seconds that flooding is enabled.

If this behavior is selected, the forwarding plane reverts to B-VPLS flooding for a configurable time period, for example, for a few seconds, then it reverts back to the MFIB entries installed by MMRP.

The following B-VPLS events initiate the switch from per I-VPLS (MMRP) MFIB entries to "B-VPLS flooding":

- Reception or local triggering of a TCN
- B-SAP failure
- Failure of a B-SDP binding
- Pseudowire activation in a primary/standby HVPLS resiliency solution
- SF/CPM switchover because of the STP reconvergence

5.2.12.4 Controlling MRP scope using MRP policies

MMRP advertises the Group B-MACs associated with ISIDs throughout the whole BVPLS context regardless of whether a specific IVPLS is present in one or all the related PEs or BEBs. When evaluating the overall scalability the resource consumption in both the control and data plane must be considered:

- **control plane**

  The control plane is responsible for MMRP processing and the number of attributes advertised.

- **data plane**

  One tree is instantiated per ISID or Group B-MAC attribute.

In a multi-domain environment, for example multiple MANs interconnected through a WAN, the BVPLS and implicitly MMRP advertisement may span across domains. The MMRP attributes are flooded throughout the BVPLS context indiscriminately, regardless of the distribution of IVPLS sites.

The solution described in this section limits the scope of MMRP control plane advertisements to a specific network domain using MRP Policy. ISID-based filters are also provided as a safety measure for BVPLS data plane.

Figure 118: Inter-domain topology

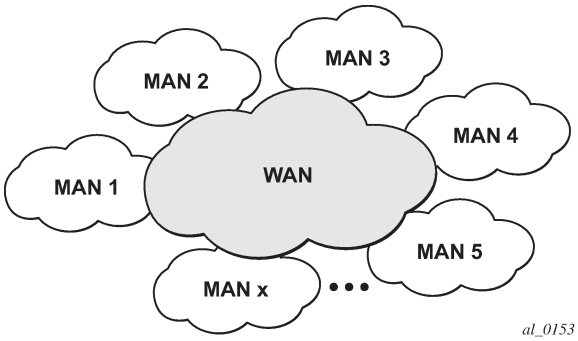
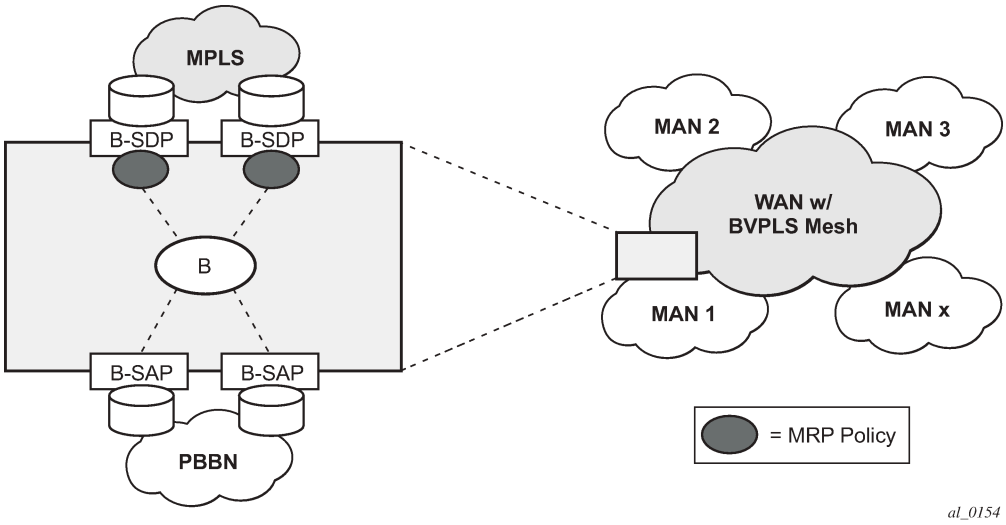


Figure 118: Inter-domain topology shows the case of an Inter-domain deployment where multiple metro domains (MANs) are interconnected through a wide area network (WAN). A BVPLS is configured across these domains running PBB M:1 model to provide infrastructure for multiple IVPLS services. MMRP is enabled in the BVPLS to build per IVPLS flooding trees. To limit the load in the core PEs or PBB BCBs, the local IVPLS instances must use MMRP and data plane resources only in the MAN regions where they have sites. A solution to the above requirements is depicted in Figure 119: Limiting the scope of MMRP advertisements. The case of native PBB metro domains inter-connected via a MPLS core is used in this example. Other technology combinations are possible.

Figure 119: Limiting the scope of MMRP advertisements



An MRP policy can be applied to the edge of MAN1 domain to restrict the MMRP advertisements for local ISIDs outside local domain. Or the MRP policy can specify the inter-domain ISIDs allowed to be advertised outside MAN1. The configuration of MRP policy is similar with the configuration of a filter. It can be specified as a template or exclusively for a specific endpoint under service mrp object. An ISID or a range of ISIDs can be used to specify one or multiple match criteria that can be used to generate the list of Group MACs to be used as filters to control which MMRP attributes can be advertised. An example of a simple mrp-policy that allows the advertisement of Group B-MACs associated with ISID range 100-150 is provided below:

```
*A:ALA-7>config>service>mrp# info
----------------------------------------------
            mrp-policy "test" create
                default-action block
                entry 1 create
                    match
                        isid 100 to 150
                    exit
        action allow
                exit
            exit
----------------------------------------------
```

A special action end-station is available under mrp-policy entry object to allow the emulation on a specific SAP/PW of an MMRP end-station. This is usually required when the operator does not want to activate MRP in the WAN domain for interoperability reasons or if it prefers to manually specify which ISID is interconnected over the WAN. In this case the MRP transmission is shutdown on that SAP/PW and the configured ISIDs are used the same way as an IVPLS connection into the BVPLS, emulating a static entry in the related BVPLS MFIB. Also if MRP is active in the BVPLS context, MMRP declares the related GB-MACs continuously over all the other BVPLS SAP/PWs until the mrp-policy end-station action is removed from the mrp-policy assigned to that BVPLS context.

The MMRP usage of the mrp-policy ensures automatically that traffic using GB-MAC is not flooded between domains. There could be though small transitory periods when traffic originated from PBB BEB with unicast B-MAC destination may be flooded in the BVPLS context as unknown unicast in the BVPLS context for both IVPLS and PBB Epipe. To restrict distribution of this traffic for local PBB services a new

ISID match criteria is added to existing mac-filters. The mac-filter configured with ISID match criteria can be applied to the same interconnect endpoints, BVPLS SAP or PW, as the mrp-policy to restrict the egress transmission any type of frames that contain a local ISID. An example of this new configuration option is as follows:

```
-----------------------------------------------
A;ALA-7>config>filter# info
-----------------------------------------------
mac-filter 90 create
description "filter-wan-man"
type isid
scope template
entry 1 create
description "drop-local-isids"
match
isid from 100 to 1000
exit
action drop
exit
-----------------------------------------------
```

These filters are applied as required on a per B-SAP or B-PW basis just in the egress direction. The ISID match criteria is exclusive with any other criteria under mac-filter. A new mac-filter type attribute is defined to control the use of ISID match criteria and must be set to isid to allow the use of isid match criteria. The ISID tag is identified using the PBB ethertype provisioned under **config**>**port**>**ethernet**>**pbb-etype**.

5.2.13 PBB and BGP-AD

BGP auto-discovery is supported only in the BVPLS to automatically instantiate the BVPLS pseudowires and SDPs.

5.2.14 PBB E-Line service

E-Line service is defined in PBB (IEEE 802.1ah) as a point-to-point service over the B-component infrastructure. The Nokia implementation offers support for PBB E-Line through the mapping of multiple Epipe services to a backbone VPLS infrastructure.

The use of Epipe scales the E-Line services as no MAC switching, learning or replication is required to deliver the point-to-point service.

All packets that ingress the customer SAP/spoke SDP are PBB encapsulated and unicasted through the B-VPLS "tunnel" using the backbone destination MAC of the remote PBB PE. The Epipe service does not support the forwarding of PBB encapsulated frames received on SAPs or spoke SDPs through their associated B-VPLS service. PBB frames are identified based on the configured PBB Ethertype (0x88e7 by default).

All the packets that ingress the B-VPLS destined for the Epipe are PBB de-encapsulated and forwarded to the customer SAP/spoke SDP.

A PBB E-Line service support the configuration of a SAP or non-redundant spoke SDP.

5.2.14.1 Non-redundant PBB Epipe spoke termination

This feature provides the capability to use non-redundant pseudowire connections on the access side of a PBB Epipe, where previously only SAPs could be configured.

5.2.15 PBB using G.8031 protected Ethernet tunnels

IEEE 802.1ah Provider Backbone Bridging (PBB) specification employs provider MSTP (PMSTP) to ensure loop avoidance in a resilient native Ethernet core. The usage of P-MSTP means failover times depend largely on the size and the connectivity model used in the network. The use of MPLS tunnels provides a way to scale the core while offering fast failover times using MPLS FRR. There are still service provider environments where Ethernet services are deployed using native Ethernet backbones. A solution based on native Ethernet backbone is required to achieve the same fast failover times as in the MPLS FRR case.

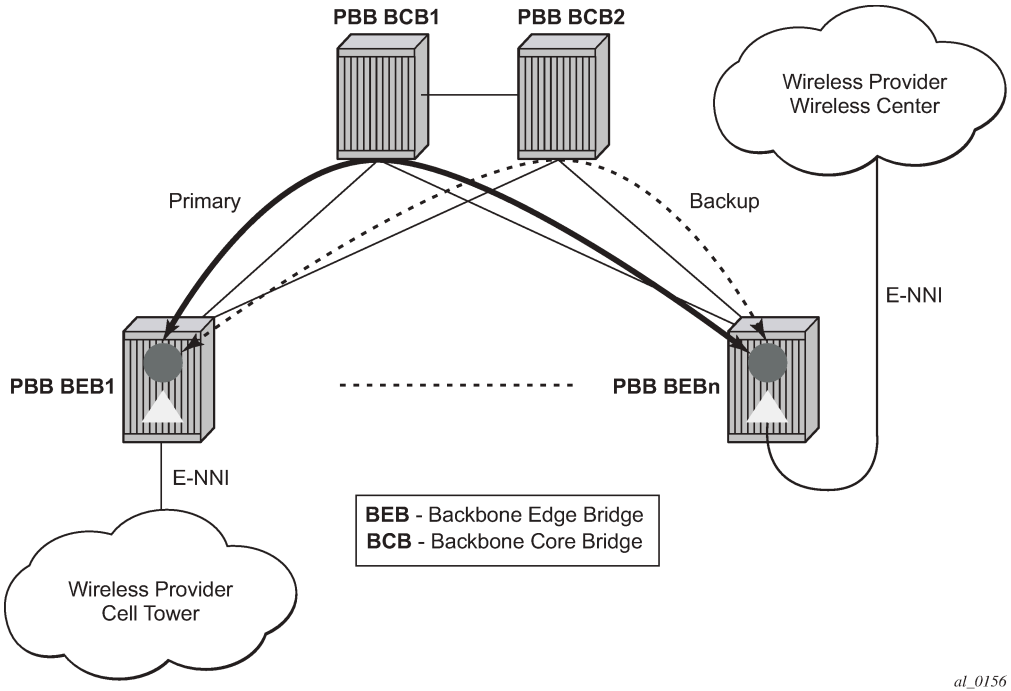
The Nokia PBB implementation offers the capability to use core Ethernet tunnels compliant with ITU-T G.8031 specification to achieve 50 ms resiliency for backbone failures. This is required to comply with the stringent SLAs provided by service providers in the current competitive environment. The implementation also allows a LAG-emulating Ethernet tunnel providing a complimentary native Ethernet E-LAN capability. The LAG-emulating Ethernet tunnels and G.8031 protected Ethernet tunnels operate independently.

The next section describes an applicability example where an Ethernet service provider using native PBB offers a carrier of carrier backhaul service for mobile operators.

5.2.15.1 Solution overview

A simplified topology example for a PBB network offering a carrier of carrier service for wireless service providers is depicted in Figure 120: Mobile backhaul use case.

Figure 120: Mobile backhaul use case



The wireless service provider in this example purchases an E-Line service between the ENNIs on PBB edge nodes, BEB1 and BEBn. PBB services are employing a type of Ethernet tunneling (Eth-tunnels) between BEBs where primary and backup member paths controlled by G.8031 1:1 protection are used to ensure faster backbone convergence. Ethernet CCMs based on IEEE 802.1ag specification may be used to monitor the liveliness for each individual member paths.

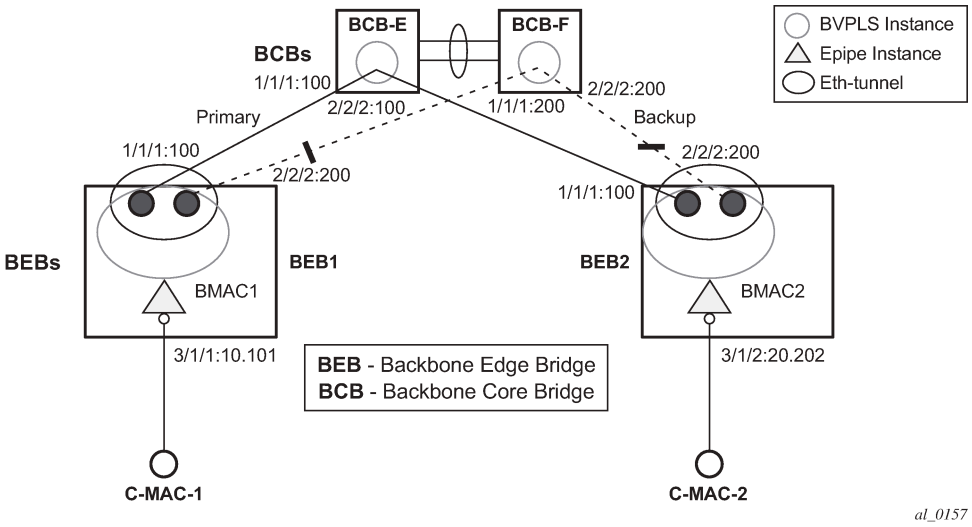
The Ethernet paths span a native Ethernet backbone where the BCBs are performing simple Ethernet switching between BEBs using an Epipe or a VPLS service.

Although the network diagram shows just the Epipe case, both PBB E-Line and E-LAN services are supported.

5.2.15.2 Detailed solution description

This section discusses the details of the Ethernet tunneling for PBB. The main solution components are depicted in Figure 121: PBB-Epipe with B-VPLS over Ethernet tunnel.

Figure 121: PBB-Epipe with B-VPLS over Ethernet tunnel



The PBB E-Line service is represented in the BEBs as a combination of an Epipe mapped to a BVPLS instance. A eth-tunnel object is used to group two possible paths defined by specifying a member port and a control tag. In our example, the blue-circle representing the eth-tunnel is associating in a protection group the two paths instantiated as (port, control-tag/bvid): a primary one of port 1/1/1, control-tag 100 and respectively a secondary one of port 2/2/2, control tag 200.

The BCBs devices stitch each BVID between different BEB-BCB links using either a VPLS or Epipe service. Epipe instances are recommended as the preferred option because of the increased tunnel scalability.

Fast failure detection on the primary and backup paths is provided using IEEE 802.1ag CCMs that can be configured to transmit at 10 msec interval. Alternatively, the link layer fault detection mechanisms like LoS/RDI or 802.3ah can be employed.

Path failover is controlled by an Ethernet protection module, based on standard G.8031 Ethernet Protection Switching. The Nokia implementation of Ethernet protection switching supports only the 1:1 model which is common practice for packet based services because it makes better use of available bandwidth. The following additional functions are provided by the protection module:

- Synchronization between BEBs such that both send and receive on the same Ethernet path in stable state.
- Revertive / non-revertive choices.
- Compliant G.8031 control plane.

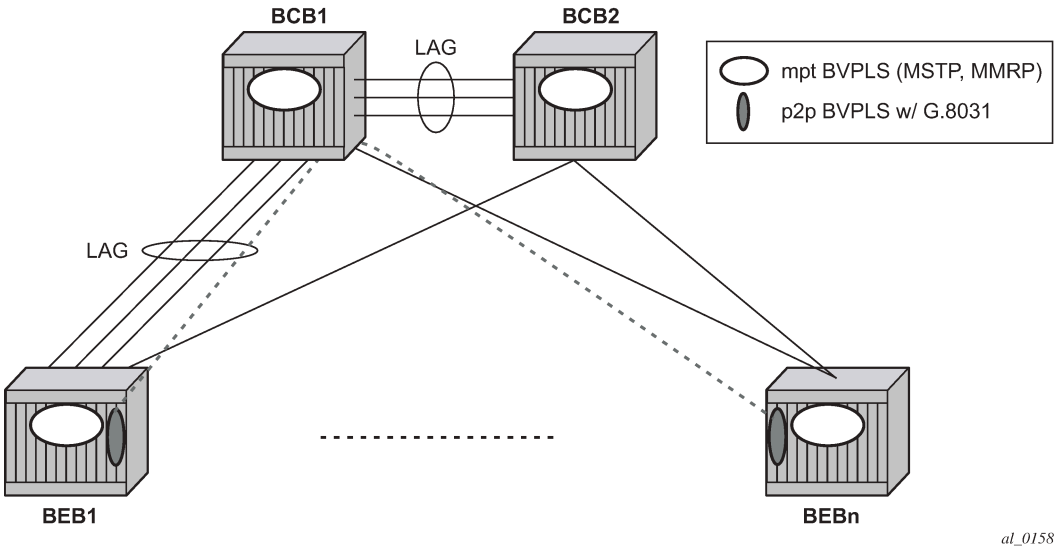
The secondary path requires a MEP to exchange the G.8031 APS PDUs. The following Ethernet CFM configuration in the **eth-tunnel>path>eth-cfm>mep** context can be used to enable the G.8031 protection without activating the Ethernet CCMs:

1. Create the domain (MD) in CFM.
2. Create the association (MA) in CFM and do not put remote MEPs.
3. Create the MEP.
4. Configure control-mep and no shutdown on the MEP.
5. Use the **no ccm-enable** command to keep the CCM transmission disabled.

If a MEP is required for troubleshooting issues on the primary path, the configuration described above for the secondary path must be used to enable the use of Link Layer OAM on the primary path.

LAG loadsharing is offered to complement G.8031 protected Ethernet tunnels for situations where unprotected VLAN services are to be offered on some or all of the same native Ethernet links.

Figure 122: G.8031 P2P tunnels and LAG-like loadsharing coexistence



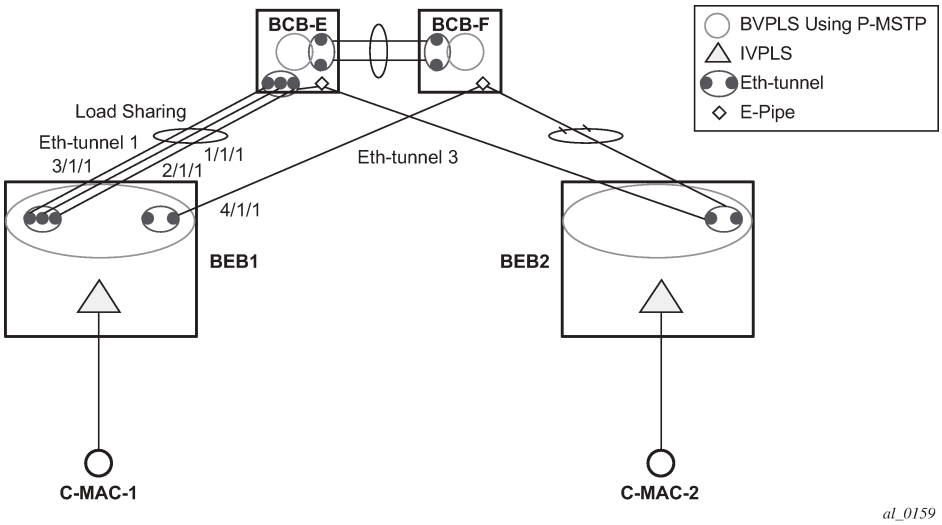
In Figure 122: G.8031 P2P tunnels and LAG-like loadsharing coexistence , the G.8031 Ethernet tunnels are used by the B-SAPs mapped to the green BVPLS entities supporting the E-Line services. A LAG-like loadsharing solution is provided for the Multipoint BVPLS (white circles) supporting the E-LAN (IVPLS) services. The green G.8031 tunnels coexist with LAG-emulating Ethernet tunnels (loadsharing mode) on both BEB-BCB and BCB-BCB physical links.

The G.8031-controlled Ethernet tunnels select an active tunnel based on G.8031 APS operation, while emulated-LAG Ethernet tunnels hash traffic within the configured links. Upon failure of one of the links the emulated-LAG tunnels rehash traffic within the remaining links and fail the tunnel when the number of links breaches the minimum required (independent of G.8031-controlled Ethernet tunnels on the links shared emulated-LAG).

5.2.15.3 Detailed PBB emulated LAG solution description

This section discusses the details of the emulated LAG Ethernet tunnels for PBB. The main solution components are depicted in Figure 123: Ethernet tunnel overlay which overlays Ethernet Tunnels services on the network from Figure 121: PBB-Epipe with B-VPLS over Ethernet tunnel.

Figure 123: Ethernet tunnel overlay



For a PBB Ethernet VLAN to make efficient use of an emulated LAG solution, a Management-VPLS (m-VPLS) is configured enabling Provider Multi-Instance Spanning Tree Protocol (P-MSTP). The m-VPLS is assigned to two SAPs; the eth-tunnels connecting BEB1 to BCB-E and BCB-F, respectively, reserving a range of VLANs for P-MSTP.

The PBB P-MSTP service is represented in the BEBs as a combination of an Epipe mapped to a BVPLS instance as before but now the PBB service is able to use the Ethernet tunnels under the P-MSTP control and load share traffic on the emulated LAN. In our example, the blue-circle representing the BVPLS is assigned to the SAPs which define two paths each. All paths are specified as primary precedence to load share the traffic.

A Management VPLS (m-VPLS) is first configured with a VLAN-range and assigned to the SAPs containing the path to the BCBs. The load shared eth-tunnel objects are defined by specifying a member ports and a control tag of zero. Then individual B-VPLS services can be assigned to the member paths of the emulated LAGs and defining the path encapsulation. Then individual services such as the IVPLS service can be assigned to the B-VPLS.

At the BCBs the tunnels are terminated the next BVPLS instance controlled by P-MSTP on the BCBs to forward the traffic.

In the event of link failure, the emulated LAG group automatically adjusts the number of paths. A threshold can be set whereby the LAG group is declared down. All emulated LAG operations are independent of any 8031-1to1 operation.

5.2.15.4 Support service and solution combinations

The following considerations apply when Ethernet tunnels are configured under a VPLS service:

- Only ports in access or hybrid mode can be configured as eth-tunnel path members. The member ports can be located on the same or different IOMs, MDAs, XCMs, or XMAs.
- Dot1q and QinQ ports are supported as eth-tunnel path members.
- The same port cannot be used as member in both a LAG and an Ethernet-tunnel.
- A mix of regular and multiple eth-tunnel SAPs and PWs can be configured in the same BVPLS.

- Split horizon groups in BVPLS are supported on eth-tunnel SAPs. The use of split horizon groups allows the emulation of a VPLS model over the native Ethernet core, eliminating the need for P-MSTP.
- STP and MMRP are not supported in a BVPLS using eth-tunnel SAPs.
- Both PBB E-Line (Epipe) and E-LAN (IVPLS) services can be transported over a BVPLS using Ethernet-tunnel SAPs.
- MC-LAG access multihoming into PBB services is supported in combination with Ethernet tunnels:
  - MC-LAG SAPs can be configured in IVPLS or Epipe instances mapped to a BVPLS that uses eth-tunnel SAPs
  - Blackhole Avoidance using native PBB MAC flush/MAC move solution is also supported
- Support is also provided for BVPLS with P-MSTP and MMRP control plane running as ships-in-the-night on the same links with the Ethernet tunneling which is mapped by a SAP to a different BVPLS.

  Epipes must be used in the BCBs to support scalable point-to-point tunneling between the eth-tunnel endpoints when management VPLS is used.
- The following solutions or features are not supported in the current implementation for the 7450 ESS and 7750 SR and are blocked:
  - Capture SAP
  - Subscriber management
  - Application assurance
  - Eth-tunnels usage as a logical port in the **config**>**redundancy**>**multi-chassis**>**peer**>**sync**>**port** context

For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Services Overview Guide*.

5.2.16 Periodic MAC notification

Virtual B-MAC learning frames (for example, the frames sent with the source MAC set to the virtual B-MAC) can be sent periodically, allowing all BCBs/BEBs to keep the virtual B-MAC in their Layer 2 forwarding database.

This periodic mechanism is useful in the following cases:

- A new BEB is added after the current mac-notification method has stopped sending learning frames.
- A new combination of [MC-LAG:SAP|A/S PW]+[PBB-Epipe]+[associated B-VPLS]+[at least one B-SDP|B-SAP] becomes active. The current mechanism only sends learning frames when the first such combination becomes active.
- A BEB containing the remote endpoint of a dual-homed PBB-Epipe is rebooted.
- Traffic is not seen for the MAC aging timeout (assuming that the new periodic sending interval is less than the aging timeout).
- There is unidirectional traffic.

In each of the above cases, all of the remote BEB/BCBs learn the virtual MAC in the worst case after the next learning frame is sent.

In addition, this allows all of the above when to be used in conjunction with discard-unknown in the B-VPLS. Currently, if discard-unknown is enabled in all related B-VPLSs (to avoid any traffic flooding), all above cases could experience an increased traffic interruption, or a permanent loss of traffic, as only traffic

toward the dual homed PBB-Epipe can restart bidirectional communication. For example, it reduces the traffic outage when:

The PBB-Epipe virtual MAC is flushed on a remote BEB/BCB because of the failover of an MC-LAG or A/S pseudowires within the customer's access network, for example, in between the dual homed PBB-Epipe peers and their remote tunnel endpoint.

There is a failure in the PBB core causing the path between the two BEBs to pass through a different BCB.

It should be noted that this does not help in the case where the remote tunnel endpoint BEB fails. In this case traffic is flooded when the remote B-MAC ages out if discard-unknown is disabled. If discard-unknown is enabled, then the traffic follows the path to the failed BEB but is eventually dropped on the source BEB when the remote B-MAC ages out on all systems.

To scale the implementation it is expected that the timescale for sending the periodic notification messages is much longer than that used for the current notification messages.

5.2.17 MAC flush

5.2.17.1 PBB resiliency for B-VPLS over pseudowire infrastructure

The following VPLS resiliency mechanisms are also supported in PBB VPLS:

- Native Ethernet resiliency supported in both I-VPLS and B-VPLS contexts
- Distributed LAG, MC-LAG, RSTP
- MSTP in a management VPLS monitoring (B- or I-) SAPs and pseudowire
- BVPLS service resiliency, loop avoidance solutions - Mesh, active/standby pseudowires and multi-chassis endpoint
- IVPLS service resiliency, loop avoidance solutions - Mesh, active/standby pseudowires (PE-rs only role), BGP multihoming

To support these resiliency options, extensive support for blackhole avoidance mechanisms is required.

5.2.17.1.1 Porting existing VPLS LDP MAC flush in PBB VPLS

Both the I-VPLS and B-VPLS components inherit the LDP MAC flush capabilities of a regular VPLS to fast age the related FDB entries for each domain: C-MACs for I-VPLS and B-MACs for B-VPLS. Both types of LDP MAC flush are supported for I-VPLS and B-VPLS domains:

- **flush-all-but-mine**

  This refers to flushing on a positive event, for example:

  – pseudowire activation (VPLS resiliency using active/standby pseudowire)

  – reception of a STP TCN

- **flush-all-from-me**

  This refers to flushing on a negative event, for example:

  – SAP failure (link down or MC-LAG out-of-sync)

  – pseudowire or endpoint failure

In addition, only for the B-VPLS domain, changing the backbone source MAC of a B-VPLS triggers an LDP MAC flush-all-from-me to be sent in the related active topology. At the receiving PBB PE, a B-MAC flush automatically triggers a flushing of the C-MACs associated with the old source B-MAC of the B-VPLS.

5.2.17.1.2 PBB blackholing issue

In the PBB VPLS solution, a B-VPLS may be used as infrastructure for one or more I-VPLS instances. B-VPLS control plane (LDP Signaling or P-MSTP) replaces I-VPLS control plane throughout the core. This is raising an additional challenge related to blackhole avoidance in the I-VPLS domain as described in this section.

To address the PBB blackholing issue, assuming that the link between PE A1 and node 5 is active, the remote PEs participating in the orange VPN (for example, PE D) learn the C-MAC X associated with backbone MAC A1. Under failure of the link between node 5 and PE A1 and activation of link to PE A2, the remote PEs (for example, PE D) blackhole the traffic destined for customer MAC X to B-MAC A1 until the aging timer expires or a packet flows from X to Y through the PE A2. This may take a long time (default aging timer is 5 minutes) and may affect a large number of flows across multiple I-VPLSs.

A similar issue occurs in the case where node 5 is connected to A1 and A2 I-VPLS using active/standby pseudowires. For example, when node 5 changes the active pseudowire, the remote PBB PE keeps sending to the old PBB PE.

Another case is when the QinQ access network dual-homed to a PBB PE uses RSTP or M-VPLS with MSTP to provide loop avoidance at the interconnection between the PBB PEs and the QinQ SWs. In the case where the access topology changes, a TCN event is generated and propagated throughout the access network. Similarly, this change needs to be propagated to the remote PBB PEs to avoid blackholing.

A solution is required to propagate the I-VPLS events through the backbone infrastructure (B-VPLS) to flush the customer MAC to B-MAC entries in the remote PBB. As there are no IVPLS control plane exchanges across the PBB backbone, extensions to B-VPLS control plane are required to propagate the I-VPLS MAC flush events across the B-VPLS.

5.2.17.1.3 LDP MAC flush solution for PBB blackholing

In the case of an MPLS core, B-VPLS uses T-LDP signaling to set up the pseudowire forwarding. The following I-VPLS events must be propagated across the core B-VPLS using LDP MAC flush-all-but-mine or flush-all-from-me indications:

For flush-all-but-mine indication ("positive flush"):

- TCN event in one or more of the I-VPLS or in the related M-VPLS for the MSTP use case.
- Pseudowire/SDP binding activation with active/standby pseudowire (standby, active or down, up)
- Reception of an LDP MAC withdraw "flush-all-but-mine" in the related I-VPLS

For flush-all-from-me indication ("negative flush"):

- MC-LAG failure does not require send-flush-on-failure to be enabled in I-VPLS.
- Failure of a local SAP requires send-flush-on-failure to be enabled in I-VPLS.
- Failure of a local pseudowires/SDP binding requires send-flush-on-failure to be enabled in I-VPLS.
- Reception of an LDP MAC withdraw flush-all-from-me in the related I-VPLS.

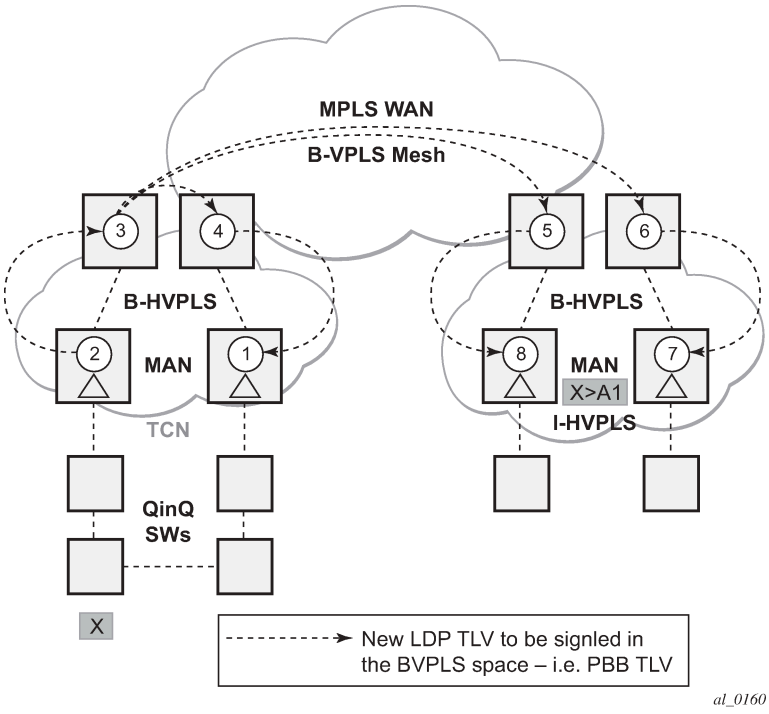
To propagate the MAC flush indications triggered by the above events, the PE that originates the LDP MAC withdraw message must be identified. In regular VPLS "mine"/"me" is represented by the pseudowire associated with the FEC and the T-LDP session on which the LDP MAC withdraw was received. In PBB, this is achieved using the B-VPLS over which the signaling was propagated and the B-MAC address of the originator PE.

Nokia PBB-VPLS solution addresses this requirement by inserting in the BVPLS LDP MAC withdraw message a new PBB-TLV (type-length-value) element. The new PBB TLV contains the source B-MAC identifying the originator ("mine"/"me") of the flush indication and the ISID list identifying the I-VPLS instances affected by the flush indication.

There are a number of advantages to this approach. Firstly, the PBB-TLV presence indicates this is a PBB MAC flush. As a result, all PEs containing only the B-VPLS instance automatically propagate the LDP MAC withdraw in the B-VPLS context respecting the split-horizon and active link topology. There is no flushing of the B-VPLS FDBs throughout the core PEs. Subsequently, the receiving PBB VPLS PEs uses the B-MAC and ISID list information to identify the specific I-VPLS FDBs and the C-MAC entries pointing to the source B-MAC included in the PBB TLV.

An example of processing steps involved in PBB MAC Flush is depicted in Figure 124: TCN triggered PBB flush-all-but-mine procedure for the case when a Topology Change Notification (TCN) is received on PBB PE 2 from a QinQ access in the I-VPLS domain.

Figure 124: TCN triggered PBB flush-all-but-mine procedure



The received TCN may be related to one or more I-VPLS domains. This generates a MAC flush in the local I-VPLS instances and if configured, it originates a PBB MAC flush-all-but-mine throughout the related B-VPLS contexts represented by the white circles 1 to 8 in our example.

A PBB-TLV is added by PE2 to the regular LDP MAC flush-all-but-mine. B-MAC2, the source B-MAC associated with B-VPLS on PE2 is carried inside the PBB TLV to indicate who "mine" is. The ISID list identifying the I-VPLS affected by the TCN is also included if the number of affected I-VPLS is 100 or less.

No ISID list is included in the PBB-TLV if more than 100 ISIDs are affected. If no ISID list is included, then the receiving PBB PE flushes all the local I-VPLS instances associated with the B-VPLS context identified by the FEC TLV in the LDP MAC withdraw message. This is done to speed up delivery and processing of the message.

Recognizing the PBB MAC flush, the B-VPLS only PEs 3, 4, 5 and 6 refrain from flushing their B-VPLS FDB tables and propagate the MAC flush message regardless of their "propagate-mac-flush" setting.

When LDP MAC withdraw reaches the terminating PBB PEs 1 and 7, the PBB-TLV information is used to flush from the I-VPLS FDBs all C-MAC entries except those associated with the originating B-MAC BM2. If specific I-VPLS ISIDs are indicated in the PBB TLV, then the PBB PEs flush only the C-MAC entries from the specified I-VPLS except those mapped to the originating B-MAC. Flush-all-but-mine indication is not propagated further in the I-VPLS context to avoid information loops.

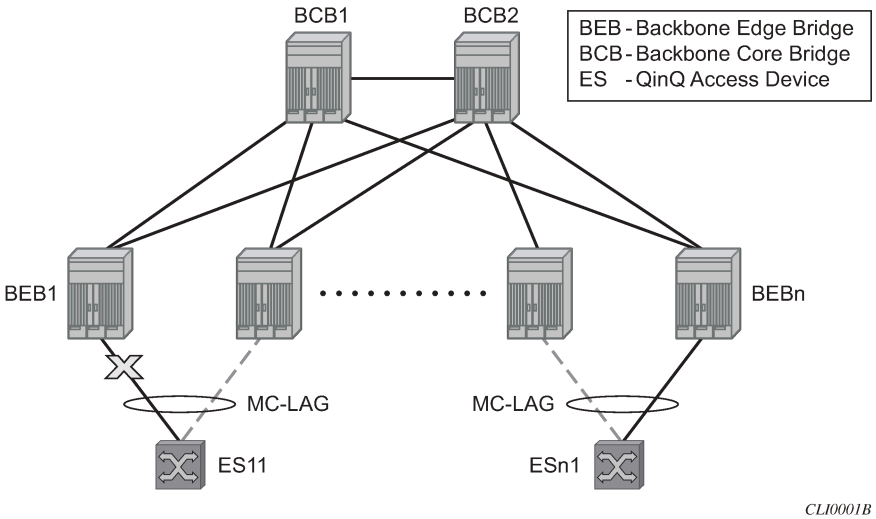
The other events that trigger Flush-all-but-mine propagation in the B-VPLS (pseudowire/SDP binding activation, Reception of an LDP MAC Withdraw) are handled similarly. The generation of PBB MAC flush-all-but-mine in the B-VPLS must be activated explicitly on a per I-VPLS basis with the **send-bvpls-flush all-but-mine** command. The generation of PBB MAC flush-all-from-me in the B-VPLS must be activated explicitly on a per I-VPLS basis with the **send-bvpls-flush all-from-me** command.

5.2.18 Access multihoming for native PBB (B-VPLS over SAP infrastructure)

Nokia PBB implementation allows the operator to use a native Ethernet infrastructure as the PBB core. Native Ethernet tunneling can be emulated using Ethernet SAPs to interconnect the related B-VPLS instances. This kind of solution may fit in specific operational environments where Ethernet services was provided in the past using QinQ solution. The drawback is that no LDP signaling is available to provide support for access multihoming for Epipe (pseudowire active/standby status) or I-VPLS services (LDP MAC Withdraw). An alternate solution is required.

A PBB network using Native Ethernet core is depicted in Figure 125: Access dual-homing into PBB BEBs - topology view. MC-LAG is used to multihome a number of edge switches running QinQ to PBB BEBs.

Figure 125: Access dual-homing into PBB BEBs - topology view



The interrupted line from the MC-LAG represents the standby, inactive link; the solid line is the active link. The BEBs are dual-homed to two core switches BCB1 and BCB2 using native Ethernet SAPs on the B-

VPLS side. Multi-point B-VPLS with MSTP for loop avoidance can be used as the PBB core tunneling. Alternatively point-to-point, G.8031 protected Ethernet tunnels can be also used to interconnect B-VPLS instances in the BEBs as described in the PBB over G.8031 protected Ethernet tunnels.

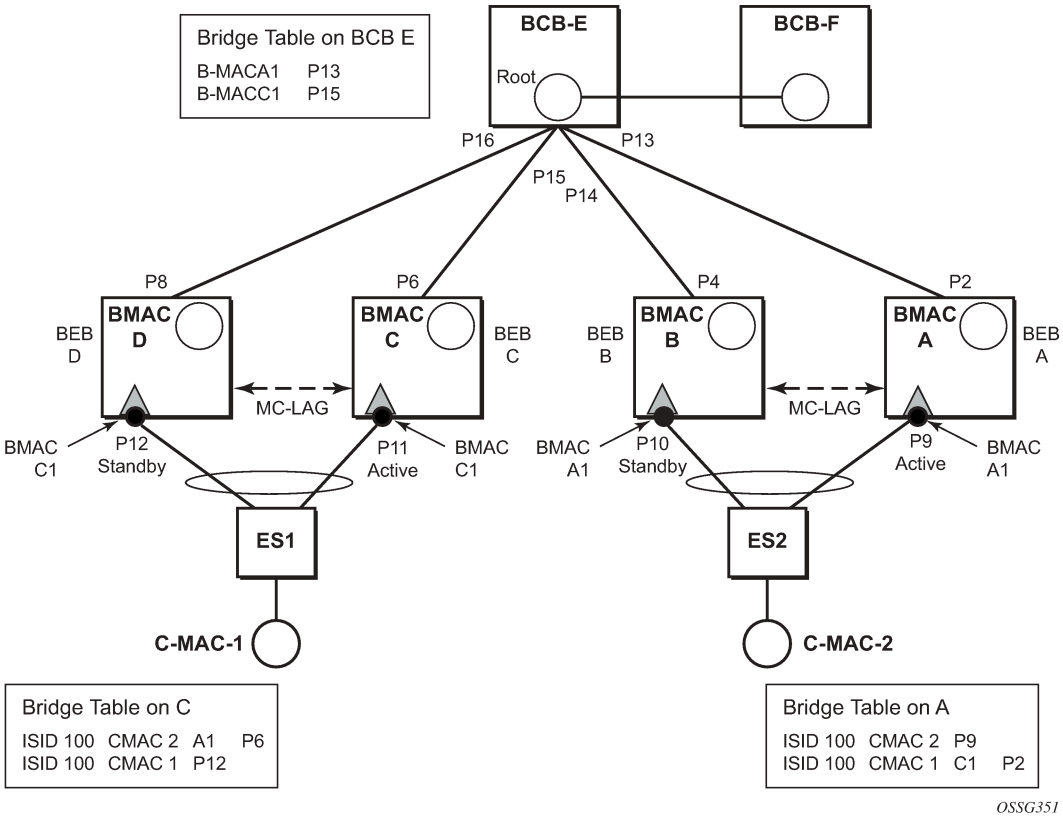
Nokia implementation provides a solution for both PBB E-Line (Epipe) and E-LAN (IVPLS) services that avoids PBB blackholing when the active ES11-BEB1 link fails. It also provides a consistent behavior for both service type and for different backbone types: for example, native Ethernet, MPLS, or a combination. Only MC-LAG is supported initially as the access-multihoming mechanism.

5.2.18.1 Solution description for I-VPLS over native PBB core

The use case described in the previous section is addressed by enhancing the existing native PBB solution to provide for blackhole avoidance.

The topology depicted in Figure 126: PBB active topology and access multihoming describes the details of the solution for the I-VPLS use case. Although the native PBB use case is used, the solution works the same for any other PBB infrastructure: for example, G.8031 Ethernet tunnels, pseudowire/MPLS, or a combination.

Figure 126: PBB active topology and access multihoming



ES1 and ES2 are dual-homed using MC-LAG into two BEB devices: ES1 to BEB C and BEB D, ES2 to BEB A and BEB B. MC-LAG P11 on BEB C and P9 on BEB A are active on each side.

In the service context, the triangles are I-VPLS instances while the small circles are B-VPLS components with the related, per BVPLS source B-MACs indicated next to each BVPLS instances. P-MSTP or RSTP

may be used for loop avoidance in the multi-point BVPLS. For simplicity, only the active SAPs (BEB P2, P4, P6 and P8) are shown in the diagram.

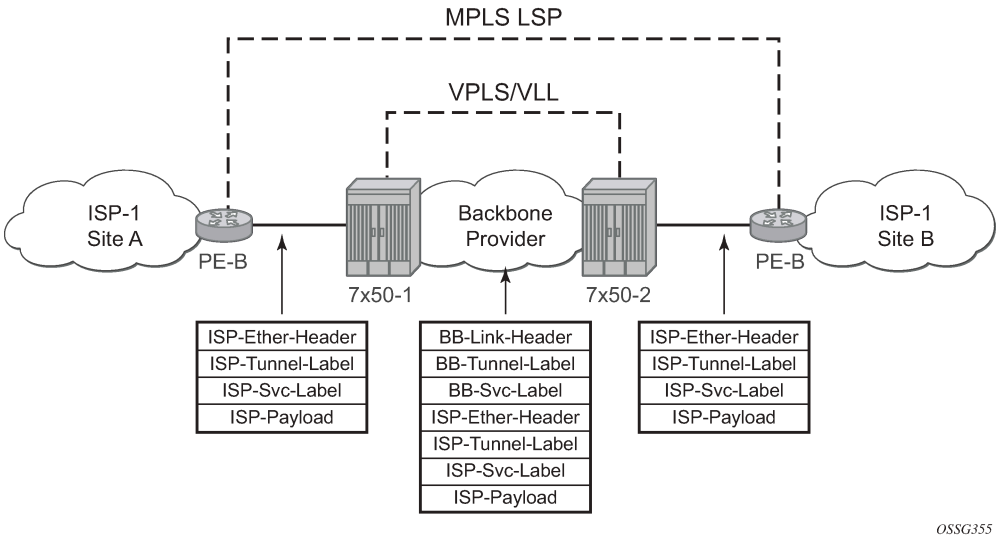
In addition to the source B-MAC associated with each BVPLS, there is an additional B-MAC associated with each MC-LAG supporting multihomed I-VPLS SAPs. The BEBs that are in a multihomed MC-LAG configuration share a common B-MAC on the related MC-LAG interfaces. For example, a common B-MAC C1 is associated in this example with ports P11 and P12 participating in the MC-LAG between BEB C and BEB D while B-MAC A1 is associated with ports P9 and P10 in the MC-LAG between BEB A and BEB B. While B-MAC C1 is associated through the I-VPLS SAPs with both BVPLS instances in BEB C and BEB D, it is actively used for forwarding to I-VPLS SAPs only on BEB C containing the active link P11.

MC-LAG protocol keeps track of which side (port or LAG) is active and which is standby for a specified MC-LAG grouping and activates the standby in case the active one fails. The source B-MAC C1 and A1 are used for PBB encapsulation as traffic arrives at the IVPLS SAPs on P11 and P9, respectively. MAC Learning in the BVPLS instances installs MAC FDB entries in BCB-E and BEB A as depicted in Figure 126: PBB active topology and access multihoming.

Active link (P11) or access node (BEB C) failures are activating through MC-LAG protocol the standby link (P12) participating in the MC-LAG on the pair MC-LAG device (BEB D).

Figure 127: Access multihoming - link failure shows the case of access link failure.

Figure 127: Access multihoming - link failure



On failure of the active link P11 on BEB C the following processing steps apply:

1. MC-LAG protocol activates the standby link P12 on the pair BEB D.
2. B-MAC C1 becomes active on BEB D and any traffic received on BEB D with destination B-MAC C1 is forwarded on the corresponding I-VPLS SAPs on P12.
3. BEB D determines the related B-VPLS instances associated with all the I-VPLS SAPs mapped to P12, the newly activated MC-LAG links/LAG components.
4. Subsequently, BEB D floods in the related B-VPLS instances an Ethernet CFM-like message using C1 as source B-MAC. A vendor CFM opcode is used followed by an Nokia OUI.
5. As a result, all the FDB entries in BCBs or BEBs along the path are automatically updated to reflect the move of B-MAC C1 to BEB D.

In this particular configuration, the entries on BEB A do not need to be updated saving MAC Flush operation.

In other topologies, it is possible that the B-MAC C1 FDB entries in the B-VPLS instance on the remote BEBs (like BEB A) need to move between B-SAPs. This involves a move of all C-MAC using as next hop B-MAC C1 and the new egress line card.

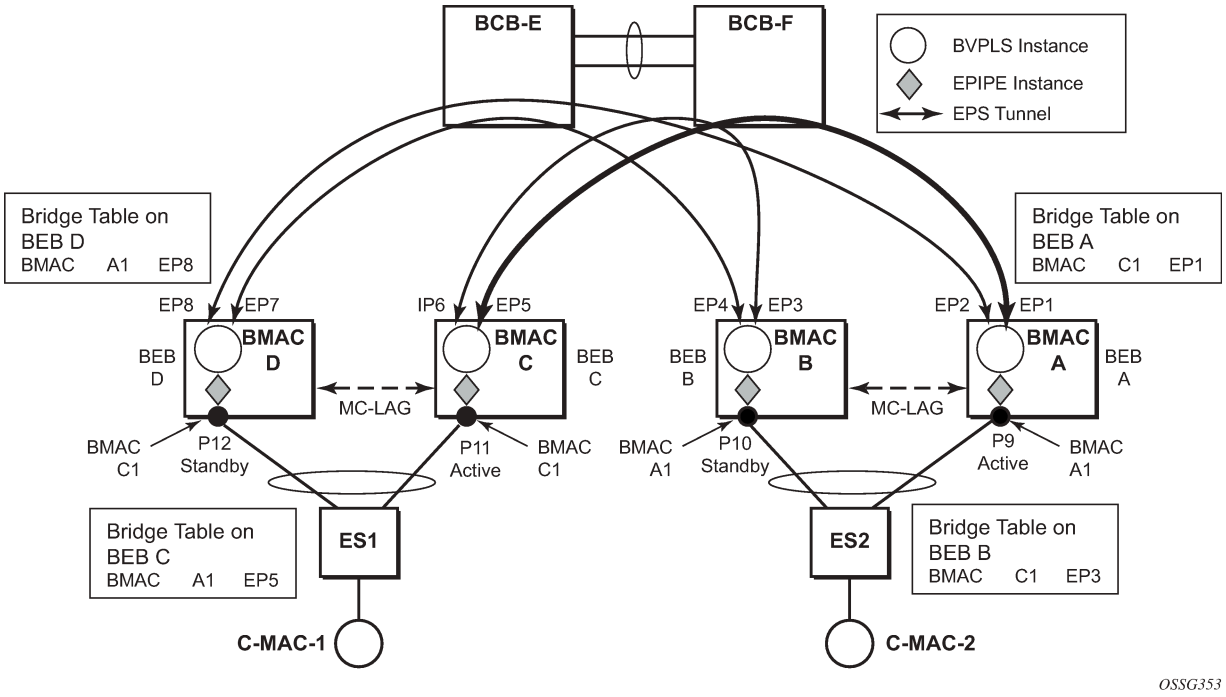
An identical procedure is used when the whole BEB C fails.

5.2.18.2 Solution description for PBB Epipe over G.8031 Ethernet tunnels

This section discusses the access multihoming solution for PBB E-Line over an infrastructure of G.8031 Ethernet tunnels. Although a specific use case is used, the solution works the same for any other PBB infrastructure: for example, native PBB, pseudowire/MPLS, or a combination.

The PBB E-Line service and the related BVPLS infrastructure are depicted in Figure 128: Access multihoming solution for PBB Epipe.

Figure 128: Access multihoming solution for PBB Epipe



The E-Line instances are connected through the B-VPLS infrastructure. Each B-VPLS is interconnected to the BEBs in the remote pair using the G.8031, Ethernet Protection Switched (EPS) tunnels. Only the active Ethernet paths are shown in the network diagram to simplify the explanation. Split Horizon Groups may be used on EPS tunnels to avoid running MSTP/RSTP in the PBB core.

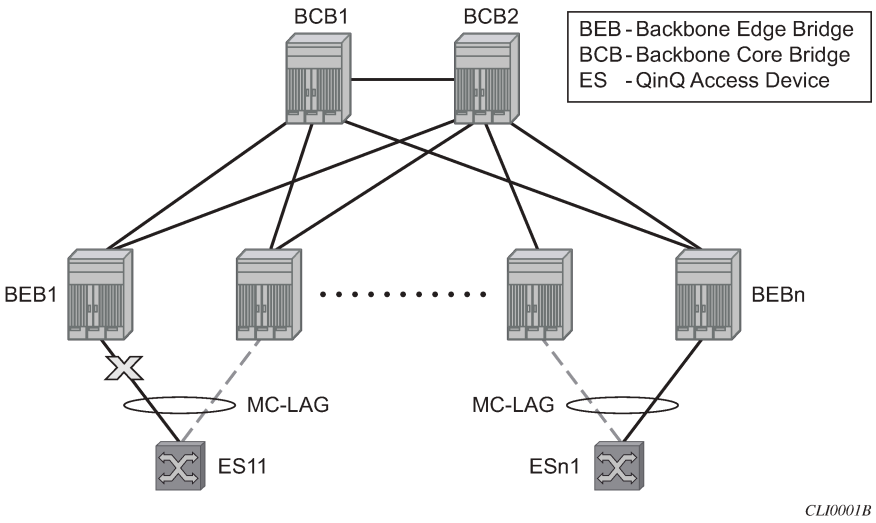
The same B-MAC addressing scheme is used as in the E-LAN case: a B-MAC per B-VPLS and additional B-MACs associated with each MC-LAG connected to an Epipe SAP. The B-MACs associated with the active MC-LAG are actively used for forwarding into B-VPLS the traffic ingressing related Epipe SAPs.

MC-LAG protocol keeps track of which side is active and which is standby for a specified MC-LAG grouping and activates the standby link in a failure scenario. The source B-MACs C1 and A1 are used

for PBB encapsulation as traffic arrives at the Epipe SAPs on P11 and P9, respectively. MAC learning in the B-VPLS instances installs MAC FDB entries in BEB C and BEB A as depicted in Figure 128: Access multihoming solution for PBB Epipe. The highlighted Ethernet tunnel (EPS) is used to forward the traffic between BEB A and BEB C.

Active link (P11) or access node (BEB C) failures are activating through MC-LAG protocol, the standby link (P12) participating in the MC-LAG on the pair MC-LAG device (BEB D). The failure of BEB C is depicted in Figure 129: Access dual-homing for PBB E-Line - BEB failure. The same procedure applies for the link failure case.

Figure 129: Access dual-homing for PBB E-Line - BEB failure



The following process steps apply:

1. BEB D loses MC-LAG communication with its peer BEB C, no more keep alives from BEB C or next-hop tracking may kick in.
2. BEB D assumes BEB C is down and activates all shared MC-LAG links, including P12.
3. B-MAC C1 becomes active on BEB D and any traffic received on BEB C with destination B-MAC C1 is forwarded on the corresponding Epipe SAPs on P12.
4. BEB D determines the related B-VPLS instances associated with all the Epipe SAPs mapped to P12, the newly activated MC-LAG links/LAG components.
5. Subsequently, BEB D floods in the related B-VPLS instances the same Ethernet CFM message using C1 as source B-MAC.
6. As a result, the FDB entries in BEB A and BEB B are automatically updated to reflect the move of B-MAC C1 from EP1 to EP2 and from EP3 to EP4, respectively.

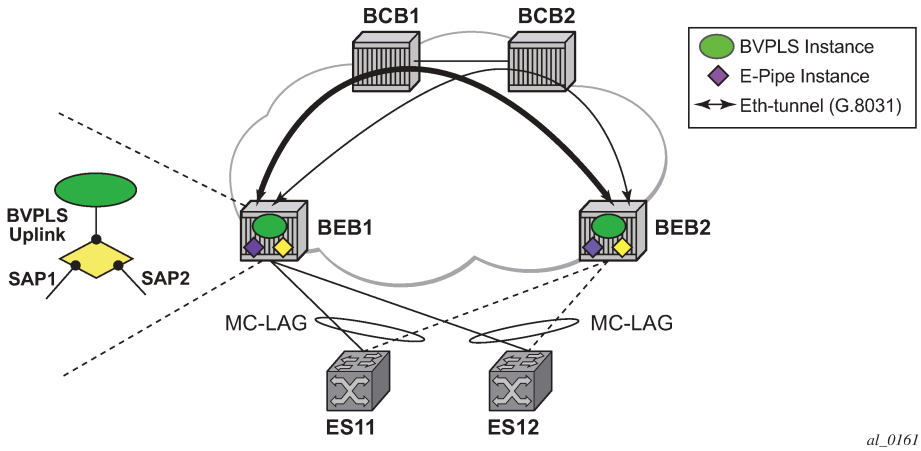
The same process is executed for all the MC-LAGs affected by BEB C failure so BEB failure can be the worst case scenario.

5.2.18.2.1 Dual-homing into PBB Epipe - local switching use case

When the service SAPs were mapped to MC-LAGs belonging to the same pair of BEBs in earlier releases, an IVPLS had to be configured even if there were just two SAPs active at any point in time. Since then, the

PBB Epipe model has been enhanced to support configuring in the same Epipe instance two SAPs and a BVPLS up link as depicted in Figure 130: Solution for access dual-homing with local switching for PBB E-Line/Epipe.

Figure 130: Solution for access dual-homing with local switching for PBB E-Line/Epipe



The PBB Epipe represented by the yellow diamond on BEB1 points through the BVPLS up link to the B-MAC associated with BEB2. The destination B-MAC can be either the address associated with the green BVPLS on BEB2 or the B-MAC of the SAP associated with the pair MC-LAG on BEB2 (preferred option).

The Epipe information model is expanded to accommodate the configuration of two SAPs (I-SAPs) and of a BVPLS up link in the same time. For this configuration to work in an Epipe environment, only two of them are active in the forwarding plane at any point in time, specifically:

- SAP1 and SAP2 when both MC-LAG links are active on the local BEB1 (see Figure 130: Solution for access dual-homing with local switching for PBB E-Line/Epipe)
- The Active SAP and the BVPLS uplink if one of the MC-LAG links is inactive on BEB1
  - PBB tunnel is considered as a backup path only when the SAP is operationally down.
  - If the SAP is administratively down, then all traffic is dropped.
- Although the CLI allows configuration of two SAPs and a BVPLS up link in the same PBB Epipe, the BVPLS up link is inactive as long as both SAPs are active.

  The traffic received through PBB tunnel is dropped if BVPLS up link is inactive. The same rules apply to BEB2.

5.2.19 BGP multihoming for I-VPLS

This section describes the application of BGP multihoming to I-VPLS services. BGP multihoming for I-VPLS uses the same mechanisms as those used when BGP multihoming is configured in a non-PBB VPLS service, which are described in detail in this guide.

The multihomed sites can be configured with either a SAP or spoke-SDP, and support both split horizon groups and fate-sharing by the use of oper-groups.

When the B-VPLS service is using LDP signaled pseudowires, blackhole protection is supported after a multihoming failover event when **send-flush-on-failure** and **send-bvpls-flush flush-all-from-me** is

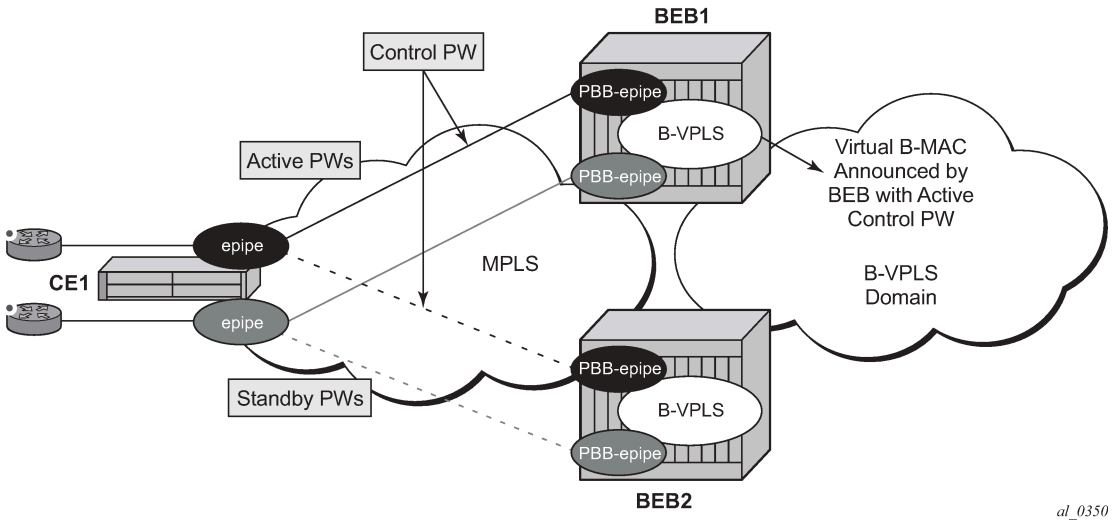
configured within the I-VPLS. This causes the system on which the site object fails to send a MAC flush all-from-me message so that customer MACs are flushed on the remote backbone edge bridges using a flush-all-from-me message. The message sent includes a PBB TLV which contains the source B-MAC identifying the originator ("mine"/"me") of the flush indication and the ISID list identifying the I-VPLS instances affected by the flush indication, see section LDP MAC flush solution for PBB blackholing.

The VPLS preference sent in BGP multihoming updates is always set to zero, however, if a non-zero value is received in a valid BGP multihoming update it is used to influence the designated forwarder (DF) election.

5.2.20 Access multihoming over MPLS for PBB Epipes

It is possible to connect backbone edge bridges (BEBs) configured with PBB Epipes to an edge device using active/standby pseudowires over an MPLS network. This is shown in Figure 131: Active/standby PW into PBB Epipes.

Figure 131: Active/standby PW into PBB Epipes



In this topology, the edge device (CE1) is configured with multiple Epipes to provide virtual lease line (VLL) connectivity across a PBB network. CE1 uses active/standby pseudowires (PWs) which terminate in PBB Epipe services on BEB1 and BEB2 and are signaled accordingly using the appropriate pseudowire status bits.

Traffic is sent from CE1 on the active pseudowires into the PBB Epipe services, then onto the remote devices through the B-VPLS service. It is important that traffic sent to CE1 is directed to the BEB that is attached to the active pseudowire connected to CE1. To achieve this, a virtual backbone MAC (vB-MAC) is associated with the services on CE1.

The vB-MAC is announced into the PBB core by the BEB connected to the active pseudowire using SPBM configured in the B-VPLS services; therefore, SPBM is mandatory. In Figure 131: Active/standby PW into PBB Epipes, the vB-MAC would be announced by BEB1; if the pseudowires failed over to BEB2, BEB1 would stop announcing the vB-MAC and BEB2 starts announcing it.

The remote services are configured to use the vB-MAC as the backbone destination MAC (backbone-dest-mac) which results in traffic being sent to the specified BEB.

The vB-MAC is configured under the SDP used to connect to the edge device's active/standby pseudowires using the command source-bmac-lsb. This command defines a sixteen (16) bit value which overrides the sixteen least-significant-bits of source backbone MAC (source-bmac) to create the vB-MAC. The operator must ensure that the vB-MACs match on the two peering BEBs for a corresponding SDP.

The PBB Epipe pseudowires are identified to be connected to an edge device active/standby pseudowire using the spoke-sdp parameter use-sdp-bmac. Enabling this parameter causes traffic forwarded from this spoke-SDP into the B-VPLS domain to use the vB-MAC as its source MAC address when both this, and the control pseudowire, are in the active state on this BEB.

PBB Epipe pseudowires connected to edge device's non-active/standby pseudowires are still able to use the same SDP.

To cater for the case where there are multiple edge device active/standby pseudowires using a specified SDP, one pseudowire must be identified to be the control pseudowire (using the source-bmac-lsb parameter control-pw-vc-id). The state of the control pseudowire determines the announcing of the vB-MAC by SPBM into the B-VPLS based on the following conditions:

- The source-bmac-lsb and control-pw-vc-id have both been configured.
- The spoke-SDP referenced by the control-pw-vc-id has use-sdp-bmac configured.
- The spoke-SDP referenced by the control-pw-vc-id is operationally up and the "Peer Pw Bits" do not include pwFwdingStandby.
- If multiple B-VPLS services are used with different SPBM Forward IDs (FIDs), the vB-MAC is advertised into any FID which has a PBB Epipe with a spoke-SDP configured with use-sdp-bmac that is using an SDP with source-bmac-lsb configured (regardless of whether the PBB Epipe spoke-SDP defined as the control pseudowire is associated with the B-VPLS).

It is expected that pseudowires configured using an SDP with source-bmac-lsb and with the parameter use-sdp-bmac are in the same state (up, down, active, standby) as the control pseudowire. If this is not the case, the following scenarios are possible (based on Figure 131: Active/standby PW into PBB Epipes):

- If any non-control pseudowires are active on BEB2 and standby on BEB1, then this continues to allow bidirectional traffic for the related services as the return traffic to CE1 is sent to BEB1, specifically to the BEB announcing the vB-MAC. As the non-control PW is in standby state it is used to send this traffic to the edge device. If this operation is not needed, it is possible to prevent traffic being sent on a standby PW using the **standby-signaling-slave** parameter under the spoke-SDP definition.
- If any non-control pseudowires are active on BEB2 but down on BEB1, then only unidirectional traffic is possible. The return traffic to CE1 is sent to BEB1, as it is announcing the vB-MAC but the pseudowire on BEB1 is down for this service.

Alarms are raised to track if, on the BEB with the control pseudowire in the standby/down state, any non-control pseudowires go active. Specifically, there is an alarm when the first non-control pseudowire becomes active and another alarm when the last non-control pseudowire becomes standby/down.

If both control pseudowires are active (neither in standby) then both BEBs would announce the vB-MAC – this would happen if the edge device was a 7450 ESS, 7750 SR, and 7950 XRS using an Epipe service without standby-signaling-master configured. Traffic from remote BEBs on any service related to the vB-MAC would be sent to the nearest SPBM BEB and it would depend on the state of the pseudowires on each BEB as to whether it could reach the edge device. Similarly, the operator must ensure that the corresponding service pseudowires on each BEB are configured as the control pseudowire, otherwise SPBM may advertise the vB-MAC from both BEBs resulting in the same consequences.

All traffic received from the edge device on a pseudowire into a PBB Epipe, on the BEB with the active control pseudowire, is forwarded by the B-VPLS using the vB-MAC as the source backbone MAC, otherwise the source-bmac is used.

The control pseudowire can be changed dynamically without shutting down the spoke-SDPs, SDP or withdrawing the SPBM advertisement of the vB-MAC; this allows a graceful change of the control pseudowire. Clearly, any change should be performed on both BEBs as closely in time as possible to avoid an asymmetric configuration, ensuring that the new control pseudowire is in the same state as the current control pseudowire on both BEBs during the change.

The following are not supported:

- active/standby pseudowires within the PBB Epipe. Consequently, the following are not supported:
  - configuration of endpoints
  - configuration of precedence under the spoke-SDP
- PW switching
- BGP-MH support, namely configuring the pseudowires to be part of a multihomed site
- network-domains
- support for the following tunneling technologies:
  - RFC 8277
  - GRE
  - L2TPv3

5.2.21 PBB and IGMP/MLD snooping

The IGMP/MLD snooping feature provided for VPLS is supported similarly in the PBB I-VPLS context, to provide efficient multicast replication in the customer domain. The difference from regular VPLS is the handling of IGMP/MLD messages arriving from the B-VPLS side over a B-VPLS SAP or SDP.

The first IGMP/MLD join message received over the local B-VPLS adds all the B-VPLS SAP and SDP components into the related multicast table associated with the I-VPLS context. This is in line with the PBB model, where the B-VPLS infrastructure emulates a backbone LAN to which every I-VPLS is connected by one virtual link.

When the querier is connected to a remote I-VPLS instance, over the B-VPLS infrastructure, its location is identified by the B-VPLS SDP and SAP on which the query was received. It is also identified by the source B-MAC address used in the PBB header for the query message. This is the B-MAC associated with the B-VPLS instance on the remote PBB PE.

It is also possible to configure that a multicast router exists in a remote I-VPLS service. This can be achieved using the **mrouter-dest** command to specify the MAC name of the destination B-MAC to be used to reach the remote I-VPLS service. This command is available in the VPLS service PBB IGMP and MLD snooping contexts.

The following are not supported in a PBB I-VPLS context with IGMP snooping or MLD snooping:

- multicast VPLS Registration (MVR)
- multicast CAC
- configuration under a default SAP

The following are not supported in a PBB I-VPLS context with MLD snooping:

- configuration of the maximum number of multicast group sources allowed per group
- configuration of the maximum number of multicast sources allowed per group

5.2.22 PBB and PIM snooping

The PIM snooping feature for IPv4 is supported in the PBB I-VPLS context to provide efficient multicast replication in the customer domain. This is similar to PIM snooping for IPv4 in a regular VPLS with the difference being the handling of PIM messages arriving from the B-VPLS side over a B-VPLS SAP or SDP.

The first PIM join message received over the local B-VPLS adds all the B-VPLS SAP and SDP components into the related multicast table associated with the I-VPLS context, and the multicast for the join is flooded throughout the B-VPLS. This is in line with the PBB model, where the B-VPLS infrastructure emulates a backbone LAN to which every I-VPLS is connected by one virtual link.

When a neighbor is located on a remote I-VPLS instance over the B-VPLS infrastructure, its location is identified by the B-VPLS SDP and SAP on which the hello message was received. The neighbor is also identified by the source B-MAC address used in the PBB header of the hello message. This is the B-MAC associated with the B-VPLS instance on the remote PBB PE.

PIM snooping for IPv4 in an I-VPLS is not supported with the following forms of default SAP:

- :*
- *.null
- *.*

5.2.23 PBB QoS

For PBB encapsulation, the configuration used for DE and dot1p in SAP and SDP policies applies to the related bits in both backbone dot1q (BTAG) and ITAG fields.

The following QoS processing rules apply for PBB B-VPLS SAPs and SDPs:

B-VPLS SAP ingress

- If dot1p, DE based classification is enabled, the BTAG fields are used by default to evaluate the internal forwarding class (fc) and discard profile if there is a BTAG field. The 802.1ah ITAG is used only if the BTAG is absent (null SAP).
- If either one of the dot1p or DE based classification is not explicitly enabled or the packets are untagged then the default fc and profile is assigned.

B-VPLS SAP egress

- If the sap-egress policy for the SAP contains an fc to dot1p/de mapping, this entry is used to set the dot1p and DE bits from the BTAG of the frame going out from the SAP. The same applies for the ITAG on frames originated locally from an I-VPLS. The mapping does not have any effect on the ITAG of frames transiting the B-VPLS.
- If no explicit mapping exists, the related dot1p DE bits are set to zero on both ITAG and BTAG if the frame is originated locally from an I-VPLS. If the frame is transiting the B-VPLS the ITAG stays unchanged, the BTAG is set according to the type of ingress SAP.
  - If the ingress SAP is tagged, the values of the dot1p, DE bits are preserved in the BTAG going out on the egress SAP.
  - If the ingress SAP is untagged, the dot1p, DE bits are set to zero in the BTAG going out on the egress SAP.

B-VPLS SDP (network) ingress policy

QoS policies for dot1p and DE bits apply only for the outer VLAN ID: this is the VLAN ID associated with the link layer and not the PBB BTAG. As a result, the dot1p DE bits are checked if an outer VLAN ID exists in the packets that ingress the SDP. If that VLAN ID is absent, nothing above the pseudowire SL is checked - for example, no dot1p bits in the BTAG or ITAG are checked. It is expected that the EXP bits are used to transport QoS information across the MPLS backbone and into the PEs.

B-VPLS SDP (network) egress policy

- When building PBB packets originating from a local I-VPLS, the BTAG and ITAG values (dot1p, DE bits) are set according to the network egress policy. The same applies for newly added BTAG (VLAN mode pseudowires) in a packet transiting the B-VPLS (SAP/SDP to SDP). If either dot1p or DE based classification is not explicitly enabled in the CLI, the values from the default fc to dot1p, DE mapping are assumed.
- Dot1p, DE bits for existing BTAGs remain unchanged - for example, applicable to packets transiting the B-VPLS and going out on SDP.

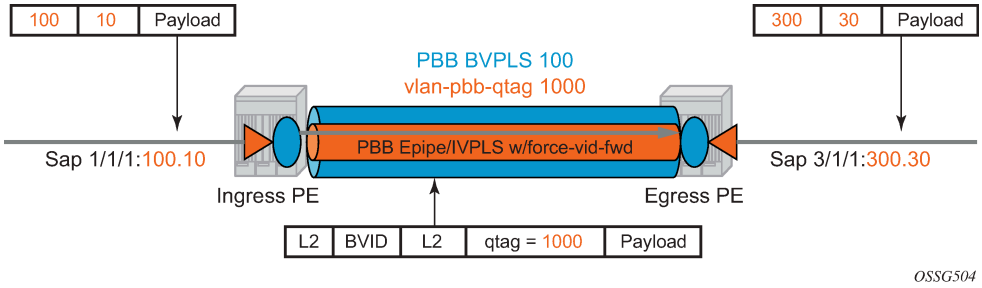
5.2.23.1 Transparency of customer QoS indication through PBB backbone

Similar to PW transport, operators want to allow their customers to preserve all eight Ethernet CoS markings (three dot1p bits) and the discard eligibility indication (DE bit) while transiting through a PBB backbone.

This means any customer CoS marking on the packets inbound to the ingress SAP must be preserved when going out on the egress SAP at the remote PBB PE even if the customer VLAN tag is used for SAP identification at the ingress.

A solution to the above requirements is depicted in Figure 132: PCP, DE bits transparency in PBB .

Figure 132: PCP, DE bits transparency in PBB



The PBB BVPLS is represented by the blue pipe in the middle with its associated CoS represented through both the service (I-tag) and tunnel CoS (BVID dot1p+DE or PW EXP bits).

The customer CoS is contained in the orange dot1q VLAN tags managed in the customer domains. There may be one (CVID) or two (CVID, SVID) tags used to provide service classification at the SAP. IVPLS or PBB Epipe instances (orange triangles) are used to provide a Carrier-of-Carrier service.

As the VLAN tags are stripped at the ingress SAP and added back at the egress SAP, the PBB implementation must provide a way to maintain the customer QoS marking. This is done using a force-qtag-forwarding configuration on a per IVPLS/Epipe basis under the node specifying the up link to the related BVPLS. When force-qtag-forwarding is enabled, a new VLAN tag is added right after the C-MAC

addresses using the configured qtag. The dot1p, DE bits from the specified outer/inner customer qtag are copied in the newly added tag.

If the **force-qtag-forwarding** is enabled in one IVPLS/PBB Epipe instance, it is enabled in all of the related instances.

At the remote PBB PE/BEB on the egress SAPs or SDPs, the first qtag after the C-MAC addresses is removed and its dot1p, DE bits are copied in the newly added customer qtags.

5.2.23.1.1 Configuration examples

This section gives usage examples for the new commands under PBB Epipe or IVPLS instances.

PBB IVPLS usage, example:

```
configure service vpls 100 ivpls
  sap 1/1/1:101
  pbb
      backbone-vpls 10 isid 100
      force-qtag-forwarding
```

PBB Epipe Usage, example:

```
configure service epipe 200
  sap 1/1/1:201
  pbb
      tunnel 10 backbone-dest-mac ab-bc-cd-ef-01-01
        isid 200
      force-qtag-forwarding
```

5.2.23.1.2 Details solution description

Figure 132: PCP, DE bits transparency in PBB shows a specific use case. Keeping the same topology, an ingress PBB PE, a PBB core and an egress PBB PE, consider the generic use case where:

1. the packet arrives on the ingress PBB PE on an I-SAP or an I-SDP binding/PW and it is assigned to a PBB service instance (Epipe/IVPLS)
2. goes next through a PBB core (native Ethernet B-SAPs or PW/MPLS based B-SDP)
3. and finally, egresses at another PBB PE through a PBB service instance on either an I-SAP or I-SDP binding/PW.

Similar to the Ethernet-VLAN VC Type, the following packet processing steps apply for different scenarios:

- Ingress PE, ingress I-SAP case with force-qtag-forwarding enabled under PBB Epipe or IVPLS

  The qtag is inserted automatically right after C-MAC addresses; an ethertype value of 8100 is used.

  – **case 1**

    SAP type = null/dot1q default (1/1/1 or 1/1/1.*) so there is no service delimiting tag used and stripped on the ingress side.

    VLAN and Dot1p+DE bits on the inserted qtag are set to zero regardless of ingress QoS policy.

  – **case 2**

SAP type = dot1q or qinq default (1/1/1.100 or 1/1/1.100.*) so there is a service delimiting tag used and stripped.

The service delimiting qtag (dot1p + DE bits and VLAN) is copied as is in the inserted qtag.

– **case 3**

SAP type = qinq (1/1/1.100.10) so there are two service delimiting tags used and stripped.

The service delimiting qtag (VLAN and dot1p + DE bits) is copied as is from the inner tag in the inserted qtag.

- Ingress PE, ingress I-SDP/PW case with force-qtag-forwarding enabled under PBB Epipe or IVPLS

  The qtag is inserted automatically right after C-MAC addresses; an ethertype value of 8100 is used.

  – **case 1**

  SDP vc-type = Ethernet (force-vlan-vc-forwarding= not supported for I-PW) so there is no service delimiting tag stripped on the ingress side.

  VLAN and Dot1p+DE bits on the inserted qtag are set to zero regardless of ingress QoS policy.

  – **case 2**

  SDP vc-type = Ethernet VLAN so there is a service delimiting tag stripped.

  VLAN and Dot1p + DE bits on the inserted qtag are preserved from the service delimiting tag.

PBB packets are tunneled through the core the same way for native ETH/MPLS cases.

- Egress PE, egress I-SAP case with force-qtag-forwarding enabled under PBB Epipe or VPLS
  – The egress QoS policy (FC->dot1p+DE bits) is used to determine the QoS settings of the added qtags. If it required to preserve the ingress QoS, no egress policy should be added.

  If QinQ SAP is used, at least qinq-mark-top-only option must be enabled to preserve the CTAG.

  – The "core qtag" (core = received over the PBB core, 1st after C-MAC addresses) is always removed after QoS information is extracted.

  If no force-qtag-forwarding is used at egress PE, the inserted qtag is maintained.

  – If egress SAP is on the ingress PE, then the dot1p+DE value is read directly from the procedures described in Ingress PE, ingress I-SAP and Ingress PE, ingress I-SDP/PW cases. The use cases below still apply:

    - **case 1**

    SAP type = null/dot1q default (2/2/2 or 2/2/2.*) so there is no service delimiting tag added on the egress side.

    Dot1p+DE bits and the VLAN value contained in the qtag are ignored.

    - **case 2**

    SAP type = dot1q/qinq default (3/1/1.300 or 3/1/1.300.*) so a service delimiting tag is added on egress.

    The FC->dot1p, DE bit entries in the SAP egress QoS policy are applied.

    If there are no such entries, then the values of the dot1p+DE bits from the stripped qtag are used.

    - **case 3**

    SAP type = qinq (3//1/1.300.30) so two service delimiting tags are added on egress.

    The FC->dot1p, DE bit entries in the SAP egress QoS policy are applied.

If the **qinq-mark-top-only** command under **vpls**>**sap**>**egress** is not enabled (default), the policy is applied to both service delimiting tags.

If the qinq-mark-top-only command is enabled, the policy is applied only to the outer service delimiting tag.

On the tags where the egress QoS policies do not apply the values of the dot1p+DE bits from the stripped qtag are used.

- Egress PE, egress I-SDP case with force-qtag-forwarding enabled under PBB Epipe or IVPLS
  - **case 1**

    I-SDP vc-type = Ethernet VLAN so there is service delimiting tag added after PW encapsulation.

    The dot1p+DE bits from the qtag received over the PBB core side are copied to the qtag added on the I-SDP.

    The VLAN value in the qtag may change to match the provisioned value for the I-SDP configuration.
  - **case 2**

    I-SDP vc-type = Ethernet (force-vlan-vc-forwarding=not supported for I-SDPs) so there is no service delimiting tag added on egress PW.

    The qtag received over the PBB core is stripped and the QoS information is lost.

5.2.24 Egress B-SAP per ISID shaping

This feature allows users to perform egress data path shaping of packets forwarded within a B-VPLS SAP. The shaping is performed within a more granular context within the SAP. The context for a B-SAP is an ISID.

5.2.24.1 B-SAP egress ISID shaping configuration

Users can enable the per-ISID shaping on the egress context of a B-VPLS SAP by configuring an encapsulation group, referred to as **encap-group** in CLI, under the QoS sub-context, referred to as **encap-defined-qos**.

config>**service**>**vpls**>**sap**>**egress**>**encap-defined-qos**>**encap-group group-name [type group-type] [qos-per-member] [create]**

The group name is unique across all member types. The **isid** type is currently the only option.

The user adds or removes members to the **encap-group**, one at a time or as a range of contiguous values. However, when the **qos-per-member** option is enabled, members must be added or removed one at a time. These members are also referred to as ISID contexts.

config>**service**>**vpls**>**sap**>**egress**>**encap-defined-qos**>**encap-group**

[no] member encap-id [to encap-id]

The user can configure one or more encap-groups in the egress context of the same B-SAP, defining different ISID values and applying each a different SAP egress QoS policy, and optionally a different scheduler policy/agg-rate-limit. ISID values are unique within the context of a B-SAP. The same ISID value cannot be re-used in another encap-group under the same B-SAP but can be re-used in an encap-group under a different B-SAP. Finally, if the user adds to an encap-group an ISID value which is already a

member of this encap-group, the command causes no effect. The same if the user attempts to remove an ISID value which is not a member of this encap-group.

When a group is created, the user assigns a SAP egress QoS policy, and optionally a scheduler policy or aggregate rate limit, using the following commands:

configure service vpls sap egress encap-defined-qos encap-group qos sap-egress-policy-id

configure service vpls sap egress encap-defined-qos encap-group scheduler-policy scheduler-policy-name

configure service vpls sap egress encap-defined-qos encap-group agg-rate-limit kilobits-per-second

A SAP egress QoS policy must first be assigned to the created encap-group before the user can add members to this group. Conversely, the user cannot perform the **no qos** command until all members are deleted from the **encap-group**.

An explicit or the default SAP egress QoS policy continues to be applied to the entire B-SAP but this serves to create the set of egress queues which is used to store and forward a packet which does not match any of the defined ISID values in any of the encap-groups for this SAP.

Only the queue definition and fc-to-queue mapping from the encap-group SAP egress QoS policy is applied to the ISID members. All other parameters configurable in a SAP egress QoS policy must be inherited from egress QoS policy applied to the B-SAP.

Furthermore, any other CLI option configured in the egress context of the B-SAP continues to apply to packets matching a member of any encap-group defined in this B-SAP.

Note also that the SAP egress QoS policy must not contain an active policer or an active queue-group queue or the application of the policy to the encap-group fails. A policer or a queue-group queue is referred to as active if one or more FC map to it in the QoS policy or the policer is referenced within the action statement of an IP or IPv6 criteria statement. Conversely, the user is not allowed to assign a FC to a policer or a queue-group queue, or reference a policer within the action statement of an IP or IPv6 criteria statement, after the QoS policy is applied to an encap-group.

The **qos-per-member** keyword allows the user to specify that a separate queue set instance and scheduler/agg-rate-limit instance are created for each ISID value in the encap-group. By default, shared instances are created for the entire encap-group.

When the B-SAP is configured on a LAG port, the ISID queue instances defined by all the encap-groups applied to the egress context of the SAP are replicated on each member link of the LAG. The set of scheduler/agg-rate-limit instances are replicated per link or per IOM or XMA depending if the adapt-qos option is set to link/port-fair mode or distribute mode. This is the same behavior as that applied to the entire B-SAP in the current implementation.

5.2.24.2 Provisioning model

The main objective of this proposed provisioning model is to separate the definition of the QoS attributes from the definition of the membership of an encap-group. The user can apply the same SAP egress QoS policy to a large number of ISID members without having to configure the QoS attributes for each member.

The following are conditions of the provisioning model:

- A SAP egress policy ID must be assigned to an **encap-group** before any member can be added regardless of the setting of the **qos-per-member** option.
- When **qos-per-member** is specified in the **encap-group** creation, the user must add or remove ISID members one at a time. The command is failed if a range is entered.

- When **qos-per-member** is specified in the **encap-group** creation, the sap-egress QoS policy ID and the scheduler policy name cannot be changed unless the group membership is empty. However, the **agg-rate-limit** parameter value can be changed or the command removed (**no agg-rate-limit**).
- When **qos-per-member** is not specified in the **encap-group** creation, the user may add or remove ISID members as a singleton or as a range of contiguous values.
- When **qos-per-member** is not specified in the **encap-group** creation, the sap-egress QoS policy ID and the scheduler policy name or **agg-rate-limit** parameter value may be changed at any time. Note however that the user cannot still remove the SAP egress QoS policy (**no qos**) while there are members defined in the encap-group.
- The QoS policy or the scheduler policy itself may be edited and modified while members are associated with the policy.
- There is a maximum number of ISID members allowed in the lifetime of an encap-group.

Operationally, the provisioning consists of the following steps:

1. Create an encap-group.
2. Define and assign a SAP egress QoS policy to the encap-group. This step is mandatory; if it is not performed, the user is allowed to add members to the **encap-group**.
3. Manage memberships for the encap-group using the **member** command (or SNMP equivalent).

   Note: The **member** command supports both range and singleton ISIDs.

   The following restrictions apply to the **member** command:

   - An ISID cannot be added if it already exists on the SAP in another encap-group. If the **member** fails for this reason, the following applies:
     - The **member** command is all-or-nothing. No ISID in a range is added if one fails.
     - The first ISID that fails is the only one identified in the error message.
     - An ISID that already exists on the SAP in another encap-group must be removed from its encap-group using the **no member** command before it can be added to a new one.
   - Specifying an ISID in a group that already exists within the group is a no-op (no failure)
   - If insufficient queues or scheduler policies or FC-to-Queue lookup table space exists to support a new member or a modified membership range, the command is fails.
4. Optionally, define and assign a scheduling policy or agg-rate-limit for the encap-group.

Logically, the encap-group membership operation can be viewed as three distinct functions:

- Creation or deletion of new queue sets and optionally scheduler/agg-rate-limit at QoS policy association time.
- Mapping or un-mapping the member ISID to either the group queue set and scheduler (group QoS) or the ISID specific queue set and scheduler (**qos-per-member**).
- Modifying the groups objective membership based on newly created or expanded ranges or singletons based on the membership operation.

5.2.24.3 Egress queue scheduling

Figure 133: Egress queue scheduling

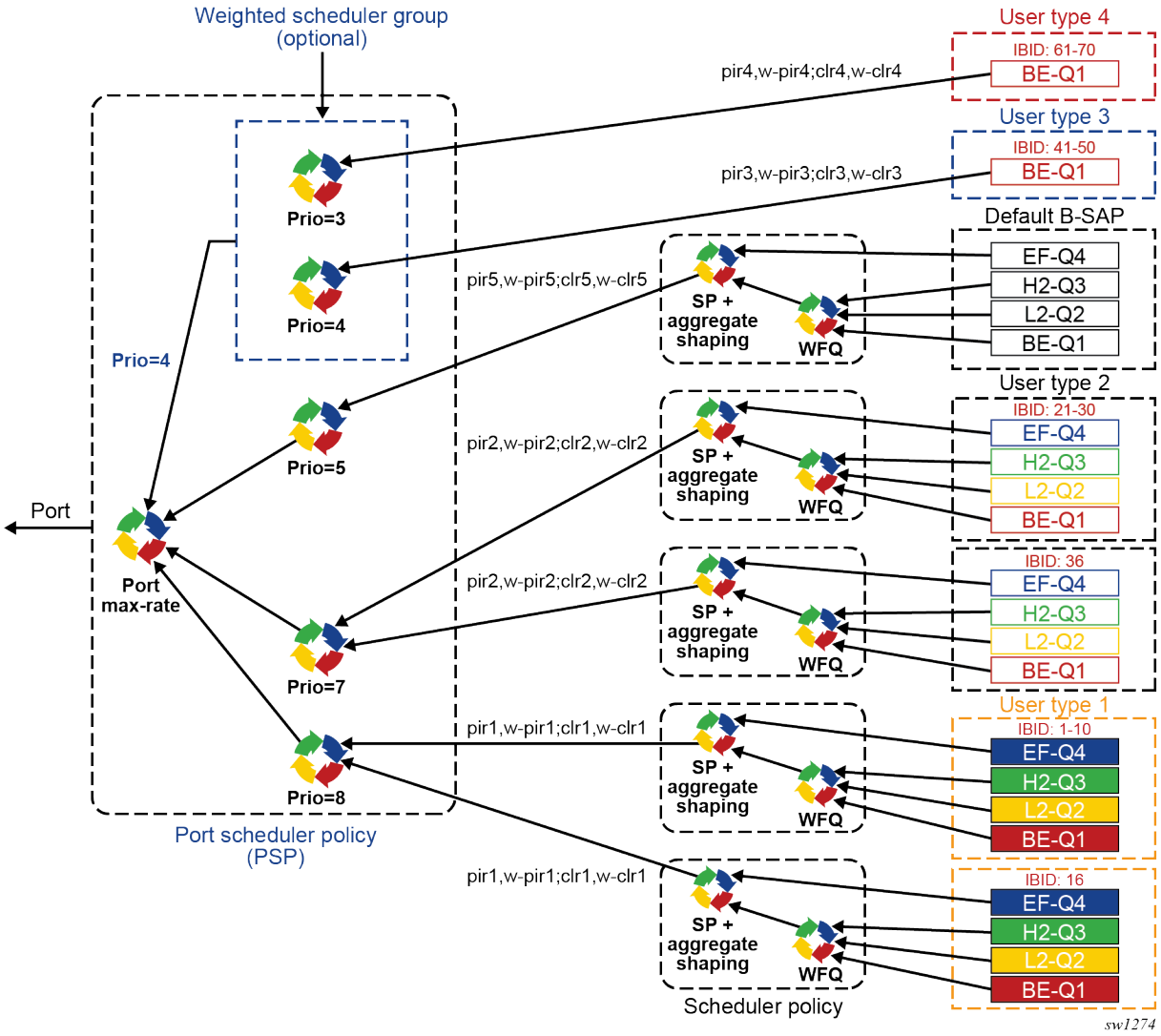


Figure 133: Egress queue scheduling displays an example of egress queue scheduling.

The queuing and scheduling re-uses existing scheduler policies and port scheduler policy with the difference that a separate set of FC queues are created for each defined ISID context according to the encap-group configured under the egress context of the B-SAP. This is in addition to the set of queues defined in the SAP egress QoS policy applied to the egress of the entire SAP.

The user type in Figure 133: Egress queue scheduling maps to a specific encap-group defined for the B-SAP in CLI. The operator has the flexibility of scheduling many user types by assigning different scheduling parameters as follows:

- A specific scheduler policy to each encap-group with a root scheduler which shapes the aggregate rate of all queues in the ISID context of the encap-group and provides strict priority scheduling to its children.

A second tier scheduler can be used as a WFQ scheduler to aggregate a subset of the ISID context FC queues. Alternatively, the operator can apply an aggregate rate limit to the ISID context instead of a scheduler policy.

- A specific priority level when parenting the ISID queues or the root of the scheduler policy serving the ISID queues to the port scheduler.
- Ability to use the weighted scheduler group to further distribute the bandwidth to the queues or root schedulers within the same priority level according to configured weights.

To make the shaping of the ISID context reflect the SLA associated with each user type, it is required to subtract the operator's PBB overhead from the Ethernet frame size. For that purpose, a **packet byte-offset** parameter is added to the context of a queue.

config>qos>sap-egress>queue>packet-byte-offset {add bytes | subtract bytes}

When a packet-byte-offset value is applied to a queue instance, it adjusts the immediate packet size. This means that the queue rates, like the operational PIR and CIR, and queue bucket updates use the adjusted packet size. In addition, the queue statistics also reflect the adjusted packet size. Scheduler policy rates, which are data rates, use the adjusted packet size.

The port scheduler **max-rate** and **priority level** rates and weights, if a Weighted Scheduler Group is used, are always "on-the-wire" rates and therefore use the actual frame size. The same applies to the agg-rate-limit on a SAP, a subscriber, or a Multi-Service Site (MSS) when the queue is port-parented.

When the user enables **frame-based-accounting** in a scheduler policy or **queue-frame-based-accounting** with agg-rate-limit in a port scheduler policy, the queue rate is capped to a user- configured "on-the-wire" rate and the **packet-byte-offset** is not included; however, the packet-byte-offset is applied to the statistics.

5.2.24.4 B-SAP per-ISID shaping configuration example

The following CLI configuration for B-SAP per-ISID shaping achieves the specific use case shown in Figure 133: Egress queue scheduling.

```
config
 qos
       port-scheduler-policy "bvpls-backbone-port-scheduler"
       group scheduler-group1 create
       rate 1000
       level 3 rate 1000 group scheduler-group1 weight w1
       level 4 rate 1000 group scheduler-group1 weight w4
       level 5 rate 1000 cir-rate 100
       level 7 rate 5000 cir-rate 5000
       level 8 rate 500 cir-rate 500
exit

       scheduler-policy "user-type1"
       tier 1
       scheduler root
port-parent level 8 rate pir1 weight w-pir1 cir-level 8 cir-rate cir1
cir-weight w-cir1
            exit
       tier 3
       scheduler wfq
           rate pir1
       parent root
            exit
        exit
```

```
exit

       scheduler-policy "user-type2"
       tier 1
       scheduler root
port-parent level 7 rate pir2 weight w-pir2 cir-level 7 cir-rate cir2
cir-weight w-cir2
            exit
       tier 3
       scheduler wfq
           rate pir2
       parent root
            exit
        exit
exit

       scheduler-policy "b-sap"
       tier 1
       scheduler root
port-parent level 5 rate pir5 weight w-pir5 cir-level 1 cir-rate cir5 cir-weight
w-cir5
            exit
       tier 3
       scheduler wfq
           rate pir5
       parent root
            exit
        exit
exit

    sap-egress 100 // user type 1 QoS policy
    queue 1
                parent wfq weight x level 3 cir-weight x cir-level 3
            packet-byte-offset subtract bytes 22
 queue 2
            packet-byte-offset subtract bytes 22
                parent wfq weight y level 3 cir-weight y cir-level 3
 queue 3
            packet-byte-offset subtract bytes 22
                parent wfq weight z level 3 cir-weight z cir-level 3
 queue 4
                parent root level 8 cir-level 8
            packet-byte-offset subtract bytes 22
 fc be queue 1
 fc l2 queue 2
 fc h2 queue 3
 fc ef queue 4
exit

       sap-egress 200 // user type 2 QoS policy
 queue 1
                parent wfq weight x level 3 cir-weight x cir-level 3
            packet-byte-offset subtract bytes 26
 queue 2
                parent wfq weight y level 3 cir-weight y cir-level 3
            packet-byte-offset subtract bytes 26
queue 3
                parent wfq weight z level 3 cir-weight z cir-level 3
            packet-byte-offset subtract bytes 26
queue 4
                parent root level 8 cir-level 8
            packet-byte-offset subtract bytes 26
 fc be queue 1
 fc l2 queue 2
```

```
 fc h2 queue 3
 fc ef queue 4
exit

       sap-egress 300 // User type 3 QoS policy
 queue 1
                port-parent level 4 rate pir3 weight w-pir3 cir-level
       4 cir-rate cir3 cir-weight w-cir3
            packet-byte-offset subtract bytes 22
 fc be queue 1
exit

       sap-egress 400 // User type 4 QoS policy
 queue 1
                port-parent level 3 rate pir4 weight w-pir4 cir-level
       3 cir-rate cir4 cir-weight w-cir4
            packet-byte-offset subtract bytes 22
 fc be queue 1
exit

       sap-egress 500 // B-SAP default QoS policy
queue 1
                parent wfq weight x level 3 cir-weight x cir-level 3
queue 2
                parent wfq weight y level 3 cir-weight y cir-level 3
 queue 3
                parent wfq weight z level 3 cir-weight z cir-level 3
 queue 4
                parent root level 8 cir-level 8
 fc be queue 1
 fc l2 queue 2
 fc h2 queue 3
 fc ef queue 4
exit
 exit
exit

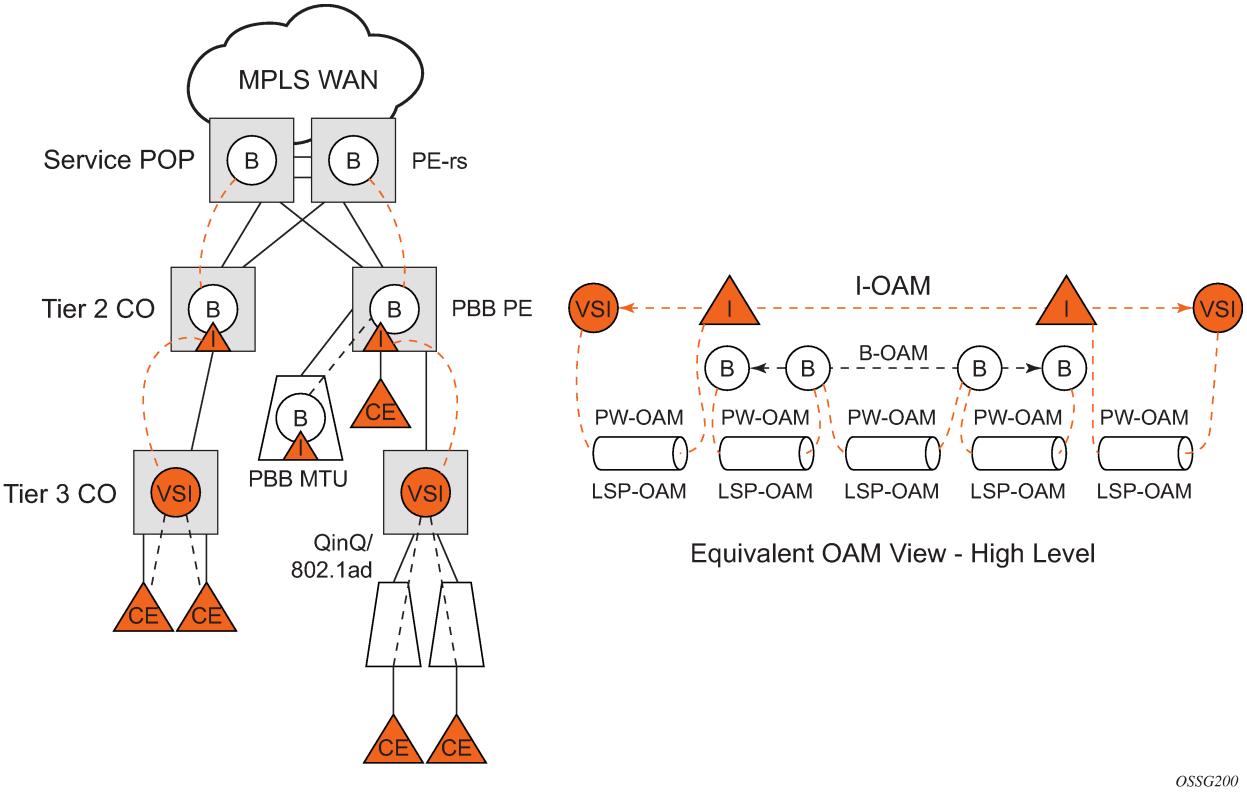
config
       service
       vpls 100 bvpls
            sap 1/1/1:100
                egress
                    encap-defined-qos
                        encap-group type1-grouped type isid
       member 1 to 10
                                qos 100
       scheduler-policy user-type1
                        exit
encap-group type1-separate type isid qos-per-member
       member 16
                                qos 100
       scheduler-policy user-type1
       exit
       encap-group type2-grouped type isid
       member 21 to 30
                                qos 200
       scheduler-policy user-type2
                        exit
encap-group type2-separate type isid qos-per-member
       member 36
                                qos 200
       scheduler-policy user-type2
       exit
       encap-group type3-grouped type isid
```

```
        member 41 to 50
                                qos 300
                        exit
                           encap-group type4-grouped type isid
        member 61 to 70
                                qos 400
        exit
                    qos 500
        scheduler-policy b-sap
        exit
             exit
         exit
     exit
exit
```

5.2.25 PBB OAM

The Nokia PBB implementation supports both MPLS and native Ethernet tunneling. In the case of an MPLS, SDP bindings are used as the B-VPLS infrastructure while T-LDP is used for signaling. As a result, the existing VPLS, MPLS diagnostic tools are supported in both I-VPLS and B-VPLS domains as depicted in Figure 134: PBB OAM view for MPLS infrastructure.

Figure 134: PBB OAM view for MPLS infrastructure



When an Ethernet switching backbone is used for aggregation between PBB PEs, a SAP is used as the B-VPLS up link instead of an SDP. No T-LDP signaling is available.

The existing IEEE 802.1ag implemented for regular VPLS SAPs may be used to troubleshoot connectivity at I-VPLS and B-VPLS layers.

5.2.25.1 Mirroring

There are no restrictions for mirroring in I-VPLS or B-VPLS.

5.2.25.2 OAM commands

All VPLS OAM commands may be used in both I-VPLS and B-VPLS instances.

I-VPLS

- The following OAM commands are meaningful only toward another I-VPLS service instance (spoke-SDP in I-VPLS):
  - LSP-ping
  - LSP-trace
  - SDP-MTU
- The following I-VPLS OAM exchanges are transparently transported over the B-VPLS core:
  - SVC-ping
  - MAC-ping
  - MAC-trace
  - MAC-populate
  - MAC-purge
  - CPE-ping (toward customer CPE)
  - 802.3ah EFM
  - SAA
- PBB up links using MPLS/SAP; there are no PBB specific OAM commands.

B-VPLS

In case of Ethernet switching backbone (B-SAPs on B-VPLS), 802.1ag OAM is supported on B-SAP, operating on:

- the customer level (C-SA/C-DA and C-type layer)
- the tunnel level (B-SA/B-DA and B-type layer)

5.2.25.3 CFM support

There is no special 802.1ag CFM (Connectivity Fault Management) support for PBB. B-component and I-components run their own maintenance domain and levels. CFM for I-components run transparently over the PBB network and appears as directly connected.

5.3 Configuration examples

Use the CLI syntax displayed to configure PBB.

5.3.1 PBB using G.8031 protected Ethernet tunnels

The following displays PBB configuration examples:

Ethernet links on BEB1:

BEB1 to BEB1 L1:

BEB1 to BCB1 L1: 1/1/1 – Member port of LAG-emulation ET1, terminate ET3

BEB1 to BCB1 L2: 2/1/1 – Member port of LAG-emulation ET1

BEB1 to BCB1 L3: 3/1/1 – Member port of LAG-emulation ET1

BEB1 to BCB2: 4/1/1 – terminate ET3

```
*A:7750_ALU>config>eth-tunnel 1
        description "LAG-emulation to BCB1 ET1"
        protection-type loadsharing
        ethernet
            mac 00:11:11:11:11:12
            encap-type dot1q
        exit
        ccm-hold-time down 5 up 10 // 50 ms down, 1 sec up
        lag-emulation
           access adapt-qos distribute
           path-threshold 1
        exit
        path 1
            member 1/1/1
            control-tag 0
            eth-cfm
                …
            exit
            no shutdown
        exit
        path 2
            member 2/1/1
            control-tag 0
            eth-cfm
                …
            exit
            no shutdown
        exit
        path 3
            member 3/1/1
            control-tag 0
            eth-cfm
                …
            exit
            no shutdown
        exit
        no shutdown
-------------------------------------------------
*A:7750_ALU>config>eth-tunnel 3
        description "G.8031 tunnel ET3"
        protection-type 8031_1to1
        ethernet
```

```
            mac 00:11:11:11:11:11
            encap-type dot1q
        exit
        ccm-hold-time down 5 // 50 ms down, no up hold-down
        path 1
            member 1/1/1
            control-tag 5
            precedence primary
            eth-cfm
                mep 2 domain 1 association 1
                    ccm-enable
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        path 2
            member 4/1/1
            control-tag 5
            eth-cfm
                mep 2 domain 1 association 2
                    ccm-enable
                    control-mep
                    no shutdown
                exit
            exit
            no shutdown
        exit
        no shutdown
-------------------------------------------------
# Service config
-------------------------------------------------
*A:7750_ALU>config>service vpls 1 customer 1 m-vpls b-vpls create
   description "m-VPLS for multipoint traffic"
   stp
      mst-name  "BVPLS"
      mode p-mstp
      mst-instance 10
         mst-priority 4096
         vlan-range 100-199
      exit
      mst-instance 20
         mst-priority 8192
         vlan-range 200-299
      exit
      no shutdown
   exit

   sap eth-tunnel-1 create // BSAP0 to BCB E
   sap 4/1/1:0 create // physical link to BCB F (NOTE 0 or 0.*)
                      // indicate untagged for m-VPLS)
   exit
   no shutdown
-------------------------------------------------
# Service config: one of the same-fate SAP over
# loadsharing tunnel
-------------------------------------------------
A:7750_ALU>config service vpls 100 customer 1 b-vpls create
   sap eth-tunnel-1:1 create //to BCB E
      // must specify tags for each path for loadsharing
      eth-tunnel
path 1 tag 100
            path 2 tag 100
```

```
            path 3 tag 100
   exit
   no shutdown   …
   sap 3/1/1:200 // to BCBF
   …

A:7750_ALU>config service vpls 1000 customer 1 i-vpls create
   pbb backbone-vpls 100 isid 1000
   sap 4/1/1:200 // access SAP to QinQ
…
-------------------------------------------------
# Service config: one of epipes into b-VPLS protected tunnel
# as per R7.0 R4
-------------------------------------------------
A:7750_ALU>config service service vpls 3 customer 1 b-vpls create
   sap eth-tunnel-3 create
   …
service epipe 2000
    pbb-tunnel 100 backbone-dest-mac to-AS20 isid 2000
    sap 3/1/1:400 create
```

Example:

```
port 1/1/1
        – ethernet
            – encap-type dot1q
    port 2/2/2
        – ethernet
            – encap-type dot1q
    config eth-tunnel 1
        – path 1
            – member 1/1/1
            – control-tag 100
            – precedence primary
            – eth-cfm
                – mep 51 domain 1 association 1 direction down
                – ccm-enable
                – low-priority-defect allDef
                – mac-address 00:AE:AE:AE:AE:AE
                – control-mep
                – no shutdown
            – no shutdown
        – path 2
            – member 2/2/2
            – control-tag 200
            – eth-cfm
                – mep
                    – mep 52 domain 1 association 2 direction down
                    – ccm-enable
                    – low-priority-defect allDef
                    – mac-address 00:BE:BE:BE:BE:BE
                    – control-mep
                    – no shutdown
            – no shutdown

    config service vpls 1 b-vpls
        – sap eth-tunnel-1
    config service epipe 1000
        – pbb-tunnel 1 backbone-dest-mac remote-beb
        – sap 3/1/1:400.10
```

5.3.2 MC-LAG multihoming for native PBB

This section describes a configuration example for BEB C configuration considering the following assumptions:

- BEB C and BEB D are MC-LAG peers
- B-VPLS 100 on BEB C and BEB D
- VPLS 1000 on BEB C and BEB D
- MC-LAG 1 on BEB C and BEB D

CLI syntax:

```
service pbb
        – source-bmac ab-ac-ad-ef-00-00
    port 1/1/1
        – ethernet
            – encap-type qinq
    lag 1
        – port 1/1/1 priority 20
        – lacp active administrative-key 32768
    redundancy
        – multi-chassis
            – peer 10.1.1.3 create
                – source-address 10.1.1.1
                – mc-lag
                    – lag 1 lacp-key 1 system-id 00:00:00:01:01:01
                    – system-priority 100
                    – source-bmac-lsb use-lacp-key
    service vpls 100 bvpls
        – sap 2/2/2:100 // bvid 100
        – mac-notification
            – no shutdown

    service vpls 101 bvpls
        – sap 2/2/2:101 // bvid 101
        – mac-notification
            – no shutdown
    // no per BVPLS source-bmac configuration, the chassis one (ab-ac-ad-ef-00-00) is used

    service vpls 1000 ivpls
        – backbone-vpls 100
        – sap lag-1:1000 //automatically associates the SAP with ab-ac-ad-ef-00-01 (first 36
 bits from BVPLS 100 sbmac+16bit source-bmac-lsb)

    service vpls 1001 ivpls
        – backbone-vpls 101
        – sap lag-1:1001 //automatically associates the SAP with ab-ac-ad-ef-00-01(first 36
 bits from BVPLS 101 sbmac+16bit source-bmac-lsb)
```

5.3.3 Access multihoming over MPLS for PBB Epipes

This section gives an example configuration for BEB1 from Figure 131: Active/standby PW into PBB Epipes.

```
*A:BEB1>config>service# info
----------------------------------------------
        pbb
            source-bmac 00:00:00:00:11:11
```

```
            mac-name "remote-BEB" 00:44:44:44:44:44
        exit
        sdp 1 mpls create
            far-end 10.1.1.4
            ldp
            keep-alive
                shutdown
            exit
            source-bmac-lsb 33:33 control-pw-vc-id 100
            no shutdown
        exit
        vpls 10 customer 1 b-vpls create
            service-mtu 1532
            stp
                shutdown
            exit
            spb 1024 fid 1 create
                no shutdown
            exit
            sap 1/1/1:10 create
                spb create
                    no shutdown
                exit
            exit
            sap 1/1/5:10 create
                spb create
                    no shutdown
                exit
            exit
            no shutdown
        exit
        epipe 100 customer 1 create
            pbb
                tunnel 10 backbone-dest-mac "remote-BEB" isid 100
            exit
            spoke-sdp 1:100 create
                use-sdp-bmac
                no shutdown
            exit
            no shutdown
        exit
        epipe 101 customer 1 create
            pbb
                tunnel 10 backbone-dest-mac "remote-BEB" isid 101
            exit
            spoke-sdp 1:101 create
                use-sdp-bmac
                no shutdown
            exit
            no shutdown
        exit
-----------------------------------------------
*A:BEB1>config>service#
```

The SDP control pseudowire information can be seen using this command:

```
*A:BEB1# show service sdp 1 detail

===============================================================================
Service Destination Point (Sdp Id : 1) Details
===============================================================================
-------------------------------------------------------------------------------
 Sdp Id 1  -10.1.1.4
```

```
-------------------------------------------------------------------------------
Description           : (Not Specified)
SDP Id               : 1                    SDP Source         : manual
...
Src B-MAC LSB        : 33-33                Ctrl PW VC ID      : 100
Ctrl PW Active       : Yes

...
===============================================================================
*A:BEB1#
```

The configuration of a pseudowire to support remote active/standby PBB Epipe operation can be seen using this command:

```
*A:BEB1# show service id 100 sdp 1:100 detail

===============================================================================
Service Destination Point (Sdp Id : 1:100) Details
===============================================================================
-------------------------------------------------------------------------------
 Sdp Id 1:100  -(10.1.1.4)
-------------------------------------------------------------------------------
Description     : (Not Specified)
SDP Id             : 1:100                  Type             : Spoke
...
Use SDP B-MAC      : True
...
===============================================================================
*A:BEB1#8.C
```

6 EVPN

6.1 Overview and EVPN applications

Ethernet Virtual Private Networks (EVPN) is an IETF technology per RFC 7432, *BGP MPLS-Based Ethernet VPN*, that uses a new BGP address family and allows VPLS services to be operated as IP-VPNs, where the MAC addresses and the information to set up the flooding trees are distributed by BGP.

EVPN is defined to fill the gaps of other L2VPN technologies such as VPLS. The main objective of the EVPN is to build E-LAN services in a similar way to RFC 4364 IP-VPNs, while supporting MAC learning within the control plane (distributed by MP-BGP), efficient multi-destination traffic delivery, and active-active multihoming.

EVPN can be used as the control plane for different data plane encapsulations. The Nokia implementation supports the following data planes:

- **EVPN for VXLAN overlay tunnels (EVPN-VXLAN)**

  EVPN for VXLAN overlay tunnels (EVPN-VXLAN), being the Data Center Gateway (DGW) function the main application for this feature. In such application VXLAN is expected within the Data Center and VPLS SDP bindings or SAPs are expected for the connectivity to the WAN. R-VPLS and VPRN connectivity to the WAN is also supported.

  The EVPN-VXLAN functionality is standardized in RFC 8365.

- **EVPN for MPLS tunnels (EVPN-MPLS)**

  EVPN for MPLS tunnels (EVPN-MPLS), where PEs are connected by any type of MPLS tunnel. EVPN-MPLS is generally used as an evolution for VPLS services in the WAN, being Data Center Interconnect one of the main applications.

  The EVPN-MPLS functionality is standardized in RFC 7432.

- **EVPN for PBB over MPLS tunnels (PBB-EVPN)**

  PEs are connected by PBB over MPLS tunnels in this data plane. It is usually used for large scale E-LAN and E-Line services in the WAN.

  The PBB-EVPN functionality is standardized in RFC 7623.

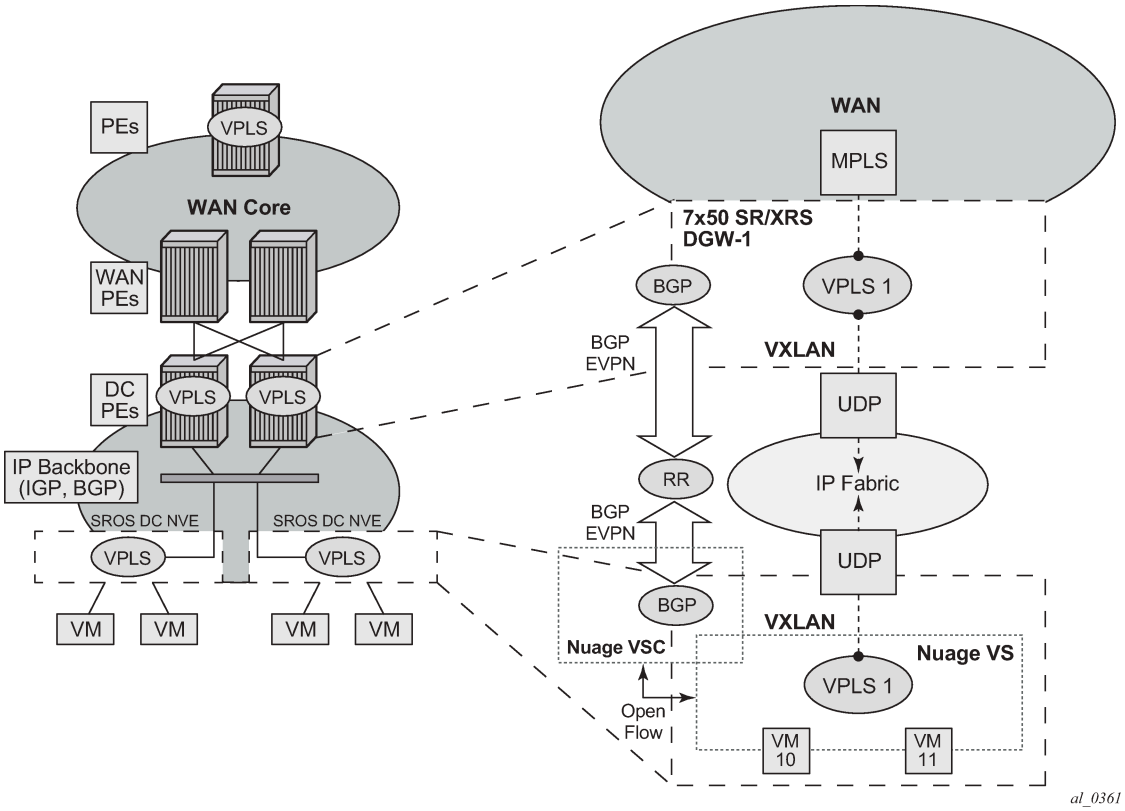
The 7750 SR, 7450 ESS, or 7950 XRS EVPN VXLAN implementation is integrated in the Nuage Data Center architecture, where the router serves as the DGW.

For more information about the Nuage Networks architecture and products, see the *Nuage Networks Virtualized Service Platform Guide*. The following sections describe the applications supported by EVPN in the 7750 SR, 7450 ESS, or 7950 XRS implementation.

6.1.1 EVPN for VXLAN tunnels in a Layer 2 DGW (EVPN-VXLAN)

Figure 135: Layer 2 DC PE with VPLS to the WAN shows the use of EVPN for VXLAN overlay tunnels on the 7750 SR, 7450 ESS, or 7950 XRS when it is used as a Layer 2 DGW.

Figure 135: Layer 2 DC PE with VPLS to the WAN



DC providers require a DGW solution that can extend tenant subnets to the WAN. Customers can deploy the NVO3-based solutions in the DC, where EVPN is the standard control plane and VXLAN is a predominant data plane encapsulation. The Nokia DC architecture uses EVPN and VXLAN as the control and data plane solutions for Layer 2 connectivity within the DC and so does the SR OS.

While EVPN VXLAN is used within the DC, some service providers use VPLS and H-VPLS as the solution to extend Layer 2 VPN connectivity. Figure 135: Layer 2 DC PE with VPLS to the WAN shows the Layer 2 DGW function on the 7750 SR, 7450 ESS, and 7950 XRS routers, providing VXLAN connectivity to the DC and regular VPLS connectivity to the WAN.

The WAN connectivity is based on VPLS where SAPs (null, dot1q, and qinq), spoke SDPs (FEC type 128 and 129), and mesh-SDPs are supported.

The DC GWs can provide multihoming resiliency through the use of BGP multihoming.

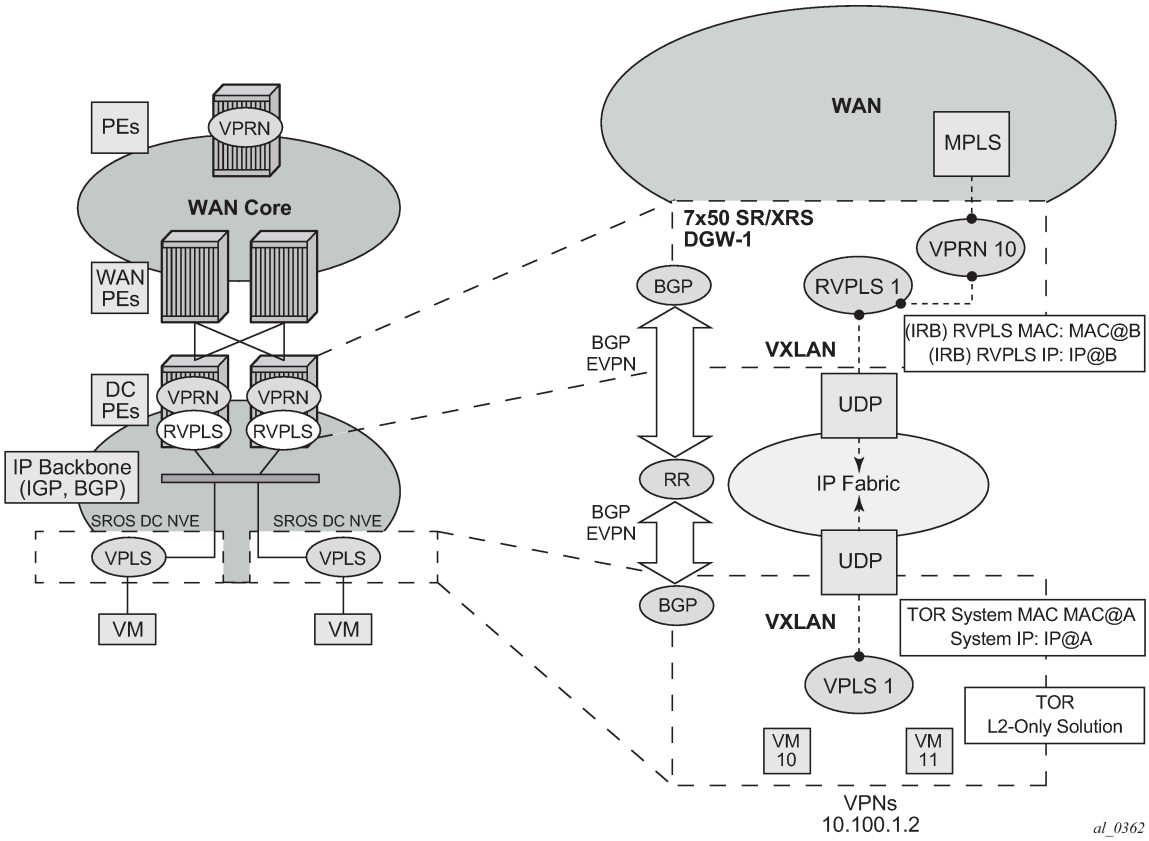
EVPN-MPLS can also be used in the WAN. In this case, the Layer 2 DGW function provides translation between EVPN-VXLAN and EVPN-MPLS. EVPN multihoming can be used to provide DGW redundancy.

If point-to-point services are needed in the DC, SR OS supports the use of EVPN-VPWS for VXLAN tunnels, including multihoming, according to RFC8214.

6.1.2 EVPN for VXLAN tunnels in a Layer 2 DC with integrated routing bridging connectivity on the DGW

Figure 136: Gateway IRB on the DC PE for an L2 EVPN/VXLAN DC shows the use of EVPN for VXLAN overlay tunnels on the 7750 SR, 7450 ESS, or 7950 XRS when the DC provides Layer 2 connectivity and the DGW can route the traffic to the WAN through an R-VPLS and linked VPRN.

Figure 136: Gateway IRB on the DC PE for an L2 EVPN/VXLAN DC

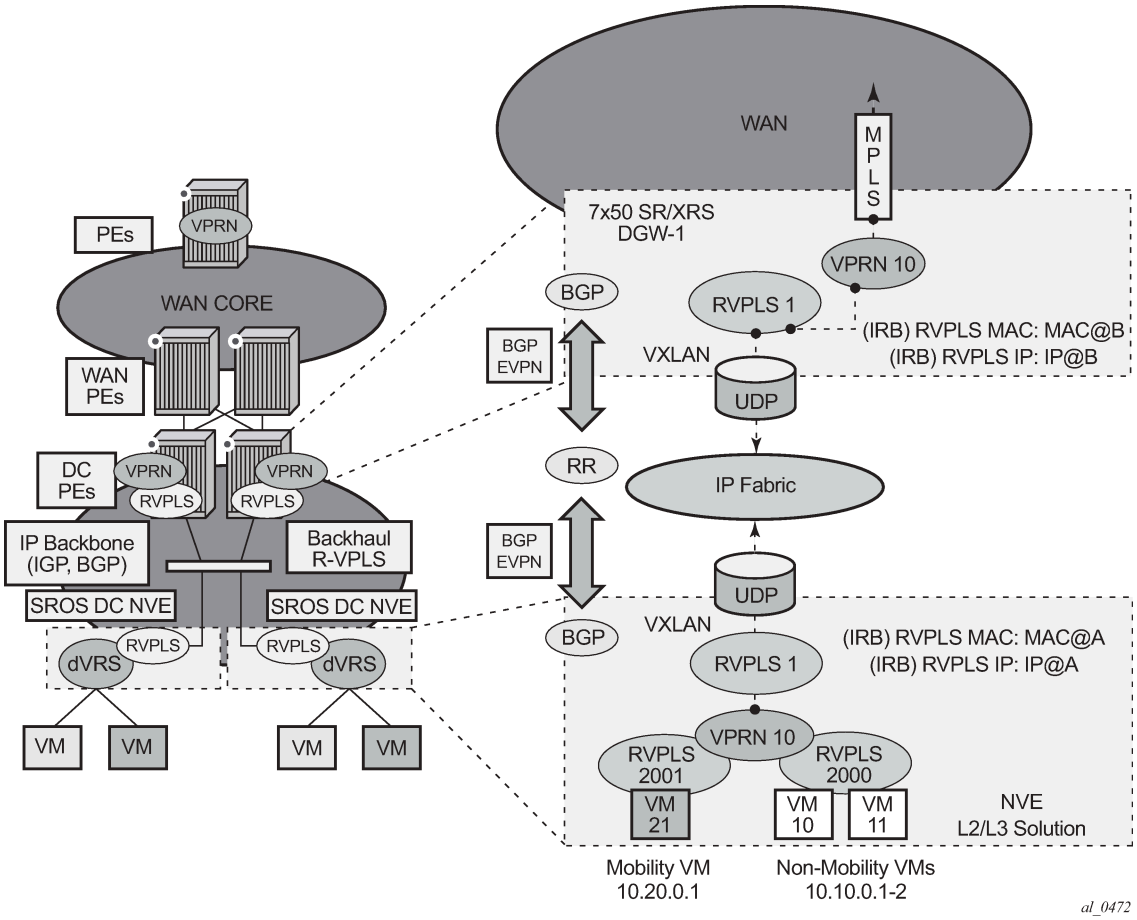


In some cases, the DGW must provide a Layer 3 default gateway function to all the hosts in a specified tenant subnet. In this case, the VXLAN data plane is terminated in an R-VPLS on the DGW, and connectivity to the WAN is accomplished through regular VPRN connectivity. The 7750 SR, 7450 ESS, and 7950 XRS support IPv4 and IPv6 interfaces as default gateways in this scenario.

6.1.3 EVPN for VXLAN tunnels in a Layer 3 DC with integrated routing bridging connectivity among VPRNs

Figure 137: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC shows the use of EVPN for VXLAN tunnels on the 7750 SR, 7450 ESS, or 7950 XRS when the DC provides distributed Layer 3 connectivity to the DC tenants.

Figure 137: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC



Each tenant has several subnets for which each DC Network Virtualization Edge (NVE) provides intra-subnet forwarding. An NVE may be a Nuage VSG, VSC/VRS, or any other NVE in the market supporting the same constructs, and each subnet normally corresponds to an R-VPLS. For example, in Figure 137: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC, subnet 10.20.0.0 corresponds to R-VPLS 2001 and subnet 10.10.0.0 corresponds to R-VPLS 2000.

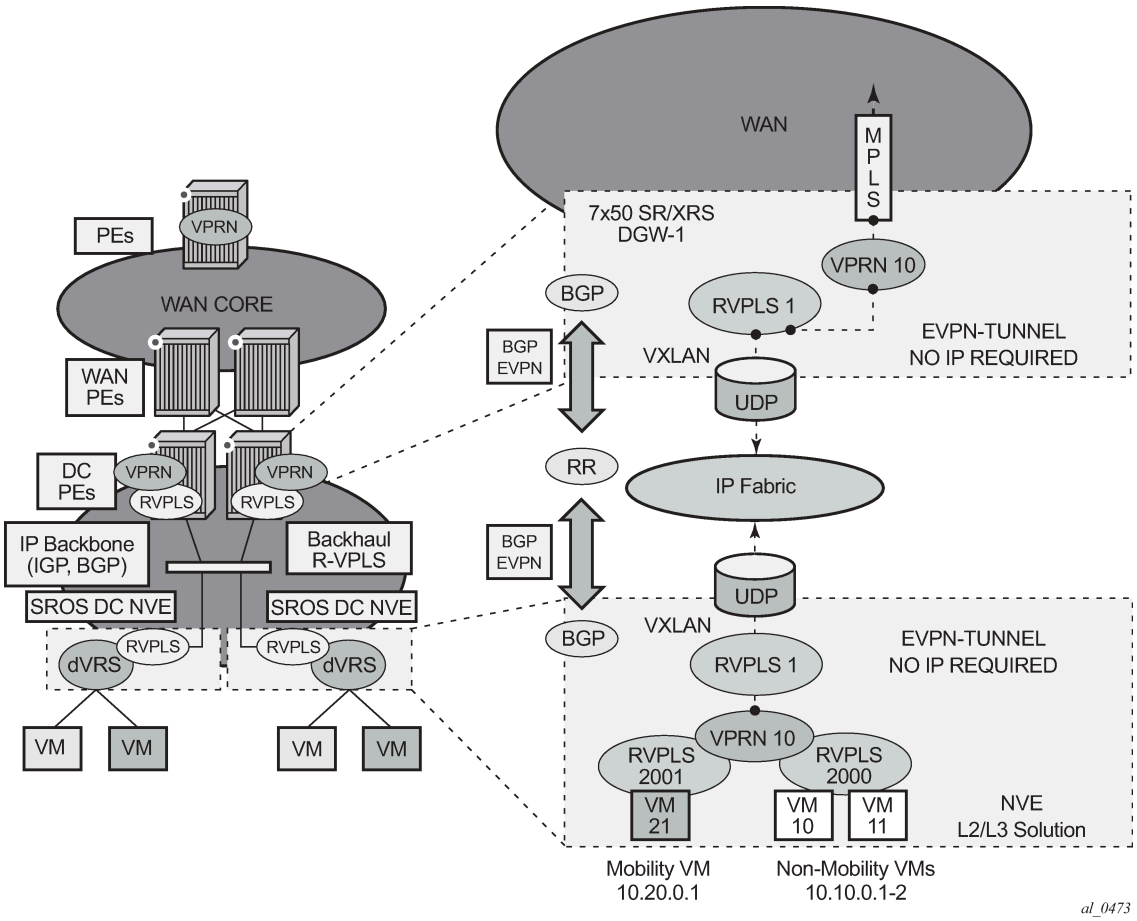
In this example, the NVE provides inter-subnet forwarding too, by connecting all the local subnets to a VPRN instance. When the tenant requires Layer 3 connectivity to the IP-VPN in the WAN, a VPRN is defined in the DGWs, which connects the tenant to the WAN. That VPRN instance is connected to the VPRNs in the NVEs by means of an IRB (Integrated Routing and Bridging) backhaul R-VPLS. This IRB backhaul R-VPLS provides a scalable solution because it allows Layer 3 connectivity to the WAN without the need for defining all of the subnets in the DGW.

The 7750 SR, 7450 ESS, and 7950 XRS DGW support the IRB backhaul R-VPLS model, where the R-VPLS runs EVPN-VXLAN and the VPRN instances exchange IP prefixes (IPv4 and IPv6) through the use of EVPN. Interoperability between the EVPN and IP-VPN for IP prefixes is also fully supported.

6.1.4 EVPN for VXLAN tunnels in a Layer 3 DC with EVPN-tunnel connectivity among VPRNs

Figure 138: EVPN-tunnel gateway IRB on the DC PE for an L3 EVPN/VXLAN DC shows the use of EVPN for VXLAN tunnels on the 7750 SR, 7450 ESS, or 7950 XRS, when the DC provides distributed Layer 3 connectivity to the DC tenants and the VPRN instances are connected through EVPN tunnels.

Figure 138: EVPN-tunnel gateway IRB on the DC PE for an L3 EVPN/VXLAN DC



The solution described in section EVPN for VXLAN tunnels in a Layer 3 DC with integrated routing bridging connectivity among VPRNs provides a scalable IRB backhaul R-VPLS service where all the VPRN instances for a specified tenant can be connected by using IRB interfaces. When this IRB backhaul R-VPLS is exclusively used as a backhaul and does not have any SAPs or SDP bindings directly attached, the solution can be optimized by using EVPN tunnels.

EVPN tunnels are enabled using the **evpn-tunnel** command under the R-VPLS interface configured on the VPRN. EVPN tunnels provide the following benefits to EVPN-VXLAN IRB backhaul R-VPLS services:

- **easier provisioning of the tenant service**

  If an EVPN tunnel is configured in an IRB backhaul R-VPLS, there is no need to provision the IRB IPv4 addresses on the VPRN. This makes the provisioning easier to automate and saves IP addresses from the tenant space.

Note: IPv6 interfaces do not require the provisioning of an IPv6 Global Address; a Link Local Address is automatically assigned to the IRB interface.

- **higher scalability of the IRB backhaul R-VPLS**

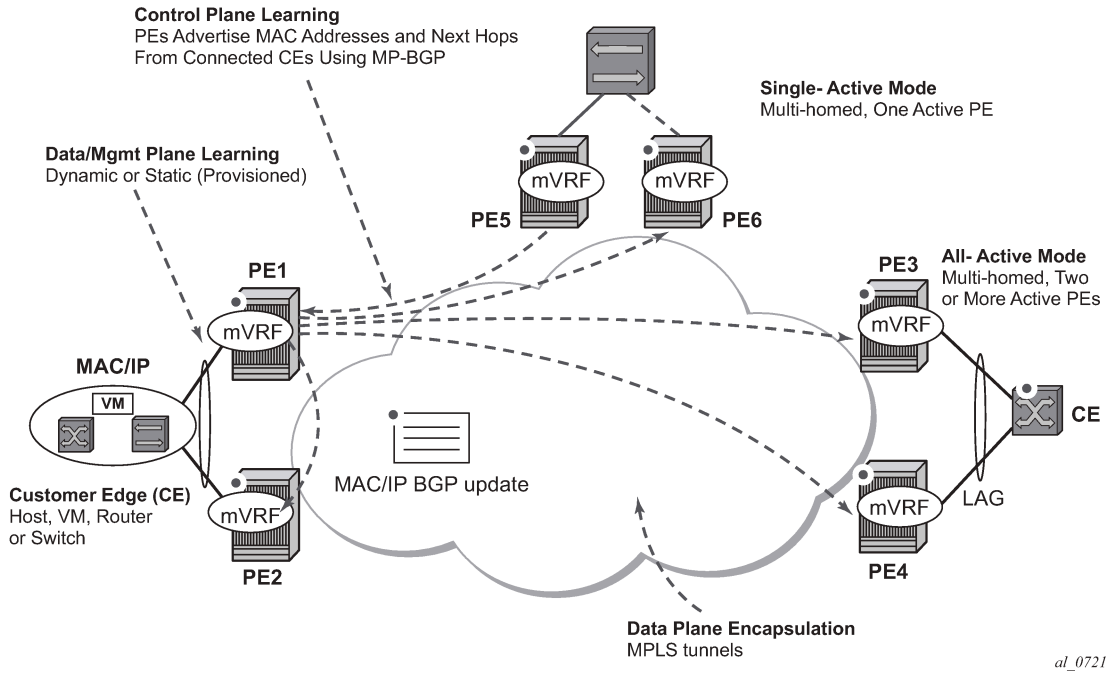
  If EVPN tunnels are enabled, multicast traffic is suppressed in the EVPN-VXLAN IRB backhaul R-VPLS service (it is not required). As a result, the number of VXLAN binds in IRB backhaul R-VPLS services with EVPN-tunnels can be much higher.

This optimization is fully supported by the 7750 SR, 7450 ESS, and 7950 XRS.

6.1.5 EVPN for MPLS tunnels in E-LAN services

Figure 139: EVPN for MPLS in VPLS services shows the use of EVPN for MPLS tunnels on the 7750 SR, 7450 ESS, and 7950 XRS. In this case, EVPN is used as the control plane for E-LAN services in the WAN.

Figure 139: EVPN for MPLS in VPLS services



EVPN-MPLS is standardized in RFC 7432 as an L2VPN technology that can fill the gaps in VPLS for E-LAN services. A significant number of service providers offering E-LAN services today are requesting EVPN for their multihoming capabilities, as well as the optimization EVPN provides. EVPN supports all-active multihoming (per-flow load-balancing multihoming) as well as single-active multihoming (per-service load-balancing multihoming).

EVPN is a standard-based technology that supports all-active multihoming, and although VPLS already supports single-active multihoming, EVPN's single-active multihoming is perceived as a superior technology because of its mass-withdrawal capabilities to speed up convergence in scaled environments.

EVPN technology provides a number of significant benefits, including:

- superior multihoming capabilities

- an IP-VPN-like operation and control for E-LAN services
- reduction and (in some cases) suppression of the BUM (broadcast, Unknown unicast, and Multicast) traffic in the network
- simple provision and management
- new set of tools to control the distribution of MAC addresses and ARP entries in the network

The SR OS EVPN-MPLS implementation is compliant with RFC 7432.

EVPN-MPLS can also be enabled in R-VPLS services with the same feature-set that is described for VXLAN tunnels in sections EVPN for VXLAN tunnels in a Layer 3 DC with integrated routing bridging connectivity among VPRNs and EVPN for VXLAN tunnels in a Layer 3 DC with EVPN-tunnel connectivity among VPRNs.

6.1.6 EVPN for MPLS tunnels in E-Line services

The MPLS network used by EVPN for E-LAN services can also be shared by E-Line services using EVPN in the control plane. EVPN for E-Line services (EVPN-VPWS) is a simplification of the RFC 7432 procedures, and it is supported in compliance with RFC 8214.

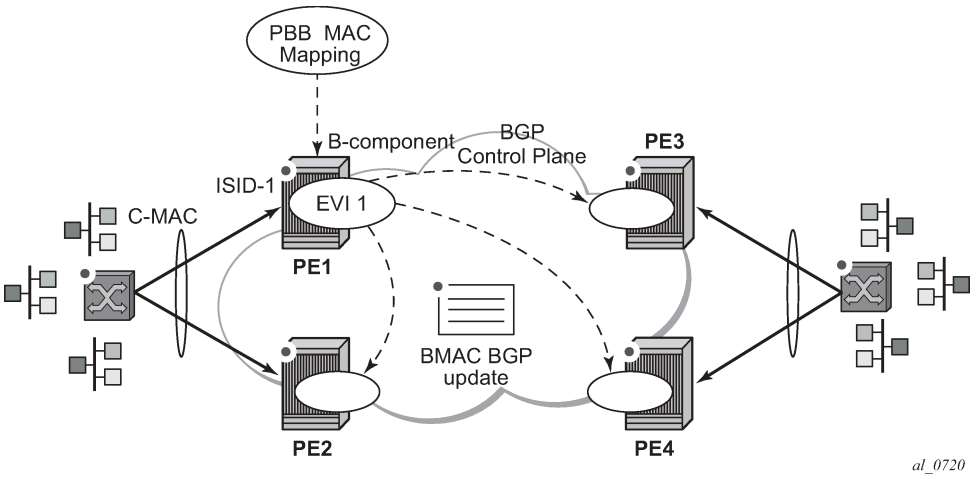
6.1.7 EVPN for MPLS tunnels in E-Tree services

The MPLS network used by E-LAN and E-Line services can also be shared by Ethernet-Tree (E-Tree) services using the EVPN control plane. EVPN E-Tree services use the EVPN control plane extensions described in IETF RFC 8317 and are supported on the 7750 SR, 7450 ESS, and 7950 XRS.

6.1.8 EVPN for PBB over MPLS tunnels (PBB-EVPN)

Figure 140: EVPN for PBB over MPLS shows the use of EVPN for MPLS tunnels on the 7750 SR, 7450 ESS, and 7950 XRS. In this case, EVPN is used as the control plane for E-LAN services in the WAN.

Figure 140: EVPN for PBB over MPLS



EVPN for PBB over MPLS (hereafter called PBB-EVPN) is specified in RFC 7623. It provides a simplified version of EVPN for cases where the network requires very high scalability and does not need all the advanced features supported by EVPN-MPLS (but still requires single-active and all-active multihoming capabilities).

PBB-EVPN is a combination of 802.1ah PBB and RFC 7432 EVPN and reuses the PBB-VPLS service model, where BGP-EVPN is enabled in the B-VPLS domain. EVPN is used as the control plane in the B-VPLS domain to control the distribution of B-MACs and setup per-ISID flooding trees for I-VPLS services. The learning of the C-MACs, either on local SAPs/SDP bindings or associated with remote B-MACs, is still performed in the data plane. Only the learning of B-MACs in the B-VPLS is performed through BGP.

The SR OS PBB-EVPN implementation supports PBB-EVPN for I-VPLS and PBB-Epipe services, including single-active and all-active multihoming.

6.2 EVPN for VXLAN tunnels and cloud technologies

This section provides information about EVPN for VXLAN tunnels and cloud technologies.

6.2.1 VXLAN

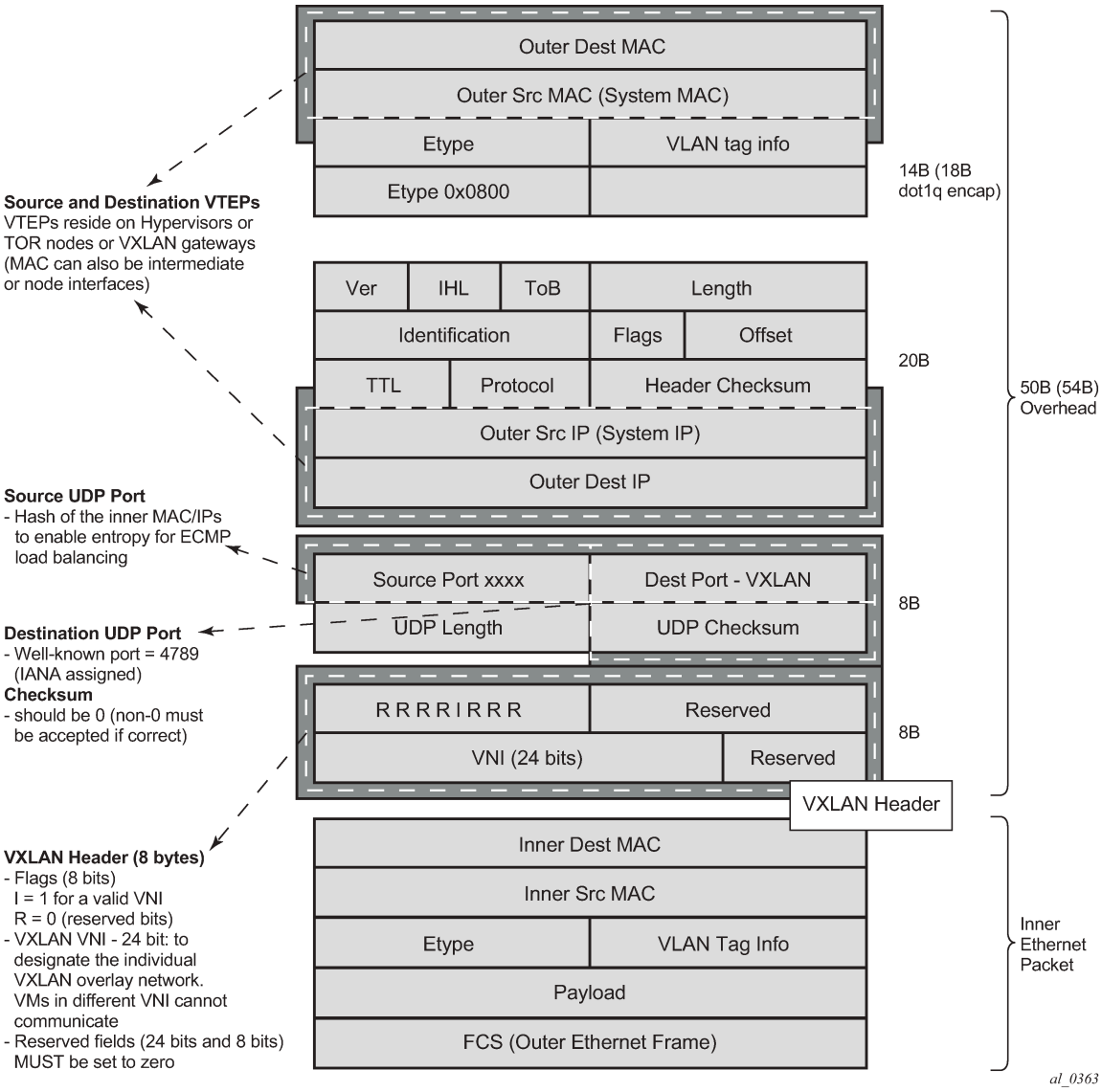
The SR OS, SR Linux and Nuage solution for DC supports VXLAN (Virtual eXtensible Local Area Network) overlay tunnels as per RFC 7348.

VXLAN addresses the data plane needs for overlay networks within virtualized data centers accommodating multiple tenants. The main attributes of the VXLAN encapsulation are:

- VXLAN is an overlay network encapsulation used to carry MAC traffic between VMs over a logical Layer 3 tunnel.
- Avoids the Layer 2 MAC explosion, because VM MACs are only learned at the edge of the network. Core nodes simply route the traffic based on the destination IP (which is the system IP address of the remote PE or VTEP-VXLAN Tunnel End Point).
- Supports multi-path scalability through ECMP (to a remote VTEP address, based on source UDP port entropy) while preserving the Layer 2 connectivity between VMs. xSTP is no longer needed in the network.
- Supports multiple tenants, each with their own isolated Layer 2 domain. The tenant identifier is encoded in the VNI field (VXLAN Network Identifier) and allows up to 16M values, as opposed to the 4k values provided by the 802.1q VLAN space.

Figure 141: VXLAN frame format shows an example of the VXLAN encapsulation supported by the Nokia implementation.

Figure 141: VXLAN frame format



As shown in Figure 141: VXLAN frame format, VXLAN encapsulates the inner Ethernet frames into VXLAN + UDP/IP packets. The main pieces of information encoded in this encapsulation are:

- VXLAN header (8 bytes)
  - Flags (8 bits) where the I flag is set to 1 to indicate that the VNI is present and valid. The rest of the flags (“Reserved” bits) are set to 0.
  - Includes the VNI field (24-bit value) or VXLAN network identifier. It identifies an isolated Layer 2 domain within the DC network.
  - The rest of the fields are reserved for future use.
- UDP header (8 bytes)
  - Where the destination port is a well-known UDP port assigned by IANA (4789).

  - The source port is derived from a hashing of the inner source and destination MAC/IP addresses that the 7750 SR, 7450 ESS, or 7950 XRS does at ingress. This creates an "entropy" value that can be used by the core DC nodes for load balancing on ECMP paths.
  - The checksum is set to zero.
- Outer IP and Ethernet headers (34 or 38 bytes)
  - The source IP and source MAC identifies the source VTEP. That is, these fields are populated with the PE's system IP and chassis MAC address.

    Note: The source MAC address is changed on all the IP hops along the path, as is usual in regular IP routing.

  - The destination IP identifies the remote VTEP (remote system IP) and be the result of the destination MAC lookup in the service Forwarding Database (FDB).

    Note: All remote MACs are learned by the EVPN BGP and associated with a remote VTEP address and VNI.

Some considerations related to the support of VXLAN on the 7750 SR, 7450 ESS, and 7950 XRS are:

- VXLAN is only supported on network or hybrid ports with null or dot1q encapsulation.
- VXLAN is supported on Ethernet/LAG and POS/APS.
- IPv4 and IPv6 unicast addresses are supported as VTEPs.
- By default, system IP addresses are supported, as VTEPs, for originating and terminating VXLAN tunnels. Non-system IPv4 and IPv6 addresses are supported by using a Forwarding Path Extension (FPE).

6.2.1.1 VXLAN ECMP and LAG

The DGW supports ECMP load balancing to reach the destination VTEP. Also, any intermediate core node in the Data Center should be able to provide further load balancing across ECMP paths because the source UDP port of each tunneled packet is derived from a hash of the customer inner packet. The following must be considered:

- ECMP for VXLAN is supported on VPLS services, but not for BUM traffic. Unicast spraying is based on the packet contents.
- ECMP for VXLAN on R-VPLS services is supported for VXLAN IPv6 tunnels.
- ECMP for VXLAN IPv4 tunnels on R-VPLS is only supported if the command **configure service vpls allow-ip-int-bind vxlan-ipv4-tep-ecmp** is enabled on the R-VPLS (as well as **config**>**router**>**ecmp**).
- ECMP for Layer 3 multicast traffic on R-VPLS services with EVPN-VXLAN destinations is only supported if the **vpls allow-ip-int-bind ip-multicast-ecmp** command is enabled (as well as **config**>**router**>**ecmp**).
- In the cases where ECMP is not supported (BUM traffic in VPLS and ECMP on R-VPLS if not enabled), each VXLAN binding is tied to a single (different) ECMP path, so that in a normal deployment with a reasonable number of remote VTEPs, there should be a fair distribution of the traffic across the paths. In other words, only per-VTEP load-balancing is supported, instead of per-flow load-balancing.

- LAG spraying based on the packet hash is supported in all the cases (VPLS unicast, VPLS BUM, and R-VPLS).

6.2.1.2 VXLAN VPLS tag handling

The following describes the behavior on the 7750 SR, 7450 ESS, and 7950 XRS with respect to VLAN tag handling for VXLAN VPLS services:

- Dot1q, QinQ, and null SAPs, as well as regular VLAN handling procedures at the WAN side, are supported on VXLAN VPLS services.
- No "vc-type vlan" like VXLAN VNI bindings are supported. Therefore, at the egress of the VXLAN network port, the router does not add any inner VLAN tag on top of the VXLAN encapsulation, and at the ingress network port, the router ignores any VLAN tag received and considers it as part of the payload.

6.2.1.3 VXLAN MTU considerations

For VXLAN VPLS services, the network port MTU must be at least 50 Bytes (54 Bytes if dot1q) greater than the Service-MTU to allow enough room for the VXLAN encapsulation.

The Service-MTU is only enforced on SAPs, (any SAP ingress packet with MTU greater than the service-mtu is discarded) and not on VXLAN termination (any VXLAN ingress packet makes it to the egress SAP regardless of the configured service-mtu).

If BGP-EVPN is enabled in a VXLAN VPLS service, the Service-MTU can be advertised in the Inclusive Multicast Ethernet Tag routes and enforce that all the routers attached to the same EVPN service have the same Service-MTU configured.

Note: The router never fragments or reassemble VXLAN packets. In addition, the router always sets the DF (Do not Fragment) flag in the VXLAN outer IP header.

6.2.1.4 VXLAN QoS

VXLAN is a network port encapsulation; therefore, the QoS settings for VXLAN are controlled from the network QoS policies.

6.2.1.4.1 Ingress

The network ingress QoS policy can be applied either to the network interface over which the VXLAN traffic arrives or under **vxlan**/**network**/**ingress** within the EVPN service.

Regardless of where the network QoS policy is applied, the ingress network QoS policy is used to classify the VXLAN packets based on the outer dot1p (if present), then the outer DSCP, to yield an FC/profile.

If the ingress network QoS policy is applied to the network interface over which the VXLAN traffic arrives then the VXLAN unicast traffic uses the network ingress queues configured on FP where the network interface resides. QoS control of BUM traffic received on the VXLAN tunnels is possible by separately redirecting these traffic types to policers within an FP ingress network queue group. This QoS control uses the per forwarding class **fp-redirect-group** parameter together with **broadcast-policer**, **unknown-policer**, and **mcast-policer** within the ingress section of a network QoS policy. This QoS control applies to all BUM

traffic received for that forwarding class on the network IP interface on which the network QoS policy is applied.

The ingress network QoS policy can also be applied within the EVPN service by referencing an FP queue group instance, as follows:

```
configure
    service
        vpls <service-id>
            vxlan vni <vni-id>
                network
                    ingress
                        qos <network-policy-id>
                            fp-redirect-group <queue-group-name>
                                     instance <instance-id>
```

In this case, the redirection to a specific ingress FP queue group applies as a single entity (per forwarding class) to all VXLAN traffic received only by this service. This overrides the QoS applied to the related network interfaces for traffic arriving on VXLAN tunnels in that service but does not affect traffic received on a spoke SDP in the same service. It is possible to also redirect unicast traffic to a policer using the per forwarding class **fp-redirect-group policer** parameter, as well as the BUM traffic as above, within the ingress section of a network QoS policy. The use of **ler-use-dscp**, **ip-criteria** and **ipv6-criteria** statements are ignored if configured in the ingress section of the referenced network QoS policy. If the instance of the named queue group template referenced in the **qos** command is not configured on an FP receiving the VXLAN traffic, then the traffic uses the ingress network queues or queue group related to the network interface.

6.2.1.4.2 Egress

On egress, there is no need to specify "remarking" in the policy to mark the DSCP. This is because the VXLAN adds a new IPv4 header, and the DSCP is always marked based on the egress network qos policy.

6.2.1.5 VXLAN ping

A new VXLAN troubleshooting tool, VXLAN Ping, is available to verify VXLAN VTEP connectivity. The **VXLAN Ping** command is available from interactive CLI and SNMP.

This tool allows the operator to specify a wide range of variables to influence how the packet is forwarded from the VTEP source to VTEP termination. The ping function requires the operator to specify a different **test-id** (equates to originator handle) for each active and outstanding test. The required local **service** identifier from which the test is launched determines the source IP (the system IP address) to use in the outer IP header of the packet. This IP address is encoded into the VXLAN header Source IP TLV. The service identifier also encodes the local VNI. The **outer-ip-destination** must equal the VTEP termination point on the remote node, and the **dest-vni** must be a valid VNI within the associated service on the remote node. The outer source IP address is automatically detected and inserted in the IP header of the packet. The outer source IP address uses the IPv4 system address by default.

If the VTEP is created using a non-system source IP address through the **vxlan-src-vtep** command, the outer source IP address uses the address specified by **vxlan-src-vtep**. The remainder of the variables are optional.

The VXLAN PDU is encapsulated in the appropriate transport header and forwarded within the overlay to the appropriate VTEP termination. The VXLAN router alert (RA) bit is set to prevent forwarding OAM PDU beyond the terminating VTEP. Because handling of the router alert bit was not defined in some early

releases of VXLAN implementations, the VNI Informational bit (I-bit) is set to "0" for OAM packets. This indicates that the VNI is invalid, and the packet should not be forwarded. This safeguard can be overridden by including the **i-flag-on** option that sets the bit to "1", valid VNI. Ensure that OAM frames meant to be contained to the VTEP are not forwarded beyond its endpoints.

The supporting VXLAN OAM ping draft includes a requirement to encode a reserved IEEE MAC address as the inner destination value. However, at the time of implementation, that IEEE MAC address had not been assigned. The inner IEEE MAC address defaults to 00:00:00:00:00:00, but may be changed using the **inner-l2** option. Inner IEEE MAC addresses that are included with OAM packets are not learned in the local Layer 2 forwarding databases.

The echo responder terminates the VXLAN OAM frame, and takes the appropriate response action, and include relevant return codes. By default, the response is sent back using the IP network as an IPv4 UDP response. The operator can choose to override this default by changing the **reply-mode** to **overlay**. The overlay return mode forces the responder to use the VTEP connection representing the source IP and source VTEP. If a return overlay is not available, the echo response is dropped by the responder.

Support is included for:

- IPv4 VTEP
- Optional specification of the outer UDP Source, which helps downstream network elements along the path with ECMP to hash to flow to the same path
- Optional configuration of the inner IP information, which helps the operator test different equal paths where ECMP is deployed on the source. A test only validates a single path where ECMP functions are deployed. The inner IP information is processed by a hash function, and there is no guarantee that changing the IP information between tests selects different paths.
- Optional end system validation for a single L2 IEEE MAC address per test. This function checks the remote FDB for the configured IEEE MAC Address. Only one end system IEEE MAC Address can be configured per test.
- Reply mode UDP (default) or Overlay
- Optional additional padding can be added to each packet. There is an option that indicates how the responder should handle the pad TLV. By default, the padding is not reflected to the source. The operator can change this behavior by including the **reflect-pad** option. The **reflect-pad** option is not supported when the reply mode is set to UDP.
- Configurable send counts, intervals, times outs, and forwarding class

The VXLAN OAM PDU includes two timestamps. These timestamps are used to report forward direction delay. Unidirectional delay metrics require accurate time of day clock synchronization. Negative unidirectional delay values are reported as "0.000". The round trip value includes the entire round trip time including the time that the remote peer takes to process that packet. These reported values may not be representative of network delay.

The following example commands and outputs show how the VXLAN Ping function can be used to validate connectivity. The echo output includes a new header to better describe the VXLAN ping packet headers and the various levels.

```
oam vxlan-ping test-id 1 service 1 dest-vni 2 outer-ip-destination 10.20.1.4
interval
0.1 send-count 10

TestID 1, Service 1, DestVNI 2, ReplyMode UDP, IFlag Off, PadSize 0, ReflectPad No,
SendCount 10, Interval 0.1, Timeout 5
Outer: SourceIP 10.20.1.3, SourcePort Dynamic, DestIP 10.20.1.4, TTL 10, FC be, Prof
ile
```

```
In
Inner: DestMAC 00:00:00:00:00:00, SourceIP 10.20.1.3, DestIP 127.0.0.1

! ! ! ! ! ! ! ! ! !
---- vxlan-id 2 ip-address 10.20.1.4 PING Statistics ----
10 packets transmitted, 10 packets received, 0.00% packet loss
   10 non-errored responses(!), 0 out-of-order(*), 0 malformed echo responses(.)
   0 send errors(.), 0 time outs(.)
   0 overlay segment not found, 0 overlay segment not operational
forward-delay min = 1.097ms, avg = 2.195ms, max = 2.870ms, stddev = 0.735ms
round-trip-delay min = 1.468ms, avg = 1.693ms, max = 2.268ms, stddev = 0.210ms



oam vxlan-ping test-id 2 service 1 dest-vni 2 outer-ip-destination 10.20.1.4 outer-
ip-source-udp 65000 outer-ip-ttl 64 inner-l2 d0:0d:1e:00:00:01 inner-ip-source
 192.168.1.2 inner-ip-destination 127.0.0.8 reply-mode overlay send-count 20
interval
 1 timeout 3 padding 1000 reflect-pad fc nc profile out

TestID 2, Service 1, DestVNI 2, ReplyMode overlay, IFlag Off, PadSize 1000, ReflectP
ad
 Yes, SendCount 20, Interval 1, Timeout 3
Outer: SourceIP 10.20.1.3, SourcePort 65000, DestIP 10.20.1.4, TTL 64, FC nc, Profil
e
 out
Inner: DestMAC d0:0d:1e:00:00:01, SourceIP 192.168.1.2, DestIP 127.0.0.8

================================================================================
rc=1 Malformed Echo Request Received, rc=2 Overlay Segment Not Present, rc=3 Overlay
 Segment Not Operational, rc=4 Ok
================================================================================
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=1 ttl=255 rtt-time=1.733ms fwd
-time=0.302ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=2 ttl=255 rtt-time=1.549ms fwd
-time=1.386ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=3 ttl=255 rtt-time=3.243ms fwd
-time=0.643ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=4 ttl=255 rtt-time=1.551ms fwd
-time=2.350ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=5 ttl=255 rtt-time=1.644ms fwd
-time=1.080ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=6 ttl=255 rtt-time=1.670ms fwd
-time=1.307ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=7 ttl=255 rtt-time=1.636ms fwd
-time=0.490ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=8 ttl=255 rtt-time=1.649ms fwd
-time=0.005ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=9 ttl=255 rtt-time=1.401ms fwd
-time=0.685ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=10 ttl=255 rtt-time=1.634ms fwd
-time=0.373ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=11 ttl=255 rtt-time=1.559ms fwd
-time=0.679ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=12 ttl=255 rtt-time=1.666ms fwd
-time=0.880ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=13 ttl=255 rtt-time=1.629ms fwd
-time=0.669ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=14 ttl=255 rtt-time=1.280ms fwd
-time=1.029ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=15 ttl=255 rtt-time=1.458ms fwd
-time=0.268ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=16 ttl=255 rtt-time=1.659ms fwd
-time=0.786ms. rc=4
```

```
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=17 ttl=255 rtt-time=1.636ms fwd
-time=1.071ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=18 ttl=255 rtt-time=1.568ms fwd
-time=2.129ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=19 ttl=255 rtt-time=1.657ms fwd
-time=1.326ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=20 ttl=255 rtt-time=1.762ms fwd
-time=1.335ms. rc=4

---- vxlan-id 2 ip-address 10.20.1.4 PING Statistics ----
20 packets transmitted, 20 packets received, 0.00% packet loss
   20 valid responses, 0 out-of-order, 0 malformed echo responses
   0 send errors, 0 time outs
   0 overlay segment not found, 0 overlay segment not operational
forward-delay min = 0.005ms, avg = 0.939ms, max = 2.350ms, stddev = 0.577ms
round-trip-delay min = 1.280ms, avg = 1.679ms, max = 3.243ms, stddev = 0.375ms



oam vxlan-ping test-id 1 service 1 dest-vni 2 outer-ip-destination 10.20.1.4 send
-count 10 end-system 00:00:00:00:00:01 interval 0.1
TestID 1, Service 1, DestVNI 2, ReplyMode UDP, IFlag Off, PadSize 0, ReflectPad No,
 EndSystemMAC 00:00:00:00:00:01, SendCount 10, Interval 0.1, Timeout 5
Outer: SourceIP 10.20.1.3, SourcePort Dynamic, DestIP 10.20.1.4, TTL 10, FC be, Prof
ile
 In
Inner: DestMAC 00:00:00:00:00:00, SourceIP 10.20.1.3, DestIP 127.0.0.1

2 2 2 2 2 2 2 2 2 2
---- vxlan-id 2 ip-address 10.20.1.4 PING Statistics ----
10 packets transmitted, 10 packets received, 0.00% packet loss
   10 non-errored responses(!), 0 out-of-order(*), 0 malformed echo responses(.)
   0 send errors(.), 0 time outs(.)
   0 overlay segment not found, 0 overlay segment not operational
   0 end-system present(1), 10 end-system not present(2)
forward-delay min = 0.467ms, avg = 0.979ms, max = 1.622ms, stddev = 0.504ms
round-trip-delay min = 1.501ms, avg = 1.597ms, max = 1.781ms, stddev = 0.088ms



oam vxlan-ping test-id 1 service 1 dest-vni 2 outer-ip-destination 10.20.1.4 send
-count 10 end-system 00:00:00:00:00:01
TestID 1, Service 1, DestVNI 2, ReplyMode UDP, IFlag Off, PadSize 0, ReflectPad No,
 EndSystemMAC 00:00:00:00:00:01, SendCount 10, Interval 1, Timeout 5
Outer: SourceIP 10.20.1.3, SourcePort Dynamic, DestIP 10.20.1.4, TTL 10, FC be, Prof
ile
 In
Inner: DestMAC 00:00:00:00:00:00, SourceIP 10.20.1.3, DestIP 127.0.0.1

================================================================================
rc=1 Malformed Echo Request Received, rc=2 Overlay Segment Not Present, rc=3 Overlay
 Segment Not Operational, rc=4 Ok
mac=1 End System Present, mac=2 End System Not Present
================================================================================

92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=1 ttl=255 rtt-time=2.883ms fwd
-time=4.196ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=2 ttl=255 rtt-time=1.596ms fwd
-time=1.536ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=3 ttl=255 rtt-time=1.698ms fwd
-time=0.000ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=4 ttl=255 rtt-time=1.687ms fwd
-time=1.766ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=5 ttl=255 rtt-time=1.679ms fwd
```

```
-time=0.799ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=6 ttl=255 rtt-time=1.678ms fwd
-time=0.000ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=7 ttl=255 rtt-time=1.709ms fwd
-time=0.031ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=8 ttl=255 rtt-time=1.757ms fwd
-time=1.441ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=9 ttl=255 rtt-time=1.613ms fwd
-time=2.570ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=10 ttl=255 rtt-time=1.631ms fwd
-time=2.130ms. rc=4 mac=2

---- vxlan-id 2 ip-address 10.20.1.4 PING Statistics ----
10 packets transmitted, 10 packets received, 0.00% packet loss
   10 valid responses, 0 out-of-order, 0 malformed echo responses
   0 send errors, 0 time outs
   0 overlay segment not found, 0 overlay segment not operational
   0 end-system present, 10 end-system not present
forward-delay min = 0.000ms, avg = 1.396ms, max = 4.196ms, stddev = 1.328ms
round-trip-delay min = 1.596ms, avg = 1.793ms, max = 2.883ms, stddev = 0.366ms
```

6.2.1.6 EVPN-VXLAN routed VPLS multicast routing support

IPv4 and IPv6 multicast routing is supported in an EVPN-VXLAN VPRN and IES routed VPLS service through its IP interface when the source of the multicast stream is on one side of its IP interface and the receivers are on either side of the IP interface. For example, the source for multicast stream G1 could be on the IP side, sending to receivers on both other regular IP interfaces and the VPLS of the routed VPLS service, while the source for group G2 could be on the VPLS side sending to receivers on both the VPLS and IP side of the routed VPLS service. See IPv4 and IPv6 multicast routing support for more details.

6.2.1.7 IGMP and MLD snooping on VXLAN

The delivery of IP multicast in VXLAN services can be optimized with IGMP and MLD snooping. IGMP and MLD snooping are supported in EVPN-VXLAN VPLS services and in EVPN-VXLAN VPRN/IES R-VPLS services. When enabled, IGMP and MLD reports are snooped on SAPs or SDP bindings, but also on VXLAN bindings, to create or modify entries in the MFIB for the VPLS service.

When configuring IGMP and MLD snooping in EVPN-VXLAN VPLS services, consider the following:

- To enable IGMP snooping in the VPLS service on VXLAN, use the **configure service vpls igmp-snooping no shutdown** command.
- To enable MLD snooping in the VPLS service on VXLAN, use the **configure service vpls mld-snooping no shutdown** command.
- The VXLAN bindings only support basic IGMP/MLD snooping functionality. Features configurable under SAPs or SDP bindings are not available for VXLAN (VXLAN bindings are configured with the default values used for SAPs and SDP bindings). By default, a specified VXLAN binding only becomes a dynamic Mrouter when it receives IGMP or MLD queries and adds a specified multicast group to the MFIB when it receives an IGMP or MLD report for that group.

  Alternatively, it is possible to configure all VXLAN bindings for a particular VXLAN instance to be Mrouter ports using the **configure service vpls vxlan igmp-snooping mrouter-port** and **configure service vpls vxlan mld-snooping mrouter-port** commands.
- The **show service id igmp-snooping**, **clear service id igmp-snooping**, **show service id mld-snooping**, and **clear service id mld-snooping** commands are also available for VXLAN bindings.

Note: MLD snooping uses MAC-based forwarding. See MAC-based IPv6 multicast forwarding for more details.

The following CLI commands show how the system displays IGMP snooping information and statistics on VXLAN bindings (the equivalent MLD output is similar).

```
*A:PE1# show service id 1 igmp-snooping port-db vxlan vtep  192.0.2.72 vni 1 detail
===============================================================================
IGMP Snooping VXLAN 192.0.2.72/1 Port-DB for service 1
===============================================================================
-------------------------------------------------------------------------------
IGMP Group 239.0.0.1
-------------------------------------------------------------------------------
Mode             : exclude              Type             : dynamic
Up Time          : 0d 19:07:05          Expires          : 137s
Compat Mode      : IGMP Version 3
V1 Host Expires  : 0s                   V2 Host Expires  : 0s
-------------------------------------------------------
Source Address  Up Time      Expires  Type     Fwd/Blk
-------------------------------------------------------
No sources.
-------------------------------------------------------------------------------
IGMP Group 239.0.0.2
-------------------------------------------------------------------------------
Mode             : include              Type             : dynamic
Up Time          : 0d 19:06:39          Expires          : 0s
Compat Mode      : IGMP Version 3
V1 Host Expires  : 0s                   V2 Host Expires  : 0s
-------------------------------------------------------
Source Address  Up Time      Expires  Type     Fwd/Blk
-------------------------------------------------------
10.0.0.232      0d 19:06:39  137s     dynamic  Fwd
-------------------------------------------------------------------------------
Number of groups: 2
===============================================================================

*A:PE1# show service id 1 igmp-snooping
statistics vxlan vtep  192.0.2.72 vni 1

===============================================================================
IGMP Snooping Statistics for VXLAN 192.0.2.72/1 (service 1)
===============================================================================
Message Type            Received      Transmitted   Forwarded
-------------------------------------------------------------------------------
General Queries         0             0             556
Group Queries           0             0             0
Group-Source Queries    0             0             0
V1 Reports              0             0             0
V2 Reports              0             0             0
V3 Reports              553           0             0
V2 Leaves               0             0             0
Unknown Type            0             N/A           0
-------------------------------------------------------------------------------
Drop Statistics
-------------------------------------------------------------------------------
Bad Length               : 0
Bad IP Checksum          : 0
Bad IGMP Checksum        : 0
Bad Encoding             : 0
No Router Alert          : 0
Zero Source IP           : 0
Wrong Version            : 0
```

```
Lcl-Scope Packets        : 0
Rsvd-Scope Packets       : 0

Send Query Cfg Drops     : 0
Import Policy Drops      : 0
Exceeded Max Num Groups  : 0
Exceeded Max Num Sources : 0
Exceeded Max Num Grp Srcs: 0
MCAC Policy Drops        : 0
===============================================================================
*A:PE1# show service id 1 mfib
===============================================================================
Multicast FIB, Service 1
===============================================================================
Source Address  Group Address         SAP or SDP Id               Svc Id   Fwd/Blk
-------------------------------------------------------------------------------
*               *                     sap:1/1/1:1             Local    Fwd
*               239.0.0.1             sap:1/1/1:1             Local    Fwd
                                      vxlan:192.0.2.72/1      Local    Fwd
10.0.0.232      239.0.0.2             sap:1/1/1:1             Local    Fwd
                                      vxlan:192.0.2.72/1      Local    Fwd
-------------------------------------------------------------------------------
Number of entries: 3
===============================================================================
```

6.2.1.8 PIM snooping on VXLAN

PIM snooping for IPv4 and IPv6 are supported in an EVPN-EVPN-VXLAN VPLS or R-VPLS service (with the R-VPLS attached to a VPRN or IES service). The snooping operation is similar to that within a VPLS service (see PIM snooping for VPLS) and supports both PIM snooping and PIM proxy modes.

PIM snooping for IPv4 is enabled using the **configure service vpls pim-snooping** command.

PIM snooping for IPv6 is enabled using the **configure service vpls pim-snooping no ipv6-multicast-disable** command.

When using PIM snooping for IPv6, the default forwarding is MAC-based with optional support for SG-based (see IPv6 multicast forwarding). SG-based forwarding requires FP3- or higher-based hardware.

It is not possible to configure **max-num-groups** for VXLAN bindings.

6.2.1.9 Static VXLAN termination in Epipe services

By default, the system IP address is used to terminate and generate VXLAN traffic. The following configuration example shows an Epipe service that supports static VXLAN termination:

```
config service epipe 1 name "epipe1" customer 1 create
  sap 1/1/1:1 create
  exit
  vxlan vni 100 create
    egr-vtep 192.0.2.1
      oper-group op-grp-1
    exit
  no shutdown
```

Where:

- **vxlan vni** *vni* **create** specifies the ingress VNI the router uses to identify packets for the service. The following considerations apply:
  - In services that use EVPN, the configured VNI is only used as the ingress VNI to identify packets that belong to the service. Egress VNIs are learned from the BGP EVPN. In the case of Static VXLAN, the configured VNI is also used as egress VNI (because there is no BGP EVPN control plane).
  - The configured VNI is unique in the system, and as a result, it can only be configured in one service (VPLS or Epipe).
- **egr-vtep ip-address** specifies the remote VTEP the router uses when encapsulating frames into VXLAN packets. The following consideration apply:
  - When the PE receives VXLAN packets, the source VTEP is not checked against the configured egress VTEP.
  - The **ip-address** must be present in the global routing table so that the VXLAN destination is operationally up.
- The **oper-group** may be added under **egr-vtep**. The expected behavior for the operational group and service status is as follows:
  - If the **egr-vtep** entry is not present in the routing table, the VXLAN destination (in the **show service id vxlan** command) and the provisioned operational group under **egr-vtep** enters into the operationally down state.
  - If the Epipe SAP goes down, the service goes down, but it is not affected if the VXLAN destination goes down.
  - If the service is **admin shutdown**, then in addition to the SAP, the VXLAN destination and the **oper-group** also enters the operationally down state.

Note: The operational group configured under **egr-vtep** cannot be monitored on the SAP of the Epipe where it is configured.

The following features are not supported by Epipe services with VXLAN destinations:

- per-service hashing
- SDP-binds
- PBB context
- BGP-VPWS
- spoke SDP-FEC
- PW-port

6.2.1.10 Static VXLAN termination in VPLS/R-VPLS services

VXLAN instances in VPLS and R-VPLS can be configured with egress VTEPs. This is referred as static vxlan-instances. The following configuration example shows a VPLS service that supports a static vxlan-instance:

```
config service vpls 1 name "vpls-1" customer 1 create
  sap 1/1/1:1 create
  exit
  vxlan instance 1 vni 100 create
```

```
    source-vtep-security
    no disable-aging /* default: disable-aging
    no disable-learning  /* default: disable-learning
    no discard-unknown-source
    no max-nbr-mac-addr <table-size>
    restrict-protected-src discard-frame
    egr-vtep 192.0.2.1 create
    exit
    egr-vtep 192.0.2.2 create
    exit
  vxlan instance 2 vni 101 create
    egr-vtep 192.0.2.3 create
    exit

 vxlan instance 2 vni 101 create
    egr-vtep 192.0.2.3 create
    exit
  no shutdown
```

Specifically the following can be stated:

- Each VPLS service can have up to two static VXLAN instances. Each instance is an implicit split-horizon-group, and up to 255 static VXLAN binds are supported in total, shared between the two VXLAN instances.
- Single VXLAN instance VPLS services with static VXLAN are supported along with SAPs and SDP bindings. Therefore:
  - VNIs configured in static VXLAN instances are "symmetric", that is, the same ingress and egress VNIs are used for VXLAN packets using that instance. Note that asymmetric VNIs are actually possible in EVPN VXLAN instances.
  - The addresses can be IPv4 or IPv6 (but not a mix within the same service).
  - A specified VXLAN instance can be configured with static egress VTEPs, or be associated with BGP EVPN, but the same instance cannot be configured to support both static and BGP-EVPN based VXLAN bindings.
- Up to two VXLAN instances are supported per VPLS (up to two).
  - When two VXLAN instances are configured in the same VPLS service, any combination of static and BGP-EVPN enabled instances are supported. That is, the two VXLAN instances can be static, or BGP-EVPN enabled, or one of each type.
  - When a service is configured with EVPN and there is a static BGP-EVPN instance in the same service, the user must configure **restrict-protected-src discard-frame** along with no **disable-learning** in the static BGP-EVPN instance, **service**>**vpls**>**vxlan**.
- MAC addresses are learned also on the VXLAN bindings of the static VXLAN instance. Therefore, they are shown in the FDB commands. Note that disable-learning and disable-aging are by default enabled in static vxlan-instance.
  - The learned MAC addresses are subject to the remote-age, and not the local-age (only MACs learned on SAPs use the local-age setting).
  - MAC addresses are learned on a VTEP as long as no disable-learning is configured, and the VXLAN VTEP is present in the base route table. When the VTEP disappears from the route table, the associated MACs are flushed.
- The **vpls vxlan source-vtep-security** command can be configured per VXLAN instance on VPLS services. When enabled, the router performs an IPv4 **source-vtep** lookup to discover if the VXLAN

packet comes from a trusted VTEP. If not, the router discards the frame. If the lookup yields a trusted source VTEP, then the frame is accepted.

- A trusted VTEP is an egress VTEP that has been statically configured, or dynamically learned (through EVPN) in any service, Epipe or VPLS
- The command **show service vxlan** shows the list of trusted VTEPs in the router.
- The command **source-vtep-security** works for static VXLAN instances or BGP-EVPN enabled VXLAN instances, but only for IPv4 VTEPs.
- The command is **mutually exclusive** with assisted-replication (replicator or leaf) in the VNI instance. AR can still be configured in a different instance.

Static VXLAN instances can use non-system IPv4/IPv6 termination.

6.2.1.11 Non-system IPv4 and IPv6 VXLAN termination in VPLS, R-VPLS, and Epipe services

Prerequisites

By default, only VXLAN packets with the same IP destination address as the system IPv4 address of the router can be terminated and processed for a subsequent MAC lookup. A router can simultaneously terminate VXLAN tunnels destined for its system IP address and three additional non-system IPv4 or IPv6 addresses, which can be on the base router or VPRN instances. This section describes the configuration requirements for services to terminate VXLAN packets destined for a non-system loopback IPv4 or IPv6 address on the base router or VPRN.

About this task

Perform the following steps to configure a service with non-system IPv4 or IPv6 VXLAN termination:

Procedure

Step 1. Create the FPE (see FPE creation)

Step 2. Associate the FPE with VXLAN termination (see FPE association with VXLAN termination)

Step 3. Configure the router loopback interface (see VXLAN router loopback interface)

Step 4. Configure VXLAN termination (non-system) VTEP addresses (see VXLAN termination VTEP addresses)

Step 5. Add the service configuration (see VXLAN services)

What to do next

The following actions must be considered when the aforementioned steps are completed.

- **FPE creation**

  A Forwarding Path Extension (FPE) is required to terminate non-system IPv4 or IPv6 VXLAN tunnels.

  In a non-system IPv4 VXLAN termination, the FPE function is used for additional processing required at ingress (VXLAN tunnel termination) only, and not at egress (VXLAN tunnel origination).

  If the IPv6 VXLAN terminates on a VPLS or Epipe service, the FPE function is used at ingress only, and not at egress.

For R-VPLS services terminating IPv6 VXLAN tunnels and also for VPRN VTEPs, the FPE is used for the egress as well as the VXLAN termination function. In the case of R-VPLS, an internal static SDP is created to allow the required extra processing.

For information about FPE configuration and functions, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*, "*Forwarding Path Extension*".

- **FPE association with VXLAN termination**

  The FPE must be associated with the VXLAN termination application. The following example configuration shows two FPEs and their corresponding association. FPE 1 uses the base router and FPE 2 is configured for VXLAN termination on VPRN 10.

```
configure
   fwd-path-ext
       fpe 1 create
             path pxc pxc-1
             vxlan-termination
       fpe 2 create
             path pxc pxc-2
             vxlan-termination router 10
```

- **VXLAN router loopback interface**

  Create the interface that terminates and originates the VXLAN packets. The interface is created as a router interface, which is added to the Interior Gateway Protocol (IGP) and used by the BGP as the EVPN NLRI next hop.

  Because the system cannot terminate the VXLAN on a local interface address, a subnet must be assigned to the loopback interface and not a host IP address that is /32 or /128. In the following example, all the addresses in subnet 11.11.11.0/24 (except 11.11.11.1, which is the interface IP) and subnet 10.1.1.0/24 (except 10.1.1.1) can be used for tunnel termination. The subnet is advertised using the IGP and is configured on either the base router or a VPRN. In the example, two subnets are assigned, in the base router and VPRN 10 respectively.

```
configure
  router
       interface "lo1"
            loopback
            address 10.11.11.1/24
       isis
            interface "lo1"
                 passive
                 no shutdown
```

```
configure
    service
        vprn 10 name "vprn10" customer 1 create
            interface "lo1"
                loopback
                address 10.1.1.1/24
            isis
                interface "lo1"
                    passive
                    no shutdown
```

A local interface address cannot be configured as a VXLAN tunnel-termination IP address in the CLI, as shown in the following example.

```
*A:PE-3# configure service system vxlan tunnel-termination 192.0.2.3 fpe 1 create
MINOR: SVCMGR #8353 VXLAN Tunnel termination IP address cannot be configured -
 IP address in use by another application or matches a local interface IP address
```

The subnet can be up to 31 bits. For example, to use 10.11.11.1 as the VXLAN termination address, the subnet should be configured and advertised as shown in the following example configuration.

```
interface "lo1"
           address 10.11.11.0/31
           loopback
           no shutdown
       exit
       isis 0
           interface "lo1"
               passive
               no shutdown
           exit
           no shutdown
       exit
```

It is not a requirement for the remote PEs and NVEs to have the specific /32 or /128 IP address in their RTM to resolve the BGP EVPN NLRI next hop or forward the VXLAN packets. An RTM with a subnet that contains the remote VTEP can also perform these tasks.

Note: The system does not check for a pre-existing local base router loopback interface with a subnet corresponding to the VXLAN tunnel termination address. If a tunnel termination address is configured and the FPE is operationally up, the system starts terminating VXLAN traffic and responding ICMP messages for that address. The following conditions are ignored in this scenario:

- the presence of a loopback interface in the base router
- the presence of an interface with the address contained in the configured subnet, and no loopback

The following example output includes an IPv6 address in the base router. It could also be configured in a VPRN instance.

```
configure
  router
      interface "lo1"
           loopback
           address 10.11.11.1/24
           ipv6
               address 2001:db8::/127
           exit
      isis
           interface "lo1"
               passive
               no shutdown
```

- **VXLAN termination VTEP addresses**

  The **service**>**system**>**vxlan**>**tunnel-termination** context allows the user to configure non-system IP addresses that can terminate the VXLAN and their corresponding FPEs.

As shown in the following example, an IP address may be associated with a new or existing FPE already terminating the VXLAN. The list of addresses that can terminate the VXLAN can include IPv4 and IPv6 addresses.

```
config service system vxlan#
       tunnel-termination 10.11.11.1 fpe 1 create
       tunnel-termination 2001:db8:1000::1 fpe 1 create

config service vprn 10 vxlan#
       tunnel-termination 10.1.1.2 fpe 2 create
```

The **tunnel-termination** command creates internal loopback interfaces that can respond to ICMP requests. In the following sample output, an internal loopback is created when the tunnel termination address is added (for 10.11.11.1 and 2001:db8:1000::1). The internal FPE router interfaces created by the VXLAN termination function are also shown in the output. Similar loopback and interfaces are created for tunnel termination addresses in a VPRN (not shown).

```
*A:PE1# show router interface
===============================================================================
Interface Table (Router: Base)
===============================================================================
Interface-Name                   Adm       Opr(v4/v6)  Mode    Port/SapId
   IP-Address                                                  PfxState
-------------------------------------------------------------------------------
_tmnx_fpe_1.a                    Up        Up/Up       Network pxc-2.a:1
   fe80::100/64                                                PREFERRED
_tmnx_fpe_1.b                    Up        Up/Up       Network pxc-2.b:1
   fe80::101/64                                                PREFERRED
_tmnx_vli_vxlan_1_131075         Up        Up/Up       Network loopback
   10.11.11.1/32                                               n/a
   2001:db8:1000::1                                            PREFERRED
   fe80::6cfb:ffff:fe00:0/64                                   PREFERRED
lo1                              Up        Up/Down     Network loopback
   10.11.11.0/31                                               n/a
system                           Up        Up/Down     Network system
   1.1.1.1/32                                                  n/a
<snip>
```

- **VXLAN services**

  By default, the VXLAN services use the system IP address as the source VTEP of the VXLAN encapsulated frames. The **vxlan-src-vtep** command in the **config**>**service**>**vpls** or **config**>**service**>**epipe** context enables the system to use a non-system IPv4 or IPv6 address as the source VTEP for the VXLAN tunnels in that service.

  A different **vxlan-src-vtep** can be used for different services, as shown in the following example where two different services use different non-system IP addresses as source VTEPs.

```
configure service vpls 1
   vxlan-src-vtep 10.11.11.1

configure service vpls 2
   vxlan-src-vtep 2001:db8:1000::1
```

  In addition, if a **vxlan-src-vtep** is configured and the service uses EVPN, the IP address is also used to set the BGP NLRI next hop in EVPN route advertisements for the service.

Note: The BGP EVPN next hop can be overridden by the use of export policies based on the following rules:

- A BGP peer policy can override a next hop pushed by the **vxlan-src-vtep** configuration.
- If the VPLS service is IPv6 (that is, the **vxlan-src-vtep** is IPv6) and a BGP peer export policy is configured with **next-hop-self**, the BGP next-hop is overridden with an IPv6 address auto-derived from the IP address of the system. The auto-derivation is based on RFC 4291. For example, ::ffff:10.20.1.3 is auto-derived from system IP 10.20.1.3.
- The policy checks the address type of the next hop provided by the **vxlan-src-vtep** command. If the command provides an IPv6 next hop, the policy is unable use an IPv4 address to override the IPv6 address provided by the **vxlan-src-vtep** command.

After the preceding steps are performed to configure a VXLAN termination, the VPLS, R-VPLS, or Epipe service can be used normally, except that the service terminates VXLAN tunnels with a non-system IPv4 or IPv6 destination address (in the base router or a VPRN instance) instead of the system IP address only.

The FPE **vxlan-termination** function creates internal router interfaces and loopbacks that are displayed by the **show** commands. When configuring IPv6 VXLAN termination on an R-VPLS service, as well as the internal router interfaces and loopbacks, the system creates internal SDP bindings for the required egress processing. The following output shows an example of an internal FPE-type SDP binding created for IPv6 R-VPLS egress processing.

```
*A:PE1# show service sdp-using
===============================================================================
SDP Using
===============================================================================
SvcId      SdpId              Type   Far End              Opr   I.Label E.Label
                                                         State
-------------------------------------------------------------------------------
2002       17407:2002         Fpe    fpe_1.b              Up    262138  262138
-------------------------------------------------------------------------------
Number of SDPs : 1
-------------------------------------------------------------------------------
===============================================================================
```

When BGP EVPN is used, the BGP peer over which the EVPN-VXLAN updates are received can be an IPv4 or IPv6 peer, regardless of whether the next-hop is an IPv4 or IPv6 address.

The same VXLAN tunnel termination address cannot be configured on different router instances; that is, on two different VPRN instances or on a VPRN and the base router.

6.2.2 EVPN for overlay tunnels

This section describes the specifics of EVPN for non-MPLS Overlay tunnels.

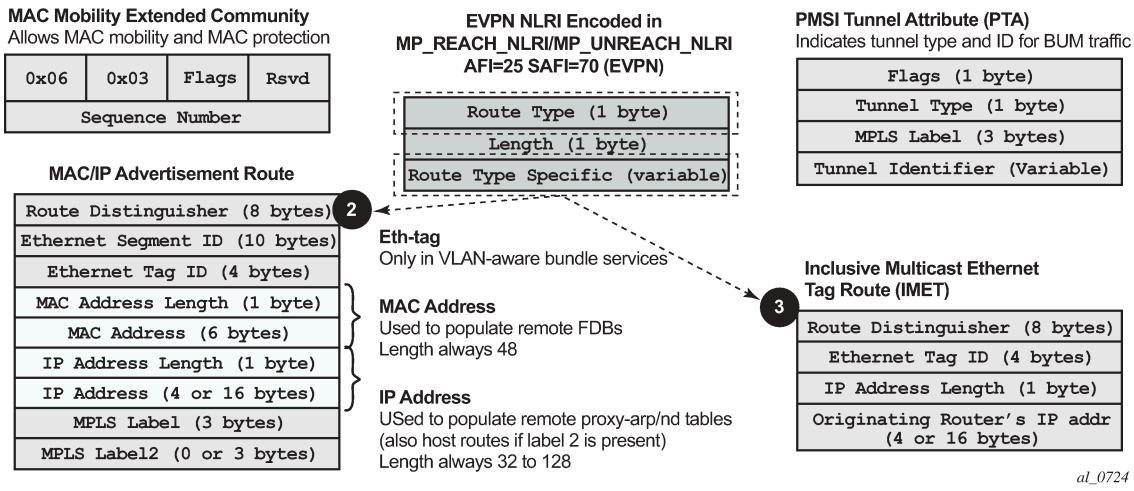
6.2.2.1 BGP-EVPN control plane for VXLAN overlay tunnels

RFC 8365 describes EVPN as the control plane for overlay-based networks. The 7750 SR, 7450 ESS, and 7950 XRS support all routes and features described in RFC 7432 that are required for the DGW function. EVPN multihoming and BGP multihoming based on the L2VPN BGP address family are both supported if redundancy is needed.

Figure 142: EVPN-VXLAN required routes and communities shows the EVPN MP-BGP NLRI, required attributes and extended communities, and two route types supported for the DGW Layer 2 applications:

route type 3	Inclusive Multicast Ethernet Tag route
route type 2	MAC/IP advertisement route

Figure 142: EVPN-VXLAN required routes and communities



EVPN route type 3 – inclusive multicast Ethernet tag route

Route type 3 is used to set up the flooding tree (BUM flooding) for a specified VPLS service in the data center. The received inclusive multicast routes add entries to the VPLS flood list in the 7750 SR, 7450 ESS, and 7950 XRS. The tunnel types supported in an EVPN route type 3 when BGP-EVPN MPLS is enabled are ingress replication, P2MP MLDP, and composite tunnels.

Ingress Replication (IR) and Assisted Replication (AR) are supported for VXLAN tunnels. See Layer 2 multicast optimization for VXLAN (Assisted-Replication) for more information about the AR.

If **ingress-repl-inc-mcast-advertisement** is enabled, a route type 3 is generated by the router per VPLS service as soon as the service is in an operationally up state. The following fields and values are used:

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context.

  Note: The RD can be configured or derived from the **bgp-evpn evi** value.

- Ethernet Tag ID is 0.
- IP address length is always 32.
- Originating router's IP address carries an IPv4 or IPv6 address.

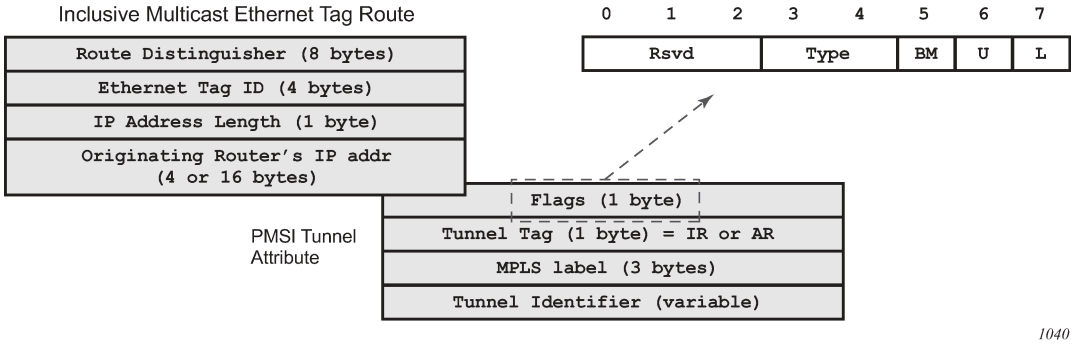
  Note: By default, the IP address of the Originating router is derived from the system IP address. However, this can be overridden by the **configure service vpls bgp-evpn incl-mcast-orig-ip ip-address** command for the Ingress Replication (and mLDP if MPLS is used) tunnel type.

- For PMSI Tunnel Attribute (PTA), tunnel type = Ingress replication (6) or Assisted Replication (10)

– Leaf not required for Flags.
– MPLS label carries the VNI configured in the VPLS service. Only one VNI can be configured per VPLS service.
– Tunnel endpoint is equal to the system IP address.

As shown in Figure 143: PMSI attribute flags field for AR, additional flags are used in the PTA when the service is configured for AR.

Figure 143: PMSI attribute flags field for AR



The Flags field is defined as a Type field (for AR) with two new flags that are defined as follows:

– T is the AR Type field (2 bits):
  • 00 (decimal 0) = RNVE (non-AR support)
  • 01 (decimal 1) = AR REPLICATOR
  • 10 (decimal 2) = AR LEAF
– The U and BM flags defined in IETF Draft *draft-ietf-bess-evpn-optimized-ir* are not used in the SR OS.

Table 19: AR-R and AR-L routes and usage describes the inclusive multicast route information sent per VPLS service when the router is configured as **assisted-replication replicator** (AR-R) or **assisted-replication leaf** (AR-L). A Regular Network Virtualization Edge device (RNVE) is defined as an EVPN-VXLAN router that does not support (or is not configured for) Assisted-Replication.

Note: For AR-R, two inclusive multicast routes may be advertised if **ingress-repl-inc-mcast-advertisement** is enabled: a route with tunnel-type IR, tunnel-id = IR IP (generally system-ip) and a route with tunnel-type AR, tunnel-id = AR IP (the address configured in the **assisted-replication-ip** command).

Table 19: AR-R and AR-L routes and usage

AR role	Function	Inclusive Mcast routes advertisement
AR-R	Assists AR-LEAFs	– IR included in the Mcast route (uses IR IP) if **ingress-repl-inc-mcast-advertisement** is enabled – AR included in the Mcast route (uses AR IP, tunnel type=AR, T=1)

AR role	Function	Inclusive Mcast routes advertisement
AR-LEAF	Sends BM only to AR-Rs	IR inclusive multicast route (IR IP, T=2) if **ingress-repl-inc-mcast-advertisement** is enabled
RNVE	Non-AR support	IR inclusive multicast route (IR IP) if **ingress-repl-inc-mcast-advertisement** is enabled

EVPN route type 2 – MAC/IP advertisement route

The 7750 SR, 7450 ESS, and 7950 XRS generates this route type for advertising MAC addresses. If mac-advertisement is enabled, the router generates MAC advertisement routes for the following:

- learned MACs on SAPs or SDP bindings
- conditional static MACs

Note: To address unknown-mac-routes, if unknown-mac-route is enabled, there is no bgp-mh site in the service or there is a (single) DF site

The route type 2 generated by a router uses the following fields and values:

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context.

  Note: The RD can be configured or derived from the **bgp-evpn evi** value.

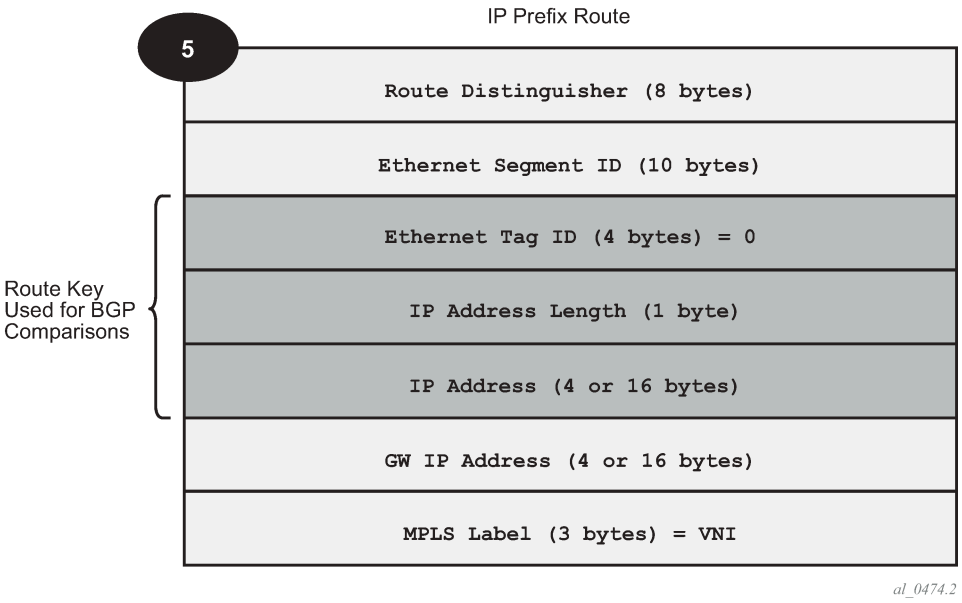
- Ethernet Segment Identifier (ESI) value = 0:0:0:0:0:0:0:0:0:0 or non-zero, depending on whether the MAC addresses are learned on an Ethernet Segment.
- Ethernet Tag ID is 0.
- MAC address length is always 48.
- MAC Address:
  - is 00:00:00:00:00:00 for the Unknown MAC route address.
  - is different from 00:…:00 for the rest of the advertised MACs.
- IP address and IP address length:
  - The length of the IP address associated with the MAC being advertised is either 32 for IPv4 or 128 for IPv6.
  - If the MAC address is the Unknown MAC route, the IP address length is zero and the IP omitted.
  - In general, any MAC route without IP has IPL=0 (IP length) and the IP is omitted.
  - When received, any IPL value not equal to zero, 32, or 128 discards the route.
- MPLS Label 1 carries the VNI configured in the VPLS service. Only one VNI can be configured per VPLS.
- MPLS Label 2 is 0.
- MAC Mobility extended community is used for signaling the sequence number in case of MAC moves and the sticky bit in case of advertising conditional static MACs. If a MAC route is received with a MAC mobility **ext-community**, the sequence number and the sticky bit are considered for the route selection.

When EVPN-VXLAN multihoming is enabled, type 1 routes (Auto-Discovery per-ES and per-EVI routes) and type 4 routes (ES routes) are also generated and processed. See BGP-EVPN control plane for MPLS tunnels for more information about route types 1 and 4.

EVPN route type 5 – IP prefix route

Figure 144: EVPN route-type 5 shows the IP prefix route or route-type 5.

Figure 144: EVPN route-type 5



The router generates this route type for advertising IP prefixes in EVPN. The router generates IP prefix advertisement routes for IP prefixes existing in a VPRN linked to the IRB backhaul R-VPLS service.

The route-type 5 generated by a router uses the following fields and values:

- Route Distinguisher: taken from the RD configured in the IRB backhaul R-VPLS service within the BGP context
- Ethernet Segment Identifier (ESI): value = 0:0:0:0:0:0:0:0:0:0
- Ethernet Tag ID: 0
- IP address length: any value in the 0 to 128 range
- IP address: any valid IPv4 or IPv6 address
- Gateway IP address: can carry two different values:
  - if different from zero, the route-type 5 carries the primary IP interface address of the VPRN behind which the IP prefix is known. This is the case for the regular IRB backhaul R-VPLS model.
  - if 0.0.0.0, the route-type 5 is sent with a MAC next-hop extended community that carries the VPRN interface MAC address. This is the case for the EVPN tunnel R-VPLS model.
- MPLS Label: carries the VNI configured in the VPLS service. Only one VNI can be configured per VPLS service.

All the routes in EVPN-VXLAN is sent with the RFC 5512 tunnel encapsulation extended community, with the tunnel type value set to VXLAN.

6.2.2.2 EVPN for VXLAN in VPLS services

The EVPN-VXLAN service is designed around the current VPLS objects and the additional VXLAN construct.

Figure 135: Layer 2 DC PE with VPLS to the WAN shows a DC with a Layer 2 service that carries the traffic for a tenant who wants to extend a subnet beyond the DC. The DC PE function is carried out by the 7750 SR, 7450 ESS, and 7950 XRS where a VPLS instance exists for that particular tenant. Within the DC, the tenant has VPLS instances in all the Network Virtualization Edge (NVE) devices where they require connectivity (such VPLS instances can be instantiated in TORs, Nuage VRS, VSG, and so on). The VPLS instances in the redundant DGW and the DC NVEs are connected by VXLAN bindings. BGP-EVPN provides the required control plane for such VXLAN connectivity.

The DGW routers are configured with a VPLS per tenant that provides the VXLAN connectivity to the Nuage VPLS instances. On the router, each tenant VPLS instance is configured with:

- The WAN-related parameters (SAPs, spoke SDPs, mesh-SDPs, BGP-AD, and so on).
- The BGP-EVPN and VXLAN (VNI) parameters. The following CLI output shows an example for an EVPN-VXLAN VPLS service.

```
*A:DGW1>config>service>vpls# info
----------------------------------------------
            description "vxlan-service"
            vxlan instance 1 vni 1 create
            exit
            bgp
                route-distinguisher 65001:1
                route-target export target:65000:1 import target:65000:1
            exit
            bgp-evpn
                unknown-mac-route
                mac-advertisement
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            sap 1/1/1:1 create
            exit
            no shutdown
----------------------------------------------
```

The **bgp-evpn** context specifies the encapsulation type (only vxlan is supported) to be used by EVPN and other parameters like the unknown-mac-route and mac-advertisement commands. These commands are typically configured in three different ways:

- If the operator configures **no unknown-mac-route** and **mac-advertisement** (default option), the router advertises new learned MACs (on the SAPs or SDP bindings) or new conditional static MACs.
- If the operator configures **unknown-mac-route** and **no mac-advertisement**, the router only advertises an unknown-mac-route as long as the service is operationally up (if no BGP-MH site is configured in the service) or the router is the DF (if BGP-MH is configured in the service).
- If the operator configures **unknown-mac-route** and **mac-advertisement**, the router advertises new learned MACs, conditional static MACs, and the unknown-mac-route. The unknown-mac-route is only advertised under the preceding described conditions.

Other parameters related to EVPN or VXLAN are:

- MAC duplication parameters

- VXLAN VNI (defines the VNI that the router uses in the EVPN routes generated for the VPLS service)

After the VPLS is configured and operationally up, the router sends or receives inclusive multicast Ethernet Tag routes, and a full-mesh of VXLAN connections are automatically created. These VXLAN "auto-bindings" can be characterized as follows:

- The VXLAN auto-binding model is based on an IP-VPN-like design, where no SDPs or SDP binding objects are created by or visible to the user. The VXLAN auto-binds are composed of remote VTEPs and egress VNIs, and can be displayed with the following command:
  - ```
    show service id 112 vxlan destinations
    ```

    Output example

    ```
    ===============================================================================
    Egress VTEP, VNI (Instance 1)
    ===============================================================================
    VTEP Address                                          Egress VNI Oper Mcast Num
                                                                     State      MACs
    -------------------------------------------------------------------------------
    192.0.2.2                                             112        Up   BUM   1
    192.0.2.3                                             112        Down BUM   0
    -------------------------------------------------------------------------------
    Number of Egress VTEP, VNI : 2
    ===============================================================================
    ```

  - ```
    show service id 112 vxlan destinations detail
    ```

    Output example

    ```
    ===============================================================================
    Egress VTEP, VNI (Instance 1)
    ===============================================================================
    VTEP Address                                          Egress VNI Oper Mcast Num
                                                                     State      MACs
    -------------------------------------------------------------------------------
    192.0.2.2                                             112        Up   BUM   1
    Oper Flags       : None
    Type             : evpn
    L2 PBR           : No
    Sup BCast Domain : No
    Last Update      : 02/03/2023 22:15:06
    192.0.2.3                                             112        Down BUM   0
    Oper Flags       : MTU-Mismatch
    Type             : evpn
    L2 PBR           : No
    Sup BCast Domain : No
    Last Update      : 01/31/2023 21:28:39
    -------------------------------------------------------------------------------
    Number of Egress VTEP, VNI : 2
    ===============================================================================
    ```

- If the following command is configured on the PEs attached to the same service, the service MTU value is advertised in the EVPN Layer-2 Attributes extended community along with the Inclusive Multicast Ethernet Tag routes.
  - **MD-CLI**

    ```
    configure service vpls bgp-evpn routes incl-mcast advertise-l2-attributes
    ```

– **classic CLI**

```
configure service vpls bgp-evpn incl-mcast-l2-attributes-advertisement
```

Upon receiving the signaled MTU from an egress PE, the ingress PE compares the MTU with the local one and, in case of mismatch, the EVPN VXLAN destination is brought operationally down. An operational flag MTU-Mismatch shows the reason why the VXLAN destination is operationally down in this case. The following command makes the router ignore the MTU signaled by the remote PE and bring up the VXLAN destination if there are no other reasons to keep it down.

```
configure service vpls bgp-evpn ignore-mtu-mismatch
```

- The VXLAN bindings observe the VPLS split-horizon rule. This is performed automatically without the need for any split-horizon configuration.
- BGP Next-Hop Tracking for EVPN is fully supported. If the BGP next-hop for a specified received BGP EVPN route disappears from the routing table, the BGP route is not marked as "used" and the respective entry in **show service id vxlan destinations** is removed.

After the flooding domain is setup, the routers and DC NVEs start advertising MAC addresses, and the routers can learn MACs and install them in the FDB. Some considerations are the following:

- All the MAC addresses associated with remote VTEP/VNIs are always learned in the control plane by EVPN. Data plane learning on VXLAN auto-bindings is not supported.
- When **unknown-mac-route** is configured, it is generated when no (BGP-MH) site is configured, or a site is configured AND the site is DF in the PE.

Note: The **unknown-mac-route** is not installed in the FDB (therefore, does not show up in the **show service id svc-id fdb detail** command).

- While the router can be configured with only one VNI (and signals a single VNI per VPLS), it can accept any VNI in the received EVPN routes as long as the route target is properly imported. The VTEPs and VNIs show up in the FDB associated with MAC addresses:

```
A:PE65# show service id 1000 fdb detail
===============================================================================
Forwarding Database, Service 1000
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1000      00:00:00:00:00:01 vxlan-1:                 Evpn     10/05/13 23:25:57
                            192.0.2.63:1063
1000      00:00:00:00:00:65 sap:1/1/1:1000           L/30     10/05/13 23:25:57
1000      00:ca:ca:ca:ca:00 vxlan-1:                 EvpnS    10/04/13 17:35:43
                            192.0.2.63:1063
-------------------------------------------------------------------------------
No. of MAC Entries: 3
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
===============================================================================
```

6.2.2.2.1 Resiliency and BGP multihoming

The DC overlay infrastructure relies on IP tunneling, that is, VXLAN; therefore, the underlay IP layer resolves failure in the DC core. The IGP should be optimized to get the fastest convergence.

From a service perspective, resilient connectivity to the WAN may be provided by BGP multihoming.

6.2.2.2.2 Use of BGP-EVPN, BGP-AD, and sites in the same VPLS service

All BGP-EVPN (control plane for a VXLAN DC), BGP-AD (control plane for MPLS-based spoke SDPs connected to the WAN), and one site for BGP multihoming (control plane for the multihomed connection to the WAN) can be configured in one service in a specified system. If that is the case, the following considerations apply:

- The configured BGP route-distinguisher and route-target are used by BGP for the two families, that is, evpn and l2vpn. If different import/export route targets are to be used per family, vsi-import/export policies must be used.
- The pw-template-binding command under BGP, does not have any effect on evpn or bgp-mh. It is only used for the instantiation of the BGP-AD spoke SDPs.
- If the same import/export route-targets are used in the two redundant DGWs, VXLAN binding as well as a fec129 spoke SDP binding is established between the two DGWs, creating a loop. To avoid creating a loop, the router allows the establishment of an EVPN VXLAN binding and an SDP binding to the same far-end, but the SDP binding is kept operationally down. Only the VXLAN binding is operationally up.

6.2.2.2.3 Use of the unknown-mac-route

This section describes the behavior of the EVPN-VXLAN service in the router when the unknown-mac-route and BGP-MH are configured at the same time.

The use of EVPN, as the control plane of NVO networks in the DC, provides a significant number of benefits as described in IETF Draft *draft-ietf-bess-evpn-overlay*.

However, there is a potential issue that must be addressed when a VPLS DCI is used for an NVO3-based DC: all the MAC addresses learned from the WAN side of the VPLS must be advertised by BGP EVPN updates. Even if optimized BGP techniques like RT-constraint are used, the number of MAC addresses to advertise or withdraw (in case of failure) from the DC GWs can be difficult to control and overwhelming for the DC network, especially when the NVEs reside in the hypervisors.

The 7750 SR, 7450 ESS, and 7950 XRS solution to this issue is based on the use of an unknown-mac-route address that is advertised by the DC PEs. By using this unknown-mac-route advertisement, the DC tenant may decide to optionally turn off the advertisement of WAN MAC addresses in the DGW, therefore, reducing the control plane overhead and the size of the FDB tables in the NVEs.

The use of the unknown-mac-route is optional and helps to reduce the amount of unknown-unicast traffic within the data center. All the receiving NVEs supporting this concept send any unknown-unicast packet to the owner of the unknown-mac-route, as opposed to flooding the unknown-unicast traffic to all other NVEs that are part of the same VPLS.

Note: Although the router can be configured to generate and advertise the unknown-mac-route, the router never honors the unknown-mac-route and floods to the TLS-flood list when an unknown-unicast packet arrives at an ingress SAP or SDP binding.

The use of the unknown-mac-route assumes the following:

- A fully virtualized DC where all the MACs are control-plane learned, and learned previous to any communication (no legacy TORs or VLAN connected servers).
- The only exception is MACs learned over the SAPs/SDP bindings that are part of the BGP-MH WAN site-id. Only one site-id is supported in this case.
- No other SAPs/SDP bindings out of the WAN site-id are supported, unless only static MACs are used on those SAPs/SDP bindings.

Therefore, when unknown-mac-route is configured, it is only generated when one of the following applies:

- No site is configured and the service is operationally up.
- A BGP-MH site is configured AND the DGW is Designated Forwarder (DF) for the site. In case of BGP-MH failover, the unknown-mac-route is withdrawn by the former DF and advertised by the new DF.

6.2.2.3 EVPN for VXLAN in R-VPLS services

Figure 136: Gateway IRB on the DC PE for an L2 EVPN/VXLAN DC shows a DC with a Layer 2 service that carries the traffic for a tenant who extends a subnet within the DC, while the DGW is the default gateway for all the hosts in the subnet. The DGW function is carried out by the 7750 SR, 7450 ESS, and 7950 XRS where an R-VPLS instance exists for that particular tenant. Within the DC, the tenant has VPLS instances in all the NVE devices where they require connectivity (such VPLS instances can be instantiated in TORs, Nuage VRS, VSG, and so on). The WAN connectivity is based on existing IP-VPN features.

In this model, the DGW routers are configured with a R-VPLS (bound to the VPRN that provides the WAN connectivity) per tenant that provides the VXLAN connectivity to the Nuage VPLS instances. This model provides inter-subnet forwarding for L2-only TORs and other L2 DC NVEs.

On the router:

- The VPRN is configured with an interface bound to the backhaul R-VPLS. That interface is a regular IP interface (IP address configured or possibly a Link Local Address if IPv6 is added).
- The VPRN can support other numbered interfaces to the WAN or even to the DC.
- The R-VPLS is configured with the BGP, BGP-EVPN and VXLAN (VNI) parameters.

The Nuage VSGs and NVEs use a regular VPLS service model with BGP EVPN and VXLAN parameters.

Consider the following:

- Route-type 2 routes with MACs and IPs are advertised. Some considerations about MAC+IP and ARP/ND entries are:
  - The 7750 SR advertises its IRB MAC+IP in a route type 2 route and possibly the VRRP vMAC+vIP if it runs VRRP and the 7750 SR is the active router. In both cases, the MACs are advertised as static MACs, therefore, protected by the receiving PEs.
  - If the 7750 SR VPRN interface is configured with one or more additional secondary IP addresses, they are all advertised in routes type 2, as static MACs.
  - The 7750 SR processes route-type 2 routes as usual, populating the FDB with the received MACs and the VPRN ARP/ND table with the MAC and IPs, respectively.

Note: ND entries received from the EVPN are installed as Router entries. The ARP/ND entries coming from the EVPN are tagged as **evpn**.

  - When a VPLS containing proxy-ARP/proxy-ND entries is bound to a VPRN (allow-ip-int-bind) all the proxy-ARP/proxy-ND entries are moved to the VPRN ARP/ND table. ARP/ND entries are also moved to proxy-ARP/proxy-ND entries if the VPLS is unbound.
  - EVPN does not program EVPN-received ARP/ND entries if the receiving VPRN has no IP addresses for the same subnet. The entries are added when the IP address for the same subnet is added.
  - Static ARP/ND entries have precedence over dynamic and EVPN ARP/ND entries.
- VPRN interface binding to VPLS service brings down the VPRN interface operational status, if the VPRN interface MAC or the VRRP MAC matches a static-mac or OAM MAC configured in the associated VPLS service. If that is the case, a trap is generated.
- Redundancy is handled by VRRP. The active 7750 SR advertises vMAC and vIP, as discussed, including the MAC mobility extended community and the sticky bit.

EVPN-enabled R-VPLS services are also supported on IES interfaces.

6.2.2.3.1 EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes

Figure 137: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC shows a Layer 3 DC model, where a VPRN is defined in the DGWs, connecting the tenant to the WAN. That VPRN instance is connected to the VPRNs in the NVEs by means of an IRB backhaul R-VPLS. Because the IRB backhaul R-VPLS provides connectivity only to all the IRB interfaces and the DGW VPRN is not directly connected to all the tenant subnets, the WAN ip-prefixes in the VPRN routing table must be advertised in EVPN. In the same way, the NVEs send IP prefixes in EVPN that are received by the DGW and imported in the VPRN routing table.

Note: To generate or process IP prefixes sent or received in EVPN route type 5, support for IP route advertisement must be enabled in BGP-EVPN using the **bgp-evpn ip-route-advertisement** command. This command is disabled by default and must be explicitly enabled. The command is tied to the **allow-ip-int-bind** command required for R-VPLS, and it is not supported on an R-VPLS linked to IES services.

Local router interface host addresses are not advertised in EVPN by default. To advertise them, the **ip-route-advertisement incl-host** command must be enabled. For example:

```
===============================================================================
Route Table (Service: 2)
===============================================================================
Dest Prefix[Flags]                            Type    Proto     Age        Pref
      Next Hop[Interface Name]                         Active     Metric
-------------------------------------------------------------------------------
10.1.1.0/24                                   Local   Local     00h00m11s  0
       if                                              Y           0
10.1.1.100/32                                 Local   Host      00h00m11s  0
       if                                              Y           0
===============================================================================
```

For the case displayed by the output above, the behavior is the following:

- **ip-route-advertisement** only local subnet (default) - 10.1.1.0/24 is advertised
- **ip-route-advertisement incl-host** local subnet, host - 10.1.1.0/24 and 10.1.1.100/32 are advertised

Below is an example of VPRN (500) with two IRB interfaces connected to backhaul R-VPLS services 501 and 502 where EVPN-VXLAN runs:

```
vprn 500 customer 1 create
            ecmp 4
            route-distinguisher 65072:500
            vrf-target target:65000:500
            interface "evi-502" create
                address 10.20.20.72/24
                vpls "evpn-vxlan-502"
                exit
            exit
            interface "evi-501" create
                address 10.10.10.72/24
                vpls "evpn-vxlan-501"
                exit
            exit
            no shutdown
vpls 501 name “evpn-vxlan-501” customer 1 create
            allow-ip-int-bind
            vxlan instance 1 vni 501 create
            exit
            bgp
                route-distinguisher 65072:501
                route-target export target:65000:501 import target:65000:501
            exit
            bgp-evpn
                ip-route-advertisement incl-host
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            no shutdown
        exit
vpls 502 name “evpn-xvlan-502” customer 1 create
            allow-ip-int-bind
            vxlan instance 1 vni 502 create
            exit
            bgp
                route-distinguisher 65072:502
                route-target export target:65000:502 import target:65000:502
            exit
            bgp-evpn
                ip-route-advertisement incl-host
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            no shutdown
        exit
```

When the above commands are enabled, the router behaves as follows:

- Receive route-type 5 routes and import the IP prefixes and associated IP next-hops into the VPRN routing table.
  - If the route-type 5 is successfully imported by the router, the prefix included in the route-type 5 (for example, 10.0.0.0/24), is added to the VPRN routing table with a next-hop equal to the gateway IP included in the route (for example, 192.0.0.1. that refers to the IRB IP address of the remote VPRN behind which the IP prefix sits).

– When the router receives a packet from the WAN to the 10.0.0.0/24 subnet, the IP lookup on the VPRN routing table yields 192.0.0.1 as the next-hop. That next-hop is resolved to a MAC in the ARP table and the MAC resolved to a VXLAN tunnel in the FDB table

Note: IRB MAC and IP addresses are advertised in the IRB backhaul R-VPLS in routes type 2.

- Generate route-type 5 routes for the IP prefixes in the associated VPRN routing table.

  For example, if VPRN-1 is attached to EVPN R-VPLS 1 and EVPN R-VPLS 2, and R-VPLS 2 has **bgp-evpn ip-route-advertisement** configured, the 7750 SR advertises the R-VPLS 1 interface subnet in one route-type 5.
- Routing policies can filter the imported and exported IP prefix routes accordingly.

The VPRN routing table can receive routes from all the supported protocols (BGP-VPN, OSPF, IS-IS, RIP, static routing) as well as from IP prefixes from EVPN, as shown below:

```
*A:PE72# show router 500 route-table
===============================================================================
Route Table (Service: 500)
===============================================================================
Dest Prefix[Flags]                            Type    Proto     Age        Pref
      Next Hop[Interface Name]                                    Metric
-------------------------------------------------------------------------------
10.20.20.0/24                                 Local   Local     01d11h10m  0
       evi-502                                                    0
10.20.20.71/32                                Remote  BGP EVPN  00h02m26s  169
       10.10.10.71                                                0
10.10.10.0/24                                Remote  Static    00h00m05s  5
       10.10.10.71                                                1
10.16.0.1/32                                 Remote  BGP EVPN  00h02m26s  169
       10.10.10.71                                                0
-------------------------------------------------------------------------------
No. of Routes: 4
```

The following considerations apply:

- The route Preference for EVPN IP prefixes is 169.

  BGP IP-VPN routes have a preference of 170 by default, therefore, if the same route is received from the WAN over BGP-VPRN and from BGP-EVPN, then the EVPN route is preferred.
- When the same route-type 5 prefix is received from different gateway IPs, ECMP is supported if configured in the VPRN.
- All routes in the VPRN routing table (as long as they do not point back to the EVPN R-VPLS interface) are advertised via EVPN.

Although the description above is focused on IPv4 interfaces and prefixes, it applies to IPv6 interfaces too. The following considerations are specific to IPv6 VPRN R-VPLS interfaces:

- IPv4 and IPv6 interfaces can be defined on R-VPLS IP interfaces at the same time (dual-stack).
- The user may configure specific IPv6 Global Addresses on the VPRN R-VPLS interfaces. If a specific Global IPv6 Address is not configured on the interface, the Link Local Address interface MAC/IP is advertised in a route type 2 as soon as IPv6 is enabled on the VPRN R-VPLS interface.
- Routes type 5 for IPv6 prefixes are advertised using either the configured Global Address or the implicit Link Local Address (if no Global Address is configured).

If more than one Global Address is configured, normally the first IPv6 address is used as gateway IP. The "first IPv6 address" refers to the first one on the list of IPv6 addresses shown through **show router** *id* **interface** *interface* **ipv6** or through SNMP.

The rest of the addresses are advertised only in MAC-IP routes (Route Type 2) but not used as gateway IP for IPv6 prefix routes.

6.2.2.3.2 EVPN for VXLAN in EVPN tunnel R-VPLS services

Figure 138: EVPN-tunnel gateway IRB on the DC PE for an L3 EVPN/VXLAN DC shows an L3 connectivity model that optimizes the solution described in EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes. Instead of regular IRB backhaul R-VPLS services for the connectivity of all the VPRN IRB interfaces, EVPN tunnels can be configured. The main advantage of using EVPN tunnels is that they do not need the configuration of IP addresses, as regular IRB R-VPLS interfaces do.

In addition to the **ip-route-advertisement** command, this model requires the configuration of the **config**>**service**>**vprn**>**if**>**vpls** <**name**> **evpn-tunnel**.

Note: EVPN tunnels can be enabled independently of the **ip-route-advertisement** command, however, no route-type 5 advertisements are sent or processed. Neither command, **evpn-tunnel** and **ip-route-advertisement**, is supported on R-VPLS services linked to IES interfaces.

The example below shows a VPRN (500) with an EVPN-tunnel R-VPLS (504):

```
vprn 500 name "vprn500" customer 1 create
            ecmp 4
            route-distinguisher 65071:500
            vrf-target target:65000:500
            interface "evi-504" create
                vpls "evpn-vxlan-504"
                    evpn-tunnel
                exit
            exit
            no shutdown
        exit
        vpls 504 name "evpn-vxlan-504" customer 1 create
            allow-ip-int-bind
            vxlan instance 1 vni 504 create
            exit
            bgp
                route-distinguisher 65071:504
                route-target export target:65000:504 import target:65000:504
            exit
            bgp-evpn
                ip-route-advertisement
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            no shutdown
        exit
```

A specified VPRN supports regular IRB backhaul R-VPLS services as well as EVPN tunnel R-VPLS services.

Note: EVPN tunnel R-VPLS services do not support SAPs or SDP-binds.

The process followed upon receiving a route-type 5 on a regular IRB R-VPLS interface differs from the one for an EVPN-tunnel type:

- IRB backhaul R-VPLS VPRN interface:
  - When a route-type 2 that includes an IP prefix is received and it becomes active, the MAC/IP information is added to the FDB and ARP tables. This can be checked with the **show router arp** command and the **show service id fdb detail** command.
  - When route-type 5 is received and becomes active for the R-VPLS service, the IP prefix is added to the VPRN routing table, regardless of the existence of a route-type 2 that can resolve the gateway IP address. If a packet is received from the WAN side and the IP lookup hits an entry for which the gateway IP (IP next-hop) does not have an active ARP entry, the system uses ARP to get a MAC. If ARP is resolved but the MAC is unknown in the FDB table, the system floods into the TLS multicast list. Routes type 5 can be checked in the routing table with the **show router route-table** and **show router fib** commands.
- EVPN tunnel R-VPLS VPRN interface:
  - When route-type 2 is received and becomes active, the MAC address is added to the FDB (only).
  - When a route-type 5 is received and active, the IP prefix is added to the VPRN routing table with next-hop equal to EVPN tunnel: GW-MAC.

    For example, ET-d8:45:ff:00:01:35, where the GW-MAC is added from the GW-MAC extended community sent along with the route-type 5.

    If a packet is received from the WAN side, and the IP lookup hits an entry for which the next-hop is a EVPN tunnel: GW-MAC, the system looks up the GW-MAC in the FDB. Usually a route-type 2 with the GW-MAC is previously received so that the GW-MAC can be added to the FDB. If the GW-MAC is not present in the FDB, the packet is dropped.
  - IP prefixes with GW-MACs as next-hops are displayed by the show router command, as shown below:

```
*A:PE71# show router 500 route-table
===============================================================================
Route Table (Service: 500)
===============================================================================
Dest Prefix[Flags]                              Type    Proto     Age        Pref
      Next Hop[Interface Name]                                    Metric
-------------------------------------------------------------------------------
10.20.20.72/32                                  Remote  BGP EVPN  00h23m50s  169
       10.10.10.72                                                  0
10.30.30.0/24                                   Remote  BGP EVPN  01d11h30m  169
       evi-504 (ET-d8:45:ff:00:01:35)                               0
10.10.10.0/24                                  Remote  BGP VPN   00h20m52s  170
       192.0.2.69 (tunneled)                                        0
10.1.0.0/16                                    Remote  BGP EVPN  00h22m33s  169
       evi-504 (ET-d8:45:ff:00:01:35)                               0
-------------------------------------------------------------------------------
No. of Routes: 4
```

The GW-MAC as well as the rest of the IP prefix BGP attributes are displayed by the **show router bgp routes evpn ip-prefix** command.

```
*A:Dut-A# show router bgp routes evpn ip-prefix prefix 3.0.1.6/32 detail
```

```
===============================================================================
BGP Router ID:10.20.1.1        AS:100         Local AS:100
===============================================================================
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
Origin codes  : i - IGP, e - EGP, ? - incomplete, > - best, b - backup

===============================================================================
BGP EVPN IP-Prefix Routes
===============================================================================
-------------------------------------------------------------------------------
Original Attributes

Network        : N/A
Nexthop        : 10.20.1.2
From           : 10.20.1.2
Res. Nexthop   : 192.168.19.1
Local Pref.    : 100                    Interface Name : NotAvailable
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : 0
AIGP Metric    : None
Connector      : None
Community      : target:100:1 mac-nh:00:00:01:00:01:02
                 bgp-tunnel-encap:VXLAN
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 10.20.1.2
Flags          : Used  Valid  Best  IGP
Route Source   : Internal
AS-Path        : No As-Path
EVPN type      : IP-PREFIX
ESI            : N/A                    Tag            : 1
Gateway Address: 00:00:01:00:01:02
Prefix         : 3.0.1.6/32             Route Dist.    : 10.20.1.2:1
MPLS Label    : 262140
Route Tag      : 0xb
Neighbor-AS    : N/A
Orig Validation: N/A
Source Class   : 0                      Dest Class     : 0

Modified Attributes

Network        : N/A
Nexthop        : 10.20.1.2
From           : 10.20.1.2
Res. Nexthop   : 192.168.19.1
Local Pref.    : 100                    Interface Name : NotAvailable
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : 0
AIGP Metric    : None
Connector      : None
Community      : target:100:1 mac-nh:00:00:01:00:01:02
                 bgp-tunnel-encap:VXLAN
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 10.20.1.2
Flags          : Used  Valid  Best  IGP
Route Source   : Internal
AS-Path        : 111
EVPN type      : IP-PREFIX
ESI            : N/A                    Tag            : 1
Gateway Address: 00:00:01:00:01:02
Prefix         : 3.0.1.6/32             Route Dist.    : 10.20.1.2:1
MPLS Label    : 262140
Route Tag      : 0xb
Neighbor-AS    : 111
```

```
Orig Validation: N/A
Source Class   : 0                     Dest Class     : 0

-------------------------------------------------------------------------------
Routes : 1
===============================================================================
```

EVPN tunneling is also supported on IPv6 VPRN interfaces. When sending IPv6 prefixes from IPv6 interfaces, the GW-MAC in the route type 5 (IP-prefix route) is always zero. If no specific Global Address is configured on the IPv6 interface, the routes type 5 for IPv6 prefixes are always sent using the Link Local Address as GW-IP. The following example output shows an IPv6 prefix received via BGP EVPN.

```
*A:PE71# show router 30 route-table ipv6

===============================================================================
IPv6 Route Table (Service: 30)
===============================================================================
Dest Prefix[Flags]                            Type    Proto     Age        Pref
      Next Hop[Interface Name]                                    Metric
-------------------------------------------------------------------------------
2001:db8:1000::/64                            Local   Local     00h01m19s  0
       int-PE-71-CE-1                                             0
2001:db8:2000::1/128                          Remote  BGP EVPN  00h01m20s  169
       fe80::da45:ffff:fe00:6a-"int-evi-301"                      0
-------------------------------------------------------------------------------
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
===============================================================================

*A:PE71# show router bgp routes evpn ipv6-prefix prefix 2001:db8:2000::1/128 hunt
===============================================================================
 BGP Router ID:192.0.2.71       AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked
 Origin codes  : i - IGP, e - EGP, ? - incomplete, > - best, b - backup

===============================================================================
BGP EVPN IP-Prefix Routes
===============================================================================
-------------------------------------------------------------------------------
RIB In Entries
-------------------------------------------------------------------------------
Network        : N/A
Nexthop        : 192.0.2.69
From           : 192.0.2.69
Res. Nexthop   : 192.168.19.2
Local Pref.    : 100                    Interface Name : int-71-69
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : 0
AIGP Metric    : None
Connector      : None
Community      : target:64500:301 bgp-tunnel-encap:VXLAN
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.0.2.69
Flags          : Used  Valid  Best  IGP
Route Source   : Internal
AS-Path        : No As-Path
```

```
EVPN type      : IP-PREFIX
ESI            : N/A                      Tag            : 301
Gateway Address: fe80::da45:ffff:fe00:*
Prefix         : 2001:db8:2000::1/128     Route Dist.    : 192.0.2.69:301
MPLS Label     : 0
Route Tag      : 0
Neighbor-AS    : N/A
Orig Validation: N/A
Source Class   : 0                        Dest Class     : 0
Add Paths Send : Default
Last Modified  : 00h41m17s

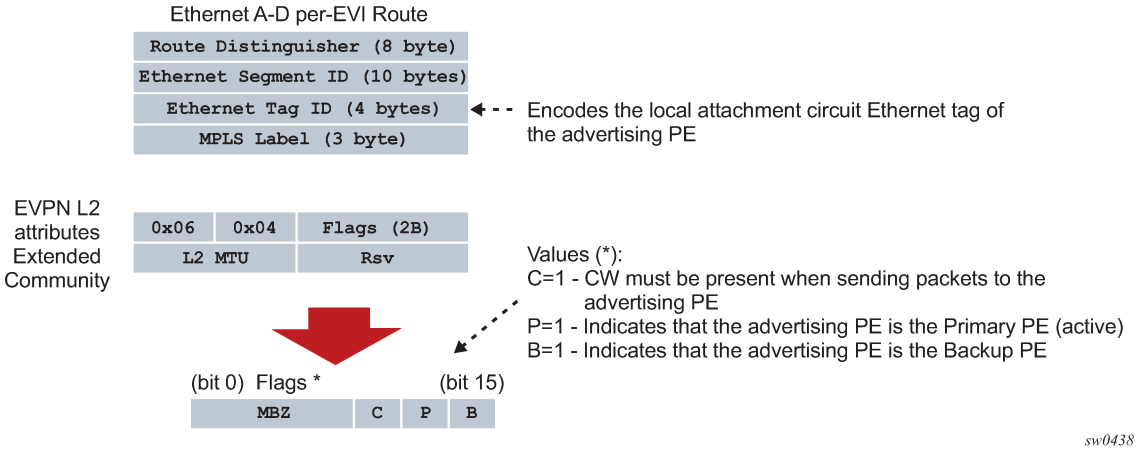
-------------------------------------------------------------------------------
RIB Out Entries
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Routes : 1
===============================================================================
```

6.2.2.4 EVPN-VPWS for VXLAN tunnels

BGP-EVPN control plane for EVPN-VPWS

EVPN-VPWS uses route-type 1 and route-type 4; it does not use route-types 2, 3 or 5. Figure 145: EVPN-VPWS BGP extensions shows the encoding of the required extensions for the Ethernet A-D per-EVI routes. The encoding follows the guidelines described in RFC 8214.

Figure 145: EVPN-VPWS BGP extensions



If the advertising PE has an access SAP-SDP or spoke SDP that is not part of an Ethernet Segment (ES), the PE populates the fields of the AD per-EVI route with the following values:

- Ethernet Tag ID field is encoded with the value configured by the user in the **service bgp-evpn local-attachment-circuit eth-tag value** command.
- RD and MPLS label values are encoded as specified in RFC 7432. For VXLAN, the MPLS field encodes the VXLAN VNI.
- ESI is 0.
- The route is sent along an EVPN L2 attributes extended community, as specified in RFC 8214, where:

- type and subtype are 0x06 and 0x04 as allocated by IANA
- flag C is set if a control word is configured in the service; C is always zero for VXLAN tunnels
- P and B flags are zero
- L2 MTU is encoded with a service MTU configured in the Epipe service

If the advertising PE has an access SAP-SDP or spoke SDP that is part of an ES, the AD per-EVI route is sent with the information described above, with the following minor differences:

- The ESI encodes the corresponding non-zero value.
- The P and B flags are set in the following cases:
  - All-active multihoming
    - All PEs that are part of the ES always set the P flag.
    - The B flag is never set in the all-active multihoming ES case.
  - Single-active multihoming
    - Only the DF PE sets the P bit for an EVI and the remaining PEs send it as P=0.
    - Only the backup DF PE sets the B bit.

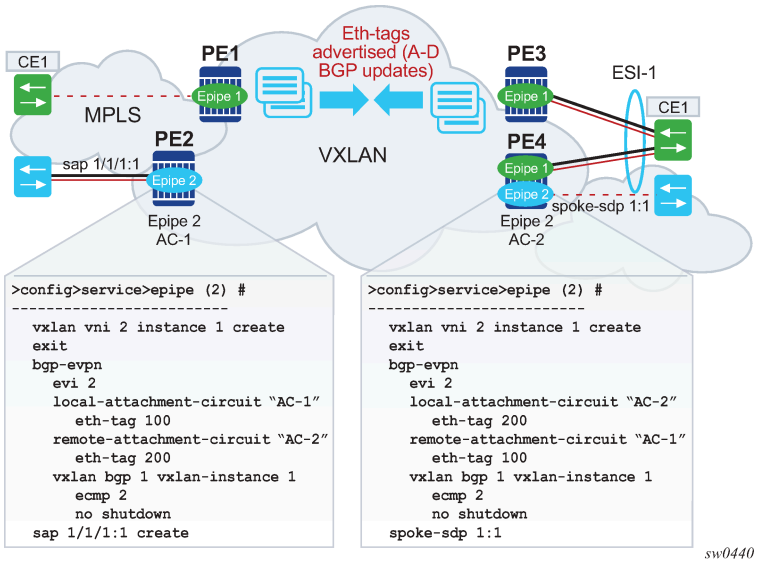
      If more than two PEs are present in the same single-active ES, the backup PE is the winner of a second DF election (excluding the DF). The remaining non-DF PEs send B=0.

Also, ES and AD per-ES routes are advertised and processed for the Ethernet-Segment, as described in RFC 7432 ESs. The ESI label sent with the AD per-ES route is used by BUM traffic on VPLS services; it is not used for Epipe traffic.

EVPN-VPWS for VXLAN tunnels in Epipe services

BGP-EVPN can be enabled in Epipe services with either SAPs or spoke SDPs at the access, as shown in Figure 146: EVPN-MPLS VPWS.

Figure 146: EVPN-MPLS VPWS



EVPN-VPWS is supported in VXLAN networks that also run EVPN-VXLAN in VPLS services. From a control plane perspective, EVPN-VPWS is a simplified point-to-point version of RFC 7432 for E-Line services for the following reasons:

- EVPN-VPWS does not use inclusive multicast, MAC/IP routes or IP-prefix routes.
- AD Ethernet per-EVI routes are used to advertise the local attachment circuit identifiers at each side of the VPWS instance. The attachment circuit identifiers are configured as local and remote Ethernet tags. When an AD per-EVI route is imported and the Ethernet tag matches the configured remote Ethernet tag, an EVPN destination is created for the Epipe.

In the following configuration example, Epipe 2 is an EVPN-VPWS service between PE2 and PE4 (as shown in Figure 146: EVPN-MPLS VPWS).

```
PE2>config>service>epipe(2)#
----------------------
vxlan vni 2 instance 1 create
exit
bgp
exit
bgp-evpn
  evi 2
  local-attachment-circuit "AC-1"
    eth-tag 100
  remote-attachment-circuit "AC-2"
    eth-tag 200
  vxlan bgp 1 vxlan-instance 1
    ecmp 2
    no shutdown
sap 1/1/1:1 create
```

```
PE4>config>service>epipe(2)#
----------------------
vxlan vni 2 instance 1 create
exit
bgp
exit
bgp-evpn
  evi 2
  local-attachment-circuit "AC-2"
    eth-tag 200
  remote-attachment-circuit "AC-1"
    eth-tag 100
  vxlan bgp 1 vxlan-instance 1
    ecmp 2
    no shutdown
spoke-sdp 1:1
```

The following considerations apply to the preceding example configuration:

- When the EVI value is lower than 65535, the EVI is used to automatically derive the route-target or route-distinguisher of the service. For EVI values greater than 65535, the route-distinguisher is not automatically derived and the route-target is automatically derived, if **evi-three-byte-auto-rt** is configured. The EVI values must be unique in the system regardless of the type of service to which they are assigned (Epipe or VPLS).
- Support for the following BGP-EVPN commands in Epipe services is the same as in VPLS services:
  - **vxlan bgp 1 vxlan-instance 1**
  - **vxlan send-tunnel-encap**

  - **vxlan shutdown**
  - **vxlan ecmp**
- The following BGP-EVPN commands identify the local and remote attachment circuits, with the configured Ethernet tags encoded in the advertised and received AD Ethernet per-EVI routes:
  - **local-attachment-circuit name**
  - **local-attachment-circuit name eth-tag tag-value**; where **tag-value** is 1 to 16777215
  - **remote-attachment-circuit name**
  - **remote-attachment-circuit name eth-tag tag-value**; where **tag-value** is 1 to 16777215

    Changes to remote Ethernet tags are allowed without shutting down BGP-EVPN VXLAN or the Epipe service. The local AC Ethernet tag value cannot be changed without BGP-EVPN VXLAN shutdown.

    Both local and remote Ethernet tags are mandatory to bring up the Epipe service.

EVPN-VPWS Epipes can also be configured with the following characteristics:

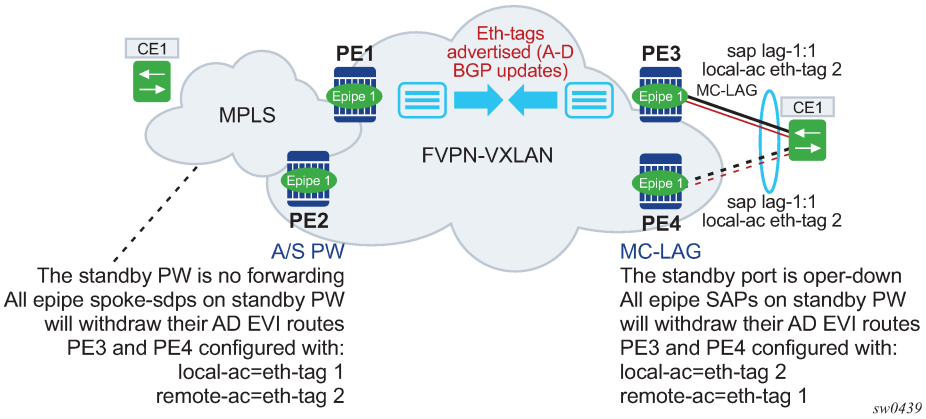
- Access attachment circuits can be SAPs or spoke SDP. Only manually-configured spoke SDP is supported; BGP-VPWS and endpoints are not supported. The VC switching configuration is not supported on BGP-EVPN enabled pipes.
- EVPN-VPWS Epipes can advertise the Layer 2 (service) MTU and check its consistency as follows:
  1. The advertised MTU value is taken from the configured service MTU in the Epipe service.
  2. The received L2 MTU is compared to the local value. In case of a mismatch between the received MTU and the configured service MTU, the system does not set up the EVPN destination; as a result, the service does not come up.

     Consider the following:
     - The system does not check the network port MTU value.
     - If the received L2 MTU value is 0, the MTU is ignored.

Using A/S PW and MC-LAG with EVPN-VPWS Epipes

The use of A/S PW (for access spoke SDP) and MC-LAG (for access SAPs) provides an alternative redundant solution for EVPN-VPWS that do not use the EVPN multi homing procedures described in RFC 8214. Figure 147: A/S PW and MC-LAG support on EVPN-VPWS shows the use of both mechanisms in a single Epipe.

Figure 147: A/S PW and MC-LAG support on EVPN-VPWS



In Figure 147: A/S PW and MC-LAG support on EVPN-VPWS, an A/S PW connects the CE to PE1 and PE2 (left side of the diagram), and an MC-LAG connects the CE to PE3 and PE4 (right side of the diagram). As EVPN multi homing is not used, there are no AD per-ES routes or ES routes. The redundancy is handled as follows:

- PE1 and PE2 are configured with Epipe-1, where a spoke SDP connects the service in each PE to the access CE. The local AC Ethernet tag is 1 and the remote AC Ethernet tag is 2 (in PE1/PE2).
- PE3 and PE4 are configured with Epipe-1, where each PE has a lag SAP that belongs to a previously-configured MC-LAG construct. The local AC Ethernet tag is 2 and the remote AC Ethernet tag is 1.
- An endpoint and A/S PW is configured on the CE on the left side of the diagram. PE1/PE2 are able to advertise Ethernet tag 1 based on the operating status or the forwarding status of the spoke SDP.

  For example, if PE1 receives a standby PW status indication from the CE and the previous status was forward, it withdraws the AD EVI route for Ethernet tag 1. If PE2 receives a forward PW status indication and the previous status was standby or down, it advertises the AD EVI route for Ethernet tag 1.
- The user can configure MC-LAG for access SAPs using the example configuration of PE3 and PE4, as shown in Figure 147: A/S PW and MC-LAG support on EVPN-VPWS. In this case, the MC-LAG determines which chassis is active and which is standby.

  If PE4 becomes the standby chassis, the entire LAG port is brought down. As a result, the SAP goes operationally down and PE4 withdraws any previous AD EVI routes for Ethernet tag 2.

  If PE3 becomes the active chassis, the LAG port becomes operationally up. As a result, the SAP and the PE3 advertise the AD per-EVI route for Ethernet tag 2.

EVPN multihoming for EVPN-VPWS services

EVPN multihoming is supported for EVPN-VPWS Epipe services with the following considerations:

- Single-active and all-active multihoming is supported for SAPs and spoke SDP.
- ESs can be shared between the Epipe (MPLS and VXLAN) and VPLS (MPLS) services for LAGs, ports, and SDPs.
- A split-horizon function is not required because there is no traffic between the Designated Forwarder (DF) and the non-DF for Epipe services. As a result, the ESI label is never used, and the **ethernet-segment multi-homing single-active no-esi-label** and **ethernet-segment source-bmac-lsb** commands do not affect Epipe services.

- The local Ethernet tag values must match on all PEs that are part of the same ES, regardless of the multi homing mode. The PEs in the ES use the AD per-EVI routes from the peer PEs to validate the PEs as DF election candidates for a specific EVI.

The DF election for Epipes that is defined in an all-active multi homing ES is not relevant because all PEs in the ES behave in the same way as follows:

- All PEs send P=1 on the AD per-EVI routes.
- All PEs can send upstream and downstream traffic, regardless of whether the traffic is unicast, multicast, or broadcast (all traffic is treated as unicast in the Epipe services).

  Therefore, the following tools command shows **N/A** when all-active multihoming is configured.

```
*A:PE-2# tools dump service system bgp-evpn ethernet-segment "ESI-12" evi 6000 df
[03/18/2016 20:31:35] All Active VPWS - DF N/A
```

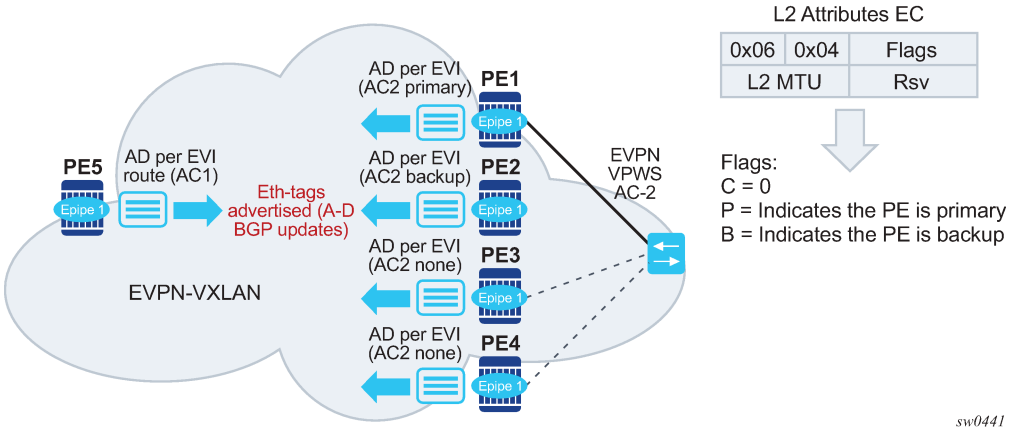
Aliasing is supported for traffic sent to an ES destination. If ECMP is enabled on the ingress PE, per-flow load balancing is performed to all PEs that advertise P=1. The PEs that advertise P=0, are not considered as next hops for an ES destination.

Note: The ingress PE load balances the traffic if shared queuing or ingress policing is enabled on the access SAPs.

Although DF election is not relevant for Epipes in an all-active multi homing ES, it is essential for the following forwarding and backup functions in a single-active multihoming ES:

- The PE elected as DF is the primary PE for the ES in the Epipe. The primary PE unblocks the SAP or spoke SDP for upstream and downstream traffic; the remaining PEs in the ES bring their ES SAPs or spoke SDPs operationally down.
- The DF candidate list is built from the PEs sending ES routes for the same ES and is pruned for a specific service, depending on the availability of the AD per-ES and per-EVI routes.
- When the SAP or spoke SDPs that are part of the ES come up, the AD per-EVI routes are sent with P=0 and B=0. The remote PEs do not start sending traffic until the DF election process is complete and the ES activation timer is expired, and the PEs advertise AD per-EVI routes with P and B bits other than zero.
- The backup PE function is supported as defined in RFC 8214. The primary PE, backup, or none status is signaled by the PEs (part of the same single-active MH ES) in the P or B flags of the EVPN L2 attributes extended community. Figure 148: EVPN-VPWS single-active multihoming shows the advertisement and use of the primary, backup, or none indication by the PEs in the ES.

Figure 148: EVPN-VPWS single-active multihoming



As specified in RFC 7432, the remote PEs in VPLS services have knowledge of the primary PE in the remote single-active ES, based on the advertisement of the MAC/IP routes because only the DF learns and advertises MAC/IP routes.

Because there are no MAC/IP routes in EVPN-VPWS, the remote PEs can forward the traffic based on the P/B bits. The process is described in the following list:

1. The DF PE for an EVI (PE1) sends P=1 and B=0.
2. For each ES or EVI, a second DF election is run among the PEs in the backup candidate list to elect the backup PE. The backup PE sends P=0 and B=1 (PE2).
3. All remaining multi homing PEs send P=0 and B=0 (PE3 and PE4).
4. At the remote PEs (PE5), the P and B flags are used to identify the primary and backup PEs within the ES destination. The traffic is then sent to the primary PE, provided that it is active.

- When a remote PE receives the withdrawal of an Ethernet AD per-ES (or per-EVI) route from the primary PE, the remote PE immediately switches the traffic to the backup PE for the affected EVIs. The backup PE takes over immediately without waiting for the ES activation timer to bring up its SAP or spoke SDP.
- The BGP-EVPN MPLS ECMP setting also governs the forwarding in single-active multi homing, regardless of the single-active multi homing bit in the AD per-ES route received at the remote PE (PE5).
  - PE5 always sends the traffic to the primary remote PE (the owner of the P=1 bit). In case of multiple primary PEs and ECMP>1, PE5 load balances the traffic to all primary PEs, regardless of the multi homing mode.
  - If the last primary PE withdraws its AD per-EVI or per-ES route, PE5 sends the traffic to the backup PE or PEs. In case of multiple backup PEs and ECMP>1, PE1 load balances the traffic to the backup PEs.

Non-system IPv4/IPv6 VXLAN termination for EVPN-VPWS services

EVPN-VPWS services support non-system IPv4/IPv6 VXLAN termination. For system configuration information, see Non-system IPv4 and IPv6 VXLAN termination in VPLS, R-VPLS, and Epipe services .

EVPN multihoming is supported when the PEs use non-system IP termination, however additional configuration steps are needed in this case:

- The **configure service system bgp-evpn eth-seg es-orig-ip ip-address** command must be configured with the non-system IPv4/IPv6 address used for the EVPN-VPWS VXLAN service. As a result, this command modifies the originating-ip field in the ES routes advertised for the Ethernet Segment, and makes the system use this IP address when adding the local PE as DF candidate.
- The **configure service system bgp-evpn eth-seg route-next-hop ip-address** command must be configured with the non-system IP address, too. The command changes the next-hop of the ES and AD per-ES routes to the configured address.
- The non-system IP address (in each of the PEs in the ES) must match in these three commands for the local PE to be considered suitable for DF election:
  - **es-orig-ip ip-address**
  - **route-next-hop ip-address**
  - **vxlan-src-vtep ip-address**

6.2.2.4.1 EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes

Figure 137: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC shows a Layer 3 DC model, where a VPRN is defined in the DGWs, connecting the tenant to the WAN. That VPRN instance is connected to the VPRNs in the NVEs by means of an IRB backhaul R-VPLS. Because the IRB backhaul R-VPLS provides connectivity only to all the IRB interfaces and the DGW VPRN is not directly connected to all the tenant subnets, the WAN ip-prefixes in the VPRN routing table must be advertised in EVPN. In the same way, the NVEs send IP prefixes in EVPN that is received by the DGW and imported in the VPRN routing table.

Note: To generate or process IP prefixes sent or received in EVPN route type 5, the support for IP route advertisement must be enabled in BGP-EVPN. This is performed through the **bgp-evpn ip-route-advertisement** command. This command s disabled by default and must be explicitly enabled. The command is tied to the **allow-ip-int-bind** command required for R-VPLS, and it is not supported on R-VPLS linked to IES services.

Local router interface host addresses are not advertised in EVPN by default. To advertise them, the **ip-route-advertisement incl-host** command must be enabled. For example:

```
===============================================================================
Route Table (Service: 2)
===============================================================================
Dest Prefix[Flags]                            Type    Proto     Age        Pref
      Next Hop[Interface Name]                         Active     Metric
-------------------------------------------------------------------------------
10.1.1.0/24                                   Local   Local     00h00m11s  0
       if                                              Y           0
10.1.1.100/32                                 Local   Host      00h00m11s  0
       if                                              Y           0
===============================================================================
```

For the case displayed by the output above, the behavior is the following:

- **ip-route-advertisement** only local subnet (default) - 10.1.1.0/24 is advertised
- **ip-route-advertisement incl-host** local subnet, host - 10.1.1.0/24 and 10.1.1.100/32 are advertised

Below is an example of VPRN (500) with two IRB interfaces connected to backhaul R-VPLS services 501 and 502 where EVPN-VXLAN runs:

```
vprn 500 customer 1 create
```

```
            ecmp 4
            route-distinguisher 65072:500
            vrf-target target:65000:500
            interface "evi-502" create
                address 10.20.20.72/24
                vpls "evpn-vxlan-502"
                exit
            exit
            interface "evi-501" create
                address 10.10.10.72/24
                vpls "evpn-vxlan-501"
                exit
            exit
            no shutdown
vpls 501 name "evpn-vxlan-501" customer 1 create
            allow-ip-int-bind
            vxlan instance 1 vni 501 create
            exit
            bgp
                route-distinguisher 65072:501
                route-target export target:65000:501 import target:65000:501
            exit
            bgp-evpn
                ip-route-advertisement incl-host
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            no shutdown
        exit
vpls 502 name "evpn-vxlan-502" customer 1 create
            allow-ip-int-bind
            vxlan instance 1 vni 502 create
            exit
            bgp
                route-distinguisher 65072:502
                route-target export target:65000:502 import target:65000:502
            exit
            bgp-evpn
                ip-route-advertisement incl-host
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            no shutdown
        exit
```

When the above commands are enabled, the router behaves as follows:

- Receive route-type 5 routes and import the IP prefixes and associated IP next-hops into the VPRN routing table.
  - If the route-type 5 is successfully imported by the router, the prefix included in the route-type 5 (for example, 10.0.0.0/24), is added to the VPRN routing table with a next-hop equal to the gateway IP included in the route (for example, 192.0.0.1. that refers to the IRB IP address of the remote VPRN behind which the IP prefix sits).
  - When the router receives a packet from the WAN to the 10.0.0.0/24 subnet, the IP lookup on the VPRN routing table yields 192.0.0.1 as the next-hop. That next-hop is resolved to a MAC in the ARP table and the MAC resolved to a VXLAN tunnel in the FDB table

Note: IRB MAC and IP addresses are advertised in the IRB backhaul R-VPLS in routes type 2.

- Generate route-type 5 routes for the IP prefixes in the associated VPRN routing table.

  For example, if VPRN-1 is attached to EVPN R-VPLS 1 and EVPN R-VPLS 2, and R-VPLS 2 has **bgp-evpn ip-route-advertisement** configured, the 7750 SR advertises the R-VPLS 1 interface subnet in one route-type 5.
- Routing policies can filter the imported and exported IP prefix routes accordingly.

The VPRN routing table can receive routes from all the supported protocols (BGP-VPN, OSPF, IS-IS, RIP, static routing) as well as from IP prefixes from EVPN, as shown below:

```
*A:PE72# show router 500 route-table
===============================================================================
Route Table (Service: 500)
===============================================================================
Dest Prefix[Flags]                            Type    Proto     Age        Pref
      Next Hop[Interface Name]                                    Metric
-------------------------------------------------------------------------------
10.20.20.0/24                                 Local   Local     01d11h10m  0
       evi-502                                                      0
10.20.20.71/32                                Remote  BGP EVPN  00h02m26s  169
       10.10.10.71                                                  0
10.10.10.0/24                                Remote  Static    00h00m05s  5
       10.10.10.71                                                  1
10.16.0.1/32                                 Remote  BGP EVPN  00h02m26s  169
       10.10.10.71                                                  0
-------------------------------------------------------------------------------
No. of Routes: 4
```

The following considerations apply:

- The route Preference for EVPN IP prefixes is 169.

  BGP IP-VPN routes have a preference of 170 by default, therefore, if the same route is received from the WAN over BGP-VPRN and from BGP-EVPN, then the EVPN route is preferred.
- When the same route-type 5 prefix is received from different gateway IPs, ECMP is supported if configured in the VPRN.
- All routes in the VPRN routing table (as long as they do not point back to the EVPN R-VPLS interface) are advertised via EVPN.

Although the description above is focused on IPv4 interfaces and prefixes, it applies to IPv6 interfaces too. The following considerations are specific to IPv6 VPRN R-VPLS interfaces:

- IPv4 and IPv6 interfaces can be defined on R-VPLS IP interfaces at the same time (dual-stack).
- The user may configure specific IPv6 Global Addresses on the VPRN R-VPLS interfaces. If a specific Global IPv6 Address is not configured on the interface, the Link Local Address interface MAC/IP is advertised in a route type 2 as soon as IPv6 is enabled on the VPRN R-VPLS interface.
- Routes type 5 for IPv6 prefixes are advertised using either the configured Global Address or the implicit Link Local Address (if no Global Address is configured).

  If more than one Global Address is configured, normally the first IPv6 address is used as gateway IP. The "first IPv6 address" refers to the first one on the list of IPv6 addresses shown through the **show router** <**id**> **interface** *interface* IPv6 or through SNMP.

The rest of the addresses are advertised only in MAC-IP routes (Route Type 2) but not used as gateway IP for IPv6 prefix routes.

6.2.2.4.2 EVPN for VXLAN in EVPN tunnel R-VPLS services

Figure 138: EVPN-tunnel gateway IRB on the DC PE for an L3 EVPN/VXLAN DC shows an L3 connectivity model that optimizes the solution described in EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes. Instead of regular IRB backhaul R-VPLS services for the connectivity of all the VPRN IRB interfaces, EVPN tunnels can be configured. The main advantage of using EVPN tunnels is that they do not need the configuration of IP addresses, as regular IRB R-VPLS interfaces do.

In addition to the **ip-route-advertisement** command, this model requires the configuration of the **config>service>vprn>if>vpls <name> evpn-tunnel**.

Note: The **evpn-tunnel** can be enabled independently of **ip-route-advertisement**, however, no route-type 5 advertisements are sent or processed in that case. Neither command, **evpn-tunnel** and **ip-route-advertisement**, is supported on R-VPLS services linked to IES interfaces.

The example below shows a VPRN (500) with an EVPN-tunnel R-VPLS (504):

```
vprn 500 customer 1 create
            ecmp 4
            route-distinguisher 65071:500
            vrf-target target:65000:500
            interface "evi-504" create
                vpls "evpn-vxlan-504"
                    evpn-tunnel
                exit
            exit
            no shutdown
        exit
        vpls 504 name “evpn-vxlan-504” customer 1 create
            allow-ip-int-bind
            vxlan instance 1 vni 504 create
            exit
            bgp
                route-distinguisher 65071:504
                route-target export target:65000:504 import target:65000:504
            exit
            bgp-evpn
                ip-route-advertisement
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            no shutdown
        exit
```

A specified VPRN supports regular IRB backhaul R-VPLS services as well as EVPN tunnel R-VPLS services.

Note: EVPN tunnel R-VPLS services do not support SAPs or SDP-binds.

The process followed upon receiving a route-type 5 on a regular IRB R-VPLS interface differs from the one for an EVPN-tunnel type:

- IRB backhaul R-VPLS VPRN interface:
  - When a route-type 2 that includes an IP prefix is received and it becomes active, the MAC/IP information is added to the FDB and ARP tables. This can be checked with the **show router arp** command and the **show service id fdb detail** command.
  - When route-type 5 is received and becomes active for the R-VPLS service, the IP prefix is added to the VPRN routing table, regardless of the existence of a route-type 2 that can resolve the gateway IP address. If a packet is received from the WAN side and the IP lookup hits an entry for which the gateway IP (IP next-hop) does not have an active ARP entry, the system uses ARP to get a MAC. If ARP is resolved but the MAC is unknown in the FDB table, the system floods into the TLS multicast list. Routes type 5 can be checked in the routing table with the **show router route-table** and **show router fib** commands.
- EVPN tunnel R-VPLS VPRN interface:
  - When route-type 2 is received and becomes active, the MAC address is added to the FDB (only).
  - When a route-type 5 is received and active, the IP prefix is added to the VPRN routing table with next-hop equal to EVPN tunnel: GW-MAC.

    For example, ET-d8:45:ff:00:01:35, where the GW-MAC is added from the GW-MAC extended community sent along with the route-type 5.

    If a packet is received from the WAN side, and the IP lookup hits an entry for which the next-hop is a EVPN tunnel: GW-MAC, the system looks up the GW-MAC in the FDB. Usually a route-type 2 with the GW-MAC is previously received so that the GW-MAC can be added to the FDB. If the GW-MAC is not present in the FDB, the packet is dropped.
  - IP prefixes with GW-MACs as next-hops are displayed by the show router command, as shown below:

```
*A:PE71# show router 500 route-table
===============================================================================
Route Table (Service: 500)
===============================================================================
Dest Prefix[Flags]                              Type    Proto     Age        Pref
      Next Hop[Interface Name]                                      Metric
-------------------------------------------------------------------------------
10.20.20.72/32                                  Remote  BGP EVPN  00h23m50s  169
       10.10.10.72                                                  0
10.30.30.0/24                                   Remote  BGP EVPN  01d11h30m  169
       evi-504 (ET-d8:45:ff:00:01:35)                               0
10.10.10.0/24                                  Remote  BGP VPN   00h20m52s  170
       192.0.2.69 (tunneled)                                        0
10.1.0.0/16                                    Remote  BGP EVPN  00h22m33s  169
       evi-504 (ET-d8:45:ff:00:01:35)                               0
-------------------------------------------------------------------------------
No. of Routes: 4
```

The GW-MAC as well as the rest of the IP prefix BGP attributes are displayed by the **show router bgp routes evpn ip-prefix** command.

```
*A:Dut-A# show router bgp routes evpn ip-prefix prefix 3.0.1.6/32 detail
===============================================================================
BGP Router ID:10.20.1.1        AS:100         Local AS:100
===============================================================================
Legend -
Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
Origin codes  : i - IGP, e - EGP, ? - incomplete, > - best, b - backup
```

```
===============================================================================
BGP EVPN IP-Prefix Routes
===============================================================================
-------------------------------------------------------------------------------
Original Attributes

Network        : N/A
Nexthop        : 10.20.1.2
From           : 10.20.1.2
Res. Nexthop   : 192.168.19.1
Local Pref.    : 100                    Interface Name : NotAvailable
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : 0
AIGP Metric    : None
Connector      : None
Community      : target:100:1 mac-nh:00:00:01:00:01:02
                 bgp-tunnel-encap:VXLAN
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 10.20.1.2
Flags          : Used  Valid  Best  IGP
Route Source   : Internal
AS-Path        : No As-Path
EVPN type      : IP-PREFIX
ESI            : N/A                    Tag            : 1
Gateway Address: 00:00:01:00:01:02
Prefix         : 3.0.1.6/32             Route Dist.    : 10.20.1.2:1
MPLS Label    : 262140
Route Tag      : 0xb
Neighbor-AS    : N/A
Orig Validation: N/A
Source Class   : 0                      Dest Class     : 0

Modified Attributes

Network        : N/A
Nexthop        : 10.20.1.2
From           : 10.20.1.2
Res. Nexthop   : 192.168.19.1
Local Pref.    : 100                    Interface Name : NotAvailable
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : 0
AIGP Metric    : None
Connector      : None
Community      : target:100:1 mac-nh:00:00:01:00:01:02
                 bgp-tunnel-encap:VXLAN
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 10.20.1.2
Flags          : Used  Valid  Best  IGP
Route Source   : Internal
AS-Path        : 111
EVPN type      : IP-PREFIX
ESI            : N/A                    Tag            : 1
Gateway Address: 00:00:01:00:01:02
Prefix         : 3.0.1.6/32             Route Dist.    : 10.20.1.2:1
MPLS Label    : 262140
Route Tag      : 0xb
Neighbor-AS    : 111
Orig Validation: N/A
Source Class   : 0                      Dest Class     : 0

-------------------------------------------------------------------------------
Routes : 1
===============================================================================
```

EVPN tunneling is also supported on IPv6 VPRN interfaces. When sending IPv6 prefixes from IPv6 interfaces, the GW-MAC in the route type 5 (IP-prefix route) is always zero. If no specific Global Address is configured on the IPv6 interface, the routes type 5 for IPv6 prefixes are always sent using the Link Local Address as GW-IP. The following example output shows an IPv6 prefix received through BGP EVPN.

```
*A:PE71# show router 30 route-table ipv6

===============================================================================
IPv6 Route Table (Service: 30)
===============================================================================
Dest Prefix[Flags]                            Type    Proto     Age        Pref
      Next Hop[Interface Name]                                    Metric
-------------------------------------------------------------------------------
2001:db8:1000::/64                            Local   Local     00h01m19s  0
       int-PE-71-CE-1                                             0
2001:db8:2000::1/128                          Remote  BGP EVPN  00h01m20s  169
       fe80::da45:ffff:fe00:6a-"int-evi-301"                      0
-------------------------------------------------------------------------------
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
       B = BGP backup route available
       L = LFA nexthop available
       S = Sticky ECMP requested
===============================================================================

*A:PE71# show router bgp routes evpn ipv6-prefix prefix 2001:db8:2000::1/128 hunt
===============================================================================
 BGP Router ID:192.0.2.71       AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked
 Origin codes  : i - IGP, e - EGP, ? - incomplete, > - best, b - backup

===============================================================================
BGP EVPN IP-Prefix Routes
===============================================================================
-------------------------------------------------------------------------------
RIB In Entries
-------------------------------------------------------------------------------
Network        : N/A
Nexthop        : 192.0.2.69
From           : 192.0.2.69
Res. Nexthop   : 192.168.19.2
Local Pref.    : 100                    Interface Name : int-71-69
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : 0
AIGP Metric    : None
Connector      : None
Community      : target:64500:301 bgp-tunnel-encap:VXLAN
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.0.2.69
Flags          : Used  Valid  Best  IGP
Route Source   : Internal
AS-Path        : No As-Path
EVPN type      : IP-PREFIX
ESI            : N/A                    Tag            : 301
Gateway Address: fe80::da45:ffff:fe00:*
Prefix         : 2001:db8:2000::1/128   Route Dist.    : 192.0.2.69:301
MPLS Label     : 0
Route Tag      : 0
```

```
Neighbor-AS    : N/A
Orig Validation: N/A
Source Class   : 0                     Dest Class     : 0
Add Paths Send : Default
Last Modified  : 00h41m17s

-------------------------------------------------------------------------------
RIB Out Entries
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Routes : 1
===============================================================================
```

6.2.3 Layer 2 multicast optimization for VXLAN (Assisted-Replication)

The Assisted-Replication feature for IPv4 VXLAN tunnels (both Leaf and Replicator functions) is supported in compliance with the non-selective mode described in IETF Draft *draft-ietf-bess-evpn-optimized-ir*.

The Assisted-Replication feature is a Layer 2 multicast optimization feature that helps software-based PE and NVEs with low-performance replication capabilities to deliver broadcast and multicast Layer 2 traffic to remote VTEPs in the VPLS service.

The EVPN and proxy-ARP/ND capabilities can reduce the amount of broadcast and unknown unicast in the VPLS service; ingress replication is sufficient for most use cases in this scenario. However, when multicast applications require a significant amount of replication at the ingress node, software-based nodes struggle because of their limited replication performance. By enabling the Assisted-Replication Leaf function on the software-based SR-series router, all the broadcast and multicast packets are sent to a 7x50 router configured as a Replicator, which replicates the traffic to all the VTEPs in the VPLS service on behalf of the Leaf. This guarantees that the broadcast or multicast traffic is delivered to all the VPLS participants without any packet loss caused by performance issues.

The Leaf or Replicator function is enabled per VPLS service by the **configure service vpls vxlan assisted-replication {replicator | leaf}** command. In addition, the Replicator requires the configuration of an Assisted-Replication IP (AR-IP) address. The AR-IP loopback address indicates whether the received VXLAN packets have to be replicated to the remote VTEPs. The AR-IP address is configured using the **configure service system vxlan assisted-replication-ip** <**ip-address**> command.

Based on the **assisted-replication {replicator | leaf}** configuration, the SR-series router can behave as a Replicator (AR-R), Leaf (AR-L), or Regular Network Virtualization Edge (RNVE) router. An RNVE router does not support the Assisted-Replication feature. Because it is configured with no assisted replication, the RNVE router ignores the AR-R and AR-L information and replicates to its flooding list where VTEPs are added based on the regular ingress replication routes.

6.2.3.1 Replicator (AR-R) procedures

An AR-R configuration is shown in the following example.

```
*A:PE-2>config>service>system>vxlan# info
----------------------------------------------
            assisted-replication-ip 10.2.2.2
----------------------------------------------
*A:PE-2>config>service>vpls# info
----------------------------------------------
            vxlan instance 1 vni 4000 create
                assisted-replication replicator
```

```
            exit
            bgp
            exit
            bgp-evpn
                evi 4000
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
<snip>
            no shutdown
-----------------------------------------------
```

In this example configuration, the BGP advertises a new inclusive multicast route with tunnel-type = AR, type (T) = AR-R, and tunnel-id = originating-ip = next-hop = assisted-replication-ip (IP address 10.2.2.2 in the preceding example). In addition to the AR route, the AR-R sends a regular IR route if **ingress-repl-inc-mcast-advertisement** is enabled.

Note: You should disable the **ingress-repl-inc-mcast-advertisement** command if the AR-R does not have any SAP or SDP bindings and is used solely for Assisted-Replication functions.

The AR-R builds a flooding list composed of ACs (SAPs and SDP bindings) and VXLAN tunnels to remote nodes in the VPLS. All objects in the flooding list are broadcast/multicast (BM) and unknown unicast (U) capable. The following example output of the **show service id vxlan** command shows that the VXLAN destinations in the flooding list are tagged as "BUM".

```
*A:PE-2# show service id 4000 vxlan
===============================================================================
Vxlan Src Vtep IP: N/A
===============================================================================
VPLS VXLAN, Ingress VXLAN Network Id: 4000
Creation Origin: manual
Assisted-Replication: replicator
RestProtSrcMacAct: none
===============================================================================
VPLS VXLAN service Network Specifics
===============================================================================
Ing Net QoS Policy : none                          Vxlan VNI Id     : 4000
Ingress FP QGrp    : (none)                        Ing FP QGrp Inst : (none)
===============================================================================
Egress VTEP, VNI
===============================================================================
VTEP Address                          Egress VNI  Num. MACs   Mcast Oper  L2
                                                                  State PBR
-------------------------------------------------------------------------------
192.0.2.3                             4000        0           BUM   Up    No
192.0.2.5                             4000        0           BUM   Up    No
192.0.2.6                             4000        0           BUM   Up    No
-------------------------------------------------------------------------------
Number of Egress VTEP, VNI : 3
-------------------------------------------------------------------------------
===============================================================================
```

When the AR-R receives a BUM packet on an AC, the AR-R forwards the packet to its flooding list (including the local ACs and remote VTEPs).

When the AR-R receives a BM packet on a VXLAN tunnel, it checks the IP DA of the underlay IP header and performs the following BM packet processing.

Note: The AR-R function is only relevant to BM packets; it does not apply to unknown unicast packets. If the AR-R receives unknown unicast packets, it sends them to the flooding list, skipping the VXLAN tunnels.

- If the destination IP matches its AR-IP, the AR-R forwards the BM packet to its flooding list (ACs and VXLAN tunnels). The AR-R performs source suppression to ensure that the traffic is not sent back to the originating Leaf.
- If the destination IP matches its regular VXLAN termination IP (IR-IP), the AR-R skips all the VXLAN tunnels from the flooding list and only replicates to the local ACs. This is the default Ingress Replication (IR) behavior.

6.2.3.2 Leaf (AR-L) procedures

An AR-L is configured as shown in the following example.

```
A:PE-3>config>service>vpls# info
----------------------------------------------
            vxlan instance 1 vni 4000 create
                assisted-replication leaf replicator-activation-time 30
            bgp
            exit
            bgp-evpn
                evi 4000
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
                mpls
                    shutdown
                exit
            exit
            stp
                shutdown
            exit
            sap 1/1/1:4000 create
                no shutdown
            exit
            no shutdown
----------------------------------------------
```

In this example configuration, the BGP advertises a new inclusive multicast route with a tunnel-type = IR, type (T) = AR-L and tunnel-id = originating-ip = next-hop = IR-IP (IP address terminating VXLAN normally, either system-ip or vxlan-src-vtep address).

The AR-L builds a single flooding list per service but controlled by the BM and U flags. These flags are displayed in the following **show service id vxlan** command example output.

```
A:PE-3# show service id 4000 vxlan
===============================================================================
Vxlan Src Vtep IP: N/A
===============================================================================
VPLS VXLAN, Ingress VXLAN Network Id: 4000
Creation Origin: manual
Assisted-Replication: leaf      Replicator-Activation-Time: 30
RestProtSrcMacAct: none
===============================================================================
VPLS VXLAN service Network Specifics
===============================================================================
Ing Net QoS Policy : none                           Vxlan VNI Id     : 4000
```

```
Ingress FP QGrp    : (none)                          Ing FP QGrp Inst : (none)
===============================================================================
Egress VTEP, VNI
===============================================================================
VTEP Address                            Egress VNI  Num. MACs   Mcast Oper  L2
                                                                      State PBR
-------------------------------------------------------------------------------
10.2.2.2                                 4000        0           BM    Up    No
10.4.4.4                                 4000        0           -     Up    No
192.0.2.2                               4000        0           U     Up    No
192.0.2.5                               4000        0           U     Up    No
192.0.2.6                               4000        0           U     Up    No
-------------------------------------------------------------------------------
Number of Egress VTEP, VNI : 5
-------------------------------------------------------------------------------
===============================================================================
```

The AR-L creates the following VXLAN destinations when it receives and selects a Replicator-AR route or the Regular-IR routes:

- A VXLAN destination to each remote PE that sent an IR route. These bindings have the U flag set.
- A VXLAN destination to the selected AR-R. These bindings have only the BM flag set; the U flag is not set.
- The non-selected AR-Rs create a binding with flag "-" (in the CPM) that is displayed by the **show service id vxlan** command. Although the VXLAN destinations to non-selected AR-Rs do not carry any traffic, the destinations count against the total limit and must be considered when accounting for consumed VXLAN destinations in the router.

The BM traffic is only sent to the selected AR-R, whereas the U (unknown unicast) traffic is sent to all the destinations with the U flag.

The AR-L performs per-service load-balancing of the BM traffic when two or more AR-Rs exist in the same service. The AR Leaf creates a list of candidate PEs for each AR-R (ordered by IP and VNI; candidate 0 being the lowest IP and VNI). The replicator is selected out of a modulo function of the service-id and the number of replicators, as shown in the following example output.

```
A:PE-3# show service id 4000 vxlan assisted-replication replicator
===============================================================================
Vxlan AR Replicator Candidates
===============================================================================
VTEP Address           Egress VNI     In Use  In Candidate List Pending Time
-------------------------------------------------------------------------------
10.2.2.2                4000           yes     yes               0
10.4.4.4                4000           no      yes               0
-------------------------------------------------------------------------------
Number of entries : 2
-------------------------------------------------------------------------------
===============================================================================
```

A change in the number of Replicator-AR routes (for example, if a route is withdrawn or a new route appears) affects the result of the hashing, which may cause a different AR-R to be selected.

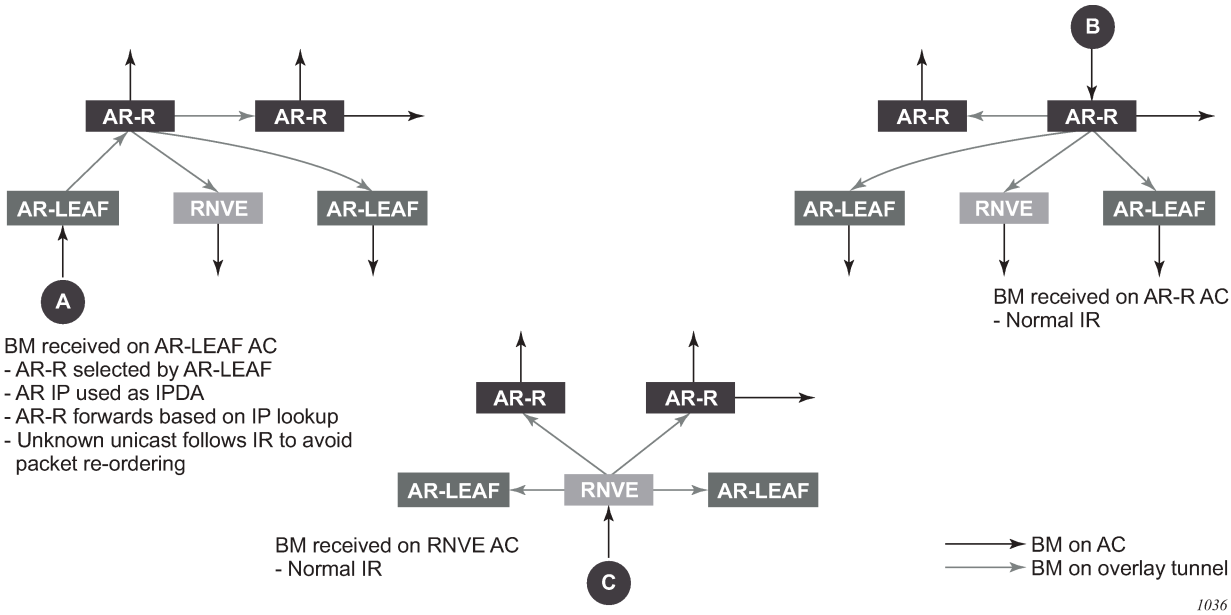
Note: An AR-L waits for the configured replicator-activation-time before sending the BM packets to the AR-R. In the interim, the AR-L uses regular ingress replication procedures. This activation time allows the AR-R to program the Leaf VTEP. If the timer is zero, the AR-R may receive packets from a not-yet-programmed source VTEP, in which case it discards the packets.

The following list summarizes other aspects of the AR-L behavior:

- When a Leaf receives a BM packet on an AC, it sends the packet to its flood list that includes access SAP or SDP bindings and VXLAN destinations with BM or BUM flags. If a single AR-R is selected, only a VXLAN destination includes the BM flags.
- Control plane-generated BM packets, such as ARP/ND (when proxy-ARP/ND is enabled) or Eth-CFM, follow the behavior of regular data plane BM packets.
- When a Leaf receives an unknown unicast packet on an AC, it sends the packet to the flood-list, skipping the AR destination because the U flag is set to 0. To avoid packet re-ordering, the unknown unicast packets do not go through the AR-R.
- When a Leaf receives a BUM packet on an overlay tunnel, it forwards the packet to the flood list, skipping the VXLAN tunnels (that is, the packet is sent to the local ACs and never to a VXLAN tunnel). This is the default IR behavior.
- When the last Replicator-AR route is withdrawn, the AR-L removes the AR destination from the flood list and falls back to ingress replication.

Figure 149: AR BM replication behavior for a BM packet shows the expected replication behavior for BM traffic when received at the access on an AR-R, AR-L, or RNVE router. Unknown unicast follows regular ingress replication behavior regardless of the role of the ingress node for the specific service.

Figure 149: AR BM replication behavior for a BM packet



6.2.3.3 Assisted-Replication interaction with other VPLS features

The Assisted-Replication feature has the following limitations:

- The following features are not supported on the same service where the Assisted-Replication feature is enabled.
  - Aggregate QoS per VNI
  - VXLAN IPv6 transport
  - IGMP/MLD/PIM-snooping

- Assisted-Replication Leaf and Replicator functions are mutually exclusive within the same VPLS service.
- The Assisted-Replication feature is supported with IPv4 non-system-ip VXLAN termination. However, the configured assisted-replication-ip (AR-IP) must be different from the tunnel termination IP address.
- The AR-IP address must be a /32 loopback interface on the base router.
- The Assisted-Replication feature is only supported in EVPN-VXLAN services (VPLS with BGP-EVPN vxlan enabled). Although services with a combination of EVPN-MPLS and EVPN-VXLAN are supported, the Assisted-Replication configuration is only relevant to the VXLAN.

6.2.4 DGW policy based forwarding/routing to an EVPN ESI

The Nuage Virtual Services Platform (VSP) supports a service chaining function that ensures traffic traverses a number of services (also known as Service Functions) between application hosts (FW, LB, NAT, IPS/IDS, and so on.) if the operator needs to do so. In the DC, tenants want the ability to specify these functions and their sequence, so that services can be added or removed without requiring changes to the underlying application.

This service chaining function is built based on a series of policy based routing/forwarding redirecting rules that are automatically coordinated and abstracted by the Nuage Virtual Services Directory (VSD). From a networking perspective, the packets are hop-by-hop redirected based on the location of the corresponding SF (Service Function) in the DC fabric. The location of the SF is specified by its VTEP and VNI and is advertised by BGP-EVPN along with an Ethernet Segment Identifier that is uniquely associated with the SF.

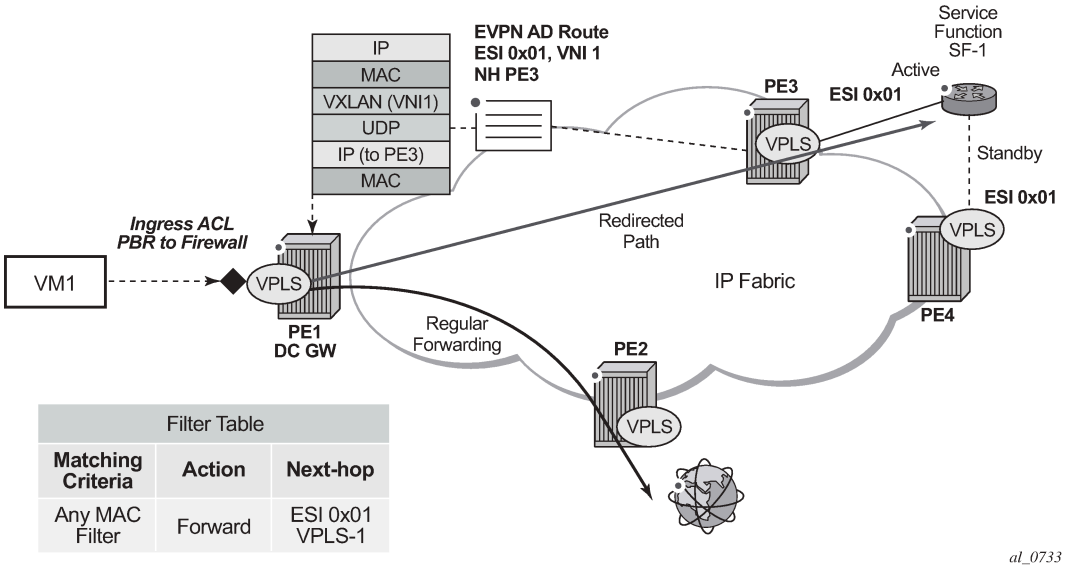
For more information about the Nuage Service Chaining solution, see the Nuage VSP documentation.

The 7750 SR, 7450 ESS, or 7950 XRS can be integrated as the first hop in the chain in a Nuage DC. This service chaining integration is intended to be used as described in the following three use cases.

6.2.4.1 Policy based forwarding in VPLS services for Nuage Service Chaining integration in L2-domains

Figure 150: PBF to ESI function shows the 7750 SR, 7450 ESS, and 7950 XRS Service Chaining integration with the Nuage VSP on VPLS services. In this example, the DC gateway, PE1, is connected to an L2-DOMAIN that exists in the DC and must redirect the traffic to the Service Function SF-1. The regular Layer 2 forwarding procedures would have taken the packets to PE2, as opposed to SF-1.

Figure 150: PBF to ESI function



An operator must configure a PBF match/action filter policy entry in an IPv4 or MAC ingress access or network filter deployed on a VPLS interface using CLI/SNMP/NETCONF management interfaces. The PBF target is the first service function in the chain (SF-1) that is identified by an ESI.

In the example shown in Figure 150: PBF to ESI function, the PBF filter redirects the matching packets to ESI 0x01 in VPLS-1.

Note: The Figure 150: PBF to ESI function represents ESI as "0x01" for simplicity; in reality, the ESI is a 10-byte number.

As soon as the redirection target is configured and associated with the vport connected to SF-1, the Nuage VSC (Virtual Services Controller, or the remote PE3 in the example) advertises the location of SF-1 via an Auto-Discovery Ethernet Tag route (route type 1) per-EVI. In this AD route, the ESI associated with SF-1 (ESI 0x01) is advertised along with the VTEP (PE3's IP) and VNI (VNI-1) identifying the vport where SF-1 is connected. PE1 sends all the frames matching the ingress filter to PE3's VTEP and VNI-1.

Note: When packets get to PE3, VNI-1 (the VNI advertised in the AD route) indicate that a cut-through switching operation is needed to deliver the packets straight to the SF-1 vport, without the need for a regular MAC lookup.

The following filter configuration shows an example of PBF rule redirecting all the frames to an ESI.

```
A:PE1>config>filter>mac-filter# info
----------------------------------------------
            default-action forward
            entry 10 create
                action
                    forward esi ff:00:00:00:00:00:00:00:00:01 service-id 301
                exit
            exit
```

When the filter is properly applied to the VPLS service (VPLS-301 in this example), it shows 'Active' in the following show commands as long as the Auto-Discovery route for the ESI is received and imported.

```
A:PE1# show filter mac 1
===============================================================================
Mac Filter
===============================================================================
Filter Id   : 1                                 Applied         : Yes
Scope       : Template                          Def. Action     : Forward
Entries     : 1                                 Type            : normal
Description : (Not Specified)
-------------------------------------------------------------------------------
Filter Match Criteria : Mac
-------------------------------------------------------------------------------
Entry       : 10                                FrameType       : Ethernet
Description : (Not Specified)
Log Id      : n/a
Src Mac     : Undefined
Dest Mac    : Undefined
Dot1p       : Undefined                         Ethertype       : Undefined
DSAP        : Undefined                         SSAP            : Undefined
Snap-pid    : Undefined                         ESnap-oui-zero  : Undefined
Match action: Forward (ESI) Active
  ESI       : ff:00:00:00:00:00:00:00:00:01
  Svc Id    : 301
PBR Down Act: Forward (entry-default)
Ing. Matches: 3 pkts
Egr. Matches: 0 pkts
===============================================================================

A:PE1# show service id 301 es-pbr
===============================================================================
L2 ES PBR
===============================================================================
ESI                          Users      Status
                                        VTEP:VNI
-------------------------------------------------------------------------------
ff:00:00:00:00:00:00:00:00:01 1         Active
                                        192.0.2.72:7272
-------------------------------------------------------------------------------
Number of entries : 1
-------------------------------------------------------------------------------
===============================================================================
```

Details of the received AD route that resolves the filter forwarding are shown in the following **show router bgp routes** command.

```
A:PE1# show router bgp routes evpn auto-
disc esi ff:00:00:00:00:00:00:00:00:01
===============================================================================
 BGP Router ID:192.0.2.71       AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked, x - stale, > - best, b - backup
 Origin codes  : i - IGP, e - EGP, ? - incomplete
===============================================================================
BGP EVPN Auto-Disc Routes
===============================================================================
Flag  Route Dist.         ESI                           NextHop
      Tag                                               Label
-------------------------------------------------------------------------------
```

```
u*>i  192.0.2.72:100      ff:00:00:00:00:00:00:00:01 192.0.2.72
      0                                              VNI 7272

-------------------------------------------------------------------------------
Routes : 1
===============================================================
```

This AD route, when used for PBF redirection, is added to the list of EVPN-VXLAN bindings for the VPLS service and shown as 'L2 PBR' type:

```
A:PE1# show service id 301 vxlan
===============================================================================
VPLS VXLAN, Ingress VXLAN Network Id: 301
===============================================================================
Egress VTEP, VNI
===============================================================================
VTEP Address           Egress VNI     Num. MACs    Mcast   Oper State   L2 PBR
-------------------------------------------------------------------------------
192.0.2.69             301            1            Yes     Up           No
192.0.2.72             301            1            Yes     Up           No
192.0.2.72             7272           0            No      Up           Yes
-------------------------------------------------------------------------------
Number of Egress VTEP, VNI : 3
-------------------------------------------------------------------------------
===============================================================================
```

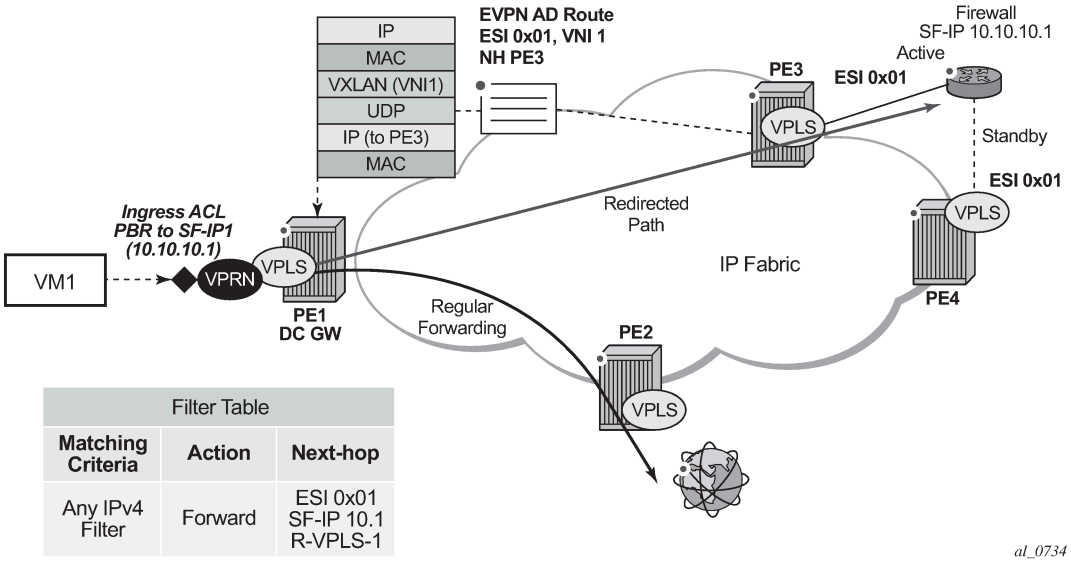
If the AD route is withdrawn, the binding disappears and the filter is inactive again. The user can control whether the matching packets are dropped or forwarded if the PBF target cannot be resolved by BGP.

Note: ES-based PBF filters can be applied only on services with the default **bgp (vxlan) instance (instance 1)**.

6.2.4.2 Policy based routing in VPRN services for Nuage Service Chaining integration in L2-DOMAIN-IRB domains

Figure 151: PBR to ESI function shows the 7750 SR, 7450 ESS, and 7950 XRS Service Chaining integration with the Nuage VSP on L2-DOMAIN-IRB domains. In this example, the DC gateway, PE1, is connected to an L2-DOMAIN-IRB that exists in the DC and must redirect the traffic to the Service Function SF-1 with IP address 10.10.10.1. The regular Layer 3 forwarding procedures would have taken the packets to PE2, as opposed to SF-1.

Figure 151: PBR to ESI function



In this case, an operator must configure a PBR match/action filter policy entry in an IPv4 ingress access or network filter deployed on IES/VPRN interface using CLI, SNMP or NETCONF management interfaces. The PBR target identifies first service function in the chain (ESI 0x01 in Figure 151: PBR to ESI function, identifying where the Service Function is connected and the IPv4 address of the SF) and EVPN VXLAN egress interface on the PE (VPRN routing instance and R-VPLS interface name). The BGP control plane together with ESI PBR configuration are used to forward the matching packets to the next-hop in the EVPN-VXLAN data center chain (through resolution to a VNI and VTEP). If the BGP control plane information is not available, the packets matching the ESI PBR entry is, by default, forwarded using regular routing. Optionally, an operator can select to drop the packets when the ESI PBR target is not reachable.

The following filter configuration shows an example of a PBR rule redirecting all the matching packets to an ESI.

```
*A:PE1>config>filter>ip-filter# info
----------------------------------------------
            default-action forward
            entry 10 create
                match
                    dst-ip 10.10.10.253/32
                exit
                action
                    forward esi ff:00:00:00:00:21:5f:00:df:e5 sf-ip 10.10.10.1 vas-
interface "evi-301" router 300
                exit
                pbr-down-action-override filter-default-action
            exit
----------------------------------------------
```

In this use case, the following are required in addition to the ESI: the **sf-ip** (10.10.10.1 in the example above), **router** instance (300), and **vas-interface**.

The **sf-ip** is used by the system to know which inner MAC DA it has to use when sending the redirected packets to the SF. The SF-IP is resolved to the SF MAC following regular ARP procedures in EVPN-VXLAN.

The **router** instance may be the same as the one where the ingress filter is configured or may be different: for instance, the ingress PBR filter can be applied on an IES interface pointing at a VPRN router instances that is connected to the DC fabric.

The **vas-interface** refers to the R-VPLS interface name through which the SF can be found. The VPRN instance may have more than one R-VPLS interface, therefore, it is required to specify which R-VPLS interface to use.

When the filter is properly applied to the VPRN or IES service (VPRN-300 in this example), it shows 'Active' in the following show commands as long as the Auto-Discovery route for the ESI is received and imported and the SF-IP resolved to a MAC address.

```
*A:PE1# show filter ip 1

===============================================================================
IP Filter
===============================================================================
Filter Id    : 1                                Applied        : Yes
Scope        : Template                         Def. Action    : Forward
System filter: Unchained
Radius Ins Pt: n/a
CrCtl. Ins Pt: n/a
RadSh. Ins Pt: n/a
PccRl. Ins Pt: n/a
Entries      : 1
Description  : (Not Specified)
-------------------------------------------------------------------------------
Filter Match Criteria : IP
-------------------------------------------------------------------------------
Entry        : 10
Description  : (Not Specified)
Log Id       : n/a
Src. IP      : 0.0.0.0/0
Src. Port    : n/a
Dest. IP     : 10.16.0.253/32
Dest. Port   : n/a
Protocol     : Undefined                        Dscp           : Undefined
ICMP Type    : Undefined                        ICMP Code      : Undefined
Fragment     : Off                              Src Route Opt  : Off
Sampling     : Off                              Int. Sampling  : On
IP-Option    : 0/0                              Multiple Option: Off
TCP-syn      : Off                              TCP-ack        : Off
Option-pres  : Off
Egress PBR   : Undefined
Match action : Forward (ESI) Active
  ESI        : ff:00:00:00:00:21:5f:00:df:e5
  SF IP      : 10.10.10.1
  VAS If name: evi-301
  Router     : 300
PBR Down Act : Forward (filter-default-action) Ing. Matches : 3 pkts (318 bytes)
Egr. Matches : 0 pkts
===============================================================================

*A:PE1# show service id 300 es-pbr
===============================================================================
L3 ES PBR
===============================================================================
SF IP              ESI                                  Users Status
                   Interface                                  MAC
                                                              VTEP:VNI
-------------------------------------------------------------------------------
10.10.10.1         ff:00:00:00:00:21:5f:00:df:e5        1     Active
                   evi-301                                    d8:47:01:01:00:0a
```

```
                                                                    192.0.2.71:7171
-------------------------------------------------------------------------------
Number of entries : 1
-------------------------------------------------------------------------------
===============================================================================
```

In the FDB for the R-VPLS 301, the MAC address is associated with the VTEP and VNI specified by the AD route, and not by the MAC/IP route anymore. When a PBR filter with a forward action to an ESI and SF-IP (Service Function IP) exists, a MAC route is auto-created by the system and this route has higher priority that the remote MAC, or IP routes for the MAC (see BGP and EVPN route selection for EVPN routes).

The following shows that the AD route creates a new EVPN-VXLAN binding and the MAC address associated with the SF-IP uses that 'binding':

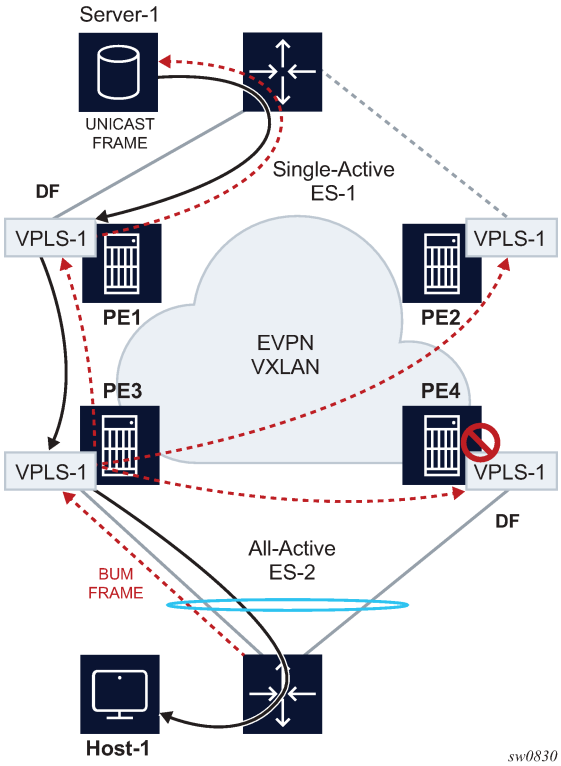
```
*A:PE1# show service id 301 vxlan
===============================================================================
VPLS VXLAN, Ingress VXLAN Network Id: 301
===============================================================================
Egress VTEP, VNI
===============================================================================
VTEP Address           Egress VNI     Num. MACs    Mcast   Oper State   L2 PBR
-------------------------------------------------------------------------------
192.0.2.69             301            1            Yes     Up           No
192.0.2.71             301            0            Yes     Up           No
192.0.2.71             7171           1            No      Up           No
-------------------------------------------------------------------------------
Number of Egress VTEP, VNI : 3
-------------------------------------------------------------------------------
===============================================================================
*A:PE1# show service id 301 fdb detail
===============================================================================
Forwarding Database, Service 301
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
301       d8:45:ff:00:00:6a vxlan-1:                 EvpnS    06/15/15 21:55:27
                            192.0.2.69:301
301       d8:47:01:01:00:0a vxlan-1:                 EvpnS    06/15/15 22:32:56
                            192.0.2.71:7171
301       d8:48:ff:00:00:6a cpm                      Intf     06/15/15 21:54:12
-------------------------------------------------------------------------------
No. of MAC Entries: 3
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
===============================================================================
```

For Layer 2, if the AD route is withdrawn or the SF-IP ARP not resolved, the filter is inactive again. The user can control whether the matching packets are dropped or forwarded if the PBF target cannot be resolved by BGP.

6.2.5 EVPN VXLAN multihoming

SR OS supports EVPN VXLAN multihoming as specified in RFC8365. Similar to EVPN-MPLS, as described in EVPN for MPLS tunnels, ESs and virtual ESs can be associated with VPLS and R-VPLS services where BGP-EVPN VXLAN is enabled. Figure 152: EVPN multihoming for EVPN-VXLAN illustrates the use of ESs in EVPN VXLAN networks.

Figure 152: EVPN multihoming for EVPN-VXLAN



As described in EVPN multihoming in VPLS services, the multihoming procedures consist of three components:

- Designated Forwarder (DF) election
- split-horizon
- aliasing

DF election is the mechanism by which the PEs attached to the same ES elect a single PE to forward all traffic (in case of single-active mode) or all BUM traffic (in case of all-active mode) to the multihomed CE. The same DF Election mechanisms described in EVPN for MPLS tunnels are supported for VXLAN services.

Split-horizon is the mechanism by which BUM traffic received from a peer ES PE is filtered so that it is not looped back to the CE that first transmitted the frame. It is applicable to all-active multihoming. This is illustrated in Figure 152: EVPN multihoming for EVPN-VXLAN, where PE4 receives BUM traffic from PE3 but, in spite of being the DF for ES-2, PE4 filters the traffic and does not send it back to host-1. While split-horizon filtering uses ESI-labels in EVPN MPLS services, an alternative procedure called "Local Bias" is applied in VXLAN services, as described in RFC 8365. In MPLS services, split-horizon filtering may be used in single-active mode to avoid in-flight BUM packets from being looped back to the CE during transient times. In VXLAN services, split-horizon filtering is only used with all-active mode.

Aliasing is the procedure by which PEs that are not attached to the ES can process non-zero MAC/IP and AD routes and create ES destinations to which per-flow ecmp can be applied. Aliasing only applies to all-active mode.

As an example, the configuration of an ES that is used for VXLAN services follows. Note that this ES can be used for VXLAN services and MPLS services (in both cases VPLS and Epipes).

```
A:PE-3# configure service system bgp-evpn ethernet-segment "ES-2"
A:PE-3>config>service>system>bgp-evpn>eth-seg# info
----------------------------------------------
                esi 01:02:00:00:00:00:00:00:00:00
                service-carving
                    mode manual
                    manual
                        preference non-revertive create
                            value 10
                        exit
                    exit
                exit
                multi-homing all-active
                lag 1
                no shutdown
----------------------------------------------
```

An example of configuration of a VXLAN service using the above ES follows:

```
A:PE-3# configure service vpls 1
A:PE-3>config>service>vpls# info
----------------------------------------------
            vxlan instance 1 vni 1 create
            exit
            bgp
            exit
            bgp-evpn
                evi 1
                vxlan bgp 1 vxlan-instance 1
                    ecmp 2
                    auto-disc-route-advertisement
                    mh-mode network
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            sap lag-1:30 create
                no shutdown
            exit
            no shutdown
----------------------------------------------
```

The **auto-disc-route-advertisement** and **mh-mode network** commands are required in all services that are attached to at least one ES, and they must be configured in both, the PEs attached to the ES locally and the remote PEs in the same service. The former enables the advertising of multihoming routes in the service, whereas the latter activates the multihoming procedures for the service, including the local bias mode for split-horizon.

In addition, the configuration of **vpls**>**bgp-evpn**>**vxlan**>**ecmp 2** (or greater) is required so that VXLAN ES destinations with two or more next hops can be used for per-flow load balancing. The following command shows how PE1, as shown in Figure 152: EVPN multihoming for EVPN-VXLAN, creates an ES destination composed of two VXLAN next hops.

```
A:PE-1# show service id 1 vxlan destinations
===============================================================================
```

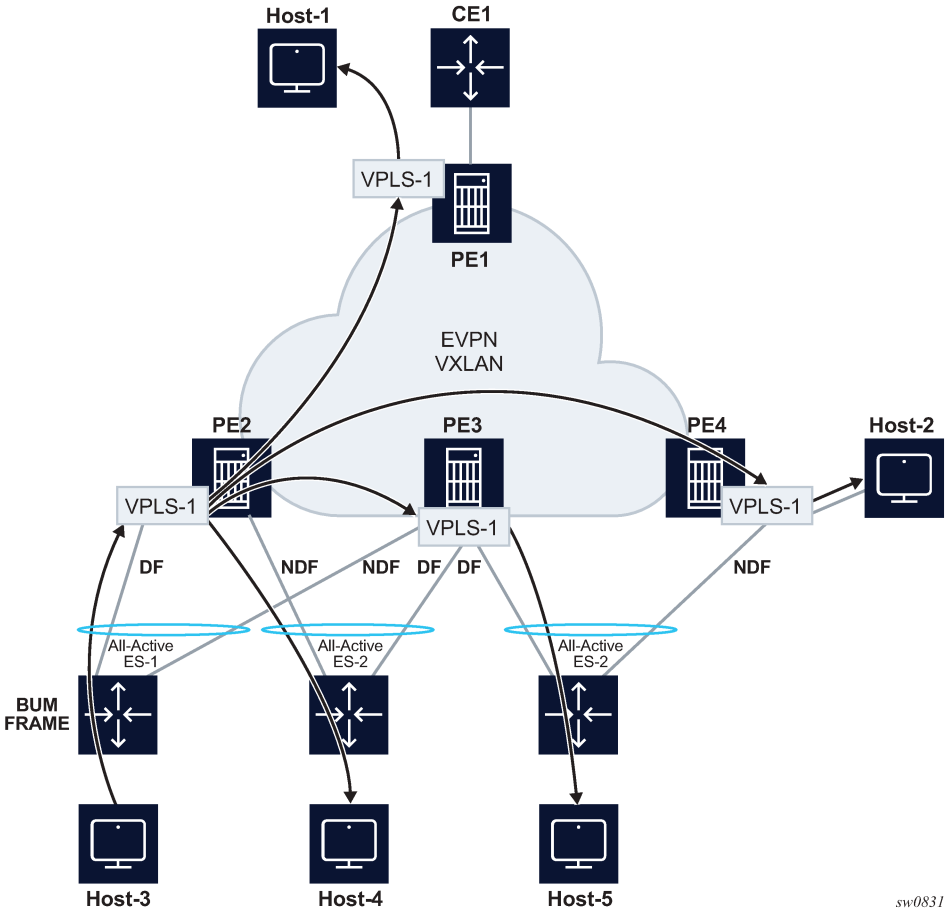
```
Egress VTEP, VNI
==============================================================================
Instance    VTEP Address                          Egress VNI  Evpn/   Num.
 Mcast       Oper State                            L2 PBR     Static  MACs
------------------------------------------------------------------------------
1           192.0.2.3                             1          evpn    0
 BUM         Up                                    No
1           192.0.2.4                             1          evpn    0
 BUM         Up                                    No
------------------------------------------------------------------------------
Number of Egress VTEP, VNI : 2
------------------------------------------------------------------------------
==============================================================================
==============================================================================
BGP EVPN-VXLAN Ethernet Segment Dest
==============================================================================
Instance  Eth SegId                       Num. Macs     Last Change
------------------------------------------------------------------------------
1         01:02:00:00:00:00:00:00:00:00   1             04/01/2019 08:54:54
------------------------------------------------------------------------------
Number of entries: 1
------------------------------------------------------------------------------
==============================================================================

A:PE-1# show service id 1 vxlan esi 01:02:00:00:00:00:00:00:00:00
==============================================================================
BGP EVPN-VXLAN Ethernet Segment Dest
==============================================================================
Instance  Eth SegId                       Num. Macs     Last Change
------------------------------------------------------------------------------
1         01:02:00:00:00:00:00:00:00:00   1             04/01/2019 08:54:54
------------------------------------------------------------------------------
Number of entries: 1
------------------------------------------------------------------------------
==============================================================================
==============================================================================
BGP EVPN-VXLAN Dest TEP Info
==============================================================================
Instance  TEP Address                   Egr VNI             Last Change
------------------------------------------------------------------------------
1         192.0.2.3                     1                   04/01/2019 08:54:54
1         192.0.2.4                     1                   04/01/2019 08:54:54
------------------------------------------------------------------------------
Number of entries : 2
------------------------------------------------------------------------------
==============================================================================
```

6.2.5.1 Local bias for EVPN VXLAN multihoming

EVPN MPLS, as described in EVPN for MPLS tunnels, uses ESI-labels to identify the BUM traffic sourced from a specified ES. The egress PE performs a label lookup to find the ESI label below the EVI label and to determine if a frame can be forwarded to a local ES. Because VXLAN does not support ESI-labels, or any MPLS label for that matter, the split-horizon filtering must be based on the tunnel source IP address. This also implies that the SAP-to-SAP forwarding rules must be changed when the SAPs belong to local ESs, irrespective of the DF state. This new forwarding is what RFC 8365 refers to as local bias. Figure 153: EVPN-VXLAN multihoming with local bias illustrates the local bias forwarding behavior.

Figure 153: EVPN-VXLAN multihoming with local bias



Local bias is based on the following principles:

- Every PE knows the IP addresses associated with the other PEs with which it has shared multihomed ESs.
- When the PE receives a BUM frame from a VXLAN bind, it looks up the source IP address in the tunnel header and filters out the frame on all local interfaces connected to ESs that are shared with the ingress PE.

With this approach, the ingress PE must perform replication locally to all directly-attached ESs (regardless of the DF Election state) for all flooded traffic coming from the access interfaces. BUM frames received on any SAP are flooded to:

- local non-ES SAPs and non-ES SDP-binds
- local all-active ES SAPs (DF and NDF)
- local single-active ES SDP-binds and SAPs (DF only)
- EVPN-VXLAN destinations

As an example, in Figure 153: EVPN-VXLAN multihoming with local bias, PE2 receives BUM traffic from Host-3 and it forwards it to the remote PEs and the local ES SAP, even though the SAP is in NDF state.

The following rules apply to egress PE forwarding for EVPN-VXLAN services:

- The source VTEP is looked up for BUM frames received on EVPN-VXLAN.
- If the source VTEP matches one of the PEs with which the local PE shares both an ES and a VXLAN service:
  - the local PE is not forwarded to the shared ES local SAPs
  - the local PE forwards normally to ES SAPs unless they are in NDF state
- Because there is no multicast label or multicast B-MAC in VXLAN, the egress PE only identifies BUM traffic using the customer MAC DA; as a result, BM or unknown MAC DAs identify BUM traffic.

For example, in Figure 153: EVPN-VXLAN multihoming with local bias, PE3 receives BUM traffic on VXLAN. PE3 identifies the source VTEP as a PE with which two ESs are shared, therefore it does not forward the BUM frames to the two shared ESs. It forwards to the non-shared ES (Host-5) because it is in DF state. PE4 receives BUM traffic and forwards it based on normal rules because it does not share any ESs with PE2.

The following command can be used to check whether the local PE has enabled the local bias procedures for a specific ES:

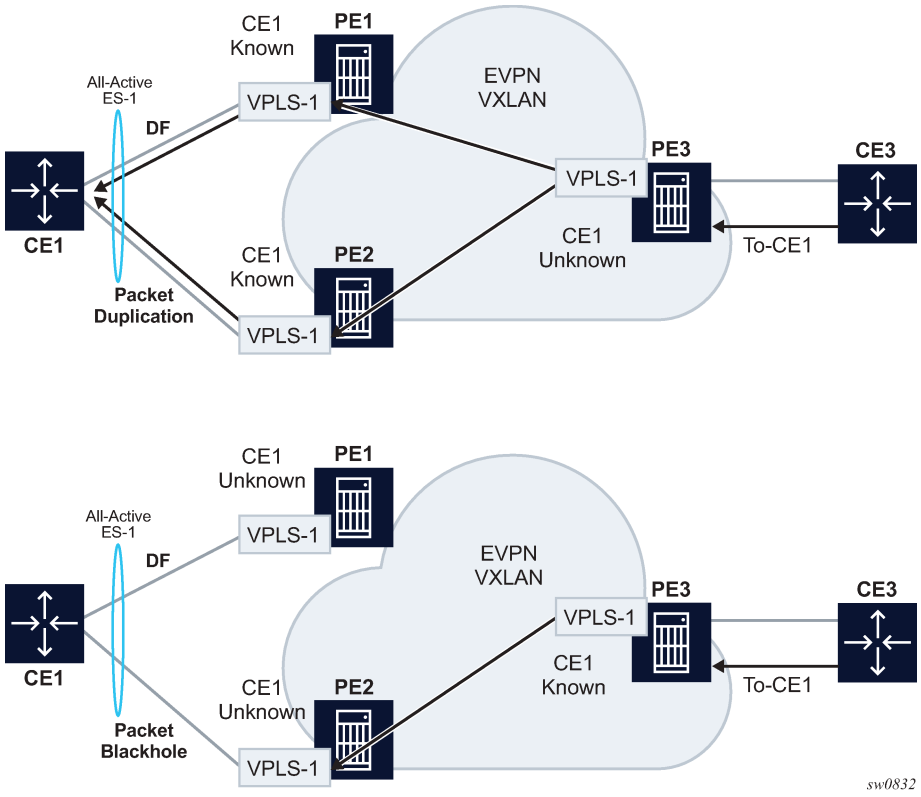
```
A:PE-2# tools dump service system bgp-evpn ethernet-segment "ES-1" local-bias
------------------------------------------------------------------------------
[04/01/2019 08:45:08] Vxlan Local Bias Information
------------------------------------------------------------------------+--------
Peer                                                                    | Enabled
------------------------------------------------------------------------+--------
192.0.2.3                                                               | Yes
------------------------------------------------------------------------------
```

6.2.5.2 Known limitations for local bias

In EVPN MPLS networks, an ingress PE that uses ingress replication to flood unknown unicast traffic pushes a BUM MPLS label that is different from a unicast label. The egress PEs use this BUM label to identify such BUM traffic to apply DF filtering for All-Active multihomed sites. In PBB-EVPN, in addition to the multicast label, the egress PE can also rely on the multicast B-MAC DA to identify customer BUM traffic.

In VXLAN there are no BUM labels or any tunnel indication that can assist the egress PE in identifying the BUM traffic. As such, the egress PE must solely rely on the C-MAC destination address, which may create some transient issues that are depicted in Figure 154: EVPN-VXLAN multihoming and unknown unicast issues.

Figure 154: EVPN-VXLAN multihoming and unknown unicast issues



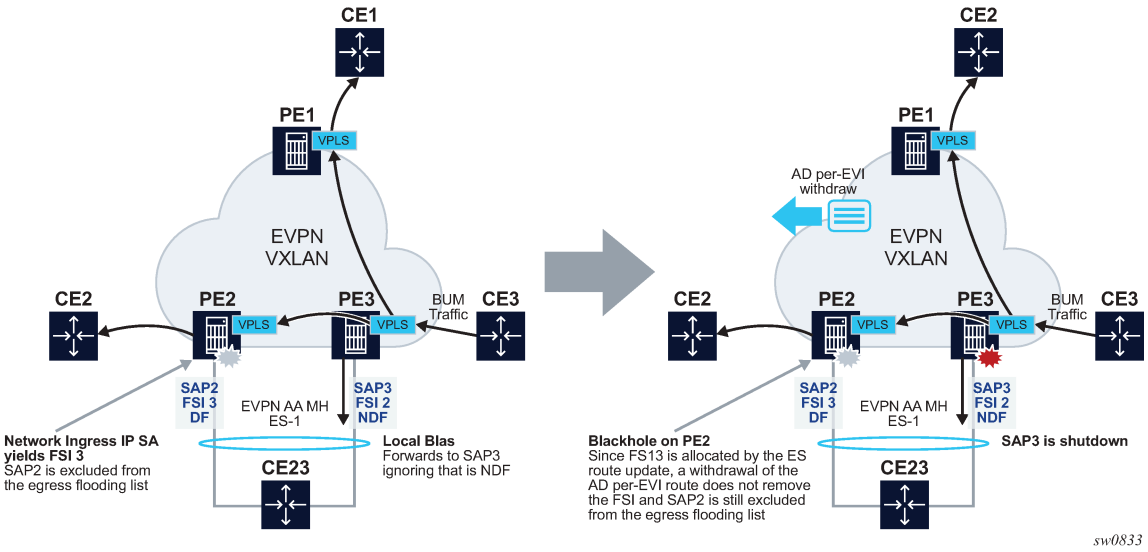
As shown in Figure 154: EVPN-VXLAN multihoming and unknown unicast issues, top diagram, in absence of the mentioned unknown unicast traffic indication there can be transient duplicate traffic to All-Active multihomed sites under the following condition: CE1's MAC address is learned by the egress PEs (PE1 and PE2) and advertised to the ingress PE3; however, the MAC advertisement has not been received or processed by the ingress PE, resulting in the host MAC address to be unknown on the ingress PE3 but known on the egress PEs. Therefore, when a packet destined for CE1 address arrives on PE3, it floods it through ingress replication to PE1 or PE2 and, because CE1's MAC is known to PE1 and PE2, multiple copies are sent to CE1.

Another issue is shown at the bottom of Figure 154: EVPN-VXLAN multihoming and unknown unicast issues. In this case, CE1's MAC address is known on the ingress PE3 but unknown on PE1 and PE2. If PE3's aliasing hashing picks up the path to the ES' NDF, a black-hole occurs.

The above two issues are solved in MPLS, as unicast known and unknown frames are identified with different labels.

Finally, another issue is described in Figure 155: Blackhole created by a remote SAP shutdown. Under normal circumstances, when CE3 sends BUM traffic to PE3, the traffic is "local-biased" to PE3's SAP3 even though it is NDF for the ES. The flooded traffic to PE2 is forwarded to CE2, but not to SAP2 because the local bias split-horizon filtering takes place.

Figure 155: Blackhole created by a remote SAP shutdown



The right side of the diagram in Figure 155: Blackhole created by a remote SAP shutdown shows an issue when SAP3 is manually shutdown. In this case, PE3 withdraws the AD per-EVI route corresponding to SAP3; however, this does not change the local bias filtering for SAP2 in PE2. Therefore, when CE3 sends BUM traffic, it can neither be forwarded to CE23 via local SAP3 nor can it be forwarded by PE2.

6.2.5.3 Non-system IPv4 and IPv6 VXLAN termination for EVPN VXLAN multihoming

EVPN VXLAN multihoming is supported on VPLS and R-VPLS services when the PEs use non-system IPv4 or IPv6 termination, however, as with EVPN VPWS services, additional configuration steps are required.

- The **configure service system bgp-evpn eth-seg es-orig-ip ip-address** command must be configured with the non-system IPv4 or IPv6 address used for the EVPN-VXLAN service. This command modifies the originating-ip field in the ES routes advertised for the Ethernet Segment, and makes the system use this IP address when adding the local PE as DF candidate.
- The **configure service system bgp-evpn eth-seg route-next-hop ip-address** command must also be configured with the non-system IP address. This command changes the next-hop of the ES and AD per-ES routes to the configured address.
- Finally, the non-system IP address (in each of the PEs in the ES) must match in these three commands for the local PE to be considered suitable for DF election:
  - **es-orig-ip ip-address**
  - **route-next-hop ip-address**
  - **vxlan-src-vtep ip-address**

6.3 EVPN for MPLS tunnels

This section provides information about EVPN for MPLS tunnels.

6.3.1 BGP-EVPN control plane for MPLS tunnels

Table 20: EVPN routes and usage lists all the EVPN routes supported in 7750 SR, 7450 ESS, or 7950 XRS SR OS and their usage in EVPN-VXLAN, EVPN-MPLS, and PBB-EVPN.

Note: Route type 1 is not required in PBB-EVPN as per RFC 7623.

Table 20: EVPN routes and usage

EVPN route	Usage	EVPN-VXLAN	EVPN-MPLS	PBB-EVPN
Type 1 - Ethernet Auto-Discovery route (A-D)	Mass-withdraw, ESI labels, Aliasing	Y	Y	—
Type 2 - MAC/IP Advertisement route	MAC/IP advertisement, IP advertisement for ARP resolution	Y	Y	Y
Type 3 - Inclusive Multicast Ethernet Tag route	Flooding tree setup (BUM flooding)	Y	Y	Y
Type 4 - ES route	ES discovery and DF election	Y	Y	Y
Type 5 - IP Prefix advertisement route	IP Routing	Y	Y	—
Type 6 - Selective Multicast Ethernet Tag route	Signal interest on a multicast group	Y	Y	—
Type 7 - Multicast Join Synch route	Join a multicast group on a multihomed ES	Y	Y	—
Type 8 - Multicast Leave Synch route	Leave a multicast group on a multihomed ES	Y	Y	—
Type 10 - Selective Provider Multicast Service Interface Auto-Discovery route	Signal and setup Selective Provider Tunnels for IP Multicast	-	Y	-

RFC 7432 describes the BGP-EVPN control plane for MPLS tunnels. If EVPN multihoming is not required, two route types are needed to set up a basic EVI (EVPN Instance): MAC/IP Advertisement and the Inclusive Multicast Ethernet Tag routes. If multihoming is required, the ES and the Auto-Discovery routes are also needed.

The route fields and extended communities for route types 2 and 3 are shown in Figure 142: EVPN-VXLAN required routes and communities. BGP-EVPN control plane for VXLAN overlay tunnels The changes compared to their use in EVPN-VXLAN are described below.

EVPN route type 3 - inclusive multicast Ethernet tag route

As in EVPN-VXLAN, route type 3 is used for setting up the flooding tree (BUM flooding) for a specified VPLS service. The received inclusive multicast routes add entries to the VPLS flood list in the 7750 SR, 7450 ESS, and 7950 XRS. Ingress replication, p2mp mLDP, and composite tunnels are supported as tunnel types in route type 3 when BGP-EVPN MPLS is enabled

The following route values are used for EVPN-MPLS services:

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context. The RD can be configured or derived from the **bgp-evpn evi** value.
- Ethernet Tag ID is 0.
- IP address length is always 32.
- Originating router's IP address carries an IPv4 or IPv6 address.
- The PMSI attribute can have different formats depending on the tunnel type enabled in the service.
  - **Tunnel type = Ingress replication (6)**

    The route is referred to as an Inclusive Multicast Ethernet Tag IR (IMET-IR) route and the PMSI Tunnel Attribute (PTA) fields are populated as follows:

    - Leaf not required for Flags.
    - MPLS label carries the MPLS label allocated for the service in the high-order 20 bits of the label field.

      Unless **bgp-evpn mpls ingress-replication-bum-label** is configured in the service, the MPLS label used is the same as that used in the MAC/IP routes for the service.
    - Tunnel endpoint is equal to the originating IP address.
  - **Tunnel type=p2mp mLDP (2)**

    The route is referred to as an IMET-P2MP route and its PTA fields are populated as follows:

    - Leaf not required for Flags.
    - MPLS label is 0.
    - Tunnel endpoint includes the route node address and an opaque number. This is the tunnel identifier that the leaf-nodes use to join the mLDP P2MP tree.
  - **Tunnel type=Composite tunnel (130)**

    The route is referred to as an IMET-P2MP-IR route and its PTA fields are populated as follows:

    - Leaf not required for Flags.
    - MPLS label 1 is 0.
    - Tunnel endpoint identifier includes the following:

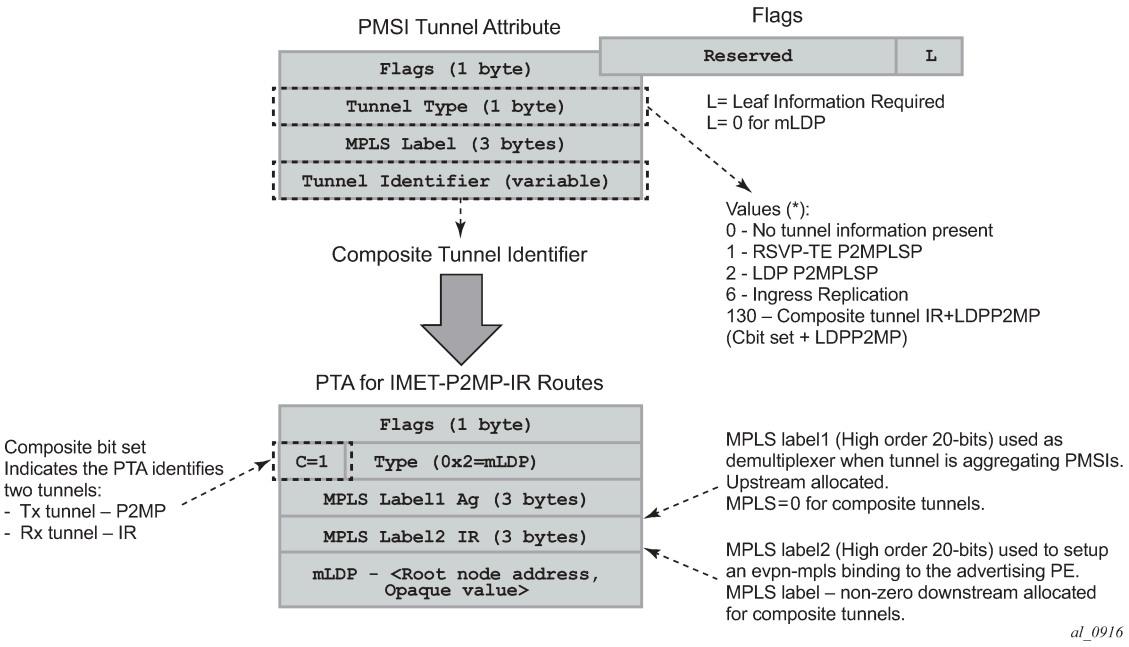
MPLS label2	non-zero, downstream allocated label (like any other IR label). The leaf-nodes use the label to set up an EVPN-MPLS destination to the root and add it to the default-multicast list.
mLDP tunnel identifier	the route node address and an opaque number. This is the tunnel identifier that the leaf-nodes use to join the mLDP P2MP tree.

IMET-P2MP-IR routes are used in EVIs with a few root nodes and a significant number of leaf-only PEs. In this scenario, a combination of P2MP and IR tunnels can be used in the network, such that the root nodes use P2MP tunnels to send broadcast, Unknown unicast, and Multicast traffic but the leaf-PE nodes use

IR to send traffic to the roots. This use case is documented in IETF RFC 8317 and the main advantage it offers is the significant savings in P2MP tunnels that the PE/P routers in the EVI need to handle (as opposed to a full mesh of P2MP tunnels among all the PEs in an EVI).

In this case, the root PEs signals a special tunnel type in the PTA, indicating that they intend to transmit BUM traffic using an mLDP P2MP tunnel but they can also receive traffic over an IR evpn-mpls binding. An IMET route with this special "composite" tunnel type in the PTA is called an IMET-P2MP-IR route and the encoding of its PTA is shown in Figure 156: Composite p2mp mLDP and IR tunnels—PTA.

Figure 156: Composite p2mp mLDP and IR tunnels—PTA



EVPN route type 2 - MAC/IP advertisement route

The 7750 SR, 7450 ESS, or 7950 XRS router generates this route type for advertising MAC addresses (and IP addresses if proxy-ARP/proxy-ND is enabled). If mac-advertisement is enabled, the router generates MAC advertisement routes for the following:

- learned MACs on SAPs or SDP bindings
- conditional static MACs

Note: The **unknown-mac-route** is not supported for EVPN-MPLS services.

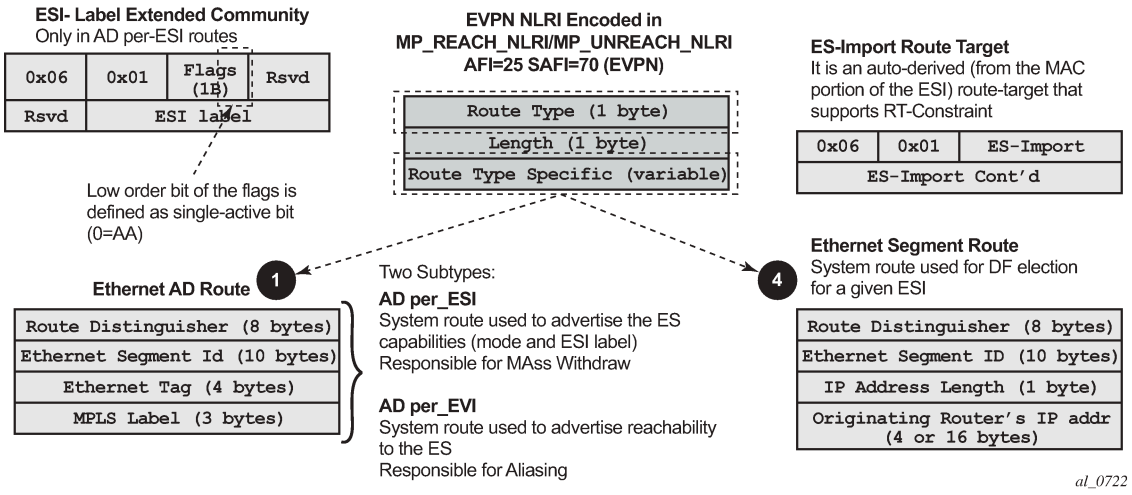
The route type 2 generated by a router uses the following fields and values:

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context. The RD can be configured or derived from the **bgp-evpn evi** value.
- Ethernet Segment Identifier (ESI) is zero for MACs learned from single-homed CEs and different from zero for MACs learned from multihomed CEs.
- Ethernet Tag ID is 0.
- MAC address length is always 48.

- MAC address can be learned or statically configured.
- IP address and IP address length:
  - It is the IP address associated with the MAC being advertised with a length of 32 (or 128 for IPv6).
  - In general, any MAC route without IP has IPL=0 (IP length) and the IP is omitted.
  - When received, any IPL value not equal to zero, 32, or 128 discards the route.
  - MPLS Label 1 carries the MPLS label allocated by the system to the VPLS service. The label value is encoded in the high-order 20 bits of the field and is the same label used in the routes type 3 for the same service unless **bgp-evpn mpls ingress-replication-bum-label** is configured in the service.
- MPLS Label 2 is 0.
- The MAC mobility extended community is used for signaling the sequence number in case of MAC moves and the sticky bit in case of advertising conditional static MACs. If a MAC route is received with a MAC mobility **ext-community**, the sequence number and the 'sticky' bit are considered for the route selection.

When EVPN multihoming is enabled in the system, two more routes are required. Figure 157: EVPN routes type 1 and 4 shows the fields in routes type 1 and 4 and their associated extended communities.

Figure 157: EVPN routes type 1 and 4



EVPN route type 1 - Ethernet auto-discovery route (AD route)

The 7750 SR, 7450 ESS, or 7950 XRS router generates this route type for advertising for multihoming functions. The system can generate two types of AD routes:

- Ethernet AD route per-ESI (Ethernet Segment ID)
- Ethernet AD route per-EVI (EVPN Instance)

The Ethernet AD per-ESI route generated by a router uses the following fields and values:

- Route Distinguisher is taken from the system level RD or service level RD.
- Ethernet Segment Identifier (ESI) contains a 10-byte identifier as configured in the system for a specified **ethernet-segment**.
- Ethernet Tag ID is MAX-ET (0xFFFFFFFF). This value is reserved and used only for AD routes per ESI.

- MPLS label is 0.
- ESI Label Extended community includes the single-active bit (0 for all-active and 1 for single-active) and ESI label for all-active multihoming split-horizon.
- Route target extended community is taken from the service level RT or an RT-set for the services defined on the Ethernet segment.

The system can either send a separate Ethernet AD per-ESI route per service, or a few Ethernet AD per-ESI routes aggregating the route-targets for multiple services. While both alternatives inter-operate, RFC 7432 states that the EVPN Auto-Discovery per-ES route must be sent with a set of route-targets corresponding to all the EVIs defined on the Ethernet Segment (ES). Either option can be enabled using the command: **config**>**service**>**system**>**bgp-evpn#ad-per-es-route-target** <[**evi-rt**] | [**evi-rt-set**]> **route-distinguisher** *ip-address* [**extended-evi-range**]

The default option **ad-per-es-route-target evi-rt** configures the system to send a separate AD per-ES route per service. When enabled, the **evi-rt-set** option supports route aggregation: a single AD per-ES route with the associated RD (**ip-address:1**) and a set of EVI route targets are advertised (up to a maximum of 128). When the number of EVIs defined in the Ethernet Segment is significant (therefore the number of route-targets), the system sends more than one route. For example:

- AD per-ES route for **evi-rt-set 1** is sent with RD **ip-address:1**
- AD per-ES route for **evi-rt-set 2** is sent with RD **ip-address:2**
- up to an AD per-ES route is sent with RD **ip-address:512**

The **extended-evi-range** option is needed for the use of **evi-rt-set** with a **comm-val** extended range of 1 through 65535. This option is recommended when EVIs greater than 65535 are configured in some services. In this case, there are more EVIs for which the route-targets must be packed in the AD per-ES routes. This command option extends the maximum number of AD per-ES routes that can be sent (since the RD now supports up to ip-address:65535) and allows many more route-targets to be included in each set.

Note: When **evi-rt-set** is configured, no vsi-export policies are possible on the services defined on the Ethernet Segment. If vsi-export policies are configured for a service, the system sends an individual AD per-ES route for that service. The maximum standard BGP update size is 4KB, with a maximum of 2KB for the route-target extended community attribute.

The Ethernet AD per-EVI route generated by a router uses the following fields and values:

- Route Distinguisher is taken from the service level RD.
- Ethernet Segment Identifier (ESI) contains a 10-byte identifier as configured in the system for a specified Ethernet Segment.
- Ethernet Tag ID is 0.
- MPLS label encodes the unicast label allocated for the service (high-order 20 bits).
- Route-target extended community is taken from the service level RT.

Note: The AD per-EVI route is not sent with the ESI label Extended Community.

EVPN route type 4 - ES route

The router generates this route type for multihoming ES discovery and DF (Designated Forwarder) election.

- Route Distinguisher is taken from the service level RD.

- Ethernet Segment Identifier (ESI) contains a 10-byte identifier as configured in the system for a specified **ethernet-segment**.
- The value of ES-import route-target community is automatically derived from the MAC address portion of the ESI. This extended community is treated as a route-target and is supported by RT-constraint (route-target BGP family).

EVPN route type 5 - IP prefix route

IP Prefix Routes are also supported for MPLS tunnels. The route fields for route type 5 are shown in Figure 144: EVPN route-type 5 . The 7750 SR, 7450 ESS, or 7950 XRS router generates this route type for advertising IP prefixes in EVPN using the same fields that are described in section BGP-EVPN control plane for VXLAN overlay tunnels, with the following exceptions:

- MPLS label carries the MPLS label allocated for the service.
- This route is sent with the RFC 5512 tunnel encapsulation extended community with the tunnel type value set to MPLS

RFC 5512 - BGP tunnel encapsulation extended community

The following routes are sent with the RFC 5512 BGP Encapsulation Extended Community: MAC/IP, Inclusive Multicast Ethernet Tag, and AD per-EVI routes. ES and AD per-ESI routes are not sent with this Extended Community.

The router processes the following BGP Tunnel Encapsulation tunnel values registered by IANA for RFC 5512:

- VXLAN encapsulation is 8.
- MPLS encapsulation is 10.

Any other tunnel value makes the route 'treat-as-withdraw'.

If the encapsulation value is MPLS, the BGP validates the high-order 20-bits of the label field, ignoring the low-order 4 bits. If the encapsulation is VXLAN, the BGP takes the entire 24-bit value encoded in the MPLS label field as the VNI.

If the encapsulation extended community (as defined in RFC 5512) is not present in a received route, BGP treats the route as an MPLS or VXLAN-based configuration of the **config**>**router**>**bgp**>**neighbor# def-recv-evpn-encap [mpls | vxlan]** command. The command is also available at the **bgp** and **group** levels.

6.3.2 EVPN for MPLS tunnels in VPLS services (EVPN-MPLS)

EVPN can be used in MPLS networks where PEs are interconnected through any type of tunnel, including RSVP-TE, Segment-Routing TE, LDP, BGP, Segment Routing IS-IS, Segment Routing OSPF, RIB-API, MPLS-forwarding-policy, SR-Policy, or MPLSoUDP. As with VPRN services, tunnel selection for a VPLS service (with BGP-EVPN MPLS enabled) is based on the **auto-bind-tunnel** command. The BGP EVPN routes next-hops can be IPv4 or IPv6 addresses and can be resolved to a tunnel in the IPv4 tunnel-table or IPv6 tunnel-table.

EVPN-MPLS is modeled similar to EVPN-VXLAN, that is, using a VPLS service where EVPN-MPLS "bindings" can coexist with SAPs and SDP bindings. The following shows an example of a VPLS service with EVPN-MPLS.

```
*A:PE-1>config>service>vpls# info
----------------------------------------------
  description "evpn-mpls-service"
```

```
bgp
exit
bgp-evpn
  evi 10
  mpls bgp 1
    no shutdown
    auto-bind-tunnel resolution any
exit
sap 1/1/1:1 create
exit
spoke-sdp 1:1 create
```

First configure a **bgp-evpn** context where VXLAN must be disabled and MPLS enabled. In addition to enabling MPLS the command, the minimum set of commands to be configured to set up the EVPN-MPLS instance are the **evi** and the **auto-bind-tunnel resolution** commands. The relevant configuration options are the following.

evi {1..16777215} — This EVPN identifier is unique in the system and is used for the service-carving algorithm used for multihoming (if configured), and for auto-deriving the route target and route distinguishers (if lower than 65535) in the service. It can be used for EVPN-MPLS and EVPN-VXLAN services.

The following options are supported:

- If this EVPN identifier is not specified, the value is zero and no route distinguisher or route target is automatically derived from it.
- If the specified EVPN identifier is lower than 65535 and no other route distinguisher or route target is configured in the service, the following applies:
  - The route distinguisher is derived from <system_ip>:evi.
  - The route target is derived from <autonomous-system>:evi.
- If the specified EVPN identifier is higher than 65535 and no other route distinguisher or route target is configured in the service, the following applies:
  - The route distinguisher cannot be automatically derived. An error is generated if enabling EVPN is attempted without a route distinguisher. A manual or an **auto-rd** route distinguisher must be configured.
  - The route target can only be automatically derived if the **evi-three-byte-auto-rt** command is configured. If configured, the route target is automatically derived in accordance with the following rules described in RFC8365.
    - The route target is composed of ASN(2-octets):A/type/D-ID/EVI.
    - The ASN is a 2-octect value configured in the system. For AS numbers exceeding the 2-byte limit, the low order 16-bit value is used.
    - The A=0 value is used for auto-derivation.
    - The type=4 (EVI-based) is used.
    - The BGP instance is encoded using D-ID= [1..2]. This allows the automatic derivation of different RTs in multi-instance services. The value is inherited from the corresponding BGP instance.
    - EVI indicates the configured EVI in the service

For example, consider a service with the following characteristics:

- ASN=64500
- VPLS with two BGP instances, bgp 1 for VXLAN-instance 1 and bgp 2 for EVPN-MPLS

- EVI=100000

The automatically derived route targets for this service are:

- bgp 1 — 64500:1090619040 (ASN:0x410186A0)
- bgp 2 — 64500:1107396256 (ASN:0x420186A0)

If this EVPN identifier is not specified, the value is zero and no route distinguisher or route targets is automatically derived from it. If specified and no other route distinguisher/route target are configured in the service:, then the following applies:

- the route distinguisher is derived from: **<system_ip>:evi**
- the route target is derived from: **<autonomous-system>:evi**

Note: When the vsi-import/export polices are configured, the route target must be configured in the policies and those values take preference over the automatically derived route targets. The operational route target for a service is displayed by the **show service id svc-id bgp** command. If the **bgp-ad vpls-id** is configured in the service, the **vpls-id** derived route target takes precedence over the evi-derived route target.

When the **evi** is configured, a **configure service vpls bgp** node (even empty) is required to allow the user to see the correct information about the **show service id 1 bgp** and **show service system bgp-route-distinguisher** commands.

The configuration of an **evi** is enforced for EVPN services with SAPs/SDP bindings in an **ethernet-segment**. See EVPN multihoming in VPLS services for more information about ESs.

The following options are specific to EVPN-MPLS (and defined in **configure service vpls bgp-evpn mpls**):

- **control word**

  Enable or disable control word capability to guarantee interoperability to other vendors. When enabled along with the following command, the control word capability is signaled in the C flag of the EVPN Layer 2 Attributes extended community, as per draft-ietf-bess-rfc7432bis;

  – **MD-CLI**

  ```
  configure service vpls bgp-evpn routes incl-mcast advertise-l2-attributes
  ```

  – **classic CLI**

  ```
  configure service vpls bgp-evpn incl-mcast-l2-attributes-advertisement
  ```

  On reception, the router compares the C flag with the local setting for **control-word**. In case of a mismatch, the EVPN destination goes operationally down with the corresponding operational flag indicating the reason.

  Note: The **control-word** is required as per RFC 7432 to avoid frame disordering.

- **auto bind tunnel**

  Select which type of MPLS transport tunnel to use for a particular instance; this command is used in the same way as in VPRN services.

  For BGP-EVPN MPLS, you must explicitly add BGP to the resolution filter in EVPN (BGP is implicit in VPRNs).

- **force VLAN VC forwarding**

  This option allows the system to preserve the VLAN ID and pbits of the service-delimiting qtag in a new tag added in the customer frame before sending it to the EVPN core.

  Note: You can use this option in conjunction with the **sap ingress vlan-translation** command. If so, the configured translated VLAN ID is sent to the EVPN binds as opposed to the service-delimiting tag VLAN ID. If the ingress SAP/binding is null-encapsulated, the output VLAN ID and pbits are zero.

- **force QinQ VC forwarding with c-tag-c-tag or s-tag-c-tag**

  This command allows the system to preserve the VLAN ID and pbits of the service-delimiting Q-tags (up to two tags) in customer frames before sending them to the EVPN core.

  Note:
  You can use this option in conjunction with the **sap ingress qinq-vlan-translation s-tag.c-tag** command. If so, the configured translated S-tag and C-tag VLAN IDs are the VLAN IDs sent to the EVPN binds as opposed to the service-delimiting tags VLAN IDs. If the ingress SAP or binding is null-encapsulated, the output VLAN ID and pbits are zero.

- **split horizon group**

  This command allows the association of a user-created split horizon group to all the EVPN-MPLS destinations. See EVPN and VPLS integration for more information.

- **ecmp**

  Set this option to a value greater than 1 to activate aliasing to the remote PEs that are defined in the same all-active multihoming ES. See EVPN all-active multihoming for more information.

- **ingress replication bum label**

  You can use this option when you want the PE to advertise a label for BUM traffic (Inclusive Multicast routes) that is different from the label advertised for unicast traffic (with the MAC/IP routes). This is useful to avoid potential transient packet duplication in all-active multihoming.

In addition to these options, the following BGP EVPN options are also available for EVPN-MPLS services:

- **mac-advertisement**
- **mac-duplication** and settings
- **incl-mcast advertise-l2-attributes** (MD-CLI)

  incl-mcast-l2-attributes-advertisement (classic CLI)

  This function enables the advertisement and processing of the EVPN Layer 2 Attributes extended community. The control-word configuration and the Service-MTU value are advertised in the extended community. On reception, the received MTU and control-word flag are compared with the local MTU and control-word configuration. In case of a mismatch in any of the two settings, the EVPN destination goes down with the corresponding operational flag indicating what the mismatch is. The absence of an IMET route from an egress PE or the absence of the EVPN L2 Attributes extended community on a received IMET route from the PE, causes the route to bring down the EVPN destinations to that PE.

- **ignore-mtu-mismatch**

  This command makes the router ignore the received Layer 2 MTU in the EVPN L2 Attributes extended community of the IMET route for a peer. If disabled, the local service MTU is compared against the received Layer 2 MTU. If there is a mismatch, the EVPN destinations to the peer stay oper-state down.

When EVPN-MPLS is established among some PEs in the network, EVPN unicast and multicast 'bindings' are created on each PE to the remote EVPN destinations. A specified ingress PE creates:

- A unicast EVPN-MPLS destination binding to a remote egress PE as soon as a MAC/IP route is received from that egress PE.
- A multicast EVPN-MPLS destination binding to a remote egress PE, if and only if the egress PE advertises an Inclusive Multicast Ethernet Tag Route with a BUM label. That is only possible if the egress PE is configured with **ingress-replication-bum-label**.

Those bindings, as well as the MACs learned on them, can be checked through the following show commands. In the following example, the remote PE(192.0.2.69) is configured with **no ingress-replication-bum-label** and PE(192.0.2.70) is configured with **ingress-replication-bum-label**. Therefore, Dut has a single EVPN-MPLS destination binding to PE(192.0.2.69) and two bindings (unicast and multicast) to PE(192.0.2.70).

```
show service id 1 evpn-mpls
```

Output example

```
===============================================================================
BGP EVPN-MPLS Dest
===============================================================================
TEP Address           Transport:Tnl     Egr Label     Oper     Mcast     Num
                                                      State              MACs

-------------------------------------------------------------------------------
192.0.2.69            ldp:65537         524118        Up       bum       0
192.0.2.70            ldp:65538         524160        Up       none      1
192.0.2.70            ldp:65538         524164        Up       bum       0
192.0.2.72            ldp:65547         524144        Up       bum       0
192.0.2.72            ldp:65547         524138        Up       none      2
192.0.2.73            ldp:65548         524148        Up       bum       1
192.0.2.254           ldp:65550         524150        Up       bum       0
-------------------------------------------------------------------------------
Number of entries : 7
-------------------------------------------------------------------------------
===============================================================================
```

```
show service id 1 fdb detail
```

Output example

```
===============================================================================
Forwarding Database, Service 1
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:ca:fe:ca:fe:69 eMpls:                   EvpnS    06/11/15 21:53:48
                            192.0.2.69:262118
1         00:ca:fe:ca:fe:70 eMpls:                   EvpnS    06/11/15 19:59:57
                            192.0.2.70:262140
1         00:ca:fe:ca:fe:72 eMpls:                   EvpnS    06/11/15 19:59:57
                            192.0.2.72:262141
-------------------------------------------------------------------------------
No. of MAC Entries: 3
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
```

```
===============================================================================
```

6.3.2.1 EVPN and VPLS integration

The 7750 SR, 7450 ESS, or 7950 XRS router SR OS EVPN implementation supports RFC 8560 so that EVPN-MPLS and VPLS can be integrated into the same network and within the same service. Because EVPN is not deployed in green-field networks, this feature is useful for the integration between both technologies and even for the migration of VPLS services to EVPN-MPLS.

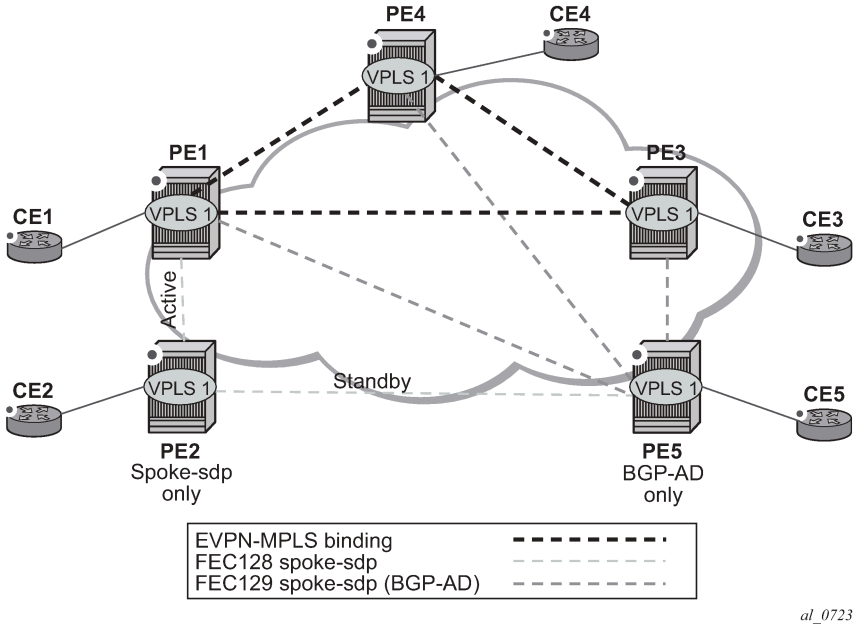
The following behavior enables the integration of EVPN and SDP bindings in the same VPLS network:

1. Systems with EVPN endpoints and SDP bindings to the same far-end bring down the SDP bindings.
   - The router allows the establishment of an EVPN endpoint and an SDP binding to the same far-end but the SDP binding is kept operationally down. Only the EVPN endpoint remains operationally up. This is true for spoke SDPs (manual, BGP-AD, and BGP-VPLS) and mesh SDPs. It is also possible between VXLAN and SDP bindings.
   - If there is an existing EVPN endpoint to a specified far-end and a spoke SDP establishment is attempted, the spoke SDP is setup but kept down with an operational flag indicating that there is an EVPN route to the same far-end.
   - If there is an existing spoke SDP and a valid/used EVPN route arrives, the EVPN endpoint is setup and the spoke SDP is brought down with an operational flag indicating that there is an EVPN route to the same far-end.
   - In the case of an SDP binding and EVPN endpoint to different far-end IPs on the same remote PE, both links are up. This can happen if the SDP binding is terminated in an IPv6 address or IPv4 address different from the system address where the EVPN endpoint is terminated.
2. The user can add spoke SDPs and all the EVPN-MPLS endpoints in the same split horizon group (SHG).
   - A CLI command is added under the **bgp-evpn**>**mpls**> context so that the EVPN-MPLS endpoints can be added to a split horizon group: **bgp-evpn**>**mpls**> **[no] split-horizon-group group-name**
   - The **bgp-evpn mpls split-horizon-group** must reference a user-configured split horizon group. User-configured split horizon groups can be configured within the service context. The same **group-name** can be associated with SAPs, spoke SDPs, pw-templates, pw-template-bindings, and EVPN-MPLS endpoints.
   - If the **split-horizon-group** command in **bgp-evpn**>**mpls**> is not used, the default split horizon group (that contains all the EVPN endpoints) is still used, but it is not possible to refer to it on SAPs/spoke SDPs.
   - SAPs and SDP bindings that share the same split horizon group of the EVPN-MPLS provider-tunnel are brought operationally down if the point-to-multipoint tunnel is operationally up.
3. The system disables the advertisement of MACs learned on spoke SDPs and SAPs that are part of an EVPN split horizon group.
   - When the SAPs and spoke SDPs (manual or BGP-AD/VPLS-discovered) are configured within the same split horizon group as the EVPN endpoints, MAC addresses are still learned on them, but they are not advertised in EVPN.
   - The preceding statement is also true if proxy-ARP/proxy-ND is enabled and an IP-MAC pair is learned on a SAP or SDP binding that belongs to the EVPN split horizon group.

- The SAPs or spoke SDPs, or both, added to an EVPN split horizon group should not be part of any EVPN multihomed ES. If that happened, the PE would still advertise the AD per-EVI route for the SAP or spoke SDP, or both, attracting EVPN traffic that could not possibly be forwarded to that SAP or SDP binding, or both.
- Similar to the preceding statement, a split horizon group composed of SAPs/SDP bindings used in a BGP-MH site should not be configured under **bgp-evpn**>**mpls**>**split-horizon-group**. This misconfiguration would prevent traffic being forwarded from the EVPN to the BGP-MH site, regardless of the DF/NDF state.

Figure 158: EVPN-VPLS integration shows an example of EVPN-VPLS integration.

Figure 158: EVPN-VPLS integration



An example CLI configuration for PE1, PE5, and PE2 is provided below.

```
*A:PE1>config>service# info
----------------------------------------------
pw-template 1 create
vpls 1 name "vpls-1" customer 1 create
  split-horizon-group "SHG-1" create
  bgp
    route-target target:65000:1
    pw-template-binding 1 split-horizon-group SHG-1
  exit
  bgp-ad
    no shutdown
    vpls-id 65000:1
  exit
  bgp-evpn
    evi 1
    mpls bgp 1
      no shutdown
      split-horizon-group SHG-1
  exit
```

```
  spoke-sdp 12:1 create
  exit
  sap 1/1/1:1 create
  exit

*A:PE5>config>service# info
----------------------------------------------
pw-template 1 create
  exit
vpls 1 customer 1 create
  bgp
    route-target target:65000:1
    pw-template-binding 1 split-horizon-group SHG-1 # auto-created SHG
  exit
  bgp-ad
    no shutdown
    vpls-id 65000:1
  exit
  spoke-sdp 52:1 create
  exit

*A:PE2>config>service# info
----------------------------------------------
vpls 1 name "vpls-1" customer 1 create
  end-point CORE create
    no suppress-standby-signaling
  exit
  spoke-sdp 21:1 end-point CORE
    precedence primary
  exit
  spoke-sdp 25:1 end-point CORE
```

- PE1, PE3, and PE4 have BGP-EVPN and BGP-AD enabled in VPLS-1. PE5 has BGP-AD enabled and PE2 has active/standby spoke SDPs to PE1 and PE5.

  In this configuration:

  – PE1, PE3, and PE4 attempt to establish BGP-AD spoke SDPs, but they are kept operationally down as long as there are EVPN endpoints active among them.

  – BGP-AD spoke SDPs and EVPN endpoints are instantiated within the same split horizon group, for example, SHG-1.

  – Manual spoke SDPs from PE1 and PE5 to PE2 are not part of SHG-1.

- EVPN MAC advertisements:

  – MACs learned on FEC128 spoke SDPs are advertised normally in EVPN.

  – MACs learned on FEC129 spoke SDPs are not advertised in EVPN (because they are part of SHG-1, which is the split horizon group used for **bgp-evpn**>**mpls**). This prevents any data plane MACs learned on the SHG from being advertised in EVPN.

- BUM operation on PE1:

  – When CE1 sends BUM, PE1 floods to all the active bindings.

  – When CE2 sends BUM, PE2 sends it to PE1 (active spoke SDP) and PE1 floods to all the bindings and SAPs.

  – When CE5 sends BUM, PE5 floods to the three EVPN PEs. PE1 floods to the active spoke SDP and SAPs, never to the EVPN PEs because they are part of the same SHG.

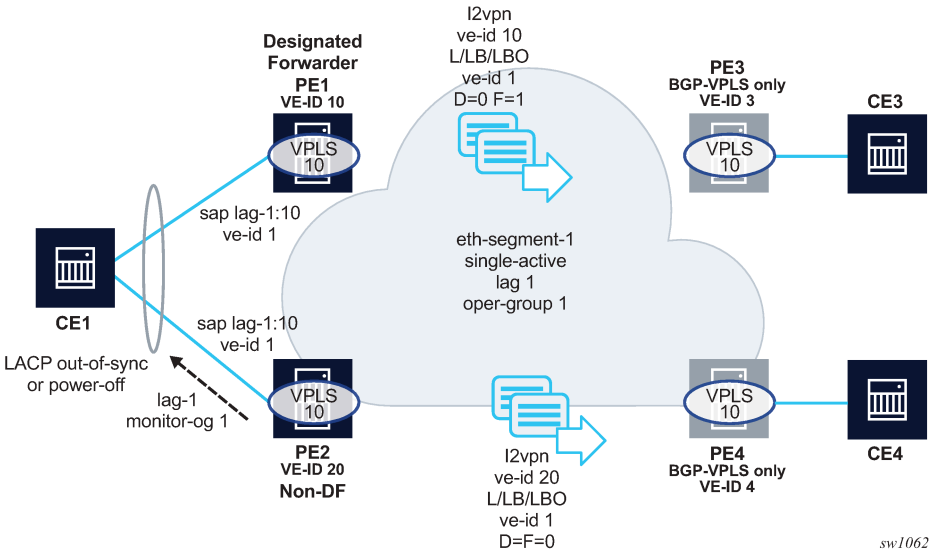
The operation in services with BGP-VPLS and BGP-EVPN is equivalent to the one described above for BGP-AD and BGP-EVPN.

6.3.2.2 EVPN single-active multihoming and BGP-VPLS integration

In a VPLS service to which multiple EVPN PEs and BGP-VPLS PEs are attached, single-active multihoming is supported on two or more of the EVPN PEs with no special considerations. All-active multihoming is not supported, because the traffic from the all-active multihomed CE could cause a MAC flip-flop effect on remote BGP-VPLS PEs, asymmetric flows, or other issues.

Figure 159: BGP-VPLS to EVPN integration and single-active MH illustrates a scenario with a single-active Ethernet-segment used in a service where EVPN PEs and BGP-VPLS are integrated.

Figure 159: BGP-VPLS to EVPN integration and single-active MH



Although other single-active examples are supported, in Figure 159: BGP-VPLS to EVPN integration and single-active MH, CE1 is connected to the EVPN PEs through a single LAG (lag-1). The LAG is associated with the Ethernet-segment 1 on PE1 and PE2, which is configured as single-active and with oper-group 1. PE1 and PE2 make use of **lag>monitor-oper-group 1** so that the non-DF PE can signal the non-DF state to CE1 (in the form of LACP out-of-synch or power-off).

In addition to the BGP-VPLS routes sent for the service ve-id, the multihoming PEs in this case need to generate additional BGP-VPLS routes per Ethernet Segment (per VPLS service) for the purpose of MAC flush on the remote BGP-VPLS PEs in case of failure.

The **sap>bgp-vpls-mh-veid number** command should be configured on the SAPs that are part of an EVPN single-active Ethernet Segment, and allows the advertisement of L2VPN routes that indicate the state of the multihomed SAPs to the remote BGP-VPLS PEs. Upon a Designated Forwarder (DF) switchover, the F and D bits of the generated L2VPN routes for the SAP ve-id are updated so that the remote BGP-VPLS PEs can perform a mac-flush operation on the service and avoid blackholes.

As an example, in case of a failure on the Ethernet-segment sap on PE1, PE1 must indicate PE3 and PE4 the need to flush MAC addresses learned from PE1 (flush-all-from-me message). Otherwise, for example, PE3 continues sending traffic with MAC DA = CE1 to PE1, and PE1 blackholes the traffic.

In the Figure 159: BGP-VPLS to EVPN integration and single-active MH example:

- Both ES peers (PE1 and PE2) should be configured with the same ve-id for the ES SAP. However, this is not mandatory.

- In addition to the regular service ve-id L2VPN route, based on the **sap**>**bgp-vpls-mh-ve-id** configuration and upon BGP VPLS being enabled, the PE advertises an L2VPN route with the following fields:
  - ve-id = **sap**>**bgp-vpls-mh-ve-id** identifier
  - RD, RT, next hop and other attributes same as the service BGP VPLS route
  - L2VPN information extended community with the following flags:
    - D=0 if the SAP is oper-up or oper-down with a flag MHStandby (for example, the PE is non-DF in single-active MH)
    - D=0 also if there is an ES oper-group and the port is down because of the oper-group
    - D=1 if the SAP is oper-down with a different flag (for example, port-down or admin-down)
    - F (DF bit) =1 if the SAP is oper-up, F=0 otherwise
- Upon a failure on the access SAP, there are only mac-flush messages triggered in case the command **bgp-vpls-mh-ve-id** is configured in the failing SAP. In case it is configured with ve-id 1:
  - If the non-DF PE has a failure on the access SAP, PE2 sends an update with ve-id=1/D=1/F=0. This is an indication for PE3/PE4 that PE2's SAP is oper-down but it should not trigger a mac-flush on PE3/PE4.
  - If the DF PE has a failure on the SAP, PE1 advertises ve-id=1/D=1/F=0. Upon receiving this update, PE3 and PE4 flushes all their MACs associated with the PE1's spoke SDP. Note, that the failure on PE1, triggers an EVPN DF Election on PE2, which becomes DF and advertises ve-id=1/D=0/F=1. This message does not trigger any mac-flush procedures.

Other considerations:

- PE3/PE4 are SR OS or any third-party PEs that support the procedures in *draft-ietf-bess-vpls-multihoming*, so that BGP-VPLS mac-flush signaling is understood.
- PE1 and PE2 are expected to run an SR OS version that supports the **sap**>**bgp-vpls-mh-veid number** configuration on the multihomed SAPs. Otherwise, the mac-flush behavior would not work as expected.
- The procedures described above are also supported if the EVPN PEs use MC-LAG instead of an ES for the CE1 redundancy. In this case, the SAP ve-id route for the standby PE is sent as ve-id=1/D=1/F=0, whereas the active chassis advertises ve-id=1/D=0/F=1. A switchover triggers mac-flush on the remote PEs as described earlier.
- The L2VPN routes generated for the ES and SAPs with the **sap bgp-vpls-mh-veid number** command are decoded in the remote nodes as bgp-mh routes (because they do not have label information) in the **show router bgp routes l2-vpn** command and debug.

6.3.2.3 Auto-derived RD in services with multiple BGP families

In a VPLS service, multiple BGP families and protocols can be enabled at the same time. When **bgp-evpn** is enabled, **bgp-ad** and **bgp-mh** are also supported. A single RD is used per service and not per BGP family or protocol.

The following rules apply:

- The VPLS RD is selected based on the following precedence:
  - Manual RD or automatic RD always take precedence when configured.
  - If no manual or automatic RD configuration, the RD is derived from the **bgp-ad**>**vpls-id**.

  - If manual RD, automatic RD, or VPLS ID are not configured, the RD is derived from the **bgp-evpn**>**evi**, except for **bgp-mh** and except when the EVI is greater than 65535. In these two cases, no EVI-derived RD is possible.
  - If manual RD, automatic RD, VPLS ID, or EVI is not configured, there is no RD and the service fails.
- The selected RD (see preceding rules) is displayed by the Oper Route Dist field of the **show service id bgp** command.
- The service supports dynamic RD changes. For example, the CLI allows the dynamic update of VPLS ID be , even if it is used to automatically derive the service RD for **bgp-ad**, **bgp-vpls**, or **bgp-mh**.

  Note: When the RD changes, the active routes for that VPLS are withdrawn and readvertised with the new RD.

- If one of the mechanisms to derive the RD for a specified service is removed from the configuration, the system selects a new RD based on the preceding rules. For example, if the VPLS ID is removed from the configuration, the routes are withdrawn, the new RD selected from the EVI, and the routes readvertised with the new RD.

  Note: This reconfiguration fails if the new RD already exists in a different VPLS or Epipe.

- Because the **vpls-id** takes precedence over the EVI when deriving the RD automatically, adding **evpn** to an existing **bgp-ad** service does not impact the existing RD. The latter is important to support **bgp-ad** to **evpn** migration.

6.3.2.4 EVPN multihoming in VPLS services

EVPN multihoming implementation is based on the concept of the **ethernet-segment**. An **ethernet-segment** is a logical structure that can be defined in one or more PEs and identifies the CE (or access network) multihomed to the EVPN PEs. An **ethernet-segment** is associated with port, LAG, PW port, or SDP objects and is shared by all the services defined on those objects. In the case of virtual ESs, individual VID or VC-ID ranges can be associated with the port, LAG, or PW port, SDP objects defined in the **ethernet-segment**.

Each **ethernet-segment** has a unique Ethernet Segment Identifier (ESI) that is 10 bytes long and is manually configured in the router.

Note: The ESI is advertised in the control plane to all the PEs in an EVPN network; therefore, it is very important to ensure that the 10-byte ESI value is unique throughout the entire network. Single-homed CEs are assumed to be connected to an Ethernet-Segment with esi = 0 (single-homed Ethernet-Segments are not explicitly configured).

This section describes the behavior of the EVPN multihoming implementation in an EVPN-MPLS service.

6.3.2.4.1 EVPN all-active multihoming

As described in RFC 7432, all-active multihoming is only supported on access LAG SAPs and it is mandatory that the CE is configured with a LAG to avoid duplicated packets to the network. Configuring the LACP is optional. SR OS also supports the association of a PW port or a normal port to an all-

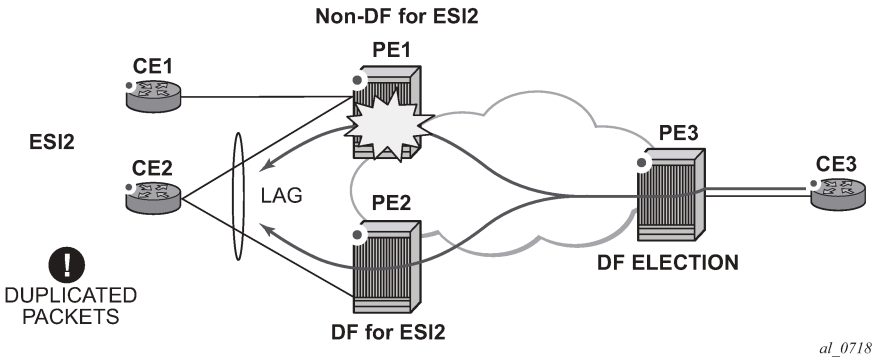
active multihoming ES. When the ES is associated with a physical port and not a LAG, the CE must be configured with a single LAG without LACP.

Three different procedures are implemented in 7750 SR, 7450 ESS, and 7950 XRS SR OS to provide all-active multihoming for a specified Ethernet-Segment:

- DF (Designated Forwarder) election
- Split-horizon
- Aliasing

Figure 160: DF election shows the need for DF election in all-active multihoming.

Figure 160: DF election

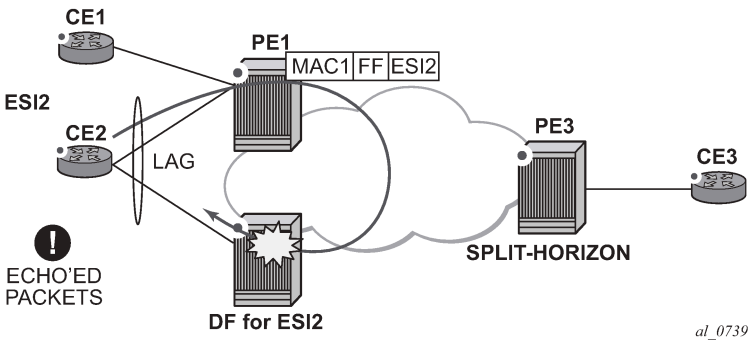


The DF election in EVPN all-active multihoming avoids duplicate packets on the multihomed CE. The DF election procedure is responsible for electing one DF PE per ESI per service; the rest of the PEs being non-DF for the ESI and service. Only the DF forwards BUM traffic from the EVPN network toward the ES SAPs (the multihomed CE). The non-DF PEs do not forward BUM traffic to the local Ethernet-Segment SAPs.

Note: The BUM traffic from the CE to the network and known unicast traffic in any direction is allowed on both the DF and non-DF PEs.

Figure 161: Split-horizon shows the EVPN split-horizon concept for all-active multihoming.

Figure 161: Split-horizon

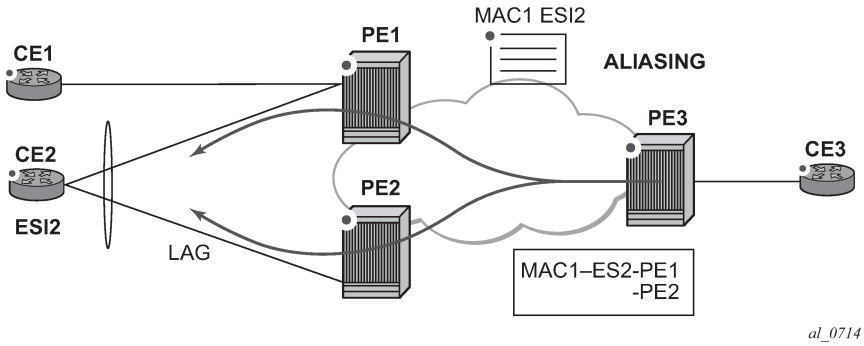


The EVPN split-horizon procedure ensures that the BUM traffic originated by the multihomed PE and sent from the non-DF to the DF, is not replicated back to the CE (echoed packets on the CE). To avoid these echoed packets, the non-DF (PE1) sends all the BUM packets to the DF (PE2) with an indication of the

source Ethernet-Segment. That indication is the ESI Label (ESI2 in the example), previously signaled by PE2 in the AD per-ESI route for the Ethernet-Segment. When PE2 receives an EVPN packet (after the EVPN label lookup), the PE2 finds the ESI label that identifies its local Ethernet-Segment ESI2. The BUM packet is replicated to other local CEs but not to the ESI2 SAP.

Figure 162: Aliasing shows the EVPN aliasing concept for all-active multihoming.

Figure 162: Aliasing



Because CE2 is multihomed to PE1 and PE2 using an all-active Ethernet-Segment, 'aliasing' is the procedure by which PE3 can load-balance the known unicast traffic between PE1 and PE2, even if the destination MAC address was only advertised by PE1 as in the example. When PE3 installs MAC1 in the FDB, it associates MAC1 not only with the advertising PE (PE1) but also with all the PEs advertising the same esi (ESI2) for the service. In this example, PE1 and PE2 advertise an AD per-EVI route for ESI2, therefore, the PE3 installs the two next-hops associated with MAC1.

Aliasing is enabled by configuring ECMP greater than 1 in the **bgp-evpn**>**mpls** context.

6.3.2.4.1.1 All-active multihoming service model

The following shows an example PE1 configuration that provides all-active multihoming to the CE2 shown in Figure 162: Aliasing .

```
*A:PE1>config>lag(1)# info
----------------------------------------------
  mode access
  encap-type dot1q
  port 1/1/2
  lacp active administrative-key 1 system-id 00:00:00:00:00:22
  no shutdown

*A:PE1>config>service>system>bgp-evpn# info
----------------------------------------------
  route-distinguisher 10.1.1.1:0
  ethernet-segment "ESI2" create
    esi 01:12:12:12:12:12:12:12:12:12
    multi-homing all-active
    service-carving
    lag 1
    no shutdown

*A:PE1>config>redundancy>evpn-multi-homing# info
----------------------------------------------
    boot-timer 120
    es-activation-timer 10
```

```
*A:PE1>config>service>vpls# info
----------------------------------------------
  description "evpn-mpls-service with all-active multihoming"
  bgp
  bgp-evpn
    evi 10
    mpls bgp 1
      no shutdown
      auto-bind-tunnel resolution any
  sap lag-1:1 create
  exit
```

In the same way, PE2 is configured as follows:

```
*A:PE1>config>lag(1)# info
----------------------------------------------
  mode access
  encap-type dot1q
  port 1/1/1
  lacp active administrative-key 1 system-id 00:00:00:00:00:22
  no shutdown

*A:PE1>config>service>system>bgp-evpn# info
----------------------------------------------
  route-distinguisher 10.1.1.1:0
  ethernet-segment "ESI12" create
    esi 01:12:12:12:12:12:12:12:12:12
    multi-homing all-active
    service-carving
    lag 1
    no shutdown

*A:PE1>config>redundancy>evpn-multi-homing# info
----------------------------------------------
    boot-timer 120
    es-activation-timer 10

*A:PE1>config>service>vpls# info
----------------------------------------------
  description "evpn-mpls-service with all-active multihoming"
  bgp
    route-distinguisher 65001:60
    route-target target:65000:60
  bgp-evpn
    evi 10
    mpls bgp 1
      no shutdown
      auto-bind-tunnel resolution any
  sap lag-1:1 create
  exit
```

The preceding configuration enables the all-active multihoming procedures. The following must be considered:

- The **ethernet-segment** must be configured with a name and a 10-byte esi:
  - **config>service>system>bgp-evpn# ethernet-segment<es_name> create**
  - **config>service> system>bgp-evpn>ethernet-segment# esi <value>**

- When configuring the esi, the system enforces the 6 high-order octets after the type to be different from zero (so that the auto-derived route-target for the ES route is different from zero). Other than that, the entire esi value must be unique in the system.
- Only a LAG or a PW port can be associated with the all-active **ethernet-segment**. This LAG is exclusively used for EVPN multihoming. Other LAG ports in the system can be still used for MC-LAG and other services.
- When the LAG is configured on PE1 and PE2, the same **admin-key**, **system-priority**, and **system-id** must be configured on both PEs, so that CE2 responds as though it is connected to the same system.
- The same **ethernet-segment** may be used for EVPN-MPLS, EVPN-VXLAN and PBB-EVPN services.

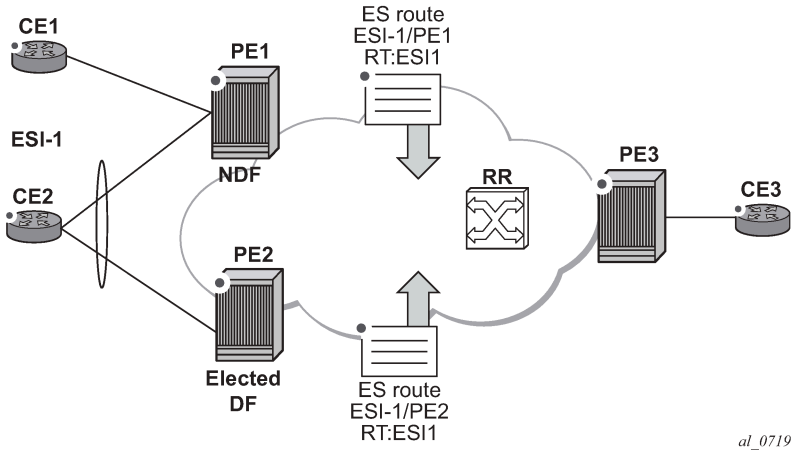
Note: The **source-bmac-lsb** attribute must be defined for PBB-EVPN (so that it is only used in PBB-EVPN, and ignored by EVPN). Other than EVPN-MPLS, EVPN-VXLAN and PBB-EVPN I-VPLS/Epipe services, no other Layer 2 services are allowed in the same **ethernet-segment** (regular VPLS defined on the **ethernet-segment** is kept operationally down).

- Only one SAP per service can be part of the same **ethernet-segment**.

6.3.2.4.1.2 ES discovery and DF election procedures

The ES discovery and DF election is implemented in three logical steps, as shown in Figure 163: ES discovery and DF election.

Figure 163: ES discovery and DF election



6.3.2.4.1.2.1 Step 1 - ES advertisement and discovery

The **ethernet-segment** ESI-1 is configured as per the previous section, with all the required parameters. When **ethernet-segment no shutdown** is executed, PE1 and PE2 advertise an ES route for ESI-1. They both include the route-target auto-derived from the MAC portion of the configured ESI. If the route-target address family is configured in the network, this allows the RR to keep the dissemination of the ES routes under control.

In addition to the ES route, PE1 and PE2 advertise AD per-ESI routes and AD per-EVI routes.

- AD per-ESI routes announce the Ethernet-Segment capabilities, including the mode (single-active or all-active) as well as the ESI label for split-horizon.
- AD per-EVI routes are advertised so that PE3 knows what services (EVIs) are associated with the ESI. These routes are used by PE3 for its aliasing and backup procedures.

6.3.2.4.1.2.2 Step 2 - DF election

When ES routes exchange between PE1 and PE2 is complete, both run the DF election for all the services in the **ethernet-segment**.

PE1 and PE2 elect a Designated Forwarder (DF) per <ESI, service>. The default DF election mechanism in 7750 SR, 7450 ESS, and 7950 XRS SR OS is **service-carving** (as per RFC 7432). The following applies when enabled on a specified PE:

- An ordered list of PE IPs where ESI-1 resides is built. The IPs are gotten from the Origin IP fields of all the ES routes received for ESI-1, as well as the local system address. The lowest IP is considered ordinal '0' in the list.
- The local IP can only be considered a "candidate" after successful **ethernet-segment no shutdown** for a specified service.

  Note: The remote PE IPs must be present in the local PE's RTM so that they can participate in the DF election.

- A PE only considers a specified remote IP address as candidate for the DF election algorithm for a specified service if, as well as the ES route, the corresponding AD routes per-ESI and per-EVI for that PE have been received and properly activated.
- All the remote PEs receiving the AD per-ES routes (for example, PE3), interpret that ESI-1 is all-active if all the PEs send their AD per-ES routes with the single-active bit = 0. Otherwise, if at least one PE sends an AD route per-ESI with the single-active flag set or the local ESI configuration is single-active, the ESI behaves as single-active.
- An **es-activation-timer** can be configured at the **redundancy**>**bgp-evpn-multi-homing**>**es-activation-timer** level or at the **service**>**system**>**bgp-evpn**>**eth-seg**>**es-activation-timer** level. This timer, which is 3 seconds by default, delays the transition from non-DF to DF for a specified service, after the DF election has run.
  - This use of the **es-activation-timer** is different from zero and minimizes the risks of loops and packet duplication because of "transient" multiple DFs.
  - The same **es-activation-timer** should be configured in all the PEs that are part of the same ESI. It is up to the user to configure either a long timer to minimize the risks of loops/duplication or even **es-activation-timer=0** to speed up the convergence for non-DF to DF transitions. When the user configures a specific value, the value configured at ES level supersedes the configured global value.
- The DF election is triggered by the following events:
  - **config**>**service**>**system**>**bgp-evpn**>**eth-seg# no shutdown** triggers the DF election for all the services in the ESI.
  - Reception of a new update/withdrawal of an ES route (containing an ESI configured locally) triggers the DF election for all the services in the ESI.

  - Reception of a new update/withdrawal of an AD per-ES route (containing an ESI configured locally) triggers the DF election for all the services associated with the list of route-targets received along with the route.
  - Reception of a new update of an AD per-ES route with a change in the ESI-label extended community (single-active bit or MPLS label) triggers the DF election for all the services associated with the list of route-targets received along with the route.
  - Reception of a new update/withdrawal of an AD route per-EVI (containing an ESI configured locally) triggers the DF election for that service.

- When the PE boots up, the boot-timer allows the necessary time for the control plane protocols to come up before bringing up the Ethernet-Segment and running the DF algorithm. The boot-timer is configured at system level - **config>redundancy>bgp-evpn-multi-homing# boot-timer** - and should use a value long enough to allow the IOMs and BGP sessions to come up before exchanging ES routes and running the DF election for each EVI/ISID.
  - The system does not advertise ES routes until the boot timer expires. This guarantees that the peer ES PEs do not run the DF election either until the PE is ready to become the DF if it needs to.
  - The following show command displays the configured boot-timer as well as the remaining timer if the system is still in boot-stage.

```
A:PE1# show redundancy bgp-evpn-multi-homing
===============================================================================
Redundancy BGP EVPN Multi-homing Information
===============================================================================
Boot-Timer              : 10 secs
Boot-Timer Remaining    : 0 secs
ES Activation Timer     : 3 secs
===============================================================================
```

- When **service-carving mode auto** is configured (default mode), the DF election algorithm runs the function [V(evi) mod N(peers) = i(ordinal)] to identify the DF for a specified service and ESI, as described in the following example.

  As shown in Figure 163: ES discovery and DF election, PE1 and PE2 are configured with ESI-1. Given that V(10) mod N(2) = 0, PE1 is elected DF for VPLS-10 (because its IP address is lower than PE2's and it is the first PE in the candidate list).

  Note: The algorithm takes the configured **evi** in the service as opposed to the service-id itself. The **evi** for a service must match in all the PEs that are part of the ESI. This guarantees that the election algorithm is consistent across all the PEs of the ESI. The **evi** must be always configured in a service with SAPs/SDP bindings that are created in an ES.

- A **manual** service-carving option is allowed so that the user can manually configure for which evi identifiers the PE is primary: **service-carving mode manual / manual evi <start-evi>** to **<end-evi>**
  - The system is the PE forwarding/multicasting traffic for the **evi** identifiers included in the configuration. The PE is secondary (non-DF) for the non-specified **evi** identifiers.
  - If a range is configured but the service-carving is not mode manual, then the range has no effect.
  - Only two PEs are supported when service-carving mode manual is configured. If a third PE is configured with service-carving mode manual for an ESI, the two non-primary PEs remain non-DF regardless of the primary status.

– For example, as shown in Figure 163: ES discovery and DF election: if PE1 is configured with service-carving manual evi 1 to 100 and PE2 with service-carving manual evi 101 to 200, then PE1 is the primary PE for service VPLS 10 and PE2 the secondary PE.

- When service-carving is disabled, the lowest originator IP wins the election for a specified service and ESI:

  config>service>system>bgp-evpn>eth-seg>service-carving> mode off

  The following show command displays the **ethernet-segment** configuration and DF status for all the EVIs and ISIDs (if PBB-EVPN is enabled) configured in the **ethernet-segment**.

```
*A:PE1# show service system bgp-evpn ethernet-segment name "ESI-1" all
===============================================================================
Service Ethernet Segment
===============================================================================
Name                    : ESI-1
Admin State             : Up                  Oper State         : Up
ESI                     : 01:00:00:00:00:71:00:00:00:01
Multi-homing            : allActive           Oper Multi-homing  : allActive
Source BMAC LSB         : 71-71
ES BMac Tbl Size        : 8                   ES BMac Entries    : 1
Lag Id                  : 1
ES Activation Timer     : 0 secs
Exp/Imp Route-Target    : target:00:00:00:00:71:00

Svc Carving             : auto
ES SHG Label            : 262142
===============================================================================
===============================================================================
EVI Information
===============================================================================
EVI                 SvcId               Actv Timer Rem      DF
-------------------------------------------------------------------------------
1                   1                   0                   no
-------------------------------------------------------------------------------
Number of entries: 1
===============================================================================
-------------------------------------------------------------------------------
DF Candidate list
-------------------------------------------------------------------------------
EVI                                     DF Address
-------------------------------------------------------------------------------
1                                       192.0.2.69
1                                       192.0.2.72
-------------------------------------------------------------------------------
Number of entries: 2
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
===============================================================================
ISID Information
===============================================================================
ISID                SvcId               Actv Timer Rem      DF
-------------------------------------------------------------------------------
20001               20001               0                   no
-------------------------------------------------------------------------------
Number of entries: 1
===============================================================================

-------------------------------------------------------------------------------
DF Candidate list
-------------------------------------------------------------------------------
ISID                                    DF Address
```

```
-------------------------------------------------------------------------------
20001                                192.0.2.69
20001                                192.0.2.72
-------------------------------------------------------------------------------
Number of entries: 2
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
===============================================================================
BMAC Information
===============================================================================
SvcId                                BMacAddress
-------------------------------------------------------------------------------
20000                                00:00:00:00:71:71
-------------------------------------------------------------------------------
Number of entries: 1
===============================================================================
```

6.3.2.4.1.2.3 Step 3 - DF and non-DF service behavior

Based on the result of the DF election or the manual service-carving, the control plane on the non-DF (PE1) instructs the data path to remove the LAG SAP (associated with the ESI) from the default flooding list for BM traffic (unknown unicast traffic may still be sent if the EVI label is a unicast label and the source MAC address is not associated with the ESI). On PE1 and PE2, both LAG SAPs learn the same MAC address (coming from the CE). For instance, in the following show commands, 00:ca:ca:ba:ce:03 is learned on both PE1 and PE2 access LAG (on ESI-1). However, PE1 learns the MAC as 'Learned' whereas PE2 learns it as 'Evpn'. This is because of the CE2 hashing the traffic for that source MAC to PE1. PE2 learns the MAC through EVPN but it associates the MAC to the ESI SAP, because the MAC belongs to the ESI.

```
*A:PE1# show service id 1 fdb detail
===============================================================================
Forwarding Database, Service 1
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:ca:ca:ba:ce:03 sap:lag-1:1              L/0      06/11/15 00:14:47
1         00:ca:fe:ca:fe:70 eMpls:                   EvpnS    06/11/15 00:09:06
                            192.0.2.70:262140
1         00:ca:fe:ca:fe:72 eMpls:                   EvpnS    06/11/15 00:09:39
                            192.0.2.72:262141
-------------------------------------------------------------------------------
No. of MAC Entries: 3
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
===============================================================================

*A:PE2# show service id 1 fdb detail
===============================================================================
Forwarding Database, Service 1
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:ca:ca:ba:ce:03 sap:lag-1:1              Evpn     06/11/15 00:14:47
1         00:ca:fe:ca:fe:69 eMpls:                   EvpnS    06/11/15 00:09:40
                            192.0.2.69:262141
1         00:ca:fe:ca:fe:70 eMpls:                   EvpnS    06/11/15 00:09:40
                            192.0.2.70:262140
```

```
-------------------------------------------------------------------------------
No. of MAC Entries: 3
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
===============================================================================
```

When PE1 (non-DF) and PE2 (DF) exchange BUM packets for **evi 1**, all those packets are sent including the ESI label at the bottom of the stack (in both directions). The ESI label advertised by each PE for ESI-1 can be displayed by the following command:

```
*A:PE1# show service system bgp-evpn ethernet-segment name "ESI-1"
===============================================================================
Service Ethernet Segment
===============================================================================
Name                    : ESI-1
Admin State             : Up                  Oper State         : Up
ESI                     : 01:00:00:00:00:71:00:00:00:01
Multi-homing            : allActive           Oper Multi-homing  : allActive
Source BMAC LSB         : 71-71
ES BMac Tbl Size        : 8                   ES BMac Entries    : 1
Lag Id                  : 1
ES Activation Timer     : 0 secs
Exp/Imp Route-Target    : target:00:00:00:00:71:00

Svc Carving             : auto
ES SHG Label            : 262142
===============================================================================
*A:PE2# show service system bgp-evpn ethernet-segment name "ESI-1"

===============================================================================
Service Ethernet Segment
===============================================================================
Name                    : ESI-1
Admin State             : Up                  Oper State         : Up
ESI                     : 01:00:00:00:00:71:00:00:00:01
Multi-homing            : allActive           Oper Multi-homing  : allActive
Source BMAC LSB         : 71-71
ES BMac Tbl Size        : 8                   ES BMac Entries    : 0
Lag Id                  : 1
ES Activation Timer     : 20 secs
Exp/Imp Route-Target    : target:00:00:00:00:71:00

Svc Carving             : auto
ES SHG Label            : 262142
===============================================================================
```

6.3.2.4.1.3 Aliasing

Following the example in Figure 163: ES discovery and DF election, if the service configuration on PE3 has ECMP > 1, PE3 adds PE1 and PE2 to the list of next-hops for ESI-1. As soon as PE3 receives a MAC for ESI-1, it starts load-balancing between PE1 and PE2 the flows to the remote ESI CE. The following command shows the FDB in PE3.

Note: MAC 00:ca:ca:ba:ce:03 is associated with the Ethernet-Segment eES:01:00:00:00:00:71:00:00:00:01 (esi configured on PE1 and PE2 for ESI-1).

```
*A:PE3# show service id 1 fdb detail
===============================================================================
```

```
Forwarding Database, Service 1
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:ca:ca:ba:ce:03 eES:                     Evpn     06/11/15 00:14:47
                            01:00:00:00:00:71:00:00:00:01
1         00:ca:fe:ca:fe:69 eMpls:                   EvpnS    06/11/15 00:09:18
                            192.0.2.69:262141
1         00:ca:fe:ca:fe:70 eMpls:                   EvpnS    06/11/15 00:09:18
                            192.0.2.70:262140
1         00:ca:fe:ca:fe:72 eMpls:                   EvpnS    06/11/15 00:09:39
                            192.0.2.72:262141
-------------------------------------------------------------------------------
No. of MAC Entries: 4
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
===============================================================================
```

The following command shows all the EVPN-MPLS destination bindings on PE3, including the ES destination bindings.

The Ethernet-Segment eES:01:00:00:00:00:71:00:00:00:01 is resolved to PE1 and PE2 addresses:

```
*A:PE3# show service id 1 evpn-mpls
===============================================================================
BGP EVPN-MPLS Dest
===============================================================================
TEP Address     Egr Label     Num. MACs   Mcast           Last Change
                 Transport
-------------------------------------------------------------------------------
192.0.2.69      262140        0           Yes             06/10/2015 14:33:30
                ldp
192.0.2.69      262141        1           No              06/10/2015 14:33:30
                ldp
192.0.2.70      262139        0           Yes             06/10/2015 14:33:30
                ldp
192.0.2.70      262140        1           No              06/10/2015 14:33:30
                ldp
192.0.2.72      262140        0           Yes             06/10/2015 14:33:30
                ldp
192.0.2.72      262141        1           No              06/10/2015 14:33:30
                ldp
192.0.2.73      262139        0           Yes             06/10/2015 14:33:30
                ldp
192.0.2.254     262142        0           Yes             06/10/2015 14:33:30
                bgp
-------------------------------------------------------------------------------
Number of entries : 8
-------------------------------------------------------------------------------
===============================================================================
===============================================================================
BGP EVPN-MPLS Ethernet Segment Dest
===============================================================================
Eth SegId                     TEP Address     Egr Label   Last Change
                                               Transport
-------------------------------------------------------------------------------
01:00:00:00:00:71:00:00:00:01 192.0.2.69      262141      06/10/2015 14:33:30
                                              ldp
01:00:00:00:00:71:00:00:00:01 192.0.2.72      262141      06/10/2015 14:33:30
                                              ldp
01:74:13:00:74:13:00:00:74:13 192.0.2.73      262140      06/10/2015 14:33:30
                                              ldp
```

```
-------------------------------------------------------------------------------
Number of entries : 3
-------------------------------------------------------------------------------
===============================================================================
```

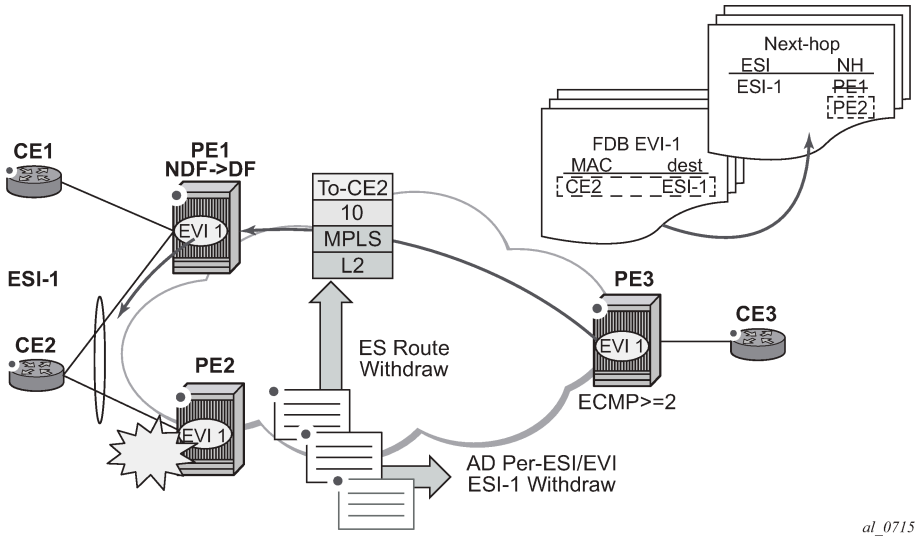
PE3 performs aliasing for all the MACs associated with that ESI. This is possible because PE1 is configured with ECMP parameter >1:

```
*A:PE3>config>service>vpls# info
----------------------------------------------
            bgp
            exit
            bgp-evpn
                evi 1
                mpls bgp 1
                    ecmp 4
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            proxy-arp
                shutdown
            exit
            stp
                shutdown
            exit
            sap 1/1/1:2 create
            exit
            no shutdown
```

6.3.2.4.1.4 Network failures and convergence for all-active multihoming

Figure 164: All-active multihoming ES failure shows the behavior on the remote PEs (PE3) when there is an **ethernet-segment** failure.

Figure 164: All-active multihoming ES failure



The unicast traffic behavior on PE3 is as follows:

1. PE3 forwards MAC DA = CE2 to both PE1 and PE2 when the MAC advertisement route came from PE1 (or PE2) and the set of Ethernet AD per-ES routes and Ethernet AD per-EVI routes from PE1 and PE2 are active at PE3.
2. If there was a failure between CE2 and PE2, PE2 would withdraw its set of Ethernet AD and ES routes, then PE3 would forward traffic destined for CE2 to PE1 only. PE3 does not need to wait for the withdrawal of the individual MAC.

   The same behavior would be followed if the failure had been at PE1.
3. If after step 2, PE2 withdraws its MAC advertisement route, then PE3 treats traffic to MAC DA = CE2 as unknown unicast, unless the MAC had been previously advertised by PE1.

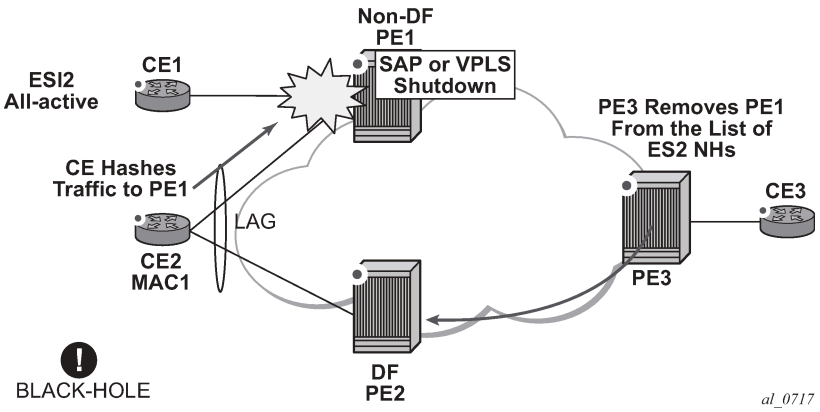
For BUM traffic, the following events would trigger a DF election on a PE and only the DF would forward BUM traffic after the **esi-activation-timer** expiration (if there was a transition from non-DF to DF).

- Reception of ES route update (local ES shutdown/no shutdown or remote route)
- New AD-ES route update/withdraw
- New AD-EVI route update/withdraw
- Local ES port/SAP/service shutdown
- Service carving range change (effecting the evi)
- Multihoming mode change (single/all active to all/single-active)

6.3.2.4.1.4.1 Logical failures on ESs and blackholes

Be aware of the effects triggered by specific 'failure scenarios'; some of these scenarios are shown in Figure 165: Blackhole caused by SAP/SVC shutdown:

Figure 165: Blackhole caused by SAP/SVC shutdown



If an individual VPLS service is **shutdown** in PE1 (the example is also valid for PE2), the corresponding LAG SAP goes **operationally down**. This event triggers the withdrawal of the AD per-EVI route for that particular SAP. PE3 removes PE1 of its list of aliased next-hops and PE2 takes over as DF (if it was not the DF already). However, this does not prevent the network from black-holing the traffic that CE2 'hashes' to the link to PE1. Traffic sent from CE2 to PE2 or traffic from the rest of the CEs to CE2 is not affected, so this situation is not easily detected on the CE.

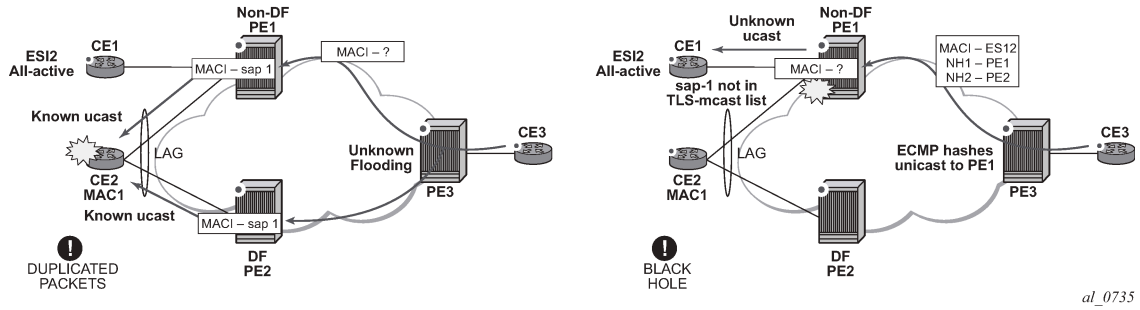
The same result occurs if the ES SAP is administratively **shutdown** instead of the service.

Note: When **bgp-evpn mpls shutdown** is executed, the SAP associated with the ES is brought operationally down (StandbyforMHprotocol) and so does the entire service if there are no other SAPs or SDP bindings in the service. However, if there are other SAPs/SDP bindings, the service remains operationally up.

6.3.2.4.1.4.2 Transient issues caused by MAC route delays

Some situations may cause potential transient issues to occur. These are shown in Figure 166: Transient issues caused by "slow" MAC learning and described below.

Figure 166: Transient issues caused by "slow" MAC learning



Transient packet duplication caused by delay in PE3 to learn MAC1:

This scenario is illustrated by the diagram on the left in Figure 166: Transient issues caused by "slow" MAC learning. In an all-active multihoming scenario, if a specified MAC address is not yet learned in a remote PE, but is known in the two PEs of the ES, for example, PE1 and PE2, the latter PEs may send duplicated packets to the CE.

In an all-active multihoming scenario, if a specified MAC address (for example, MAC1), is not learned yet in a remote PE (for example, PE3), but it is known in the two PEs of the ES (for example, PE1 and PE2), the latter PEs may send duplicated packets to the CE.

This issue is solved by the use of **ingress-replication-bum-label** in PE1 and PE2. If configured, PE1/PE2 knows that the received packet is an unknown unicast packet, therefore, the NDF (PE1) does not send the packets to the CE and there is not duplication.

Note: Even without the **ingress-replication-bum-label**, this is only a transient situation that would be solved as soon as MAC1 is learned in PE3.

Transient blackhole caused by delay in PE1 to learn MAC1:

This case is illustrated by the diagram on the right in Figure 166: Transient issues caused by "slow" MAC learning. In an all-active multihoming scenario, MAC1 is known in PE3 and aliasing is applied to MAC1. However, MAC1 is not known yet in PE1, the NDF for the ES. If PE3 hashing picks up PE1 as the destination of the aliased MAC1, the packets are blackholed. This case is solved on the NDF by not blocking unknown unicast traffic that arrives with a unicast label. If PE1 and PE2 are configured using **ingress-replication-bum-label**, PE3 sends unknown unicast with a BUM label and known unicast with a unicast label. In the latter case, PE1 considers it is safe to forward the frame to the CE, even if it is unknown unicast. It is important to note that this is a transient issue and as soon as PE1 learns MAC1 the frames are forwarded as known unicast.

6.3.2.4.2 EVPN single-active multihoming

The 7750 SR, 7450 ESS, and 7950 XRS SR OS supports single-active multihoming on access LAG SAPs, regular SAPs, and spoke SDPs for a specified VPLS service.

The following SR OS procedures support EVPN single-active multihoming for a specified Ethernet-Segment:

- **DF (Designated Forwarder) election**

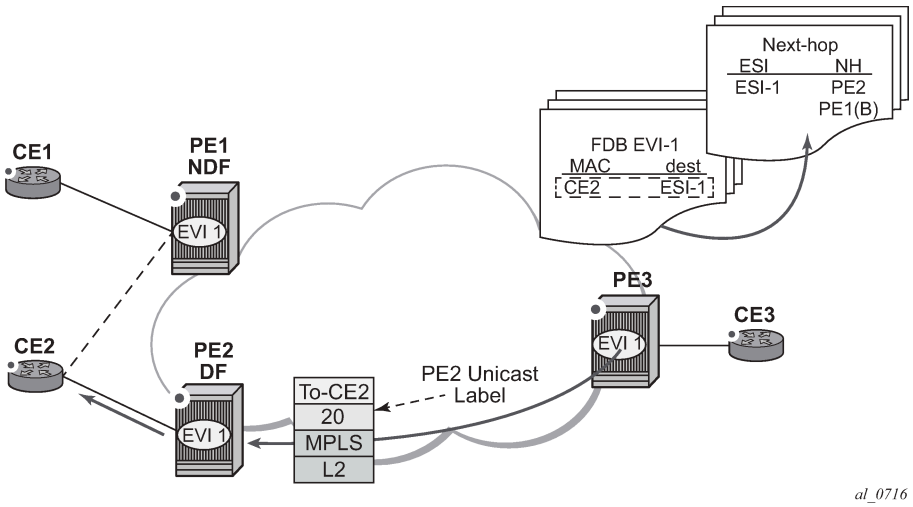
  As in all-active multihoming, DF election in single-active multihoming determines the forwarding for BUM traffic from the EVPN network to the Ethernet-Segment CE. Also, in single-active multihoming, DF election also determines the forwarding of any traffic (unicast/BUM) and in any direction (to/from the CE).

- **backup PE**

  In single-active multihoming, the remote PEs do not perform aliasing to the PEs in the Ethernet-Segment. The remote PEs identify the DF based on the MAC routes and send the unicast flows for the Ethernet-Segment to the PE in the DF and program a backup PE as an alternative next-hop for the remote ESI in case of failure.

  This RFC 7432 procedure is known as 'Backup PE' and is shown in Figure 167: Backup PE for PE3.

  Figure 167: Backup PE



6.3.2.4.2.1 Single-active multihoming service model

The following shows an example of PE1 configuration that provides single-active multihoming to CE2, as shown in Figure 167: Backup PE.

```
*A:PE1>config>service>system>bgp-evpn# info
----------------------------------------------
  route-distinguisher 10.1.1.1:0
  ethernet-segment "ESI2" create
    esi 01:12:12:12:12:12:12:12:12:12
    multi-homing single-active
    service-carving
```

```
    sdp 1
    no shutdown

*A:PE1>config>redundancy>evpn-multi-homing# info
----------------------------------------------
    boot-timer 120
    es-activation-timer 10

*A:PE1>config>service>vpls# info
----------------------------------------------
  description "evpn-mpls-service with single-active multihoming"
  bgp
  bgp-evpn
    evi 10
    mpls bgp 1
      no shutdown
      auto-bind-tunnel resolution any
  spoke-sdp 1:1 create
  exit
```

The PE2 example configuration for this scenario is as follows:

```
*A:PE1>config>service>system>bgp-evpn# info
----------------------------------------------
  route-distinguisher 10.1.1.1:0
  ethernet-segment "ESI2" create
    esi 01:12:12:12:12:12:12:12:12:12
    multi-homing single-active
    service-carving
    sdp 2
    no shutdown

*A:PE1>config>redundancy>evpn-multi-homing# info
----------------------------------------------
    boot-timer 120
    es-activation-timer 10

*A:PE1>config>service>vpls# info
----------------------------------------------
  description "evpn-mpls-service with single-active multihoming"
  bgp
  bgp-evpn
    evi 10
    mpls bgp 1
      no shutdown
      auto-bind-tunnel resolution any
  spoke-sdp 2:1 create
  exit
```

In single-active multihoming, the non-DF PEs for a specified ESI block unicast and BUM traffic in both directions (upstream and downstream) on the object associated with the ESI. Other than that, single-active multihoming is similar to all-active multihoming with the following differences:

- The **ethernet-segment** is configured for single-active: **service**>**system**>**bgp-evpn**>**eth-seg**>**multi-homing single-active**.
- The advertisement of the ESI-label in an AD per-ESI is optional for **single-active** Ethernet-Segments. The user can control the no advertisement of the ESI label by using the **service system bgp-evpn eth-seg multi-homing single-active no-esi-label** command. By default, the ESI label is used for single-active ESs too.

- For single-active multihoming, the Ethernet-Segment can be associated with a **port** and **sdp**, as well as a **lag-id**, as shown in Figure 167: Backup PE, where:
  - **port** is used for single-active SAP redundancy without the need for lag.
  - **sdp** is used for single-active spoke SDP redundancy.
  - **lag** is used for single-active LAG redundancy

  Note: In this case, key, system-id, and system-priority must be different on the PEs that are part of the Ethernet-Segment).

- For single-active multihoming, when the PE is non-DF for the service, the SAPs/spoke SDPs on the Ethernet-Segment are down and show StandByForMHProtocol as the reason.
- From a service perspective, single-active multihoming can provide redundancy to CEs (MHD, Multi-Homed Devices) or networks (MHN, Multi-Homed Networks) with the following setup:
  - **LAG with or without LACP**

    In this case, the multihomed ports on the CE are part of the different LAGs (a LAG per multihomed PE is used in the CE). The non-DF PE for each service can signal that the SAP is operationally down if eth-cfm fault-propagation-enable {use-if-tlv | suspend-ccm} is configured.
  - **regular Ethernet 802.1q/ad ports**

    In this case, the multihomed ports on the CE/network are not part of any LAG. Eth-cfm can also be used for non-DF indication to the multihomed device/network.
  - **active-standby PWs**

    In this case, the multihomed CE/network is connected to the PEs through an MPLS network and an active/standby spoke SDP per service. The non-DF PE for each service makes use of the LDP PW status bits to signal that the spoke SDP is operationally down on the PE side.

6.3.2.4.2.2 ES and DF election procedures

In all-active multihoming, the non-DF keeps the SAP up, although it removes it from the default flooding list. In the single-active multihoming implementation the non-DF brings the SAP or SDP binding operationally down. See ES discovery and DF election procedures.

The following **show** commands display the status of the single-active ESI-7413 in the non-DF. The associated spoke SDP is operationally down and it signals PW Status standby to the multihomed CE:

```
*A:PE1# show service system bgp-evpn ethernet-segment name "ESI-7413"

===============================================================================
Service Ethernet Segment
===============================================================================
Name                    : ESI-7413
Admin State             : Up                 Oper State         : Up
ESI                     : 01:74:13:00:74:13:00:00:74:13
Multi-homing            : singleActive       Oper Multi-homing  : singleActive
Source BMAC LSB         : <none>
Sdp Id                  : 4
ES Activation Timer     : 0 secs
Exp/Imp Route-Target    : target:74:13:00:74:13:00

Svc Carving             : auto
ES SHG Label            : 262141
```

```
===============================================================================
*A:PE1# show service system bgp-evpn ethernet-segment name "ESI-7413" evi 1
===============================================================================
EVI DF and Candidate List
===============================================================================
EVI           SvcId         Actv Timer Rem      DF  DF Last Change
-------------------------------------------------------------------------------
1             1             0                   no  06/11/2015 20:05:32
===============================================================================
===============================================================================
DF Candidates                           Time Added
-------------------------------------------------------------------------------
192.0.2.70                              06/11/2015 20:05:20
192.0.2.73                              06/11/2015 20:05:32
-------------------------------------------------------------------------------
Number of entries: 2
===============================================================================
*A:PE1# show service id 1 base
===============================================================================
Service Basic Information
===============================================================================
Service Id        : 1                   Vpn Id            : 0
Service Type      : VPLS
Name              : (Not Specified)
Description       : (Not Specified)

<snip>
-------------------------------------------------------------------------------
Service Access & Destination Points
-------------------------------------------------------------------------------
Identifier                               Type         AdmMTU  OprMTU  Adm  Opr
-------------------------------------------------------------------------------
sap:1/1/1:1                              q-tag        9000    9000    Up   Up
sdp:4:13 S(192.0.2.74)                   Spok         0       8978    Up   Down
===============================================================================
* indicates that the corresponding row element may have been truncated.


*A:PE1# show service id 1 all | match Pw
Local Pw Bits      : pwFwdingStandby
Peer Pw Bits       : None

*A:PE1# show service id 1 all | match Flag
Flags              : StandbyForMHProtocol
Flags              : None
```

6.3.2.4.2.3 Backup PE function

A remote PE (PE3 in Figure 167: Backup PE) imports the AD routes per ESI, where the single-active flag is set. PE3 interprets that the Ethernet-Segment is single-active if at least one PE sends an AD route per-ESI with the single-active flag set. MACs for a specified service and ESI are learned from a single PE, that is, the DF for that <ESI, EVI>.

The remote PE installs a single EVPN-MPLS destination (TEP, label) for a received MAC address and a backup next-hop to the PE for which the AD routes per-ESI and per-EVI are received. For instance, in the following command, 00:ca:ca:ba:ca:06 is associated with the remote `ethernet-segment eES`

01:74:13:00:74:13:00:00:74:13. That eES is resolved to PE (192.0.2.73), which is the DF on the ES.

```
*A:PE3# show service id 1 fdb detail
===============================================================================
Forwarding Database, Service 1
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:ca:ca:ba:ca:02 sap:1/1/1:2              L/0      06/12/15 00:33:39
1         00:ca:ca:ba:ca:06 eES:                     Evpn     06/12/15 00:33:39
                            01:74:13:00:74:13:00:00:74:13
1         00:ca:fe:ca:fe:69 eMpls:                   EvpnS    06/11/15 21:53:47
                            192.0.2.69:262118
1         00:ca:fe:ca:fe:70 eMpls:                   EvpnS    06/11/15 19:59:57
                            192.0.2.70:262140
1         00:ca:fe:ca:fe:72 eMpls:                   EvpnS    06/11/15 19:59:57
                            192.0.2.72:262141
-------------------------------------------------------------------------------
No. of MAC Entries: 5
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
===============================================================================

*A:PE3# show service id 1 evpn-mpls
===============================================================================
BGP EVPN-MPLS Dest
===============================================================================
TEP Address     Egr Label     Num. MACs   Mcast           Last Change
                 Transport
-------------------------------------------------------------------------------
192.0.2.69      262118        1           Yes             06/11/2015 19:59:03
                ldp
192.0.2.70      262139        0           Yes             06/11/2015 19:59:03
                ldp
192.0.2.70      262140        1           No              06/11/2015 19:59:03
                ldp
192.0.2.72      262140        0           Yes             06/11/2015 19:59:03
                ldp
192.0.2.72      262141        1           No              06/11/2015 19:59:03
                ldp
192.0.2.73      262139        0           Yes             06/11/2015 19:59:03
                ldp
192.0.2.254     262142        0           Yes             06/11/2015 19:59:03
                bgp
-------------------------------------------------------------------------------
Number of entries : 7
-------------------------------------------------------------------------------
===============================================================================
===============================================================================
BGP EVPN-MPLS Ethernet Segment Dest
===============================================================================
Eth SegId                     TEP Address     Egr Label   Last Change
                                               Transport
-------------------------------------------------------------------------------
01:74:13:00:74:13:00:00:74:13 192.0.2.73      262140      06/11/2015 19:59:03
                                              ldp
-------------------------------------------------------------------------------
Number of entries : 1
-------------------------------------------------------------------------------
===============================================================================
```

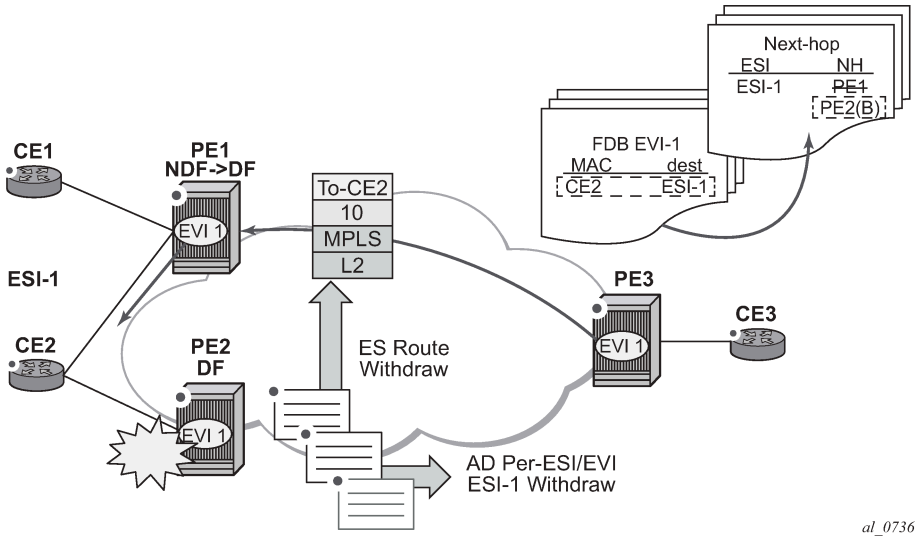
If PE3 sees only two single-active PEs in the same ESI, the second PE is the backup PE. Upon receiving an AD per-ES/per-EVI route withdrawal for the ESI from the primary PE, the PE3 starts sending the unicast traffic to the backup PE immediately.

If PE3 receives AD routes for the same ESI and EVI from more than two PEs, the PE does not install any backup route in the data path. Upon receiving an AD per-ES/per-EVI route withdrawal for the ESI, it flushes the MACs associated with the ESI.

6.3.2.4.2.4 Network failures and convergence for single-active multihoming

Figure 168: Single-active multihoming ES failure shows the remote PE (PE3) behavior when there is an Ethernet-Segment failure.

Figure 168: Single-active multihoming ES failure



The PE3 behavior for unicast traffic is as follows:

1. PE3 forwards MAC DA = CE2 to PE2 when the MAC advertisement route came from PE2 and the set of Ethernet AD per-ES routes and Ethernet AD per-EVI routes from PE1 and PE2 are active at PE3.
2. If there was a failure between CE2 and PE2, PE2 would withdraw its set of Ethernet AD and ES routes, then PE3 would immediately forward the traffic destined for CE2 to PE1 only (the backup PE). PE3 does not need to wait for the withdrawal of the individual MAC.
3. If after step 2, PE2 withdraws its MAC advertisement route, PE3 treats traffic to MAC DA = CE2 as unknown unicast, unless the MAC has been previously advertised by PE1.

Also, a DF election on PE1 is triggered. In general, a DF election is triggered by the same events as for all-active multihoming. In this case, the DF forwards traffic to CE2 when the **esi-activation-timer** expiration occurs (the timer kicks in when there is a transition from non-DF to DF).

6.3.2.4.3 EVPN ESI type 1 support

According to RFC 7432, specific Ethernet Segment Identifier (ESI) types support auto-derivation and the 10-byte ESI value does not need to be configured. SR OS supports the manual configuration of 10-byte ESI for the Ethernet segment, or alternatively, the auto-derivation of EVPN type 1 ESIs.

The **auto-esi {none|type-1}** command is supported in the Ethernet segment configuration. The default mode is **none** and it forces the user to configure a manual ESI. When **type-1** is configured, a manual ESI cannot be configured in the ES and the ESI is auto-derived in accordance with the RFC 7432 ESI type 1 definition. An ESI type 1 encodes 0x01 in the ESI type octet (T=0x01) and indicates that IEEE 802.1AX LACP is used between the PEs and CEs.

The ESI is auto-derived from the CE's LACP PDUs by concatenating the following parameters:

- **CE LACP system MAC address (6 octets)**

  The CE LACP system MAC address is encoded in the high-order 6 octets of the ESI value field.

- **CE LACP port key (2 octets)**

  The CE LACP port key is encoded in the 2 octets next to the system MAC address.

The remaining octet is set to 0x00.

The following usage considerations apply to auto-ESI type 1:

- ESI type 1 is only supported on non-virtual Ethernet segments associated with LAGs when LACP is enabled.
- Single-active or all-active modes are supported. When used with a single-active node, the CE must be attached to the PEs by a single LAG, which allows the multihomed PEs to auto-derive the same ESI.
- Changing the **auto-esi** command requires an ES shutdown.
- When the ES is enabled but the ESI has not yet been auto-derived, no multihoming routes are advertised for the ES. ES and AD routes are advertised only after ESI type 1 is auto-derived and the ES is enabled.
- When the ES LAG is operationally down as a result of the ports or LACP going down, the previously auto-derived ESI is retained. Consequently, convergence is not impacted when the LAG comes back up; if the CE's LACP information is changed, the ES goes down and a new auto-derived type 1 ESI is generated.

6.3.3 P2MP mLDP tunnels for BUM traffic in EVPN-MPLS services

P2MP mLDP tunnels for BUM traffic in EVPN-MPLS services are supported and enabled through the use of the provider-tunnel context. If EVPN-MPLS takes ownership over the provider-tunnel, **bgp-ad** is still supported in the service but it does not generate BGP updates, including the PMSI Tunnel Attribute. The following CLI example shows an EVPN-MPLS service that uses P2MP mLDP LSPs for BUM traffic.

```
*A:PE-1>config>service>vpls(vpls or b-vpls)# info
----------------------------------------------
  description "evpn-mpls-service with p2mp mLDP"
  bgp-evpn
    evi 10
    no ingress-repl-inc-mcast-advertisement
    mpls bgp 1
      no shutdown
      auto-bind-tunnel resolution any
  exit
```

```
  provider-tunnel
    inclusive
      owner bgp-evpn-mpls
      root-and-leaf
      mldp
      no shutdown
      exit
    exit
  sap 1/1/1:1 create
  exit
  spoke-sdp 1:1 create
 exit
```

When **provider-tunnel inclusive** is used in EVPN-MPLS services, the following commands can be used in the same way as for BGP-AD or BGP-VPLS services:

- **data-delay-interval**
- **root-and-leaf**
- **mldp**
- **shutdown**

The following commands are used by **provider-tunnel** in BGP-EVPN MPLS services:

- **[no] ingress-repl-inc-mcast-advertisement**

  This command allows you to control the advertisement of IMET-IR and IMET-P2MP-IR routes for the service. See BGP-EVPN control plane for MPLS tunnels for a description of the IMET routes. The following considerations apply:

  – If configured as **no ingress-repl-inc-mcast-advertisement**, the system does not send the IMET-IR or IMET-P2MP-IR routes, regardless of the service being enabled for BGP-EVPN MLPLS or BGP-EVPN VXLAN.

  – If configured as **ingress-repl-inc-mcast-advertisement** and the PE is **root-and-leaf**, the system sends an IMET-P2MP-IR route.

  – If configured as **ingress-repl-inc-mcast-advertisement** and the PE is **no root-and-leaf**, the system sends an IMET-IR route.

  – Default value is **ingress-repl-inc-mcast-advertisement**.

- **[no] owner {bgp-ad | bgp-vpls | bgp-evpn-mpls}**

  The owner of the provider tunnel must be configured. The default value is **no owner**. The following considerations apply:

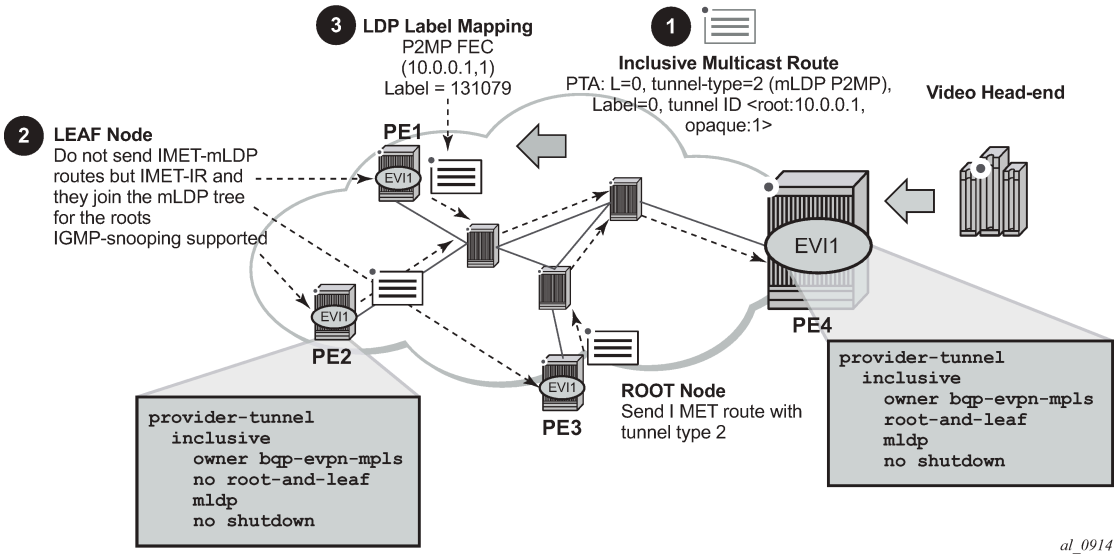
  – Only one of the protocols supports a provider tunnel in the service and it must be explicitly configured.

  – **bgp-vpls** and **bgp-evpn** are mutually exclusive.

  – While **bgp-ad** and **bgp-evpn** can coexist in the same service, only **bgp-evpn** can be the provider-tunnel owner in such cases.

Figure 169: EVPN services with p2mp mLDP—control plane shows the use of P2MP mLDP tunnels in an EVI with a root node and a few leaf-only nodes.

Figure 169: EVPN services with p2mp mLDP—control plane



Consider the use case of a root-and-leaf PE4 where the other nodes are configured as leaf-only nodes (**no root-and-leaf**). This scenario is handled as follows:

1. If **ingress-repl-inc-mcast-advertisement** is configured, then as soon as the **bgp-evpn mpls** option is enabled, the PE4 sends an IMET-P2MP route (tunnel type mLDP), or optionally, an IMET-P2MP-IR route (tunnel type composite). IMET-P2MP-IR routes allow leaf-only nodes to create EVPN-MPLS multicast destinations and send BUM traffic to the root.
2. If **ingress-repl-inc-mcast-advertisement** is configured, PE1/2/3 do not send IMET-P2MP routes; only IMET-IR routes are sent.
   - The **root-and-leaf** node imports the IMET-IR routes from the leaf nodes but it only sends BUM traffic to the P2MP tunnel as long as it is active.
   - If the P2MP tunnel goes operationally down, the **root-and-leaf** node starts sending BUM traffic to the evpn-mpls multicast destinations
3. When PE1/2/3 receive and import the IMET-P2MP or IMET-P2MP-IR from PE4, they join the mLDP P2MP tree signaled by PE4. They issue an LDP label-mapping message including the corresponding P2MP FEC.

As described in IETF Draft *draft-ietf-bess-evpn-etree*, mLDP and Ingress Replication (IR) can work in the same network for the same service; that is, EVI1 can have some nodes using mLDP (for example, PE1) and others using IR (for example, PE2). For scaling, this is significantly important in services that consist of a pair of root nodes sending BUM in P2MP tunnels and hundreds of leaf-nodes that only need to send BUM traffic to the roots. By using IMET-P2MP-IR routes from the roots, the operator makes sure the leaf-only nodes can send BUM traffic to the root nodes without the need to set up P2MP tunnels from the leaf nodes.

When both static and dynamic P2MP mLDP tunnels are used on the same router, Nokia recommends that the static tunnels use a tunnel ID lower than 8193. If a tunnel ID is statically configured with a value equal to or greater than 8193, BGP-EVPN may attempt to use the same tunnel ID for services with **enabled provider-tunnel**, and fail to set up an mLDP tunnel.

Inter-AS option C or seamless-MPLS models for non-segmented mLDP trees are supported with EVPN for BUM traffic. The leaf PE that joins an mLDP EVPN root PE supports Recursive and Basic Opaque FEC elements (types 7 and 1, respectively). Therefore, packet forwarding is handled as follows:

- The ABR or ASBR may leak the root IP address into the leaf PE IGP, which allows the leaf PE to issue a Basic opaque FEC to join the root.
- The ABR or ASBR may distribute the root IP using BGP label-ipv4, which results in the leaf PE issuing a Recursive opaque FEC to join the root.

For more information about mLDP opaque FECs, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Layer 3 Services Guide: IES and VPRN* and the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide*.

All-active multihoming and single-active with an ESI label multihoming are supported in EVPN-MPLS services together with P2MP mLDP tunnels. Both use an upstream-allocated ESI label, as described in RFC 7432 section 8.3.1.2, which is popped at the leaf PEs, resulting in the requirement that, in addition to the root PE, all EVPN-MPLS P2MP leaf PEs must support this capability (including the PEs not connected to the multihoming ES).

6.3.4 PBB-EVPN

This section contains information about PBB-EVPN.

6.3.4.1 BGP-EVPN control plane for PBB-EVPN

PBB-EVPN uses a reduced subset of the routes and procedures described in RFC 7432. The supported routes are:

- ES routes
- MAC/IP routes
- Inclusive Multicast Ethernet Tag routes

6.3.4.1.1 EVPN route type 3 - inclusive multicast Ethernet tag route

This route is used to advertise the ISIDs that belong to I-VPLS services as well as the default multicast tree. PBB-Epipe ISIDs are not advertised in Inclusive Multicast routes. The following fields are used:

- Route Distinguisher is taken from the RD of the B-VPLS service within the BGP context. The RD can be configured or derived from the value of the **bgp-evpn evi**.
- Ethernet Tag ID encodes the ISID for a specified I-VPLS.
- IP address length is always 32.
- Originating router's IP address carries an IPv4 or IPv6 address.
- PMSI attribute:
  - Tunnel type = Ingress replication (6).
  - Flags = Leaf not required.
  - MPLS label carries the MPLS label allocated for the service in the high-order 20 bits of the label field.

Note: This label is the same label used in the B-MAC routes for the same B-VPLS service unless **bgp-evpn mpls ingress-replication-bum-label** is configured in the B-VPLS service.

– Tunnel endpoint = equal to the originating IP address.

Note: The mLDP P2MP tunnel type is supported on PBB-EPVN services, but it can be used in the default multicast tree only.

6.3.4.1.2 EVPN route type 2 - MAC/IP advertisement route (or B-MAC routes)

The 7750 SR, 7450 ESS, or 7950 XRS generates this route type for advertising B-MAC addresses for the following:

- Learned MACs on B-SAPs or B-SDP bindings (if mac-advertisement is enabled)
- Conditional static MACs (if mac-advertisement is enabled)
- B-VPLS shared B-MACs (**source-bmacs**) and dedicated B-MACs (**es-bmacs**).

The route type 2 generated by the router uses the following fields and values:

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context. The RD can be configured or derived from the **bgp-evpn evi** value.
- Ethernet Segment Identifier (ESI):
  - ESI = 0 for the advertisement of source-bmac, es-bmacs, sap-bmacs, or sdp-bmacs if no multihoming or single-active multihoming is used.
  - ESI=MAX-ESI (0xFF..FF) in the advertisement of es-bmacs used for all-active multihoming.
  - ESI different from zero or MAX-ESI for learned B-MACs on B-SAPs/SDP bindings if EVPN multihoming is used on B-VPLS SAPs and SDP bindings.
- Ethernet Tag ID is 0.

  Note: A different Ethernet Tag value may be used only when **send-bvpls-evpn-flush** is enabled.

- MAC address length is always 48.
- B-MAC address (learned, configured, or system-generated).
- IP address length zero and IP address omitted.
- MPLS Label 1 carries the MPLS label allocated by the system to the B-VPLS service. The label value is encoded in the high-order 20 bits of the field and is the same label used in the routes type 3 for the same service unless BGP-EVPN MPLS **ingress-replication-bum-label** is configured in the service.
- The MAC Mobility extended community:
  - The MAC mobility extended community is used in PBB-EVPN for C-MAC flush purposes if per ISID load balancing (single-active multihoming) is used and a source-bmac is used for traffic coming from the ESI.

If there is a failure in one of the ES links, C-MAC flush through the withdrawal of the B-MAC cannot be done (other ESIs are still working); therefore, the MAC mobility extended community is used to signal C-MAC flush to the remote PEs.

- When a dedicated es-bmac per ESI is used, the MAC flush can be based on the withdrawal of the B-MAC from the failing node.
- es-bmacs are advertised as static (sticky bit set).
- Source-bmacs are advertised as static MACs (sticky bit set). In the case of an update, if advertised to indicate that C-MAC flush is needed, the MAC mobility extended community is added to the B-MAC route including a higher sequence number (than the one previously advertised) in addition to the sticky bit.

6.3.4.1.3 EVPN route type 4 - ES route

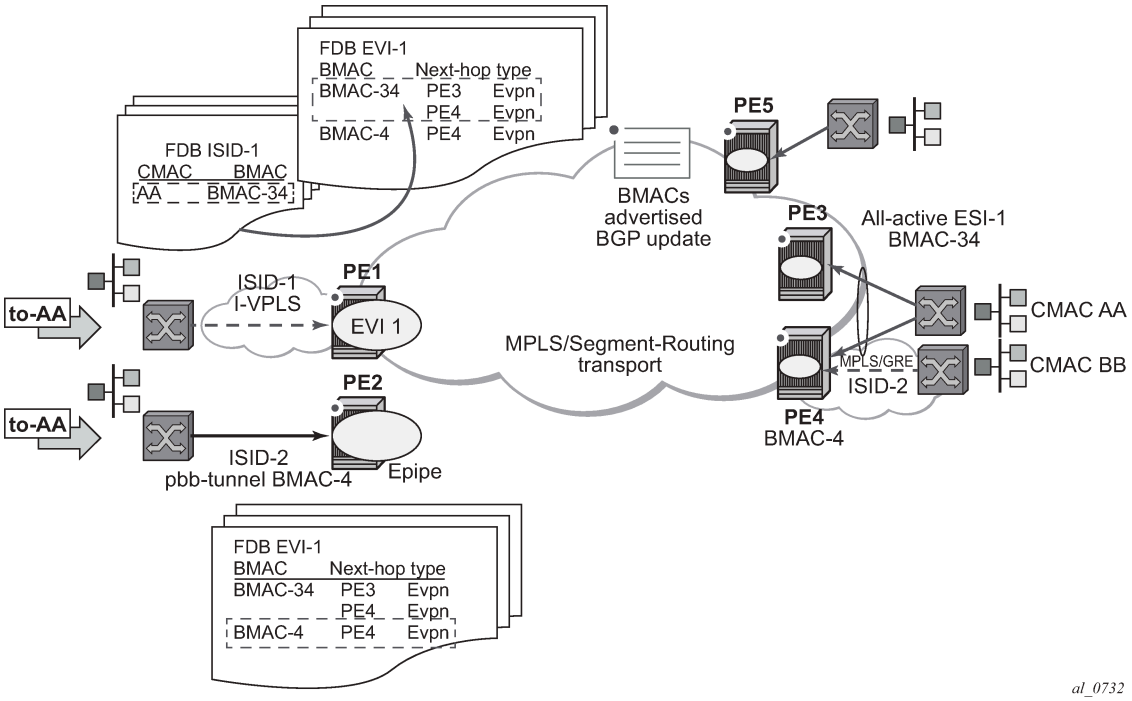
This route type is used for DF election as described in section BGP-EVPN control plane for MPLS tunnels.

Note: The EVPN route type 1—Ethernet Auto Discovery route is not used in PBB-EVPN.

6.3.4.2 PBB-EVPN for I-VPLS and PBB Epipe services

The 7750 SR, 7450 ESS, and 7950 XRS SR OS implementation of PBB-EVPN reuses the existing PBB-VPLS model, where N I-VPLS (or Epipe) services can be linked to a B-VPLS service. BGP-EVPN is enabled in the B-VPLS and the B-VPLS becomes an EVI (EVPN Instance). Figure 170: PBB-EVPN for I-VPLS and PBB Epipe services shows the PBB-EVPN model in the SR OS.

Figure 170: PBB-EVPN for I-VPLS and PBB Epipe services



Each PE in the B-VPLS domain advertises its **source-bmac** as either configured in **vpls>pbb>source-bmac** or auto-derived from the chassis MAC. The remote PEs install the advertised B-MACs in the B-VPLS FDB. If a specified PE is configured with an **ethernet-segment** associated with an I-VPLS or PBB Epipe, it may also advertise an **es-bmac** for the Ethernet-Segment.

In the example shown in Figure 170: PBB-EVPN for I-VPLS and PBB Epipe services, when a frame with MAC DA = AA gets to PE1, a MAC lookup is performed on the I-VPLS FDB and B-MAC-34 is found. A B-MAC lookup on the B-VPLS FDB yields the next-hop (or next-hops if the destination is in an all-active Ethernet-Segment) to which the frame is sent. As in PBB-VPLS, the frame is encapsulated with the corresponding PBB header. A label specified by EVPN for the B-VPLS and the MPLS transport label are also added.

If the lookup on the I-VPLS FDB fails, the system sends the frame encapsulated into a PBB packet with B-MAC DA = Group B-MAC for the ISID. That packet is distributed to all the PEs where the ISID is defined and contains the EVPN label distributed by the Inclusive Multicast routes for that ISID, as well as the transport label.

For PBB-Epipes, all the traffic is sent in a unicast PBB packet to the B-MAC configured in the **pbb-tunnel**.

The following CLI output shows an example of the configuration of an I-VPLS, PBB-Epipe, and their corresponding B-VPLS.

```
*A:PE-1>config#

service vpls 1 name "b-vpls1" b-vpls create
  description "pbb-evpn-service"
  service-mtu 2000
  pbb
    source-bmac 00:00:00:00:00:03
  bgp
  bgp-evpn
```

```
    evi 1
    mpls bgp 1
      no shutdown
      auto-bind-tunnel resolution any
  sap 1/1/1:1 create
  exit
  spoke-sdp 1:1 create


*A:PE-1>config#

service vpls 101 name "vpls101" i-vpls create
  pbb
    backbone-vpls 1
  sap 1/2/1:101 create
  spoke-sdp 1:102 create

*A:PE-1>config#

service epipe 102 name "epipe102" create
  pbb
    tunnel 1 backbone-dest-mac 00:00:00:00:00:01 isid 102
  sap 1/2/1:102 create
```

Configure the **bgp-evpn** context as described in section EVPN for MPLS tunnels in VPLS services (EVPN-MPLS).

Some EVPN configuration options are not relevant to PBB-EVPN and are not supported when BGP-EVPN is configured in a B-VPLS; these are as follows:

- **bgp-evpn> [no] ip-route-advertisement**
- **bgp-evpn> [no] unknown-mac-route**
- **bgp-evpn> vxlan [no] shutdown**
- **bgp-evpn>mpls>force-vlan-vc-forwarding**

When **bgp-evpn>mpls no shutdown** is added to a specified B-VPLS instance, the following considerations apply:

- BGP-AD is supported along with EVPN in the same B-VPLS instance.
- The following B-VPLS and BGP-EVPN commands are fully supported:
  - **vpls>backbone-vpls**
  - **vpls>backbone-vpls>send-flush-on-bvpls-failure**
  - **vpls>backbone-vpls>source-bmac**
  - **vpls>backbone-vpls>use-sap-bmac**
  - **vpls>backbone-vpls>use-es-bmac** (For more information, see PBB-EVPN multihoming in I-VPLS and PBB Epipe services)
  - **vpls>isid-policies**
  - **vpls>static-mac**
  - **vpls>SAP or SDP-binding>static-isid**
  - **bgp-evpn>mac-advertisement -** this command affects the 'learned' B-MACs on SAPs or SDP bindings and not on the system B-MAC or SAP/es-bmacs being advertised.
  - **bgp-evpn>mac-duplication** and settings.

- **bgp-evpn>mpls>auto-bind-tunnel** and options.
- **bgp-evpn>mpls>ecmp**
- **bgp-evpn>mpls>control-word**
- **bgp-evpn>evi**
- **bgp-evpn>mpls>ingress-replication-bum-label**

6.3.4.2.1 Flood containment for I-VPLS services

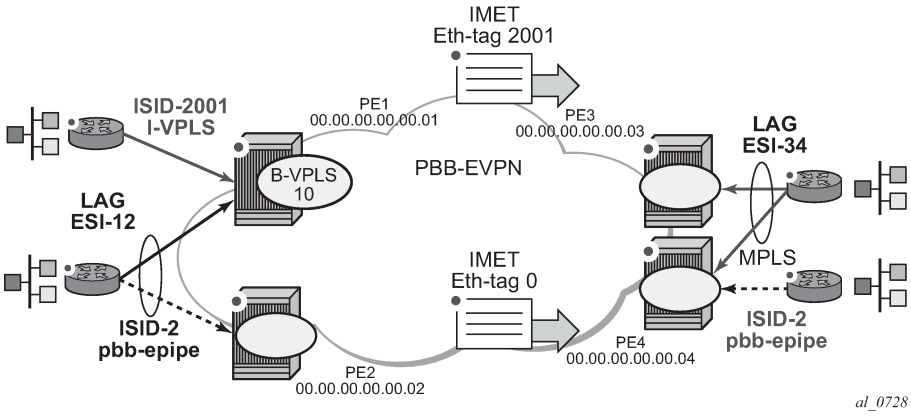
In general, PBB technologies in the 7750 SR, 7450 ESS, or 7950 XRS SR OS support a way to contain the flooding for a specified I-VPLS ISID, so that BUM traffic for that ISID only reaches the PEs where the ISID is locally defined. Each PE creates an MFIB per I-VPLS ISID on the B-VPLS instance. That MFIB supports SAP or SDP bindings endpoints that can be populated by:

- MMRP in regular PBB-VPLS
- IS-IS in SPBM

In PBB-EVPN, B-VPLS EVPN endpoints can be added to the MFIBs using EVPN Inclusive Multicast Ethernet Tag routes.

The example in Figure 171: PBB-EVPN and I-VPLS flooding containment shows how the MFIBs are populated in PBB-EVPN.

Figure 171: PBB-EVPN and I-VPLS flooding containment



When the B-VPLS 10 is enabled, PE1 advertises as follows:

- A B-MAC route containing PE1's system B-MAC (00:01 as configured in **pbb>source-bmac**) along with an MPLS label.
- An Inclusive Multicast Ethernet Tag route (IMET route) with Ethernet-tag = 0 that allows the remote B-VPLS 10 instances to add an entry for PE1 in the default multicast list.

Note: The MPLS label that is advertised for the MAC routes and the inclusive multicast routes for a specified B-VPLS can be the same label or a different label. As in regular EVPN-MPLS, this depends on the **[no] ingress-replication-bum-label** command.

When I-VPLS 2001 (ISID 2001) is enabled as per the CLI in the preceding section, PE1 advertises as follows:

An additional inclusive multicast route with Ethernet-tag = 2001. This allows the remote PEs to create an MFIB for the corresponding ISID 2001 and add the corresponding EVPN binding entry to the MFIB.

This default behavior can be modified by the configured **isid-policy**. For instance, for ISIDs 1-2000, configure as follows:

```
isid-policy
    entry 10 create
      no advertise-local
      range 1 to 2000
      use-def-mcast
```

This configuration has the following effect for the ISID range:

- **no advertise-local** instructs the system to not advertise the local active ISIDs contained in the 1 to 2001 range.
- **use-def-mcast** instructs the system to use the default flooding list as opposed to the MFIB.

The ISID flooding behavior on B-VPLS SAPs and SDP bindings is as follows:

- B-VPLS SAPs and SDP bindings are only added to the TLS-multicast list and not to the MFIB list (unless **static-isids** are configured, which is only possible for SAPs/SDP bindings and not BGP-AD spoke SDPs).

  As a result, if the system needs to flood ISID BUM traffic and the ISID is also defined in remote PEs connected through SAPs or spoke SDPs without **static-isids**, then an **isid-policy** must be configured for the ISID so that the ISID uses the default multicast list.
- When an **isid-policy** is configured and a range of ISIDs use the default multicast list, the remote PBB-EVPN PEs are added to the default multicast list as long as they advertise an IMET route with an ISID included in the policy's ISID range. PEs advertising IMET routes with Ethernet-tag = 0 are also added to the default multicast list (7750 SR, 7450 ESS, or 7950 XRS SR OS behavior).
- The B-VPLS 10 also allows the ISID flooding to legacy PBB networks via B-SAPs or B-SDPs. The legacy PBB network B-MACs are dynamically learned on those SAPs/binds or statically configured through the use of conditional **static-macs**. The use of **static-isids** is required so that non-local ISIDs are advertised.

```
sap 1/1/1:1 create
exit
spoke-sdp 1:1 create
  static-mac
      mac 00:fe:ca:fe:ca:fe create sap 1/1/1:1 monitor fwd-status
  static-isid
      range 1 isid 3000 to 5000 create
```

Note: The configuration of PBB-Epipes does not trigger any IMET advertisement.

6.3.4.2.2 PBB-EVPN and PBB-VPLS integration

The 7750 SR, 7450 ESS, and 7950 XRS SR OS EVPN implementation supports RFC 8560 so that PBB-EVPN and PBB-VPLS can be integrated into the same network and within the same B-VPLS service.

All the concepts described in section EVPN and VPLS integration are also supported in B-VPLS services so that B-VPLS SAP or SDP bindings can be integrated with PBB-EVPN destination bindings. The features

described in that section also facilitate a smooth migration from B-VPLS SDP bindings to PBB-EVPN destination bindings.

6.3.4.2.3 PBB-EVPN multihoming in I-VPLS and PBB Epipe services

The 7750 SR, 7450 ESS, and 7950 XRS SR OS PBB-EVPN implementation supports all-active and single-active multihoming for I-VPLS and PBB Epipe services.

PBB-EVPN multihoming reuses the **ethernet-segment** concept described in section EVPN multihoming in VPLS services. However, unlike EVPN-MPLS, PBB-EVPN does not use AD routes; it uses B-MACs for split-horizon checks and aliasing.

6.3.4.2.3.1 System B-MAC assignment in PBB-EVPN

RFC 7623 describes two types of B-MAC assignments that a PE can implement:

- shared B-MAC addresses that can be used for single-homed CEs and a number of multihomed CEs connected to Ethernet-Segments
- dedicated B-MAC addresses per Ethernet-Segment

In this document and in 7750 SR, 7450 ESS, and 7950 XRS terminology:

- A **shared-bmac** (in IETF) is a **source-bmac** as configured in **service>(b)vpls>pbb>source-bmac**
- A **dedicated-bmac** per ES (in IETF) is an **es-bmac** as configured in **service>pbb>use-es-bmac**

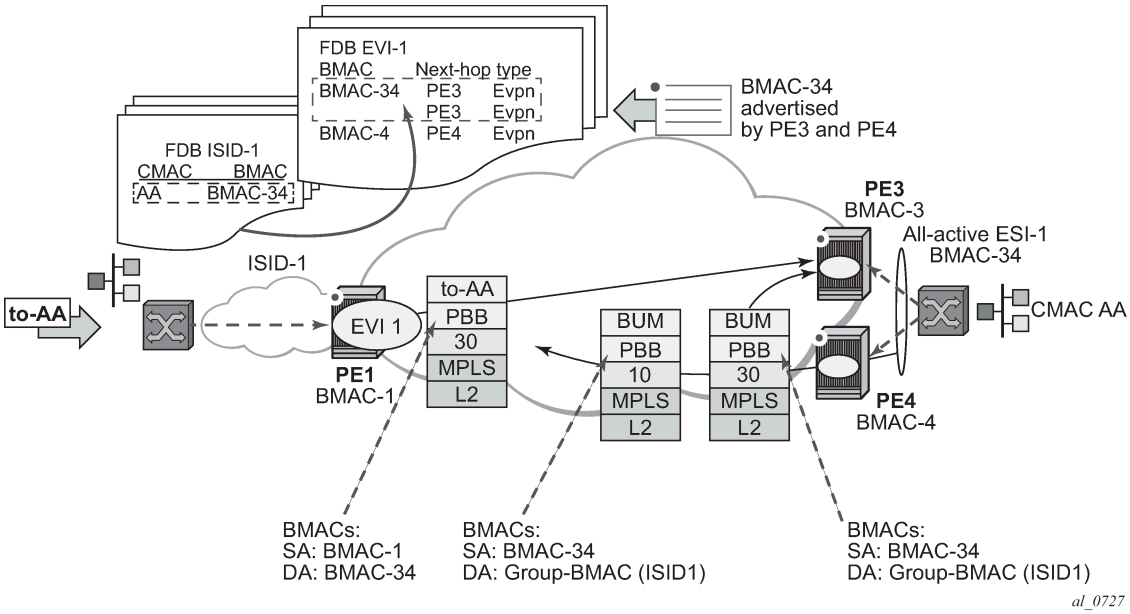
B-MAC selection and use depends on the multihoming model; for single-active mode, the type of B-MAC impacts the flooding in the network as follows:

- All-active multihoming requires **es-bmacs**.
- Single-active multihoming can use **es-bmacs** or **source-bmacs**.
  - The use of **source-bmacs** minimizes the number of B-MACs being advertised but has a larger impact on C-MAC flush upon ES failures.
  - The use of **es-bmacs** optimizes the C-MAC flush upon ES failures at the expense of advertising more B-MACs.

6.3.4.2.3.2 PBB-EVPN all-active multihoming service model

Figure 172: PBB-EVPN all-active multihoming shows the use of all-active multihoming in the 7750 SR, 7450 ESS, and 7950 XRS SR OS PBB-EVPN implementation.

Figure 172: PBB-EVPN all-active multihoming



For example, the following shows the ESI-1 and all-active configuration in PE3 and PE4. As in EVPN-MPLS, all-active multihoming is only possible if a LAG is used at the CE. All-active multihoming uses es-bmacs, that is, each ESI is assigned a dedicated B-MAC. All the PEs part of the ES source traffic using the same **es-bmac**.

In Figure 172: PBB-EVPN all-active multihoming and the following configuration, the **es-bmac** used by PE3 and PE4 is B-MAC-34 (for example, 00:00:00:00:00:34). The **es-bmac** for a specified **ethernet-segment** is configured by the **source-bmac-lsb** along with the **(b-)vpls>pbb>use-es-bmac** command.

Configuration in PE3:

```
*A:PE3>config>lag(1)# info
----------------------------------------------
  mode access
  encap-type dot1q
  port 1/1/1
  lacp active administrative-key 32768
  no shutdown

*A:PE3>config>service>system>bgp-evpn# info
----------------------------------------------
  route-distinguisher 10.3.3.3:0
  ethernet-segment ESI-1 create
    esi 00:34:34:34:34:34:34:34:34:34
    multi-homing all-active
    service-carving auto
    lag 1
    source-bmac-lsb 00:34 es-bmac-table-size 8
    no shutdown

*A:PE3>config>service>vpls 1(b-vpls)# info
----------------------------------------------
  bgp
  exit
  bgp-evpn
```

```
    evi 1
    mpls bgp 1
      no shutdown
      ecmp 2
      auto-bind-tunnel resolution any
  exit
  pbb
    source-bmac 00:00:00:00:00:03
    use-es-bmac

*A:PE3>config>service>vpls (i-vpls)# info
----------------------------------------------
  pbb
    backbone-vpls 1
  sap lag-1:101 create

*A:PE1>config>service>epipe (pbb)# info
----------------------------------------------
  pbb
    tunnel 1 backbone-dest-mac 00:00:00:00:00:01 isid 102
  sap lag-1:102 create
```

Configuration in PE4:

```
*A:PE4>config>lag(1)# info
----------------------------------------------
  mode access
  encap-type dot1q
  port 1/1/1
  lacp active administrative-key 32768
  no shutdown

*A:PE4>config>service>system>bgp-evpn# info
----------------------------------------------
  route-distinguisher 10.4.4.4:0
  ethernet-segment ESI-1 create
    esi 00:34:34:34:34:34:34:34:34:34
    multi-homing all-active
    service-carving auto
    lag 1
    source-bmac-lsb 00:34 es-bmac-table-size 8
    no shutdown


*A:PE4>config>service>vpls 1(b-vpls)# info
----------------------------------------------
  bgp
  exit
  bgp-evpn
    evi 1
    mpls bgp 1
      no shutdown
      ecmp 2
      auto-bind-tunnel resolution any
  exit
  pbb
    source-bmac 00:00:00:00:00:04
    use-es-bmac

*A:PE4>config>service>vpls (i-vpls)# info
----------------------------------------------
  pbb
```

```
    backbone-vpls 1
  sap lag-1:101 create

*A:PE4>config>service>epipe (pbb)# info
----------------------------------------------
  pbb
    tunnel 1 backbone-dest-mac 00:00:00:00:00:01 isid 102
  sap lag-1:102 create
```

The above configuration enables the all-active multihoming procedures for PBB-EVPN.

Note: The **ethernet-segment ESI-1** can also be used for regular VPLS services.

The following considerations apply when the ESI is used for PBB-EVPN:

- **ESI association**

  Only LAG is supported for all-active multihoming. The following commands are used for the LAG to ESI association:

  – **config>service>system>bgp-evpn>ethernet-segment# lag <id>**

  – **config>service>system>bgp-evpn>ethernet-segment# source-bmaclsb <MAC-lsb> [es-bmac-table-size <size>]**

  – Where:

    - The same ESI may be used for EVPN and PBB-EVPN services.
    - For PBB-EVPN services, the **source-bmac-lsb** attribute is mandatory and ignored for EVPN-MPLS services.
    - The **source-bmac-lsb** attribute must be set to a specific 2-byte value. The value must match on all the PEs part of the same ESI, for example, PE3 and PE4 for ESI-1. This means that the configured **pbb>source-bmac** on the two PEs for B-VPLS 1 must have the same 4 most significant bytes.
    - The **es-bmac-table-size** parameter modifies the default value (8) for the maximum number of virtual B-MACs that can be associated with the **ethernet-segment** (for example, **es-bmacs**). When the **source-bmac-lsb** is configured, the associated **es-bmac-table-size** is reserved out of the total FDB space.
    - When **multi-homing all-active** is configured within the **ethernet-segment**, only a LAG can be associated with it. The association of a port or an sdp is restricted by the CLI.

- **service-carving**

  If **service-carving** is configured in the ESI, the DF election algorithm is a modulo function of the ISID and the number of PEs part of the ESI, as opposed to a modulo function of evi and number of PEs (used for EVPN-MPLS).

- **service-carving mode manual**

  A **service-carving mode manual** option is added so that the user can control what PE is DF for a specified ISID. The PE is DF for the configured ISIDs and non-DF for the non-configured ISIDs.

- **DF election**

  An all-active Designated Forwarder (DF) election is also carried out for PBB-EVPN. In this case, the DF election defines which of the PEs of the ESI for a specified I-VPLS is the one able to send the downstream BUM traffic to the CE. Only one DF per ESI is allowed in the I-VPLS service, and the non-DF only blocks BUM traffic and in the downstream direction.

- **split-horizon function**

  In PBB-EVPN, the split-horizon function to avoid echoed packets on the CE is based on an ingress lookup of the ES B-MAC (as opposed to the ESI label in EVPN-MPLS). In Figure 172: PBB-EVPN all-active multihoming PE3 sends packets using B-MAC SA = BMAC-34. PE4 does not send those packets back to the CE because BMAC-34 is identified as the **es-bmac** for ESI-1.

- **aliasing**

  In PBB-EVPN, aliasing is based on the ES B-MAC sent by all the PEs part of the same ESI. See the following section for more information. In Figure 172: PBB-EVPN all-active multihoming PE1 performs load balancing between PE3 and PE4 when sending unicast flows to BMAC-34 (es-bmac for ESI-1).

In the configuration above, a PBB-Epipe is configured in PE3 and PE4, both pointing at the same remote **pbb tunnel backbone-dest-mac**. On the remote PE, for example PE1, the configuration of the PBB-Epipe points at the **es-bmac**:

```
*A:PE1>config>service>epipe (pbb)# info
----------------------------------------------
  pbb
    tunnel 1 backbone-dest-mac 00:00:00:00:00:34 isid 102
  sap 1/1/1:102 create
```

When PBB-Epipes are used in combination with all-active multihoming, Nokia recommends using **bgp-evpn mpls ingress-replication-bum-label** in the PEs where the **ethernet-segment** is created, that is in PE3 and PE4. This guarantees that in case of flooding in the B-VPLS service for the PBB Epipe, only the DF forwards the traffic to the CE.

Note: The PBB-Epipe traffic always uses B-MAC DA = unicast; therefore, the DF cannot check whether the inner frame is unknown unicast or not based on the group B-MAC. Therefore, the use of an EVPN BUM label is highly recommended.

Aliasing for PBB-Epipes with all-active multihoming only works if shared-queuing or ingress policing is enabled on the ingress PE Epipe. In any other case, the IOM sends the traffic to a single destination (no ECMP is used in spite of the **bgp-evpn mpls ecmp** setting).

All-active multihomed **es-bmacs** are treated by the remote PEs as **eES:MAX-ESI BMACs**. The following example shows the FDB in B-VPLS 1 in PE1 as shown in Figure 172: PBB-EVPN all-active multihoming:

```
*A:PE1# show service id 1 fdb detail

===============================================================================
Forwarding Database, Service 1
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:00:00:00:00:03 eMpls:                   EvpnS    06/12/15 15:35:39
                            192.0.2.3:262138
1         00:00:00:00:00:04 eMpls:                   EvpnS    06/12/15 15:42:52
                            192.0.2.4:262130
1         00:00:00:00:00:34 eES:                     EvpnS    06/12/15 15:35:57
                            MAX-ESI
-------------------------------------------------------------------------------
No. of MAC Entries: 3
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
===============================================================================
```

The **show service id evpn-mpls** on PE1 shows that the remote **es-bmac** (that is, 00:00:00:00:00:34) has two associated next-hops (for example, PE3 and PE4):

```
*A:PE1# show service id 1 evpn-mpls

===============================================================================
BGP EVPN-MPLS Dest
===============================================================================
TEP Address     Egr Label     Num. MACs   Mcast           Last Change
                 Transport
-------------------------------------------------------------------------------
192.0.2.3       262138        1           Yes             06/12/2015 15:34:48
                ldp
192.0.2.4       262130        1           Yes             06/12/2015 15:34:48
                ldp
-------------------------------------------------------------------------------
Number of entries : 2
-------------------------------------------------------------------------------
===============================================================================

===============================================================================
BGP EVPN-MPLS Ethernet Segment Dest
===============================================================================
Eth SegId                     TEP Address     Egr Label   Last Change
                                               Transport
-------------------------------------------------------------------------------
No Matching Entries
===============================================================================

===============================================================================
BGP EVPN-MPLS ES BMAC Dest
===============================================================================
VBMacAddr                   TEP Address     Egr Label     Last Change
                                             Transport
-------------------------------------------------------------------------------
00:00:00:00:00:34           192.0.2.3       262138        06/12/2015 15:34:48
                                            ldp
00:00:00:00:00:34           192.0.2.4       262130        06/12/2015 15:34:48
                                            ldp
-------------------------------------------------------------------------------
Number of entries : 2
-------------------------------------------------------------------------------
===============================================================================
```

6.3.4.2.3.3 Network failures and convergence for all-active multihoming

ES failures are resolved by the PEs withdrawing the **es-bmac**. The remote PEs withdraw the route and update their list of next-hops for a specified **es-bmac**.

No mac-flush of the I-VPLS FDB tables is required as long as the **es-bmac** is still in the FDB.

When the route corresponding to the last next-hop for a specified **es-bmac** is withdrawn, the es-bmac is flushed from the B-VPLS FDB and all the C-MACs associated with it are flushed too.

The following events trigger a withdrawal of the **es-bmac** and the corresponding next-hop update in the remote PEs:

- B-VPLS transition to operationally down status.
- Change of **pbb**>**source-bmac**.

- Change of **es-bmac** (or removal of **pbb use-es-bmac**).
- Ethernet-segment transition to operationally down status.

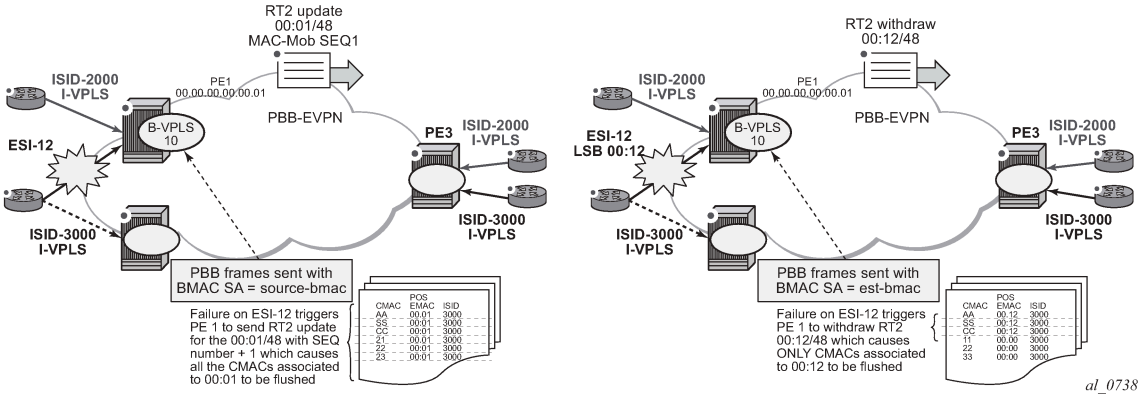
Note: Individual SAPs going operationally down in an ES do not generate any BGP withdrawal or indication so that the remote nodes can flush their C-MACs. This is solved in EVPN-MPLS by the use of AD routes per EVI; however, there is nothing similar in PBB-EVPN for indicating a partial failure in an ESI.

6.3.4.2.3.4 PBB-EVPN single-active multihoming service model

In single-active multihoming, the non-DF PEs for a specified ESI block unicast and BUM traffic in both directions (upstream and downstream) on the object associated with the ESI. Other than that, single-active multihoming follows the same service model defined in the PBB-EVPN all-active multihoming service model section with the following differences:

- The **ethernet-segment** is configured for **single-active**: **service>system>bgp-evpn>eth-seg>multi-homing single-active**.
- For single-active multihoming, the **ethernet-segment** can be associated with a port and sdp, as well as a **lag**.
- From a service perspective, single-active multihoming can provide redundancy to the following services and access types:
  - I-VPLS LAG and regular SAPs
  - I-VPLS active/standby spoke SDPs
  - EVPN single-active multihoming is supported for PBB-Epipes only in two-node scenarios with local switching.
- While all-active multihoming only uses **es-bmac** assignment to the ES, single-active multihoming can use source-bmac or **es-bmac** assignment. The system allows the following user choices per B-VPLS and ES:
  - A dedicated **es-bmac** per ES can be used. In that case, the **pbb use-es-bmac** command is configured in the B-VPLS and the same procedures described in PBB-EVPN all-active multihoming service model follow with one difference. While in all-active multihoming all the PEs part of the ESI source the PBB packets with the same source es-bmac, single-active multihoming requires the use of a different **es-bmac** per PE.
  - A non-dedicated **source-bmac** can be used. In this case, the user does not configure **pbb>use-es-bmac** and the regular **source-bmac** is used for the traffic. A different **source-bmac** has to be advertised per PE.
  - The use of **source-bmacs** or **es-bmacs** for single-active multihomed ESIs has a different impact on C-MAC flushing, as shown in Figure 173: Source-bmac versus es-bmac C-MAC flushing .

Figure 173: Source-bmac versus es-bmac C-MAC flushing



– If **es-bmacs** are used as shown in the representation on the right in Figure 173: Source-bmac versus es-bmac C-MAC flushing , a less-impacting C-MAC flush is achieved, therefore, minimizing the flooding after ESI failures. In case of ESI failure, PE1 withdraws the **es-bmac** 00:12 and the remote PE3 only flushes the C-MACs associated with that **es-bmac** (only the C-MACs behind the CE are flushed).

– If **source-bmacs** are used, as shown on the left side of Figure 173: Source-bmac versus es-bmac C-MAC flushing , in case of ES failure, a BGP update with higher sequence number is issued by PE1 and the remote PE3 flushes all the C-MACs associated with the **source-bmac**. Therefore, all the C-MACs behind the PE's B-VPLS are flushed, as opposed to only the C-MACs behind the ESI's CE.

- As in EVPN-MPLS, the non-DF status can be notified to the access CE or network:

  – **LAG with or without LACP**

    In this case, the multihomed ports on the CE are not part of the same LAG. The non-DF PE for each service may signal that the LAG SAP is operationally down by using **eth-cfm fault-propagation-enable {use-if-tlv|suspend-ccm}**.

  – **regular Ethernet 802.1q/ad ports**

    In this case, the multihomed ports on the CE/network are not part of any LAG. The non-DF PE for each service signals that the SAP is operationally down by using **eth-cfm fault-propagation-enable {use-if-tlv|suspend-ccm}**.

  – **active-standby PWs**

    In this case, the multihomed CE/network is connected to the PEs through an MPLS network and an active/standby spoke SDP per service. The non-DF PE for each service makes use of the LDP PW status bits to signal that the spoke SDP is standby at the PE side. Nokia recommends that the CE suppresses the signaling of PW status standby.

6.3.4.2.3.5 Network failures and convergence for single-active multihoming

ESI failures are resolved depending on the B-MAC address assignment chosen by the user:

- If the B-MAC address assignment is based on the use of **es-bmacs**, DF and non-DFs do send the **es-bmac/ESI=0** for a specified ESI. Each PE has a different **es-bmac** for the same ESI (as opposed to the same **es-bmac** on all the PEs for all-active).

In case of an ESI failure, the PE withdraws its **es-bmac** route triggering a mac-flush of all the C-MACs associated with it in the remote PEs.

- If the B-MAC address assignment is based on the use of **source-bmac**, DF and non-DFs advertise their respective **source-bmacs**. In case of an ES failure:
  - The PE re-advertises its **source-bmac** with a higher sequence number (the new DF does not re-advertise its **source-bmac**).
  - The far-end PEs interpret a **source-bmac** advertisement with a different sequence number as a flush-all-from-me message from the PE detecting the failure. They flush all the C-MACs associated with that B-MAC in all the ISID services.

The following events trigger a C-MAC flush notification. A 'C-MAC flush notification' means the withdrawal of a specified B-MAC or the update of B-MAC with a higher sequence number (SQN). Both BGP messages make the remote PEs flush all the C-MACs associated with the indicated B-MAC:

- B-VPLS transition to operationally down status. This triggers the withdrawal of the associated B-MACs, regardless of the **use-es-bmac** setting.
- Change of **pbb**>**source-bmac**. This triggers the withdrawal and re-advertisement of the **source-bmac**, causing the corresponding C-MAC flush in the remote PEs.
- Change of **es-bmac** (removal of **pbb use-es-bmac**). This triggers the withdrawal of the **es-bmac** and re-advertisement of the new **es-bmac**.
- Ethernet-Segment (ES) transition to operationally down or admin-down status. This triggers an **es-bmac** withdrawal (if **use-es-bmac** is used) or an update of the source-bmac with a higher SQN (if **no use-es-bmac** is used).
- Service Carving Range change for the ES. This triggers an **es-bmac** update with higher SQN (if **use-es-bmac** is used) or an update of the source-bmac with a higher SQN (if **no use-es-bmac** is used).
- Change in the number of candidate PEs for the ES. This triggers an **es-bmac** update with higher SQN (if **use-es-bmac** is used) or an update of the source-bmac with a higher SQN (if **no use-es-bmac** is used).
- In an ESI, individual SAPs/SDP bindings or individual I-VPLS going operationally down do not generate any BGP withdrawal or indication so that the remote nodes can flush their C-MACs. This is solved in EVPN-MPLS by the use of AD routes per EVI; however, there is nothing similar in PBB-EVPN for indicating a partial failure in an ESI.

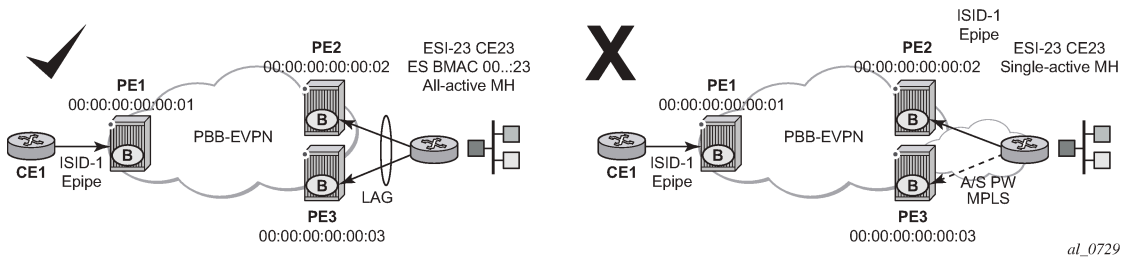
6.3.4.2.3.6 PBB-Epipes and EVPN multihoming

EVPN multihoming is supported with PBB-EVPN Epipes, but only in a limited number of scenarios. In general, the following applies to PBB-EVPN Epipes:

- PBB-EVPN Epipes do not support spoke SDPs that are associated with EVPN ESs.
- PBB-EVPN Epipes support all-active EVPN multihoming as long as no local-switching is required in the Epipe instance where the ES is defined.
- PBB-EVPN Epipes support single-active EVPN multihoming only in a two-node case scenario.

Figure 174: PBB-EVPN MH in a three-node scenario shows the EVPN MH support in a three-node scenario.

Figure 174: PBB-EVPN MH in a three-node scenario

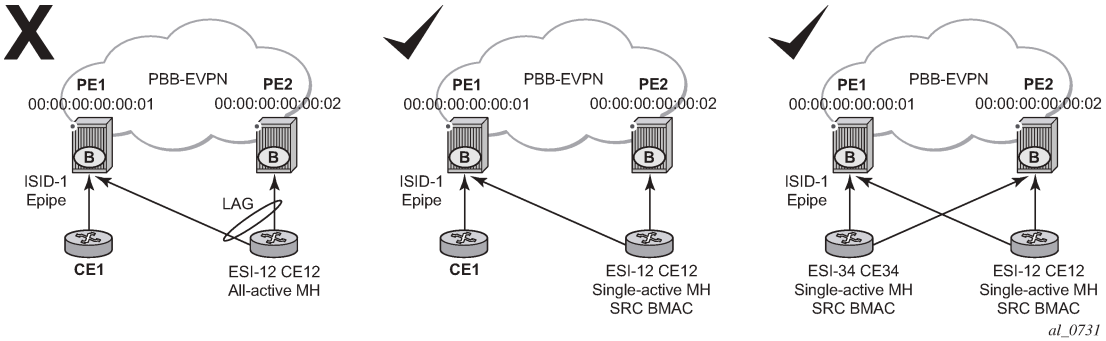


EVPN MH support in a three-node scenario has the following characteristics:

- All-active EVPN multihoming is fully supported (diagram on the left in Figure 174: PBB-EVPN MH in a three-node scenario). CE1 may also be multihomed to other PEs, as long as those PEs are not PE2 or PE3. In this case, PE1 Epipe's **pbb-tunnel** would be configured with the remote ES B-MAC.
- Single-active EVPN multihoming is not supported in a three (or more)-node scenario (diagram on the right in Figure 174: PBB-EVPN MH in a three-node scenario). Because PE1's Epipe **pbb-tunnel** can only point at a single remote B-MAC and single-active multihoming requires the use of separate B-MACs on PE2 and PE3, the scenario is not possible and not supported regardless of the ES association to port/LAG/sdps.
- Regardless of the EVPN multihoming type, the CLI prevents the user from adding a spoke SDP to an Epipe, if the corresponding SDP is part of an ES.

Figure 175: PBB-EVPN MH in a two-node scenario shows the EVPN MH support in a two-node scenario.

Figure 175: PBB-EVPN MH in a two-node scenario



EVPN MH support in a two-node scenario has the following characteristics, as shown in Figure 175: PBB-EVPN MH in a two-node scenario:

- All-active multihoming is not supported for redundancy in this scenario because PE1's **pbb-tunnel** cannot point at a locally defined ES B-MAC. This is represented in the left-most scenario in Figure 175: PBB-EVPN MH in a two-node scenario.
- Single-active multihoming is supported for redundancy in a two-node three or four SAP scenario, as displayed by the two right-most scenarios in Figure 175: PBB-EVPN MH in a two-node scenario).

  In these two cases, the Epipe **pbb-tunnel** is configured with the source B-MAC of the remote PE node.

  When two SAPs are active in the same Epipe, local-switching is used to exchange frames between the CEs.

6.3.4.2.4 PBB-EVPN and use of P2MP mLDP tunnels for default multicast list

P2MP mLDP tunnels can also be used in PBB-EVPN services. The use of provider-tunnel inclusive MLDP is only supported in the B-VPLS default multicast list; that is, no per-ISID IMET-P2MP routes are supported. IMET-P2MP routes in a B-VPLS are always advertised with Ethernet tag zero. All-active EVPN multihoming is supported in PBB-EVPN services together with P2MP mLDP tunnels; however, single-active multihoming is not supported. This capability is only required on the P2MP root PEs within PBB-EVPN services using all-active multihoming.

B-VPLS supports the use of MFIBs for ISIDs using ingress replication. The following considerations apply when **provider-tunnel** is enabled in a B-VPLS service:

- Local I-VPLS or static-ISIDs configured on the B-VPLS generate IMET-IR routes and MFIBs are created per ISID by default.
- The default IMET-P2MP or IMET-P2MP-IR route sent with Ethernet-tag = 0 is issued depending on the **ingress-repl-inc-mcast-advertisement** command.
- The following considerations apply if an **isid-policy** is configured in the B-VPLS.
  - A range of ISIDs configured with **use-def-mcast** make use of the P2MP tree, assuming the node is configured as **root-and-leaf**.
  - A range of ISIDs configured with **advertise-local** make the system advertise IMET-IR routes for the local ISIDs included in the range.

The following example CLI output shows a range of ISIDs (1000-2000) that use the P2MP tree and the system does not advertise the IMET-IR routes for those ISIDs. Other local ISIDs advertise the IMET-IR and use the MFIB to forward BUM packets to the EVPN-MPLS destinations created by the IMET-IR routes.

```
*A:PE-1>config>service>vpls(b-vpls)# info
----------------------------------------------
  service-mtu 2000
  bgp-evpn
    evi 10
    mpls bgp 1
      no shutdown
      auto-bind-tunnel resolution any
  isid-policy
    entry 10 create
      use-def-mcast
      no advertise-local
      range 1000 to 2000
      exit
    exit
  provider-tunnel
    inclusive
      owner bgp-evpn-mpls
      root-and-leaf
      mldp
      no shutdown
      exit
    exit
  sap 1/1/1:1 create
  exit
  spoke-sdp 1:1 create
 exit
```

6.3.4.2.5 PBB-EVPN ISID-based C-MAC flush

SR OS supports ISID-based C-MAC flush procedures for PBB-EVPN I-VPLS services where no single-active ESs are used. SR OS also supports C-MAC flush procedure where other redundancy mechanisms, such as BGP-MH, need these procedures to avoid blackholes caused by a SAP or spoke SDP failure.

The C-MAC flush procedures are enabled on the I-VPLS service using the **config**>**service**>**vpls**>**pbb**>**send-bvpls-evpn-flush** CLI command. The feature can be disabled on a per-SAP or per-spoke SDP basis by using the **disable-send-bvpls-evpn-flush** command in the **config**>**service**>**vpls**>**sap** or **config**>**service**>**vpls**>**spoke-sdp** context.

With the feature enabled on an I-VPLS service and a SAP or spoke SDP, if there is a SAP or spoke SDP failure, the router sends a C-MAC flush notification for the corresponding B-MAC and ISID. The router receiving the notification flushes all the C-MACs associated with the indicated B-MAC and ISID when the **config**>**service**>**vpls**>**bgp-evpn**>**accept-ivpls-evpn-flush** command is enabled for the B-VPLS service.

The C-MAC flush notification consists of an EVPN B-MAC route that is encoded as follows: the ISID to be flushed is encoded in the Ethernet Tag field and the sequence number is incremented with respect to the previously advertised route.

If **send-bvpls-evpn-flush** is configured on an I-VPLS with SAPs or spoke SDPs, one of the following rules must be observed:

- The **disable-send-bvpls-evpn-flush** option is configured on the SAPs or spoke SDPs.
- The SAPs or spoke SDPs are not on an ES.
- The SAPs or spoke SDPs are on an ES or vES with **no src-bmac-lsb** enabled.
- The **no use-es-bmac** is enabled on the B-VPLS.

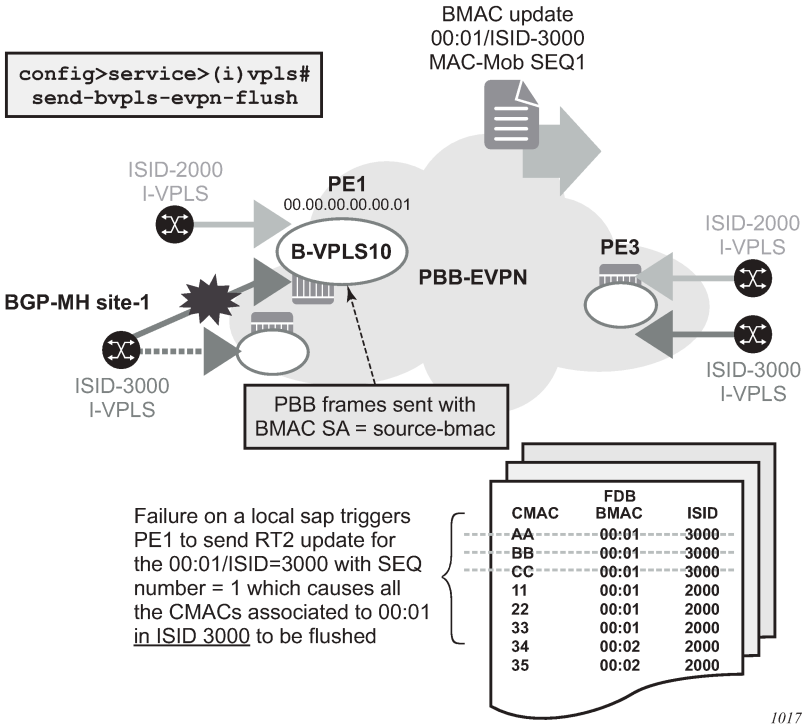
ISID-based C-MAC flush can be enabled in I-VPLS services with ES or vES. If enabled, the expected interaction between the RFC 7623-based C-MAC flush and the ISID-based C-MAC flush is as follows.

- If **send-bvpls-evpn-flush** is enabled in an I-VPLS service, the ISID-based C-MAC flush overrides (replaces) the RFC 7623-based C-MAC flushing.
- For each ES, vES, or B-VPLS, the system checks for at least one I-VPLS service that does not have **send-bvpls-evpn-flush** enabled.
  - If ISID-based C-MAC flush is enabled for all I-VPLS services, RFC 7623-based C-MAC flushing is not triggered; only ISID-based C-MAC flush notifications are generated.
  - If at least one I-VPLS service is found with no ISID-based C-MAC flush enabled, then RFC 7623-based C-MAC flushing notifications are triggered based on ES events.

    ISID-based C-MAC flush notifications are also generated for I-VPLS services that have **send-bvpls-evpn-flush** enabled.

Figure 176: Per-ISID C-MAC flush following a SAP failure shows an example where the ISID-based C-MAC flush prevents blackhole situations for a CE that is using BGP-MH as the redundancy mechanism in the I-VPLS with an ISID of 3000.

Figure 176: Per-ISID C-MAC flush following a SAP failure



When **send-bvpls-evpn-flush** is enabled, the I-VPLS service is ready to send per-ISID C-MAC flush messages in the form of B-MAC/ISID routes. The first B-MAC/ISID route for an I-VPLS service is sent with sequence number zero; subsequent updates for the same route increment the sequence number. A B-MAC/ISID route for the I-VPLS is advertised or withdrawn during the following cases:

- I-VPLS **send-bvpls-evpn-flush** configuration and deconfiguration
- I-VPLS association and disassociation from the B-VPLS service
- I-VPLS operational status change (up/down)
- B-VPLS operational status change (up/down)
- B-VPLS **bgp-evpn mpls** status change (no shutdown/shutdown)
- B-VPLS operational source B-MAC change

If **no disable-send-bvpls-evpn-flush** is configured for a SAP or spoke SDP, upon a failure on that SAP or spoke SDP, the system sends a per-ISID C-MAC flush message; that is, a B-MAC/ISID route update with an incremented sequence number.

If the user explicitly configures **disable-send-bvpls-evpn-flush** for a SAP or spoke SDP, the system does not send per-ISID C-MAC flush messages for failures on that SAP or spoke SDP.

The B-VPLS on the receiving node must be configured with **bgp-evpn>accept-ivpls-evpn-flush** to accept and process C-MAC flush non-zero Ethernet-tag MAC routes. If the **accept-ivpls-evpn-flush** command is enabled (the command is disabled by default), the node accepts non-zero Ethernet-tag MAC routes (B-MAC/ISID routes) and processes them. When a new B-MAC/ISID update (with an incremented sequence number) for an existing route is received, the router flushes all the C-MACs associated with that B-MAC and ISID. The B-MAC/ISID route withdrawals also cause a C-MAC flush.

Note: Only B-MAC routes with the Ethernet Tag field set to zero are considered for B-MAC installation in the FDB.

The following CLI example shows the commands that enable the C-MAC flush feature on PE1 and PE3.

```
*A:PE-1>config>service>vpls(i-vpls)# info
----------------------------------------------
  pbb
    backbone-vpls 10
    send-bvpls-evpn-flush
    exit
  exit
  bgp
    route-distinguisher 65000:1
    vsi-export "vsi_export"
    vsi-import "vsi_import"
  exit
  site "CE-1" create
    site-id 1
    sap lag-1:1
    site-activation-timer 3
    no shutdown
  exit
  sap lag-1:1 create
    no disable-send-bvpls-evpn-flush
    no shutdown
    exit
<snip>
*A:PE-3>config>service>vpls(b-vpls 10)# info
----------------------------------------------
<snip>
  bgp-evpn
    accept-ivpls-evpn-flush
```

In the preceding example, with **send-bvpls-evpn-flush** enabled on the I-VPLS service of PE1, a B-MAC/ISID route (for pbb source-bmac address B-MAC 00:..:01 and ISID 3000) is advertised. If the SAP goes operationally down, PE1 sends an update of the source B-MAC address (00:..:01) for ISID 3000 with a higher sequence number.

With **accept-ivpls-evpn-flush** enabled on PE3's B-VPLS service, PE3 flushes all C-MACs associated with B-MAC 00:01 and ISID 3000. The C-MACs associated with other B-MACs or ISIDs are retained in PE3's FDB.

6.3.4.2.6 PBB-EVPN ISID-based route targets

Routers with PBB-EVPN services use the following route types to advertise the ISID of a specific service:

- Inclusive Multicast Ethernet Tag routes (IMET-ISID routes) are used to auto-discover ISIDs in the PBB-EVPN network. The routes encode the service ISID in the Ethernet Tag field.
- BMAC-ISID routes are only used when ISID-based C-MAC flush is configured. The routes encode the ISID in the Ethernet Tag field.

Although the preceding routes are only relevant for routers where the advertised ISID is configured, they are sent with the B-VPLS route-target by default. As a result, the routes are unnecessarily disseminated to all the routers in the B-VPLS network.

SR OS supports the use of per-ISID or group of ISID route-targets, which limits the dissemination of IMET-ISID or BMAC-ISID routes for a specific ISID to the PEs where the ISID is configured.

The **config>service>(b-)vpls>isid-route-target>isid-range from** [**to to**] [**auto-rt** | **route-target rt**] command allows the user to determine whether the IMET-ISID and BMAC-ISID routes are sent with the B-VPLS route-target (default option, **no** command), or a route-target specific to the ISID or range of ISIDs.

The following configuration example shows how to configure ISID ranges as **auto-rt** or with a specific **route-target**.

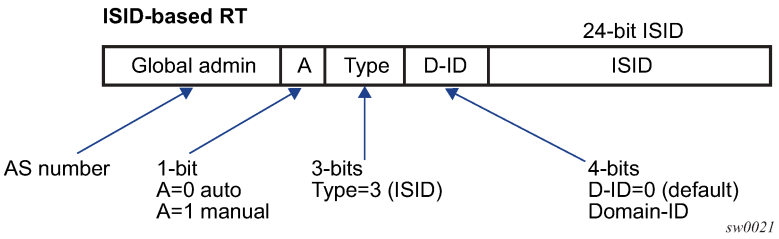
```
*A:PE-3>config>service>(b-)vpls>bgp-evpn#
isid-route-target
[no] isid-range <from> [to <to>] {auto-rt|route-target <rt>}
/* For example:
*A:PE-3>config>service>(b-)vpls>bgp-evpn#
isid-route-target
 isid-range 1000 to 1999 auto-rt
 isid-range 2000 route-target target:65000:2000
```

The **auto-rt** option auto-derives a route-target per ISID in the following format:

<2-byte-as-number>:<4-byte-value>

Where: 4-byte-value = 0x30+ISID, as described in RFC 7623. Figure 177: PBB-EVPN auto-rt ISID-based route target format shows the format of the **auto-rt** option.

Figure 177: PBB-EVPN auto-rt ISID-based route target format



Where:

- If it is 2 bytes, then the AS number is obtained from the **config>router>autonomous-system** command. If the AS number exceeds the 2 byte limit, then the low order 16-bit value is used.
- A = 0 for auto-derivation
- Type = 3, which corresponds to an ISID-type route-target
- ISID is the 24-bit ISID
- The type and sub-type are 0x00 and 0x02.

If **isid-route-target** is enabled, the export and import directions for IMET-ISID and BMAC-ISID route processing are modified as follows:

- Exported IMET-ISID and BMAC-ISID routes
  - For local I-VPLS ISIDs and static ISIDs, IMET-ISID routes are sent individually with an ISID-based route-target (and without a B-VPLS route-target) unless the ISID is contained in an ISID policy for which **no advertise-local** is configured.
  - If both **isid-route-target** and **send-bvpls-evpn-flush** options are enabled for an I-VPLS, the BMAC-ISID route is also sent with the ISID-based route-target and no B-VPLS route-target.
  - The **isid-route-target** command affects the IMET-ISID and BMAC-ISID routes only. The BMAC-0, IMET-0 (B-MAC and IMET routes with Ethernet Tag == 0), and ES routes are not impacted by the command.

- Imported IMET-ISID and BMAC-ISID routes
  - Upon enabling **isid-route-target** for a specific I-VPLS, the BGP starts importing IMET-ISID routes with ISID-based route-targets, and (assuming the **bgp-evpn accept-ivpls-evpn-flush** option is enabled) BMAC-ISID routes with ISID-based route-targets.
  - The new ISID-based RTs are added for import operations when the I-VPLS is associated with the B-VPLS service (and not based on the I-VPLS operational status), or when the **static-isid** is added.
  - The system does not maintain a mapping of the route-targets and ISIDs for the imported routes. For example, if I-VPLS 1 and 2 are configured with the **isid-route-target** option and IMET-ISID=2 route is received with a **route-target** corresponding to ISID=1, then BGP imports the route and the router processes it.
  - The router does not check the format of the received auto-derived route-targets. The route is imported as long as the route-target is on the list of RTs for the B-VPLS.
- If the **isid-route-target** option is configured for one or more I-VPLS services, the **vsi-import** and **vsi-export** policies are blocked in the B-VPLS. BGP peer import and export policies are still allowed. Matching on the export ISID-based route-target is supported.

6.3.5 EVPN-VPWS for MPLS tunnels

This section contains information about EVPN-VPWS for MPLS tunnels.

6.3.5.1 BGP-EVPN control plane for EVPN-VPWS

EVPN-VPWS for MPLS tunnels uses the RFC 8214 BGP extensions described in EVPN-VPWS for VXLAN tunnels, with the following differences for the Ethernet AD per-EVI routes:

- The MPLS field encodes an MPLS label as opposed to a VXLAN VNI.
- The C flag is set if the control word is configured in the service.

6.3.5.2 EVPN for MPLS tunnels in Epipe services (EVPN-VPWS)

The use and configuration of EVPN-VPWS services is described in EVPN-VPWS for VXLAN tunnels with the following differences when the EVPN-VPWS services use MPLS tunnels instead of VXLAN.

When MPLS tunnels are used, the **bgp-evpn**>**mpls** context must be configured in the Epipe. As an example, if Epipe 2 is an EVPN-VPWS service that uses MPLS tunnels between PE2 and PE4, this would be its configuration:

```
PE2>config>service>epipe(2)#
-----------------------
bgp
exit
bgp-evpn
  evi 2
  local-attachment-circuit "AC-1"
    eth-tag 200
exit
 remote-attachment-circuit "AC-2"
    eth-tag 200
exit
  mpls bgp 1
```

```
    ecmp 2
    no shutdown
exit
sap 1/1/1:1 create
```

```
PE4>config>service>epipe(2)#
----------------------
bgp
exit
bgp-evpn
  evi 2
  local-attachment-circuit "AC-2"
    eth-tag 200
exit
  remote-attachment-circuit "AC-1"
    eth-tag 100
exit
  mpls bgp 1
    ecmp 2
    no shutdown
exit
spoke-sdp 1:1
```

Where the following BGP-EVPN commands, specific to MPLS tunnels, are supported in the same way as in VPLS services:

- **mpls auto-bind-tunnel**
- **mpls control-word**
- **mpls entropy-label**
- **mpls force-vlan-vc-forwarding**
- **mpls shutdown**

EVPN-VPWS Epipes with MPLS tunnels can also be configured with the following characteristics:

- Access attachment circuits can be SAPs or spoke SDPs. Manually configured and BGP-VPWS spoke SDPs are supported. The VC switching configuration is not supported on BGP-EVPN-enabled pipes.
- EVPN-VPWS Epipes using null SAPs can be configured with **sap**>**ethernet**>**llf**. When enabled, upon removing the EVPN destination, the port is brought oper-down with flag LinkLossFwd, however the AD per EVI route for the SAP is still advertised (the SAP is kept oper-up). When the EVPN destination is created, the port is brought oper-up and the flag cleared.
- EVPN-VPWS Epipes for MPLS tunnels support **endpoints**. The parameter **endpoint endpoint name** is configurable along with **bgp-evpn**>**local-attachment-circuit** and **bgp-evpn**>**remote-attachment-circuit**. The following conditions apply to endpoints on EVPN-VPWS Epipes with MPLS tunnels:
  - Up to two explicit endpoints are allowed per Epipe service with BGP-EVPN configured.
  - A limited endpoint configuration is allowed in Epipes with BGP-EVPN. Specifically, neither active-hold-delay nor revert-time are configurable.
  - When **bgp-evpn**>**remote-attachment-circuit** is added to an explicit endpoint with a spoke SDP, the **spoke-sdp>precedence** command is not allowed. The spoke SDP always has a precedence of four, which is always higher than the EVPN precedence. Therefore, the EVPN-MPLS destination is used for transmission if it is created, and the spoke SDP is only used when the EVPN-MPLS destination is removed.
- EVPN-VPWS Epipes for MPLS tunnels support control word and entropy labels. When a control word is configured, the PE sets the C bit in its AD per-EVI advertisement and sends the control word in the

data path. In this case, the PE expects the control word to be received. If there is a mismatch between the received control word and the configured control word, the system does not set up the EVPN destination; as a result, the service does not come up.

- EVPN-VPWS Epipes support **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** command under **bgp-evpn mpls** and the **qinq-vlan-translation s-tag.c-tag** command on ingress QinQ SAPs.

  When QinQ VLAN translation is configured at the ingress QinQ or dot1q SAP, the service-delimiting outer and inner VLAN values can be translated to the configured values. The **force-qinq-vc-forwarding s-tag-c-tag** command must be configured to preserve the translated QinQ tags in the payload when sending EVPN packets. This translation and preservation behavior is aligned with the "normalization" concept described in *draft-ietf-bess-evpn-vpws-fxc*. The VLAN tag processing described in Epipe service pseudowire VLAN tag processing applies to EVPN destinations in EVPN-VPWS services too.

The following features, described in EVPN-VPWS for VXLAN tunnels, are also supported for MPLS tunnels:

- Advertisement of the Layer-2 MTU and consistency checking of the MTU of the AD per-EVI routes.
- Use of A/S PW and MC-LAG at access.
- EVPN multihoming, including:
  - Single-active and all-active
  - Regular or virtual ESs
  - All existing DF election modes

6.3.5.3 EVPN-VPWS services with local-switching support

Epipes with BGP-EVPN MPLS support the following configurations:

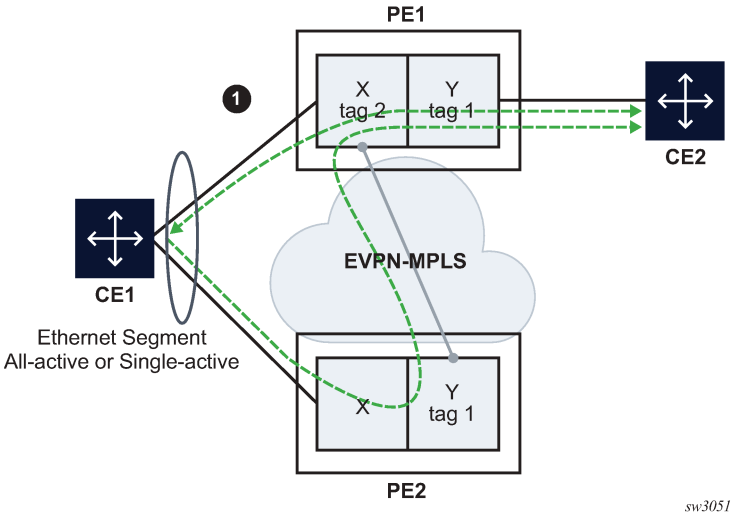
- up to two endpoints
- up to two SAPs associated with a different configured endpoint each
- two pairs of local/remote attachment circuit Ethernet tags, also associated with different configured endpoints
- EVPN destinations that can be used as Inter-Chassis Backup (ICB) links

The support of endpoints and up to two SAPs with local-switching allows two and three-node topologies for EVPN-VPWS. Figure 178: EVPN-VPWS endpoints example 1, Figure 179: EVPN-VPWS endpoints example 2, and Figure 180: EVPN-VPWS endpoints example 3 show examples of these topologies.

6.3.5.3.1 Example 1

The following figure shows an example of EVPN-VPWS endpoints.

Figure 178: EVPN-VPWS endpoints example 1



In Figure 178: EVPN-VPWS endpoints example 1, PE1 is configured with the following Epipe services:

```
endpoint X create
exit
endpoint Y create
exit
bgp-evpn
  evi 350
  local-attachment-circuit "CE-1" endpoint "Y" create
    eth-tag 1
  exit
  remote-attachment-circuit "ICB-1" endpoint "Y" create
    eth-tag 2
  exit
  local-attachment-circuit "CE-2" endpoint "X" create
    eth-tag 2
  exit
  remote-attachment-circuit "ICB-2" endpoint "X" create
    eth-tag 1
  exit
  mpls bgp 1
    auto-bind-tunnel
      resolution any
      exit
    no shutdown
    exit
  exit
sap lag-1:1 endpoint X create
exit
sap 1/1/1:1 endpoint Y create
exit
```

In Figure 178: EVPN-VPWS endpoints example 1, PE2 is configured with the following Epipe services:

```
bgp-evpn
  evi 350
  local-attachment-circuit "CE-1" create
    eth-tag 1
  exit
```

```
  remote-attachment-circuit "ICB-1" create
    eth-tag 2
  exit
// implicit endpoint "Y"
  mpls bgp 1
    auto-bind-tunnel
  resolution any
  exit
no shutdown
exit
  exit
sap lag-1:1 create
exit
// implicit endpoint "X"
```

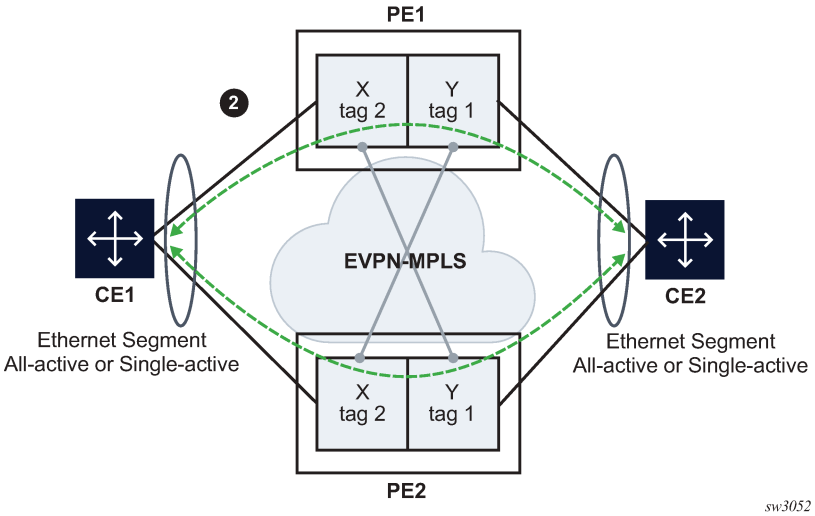
In this example, if we assume multihoming on CE1, the following applies:

- PE1 advertises two AD per-EVI routes, for tags 1 and 2, respectively. PE2 advertises only the route for tag 1.
  - AD per-EVI routes for tag 1 are advertised based on CE1 SAPs' states
  - AD per-EVI route for tag 2 is advertised based on CE2 SAP state
- PE1 creates endpoint X with sap lag-1:1 and ES-destination to tag 1 in PE2
- PE2 creates the usual destination to tag 2 in PE1
- In case of all-active MH:
  - traffic from CE1 to PE1 is forwarded to CE2 directly
  - traffic from CE1 to PE2 is forwarded to PE1 with the label that identifies CE2's SAP
  - traffic from CE2 is forwarded to CE1 directly because CE1's SAP is the endpoint Tx; in case of failure on CE1's SAP, PE1 changes the Tx object to the ES-destination to PE2
- In case of single-active MH, traffic flows in the same way, except that a non-DF SAP is operationally down and therefore cannot be an endpoint Tx object.

6.3.5.3.2 Example 2

The following figure shows an example of EVPN-VPWS endpoints.

Figure 179: EVPN-VPWS endpoints example 2



In Figure 179: EVPN-VPWS endpoints example 2, PE1 is configured with the following Epipe services.

```
endpoint X create
exit
endpoint Y create
exit
bgp-evpn
  evi 350
  local-attachment-circuit "CE-1" endpoint "Y" create
    eth-tag 1
  exit
  remote-attachment-circuit "ICB-1" endpoint "Y" create
    eth-tag 2
  exit
  local-attachment-circuit "CE-2" endpoint "X" create
    eth-tag 2
  exit
  remote-attachment-circuit "ICB-2" endpoint "X" create
    eth-tag 1
  exit
  mpls bgp 1
    auto-bind-tunnel
      resolution any
      exit
    no shutdown
    exit
  exit
sap lag-1:1 endpoint X create
exit
sap lag-2:1 endpoint Y create
exit
```

In Figure 179: EVPN-VPWS endpoints example 2, PE2 is configured with the following Epipe services.

```
endpoint X create
exit
endpoint Y create
exit
bgp-evpn
```

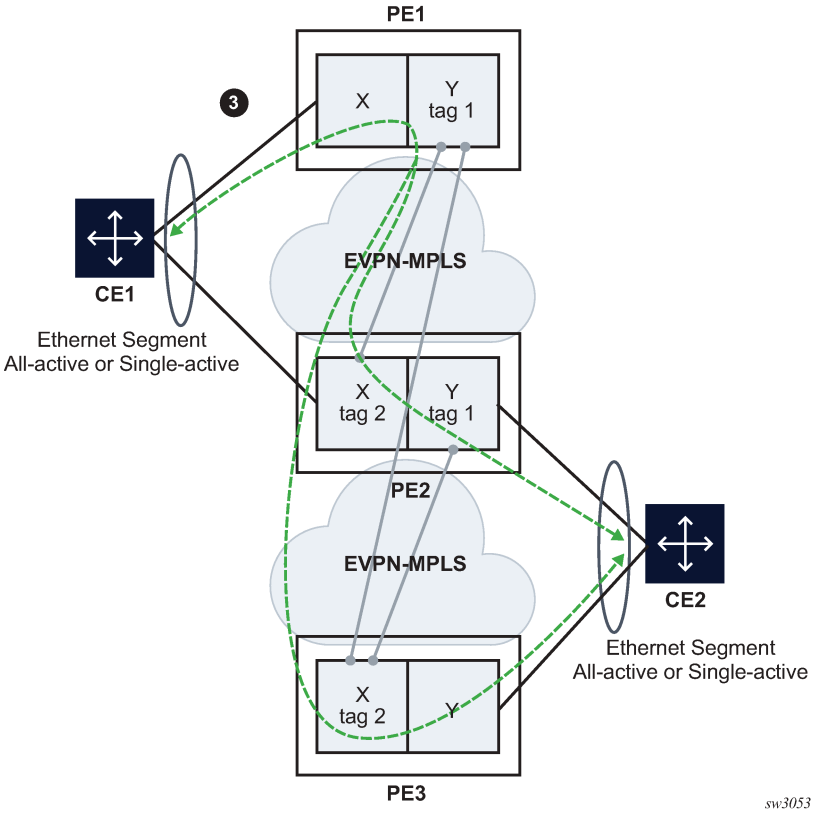
```
  evi 350
  local-attachment-circuit "CE-1" endpoint "Y" create
    eth-tag 1
  exit
  remote-attachment-circuit "ICB-1" endpoint "Y" create
    eth-tag 2
  exit
  local-attachment-circuit "CE-2" endpoint "X" create
    eth-tag 2
  exit
  remote-attachment-circuit "ICB-2" endpoint "X" create
    eth-tag 1
  exit
  mpls bgp 1
    auto-bind-tunnel
      resolution any
      exit
    no shutdown
    exit
  exit
sap lag-1:1 endpoint X create
exit
sap lag-2:1 endpoint Y create
exit
```

This example is similar to the Figure 178: EVPN-VPWS endpoints example 1 example, except that the two PEs are multihomed to both CEs. In Figure 178: EVPN-VPWS endpoints example 1, if CE2 goes down, then, no traffic exists between PEs because the two PEs lose all the objects in the endpoint connected to CE2. Traffic that arrives on EVPN is only forwarded to a SAP on a different endpoint.

6.3.5.3.3 Example 3

The following figure shows an example of EVPN-VPWS endpoints.

Figure 180: EVPN-VPWS endpoints example 3



In Figure 180: EVPN-VPWS endpoints example 3, PE1 is configured with the following Epipe services.

```
bgp-evpn
  evi 350
  local-attachment-circuit "CE-1"
    eth-tag 1
  exit
  remote-attachment-circuit "ICB-1"
    eth-tag 2
  exit
// implicit endpoint "Y"
  mpls bgp 1
    auto-bind-tunnel
      resolution any
      exit
    no shutdown
    exit
  exit
sap lag-1:1 create
// implicit endpoint "X"
exit
```

In Figure 180: EVPN-VPWS endpoints example 3, PE2 is configured with the following Epipe services.

```
endpoint X create
exit
endpoint Y create
```

```
exit
bgp-evpn
  evi 350
  local-attachment-circuit "CE-1" endpoint "Y"
    eth-tag 1
  exit
  remote-attachment-circuit "ICB-1" endpoint "Y"
    eth-tag 2
  exit
  local-attachment-circuit "CE-2" endpoint "X"
    eth-tag 2
  exit
  remote-attachment-circuit "ICB-2" endpoint "X"
    eth-tag 1
  exit
  mpls bgp 1
    auto-bind-tunnel
      resolution any
      exit
    no shutdown
    exit
  exit
sap lag-1:1 endpoint X create
exit
sap lag-2:1 endpoint Y create
exit
```

In Figure 180: EVPN-VPWS endpoints example 3, PE3 is configured with the following Epipe services.

```
bgp-evpn
  evi 350
  local-attachment-circuit "CE-2"
    eth-tag 2
  exit
  remote-attachment-circuit "ICB-2"
    eth-tag 1
  exit
// implicit endpoint "X"
  mpls bgp 1
    auto-bind-tunnel
      resolution any
      exit
    no shutdown
    exit
  exit
sap lag-1:1 create
// implicit endpoint "Y"
exit
```

This example is similar to the Figure 179: EVPN-VPWS endpoints example 2 example, except that a third node is added. Nodes PE1 and PE3 have implicit endpoints. Only node PE2 requires the configuration of endpoints.

6.3.6 EVPN for MPLS tunnels in routed VPLS services

EVPN-MPLS and IP-prefix advertisement (enabled by the **ip-route-advertisement** command) are fully supported in routed VPLS services and provide the same feature-set as EVPN-VXLAN. The following capabilities are supported in a service where **bgp-evpn mpls** is enabled:

- R-VPLS with VRRP support on the VPRN or IES interfaces

- R-VPLS support including **ip-route-advertisement** with regular interfaces

  This includes the advertisement and process of ip-prefix routes defined in IETF Draft *draft-ietf-bess-evpn-prefix-advertisement* with the appropriate encoding for EVPN-MPLS.
- R-VPLS support including **ip-route-advertisement** with **evpn-tunnel** interfaces
- R-VPLS with IPv6 support on the VPRN or IES IP interface

IES interfaces do not support either **ip-route-advertisement** or **evpn-tunnel**.

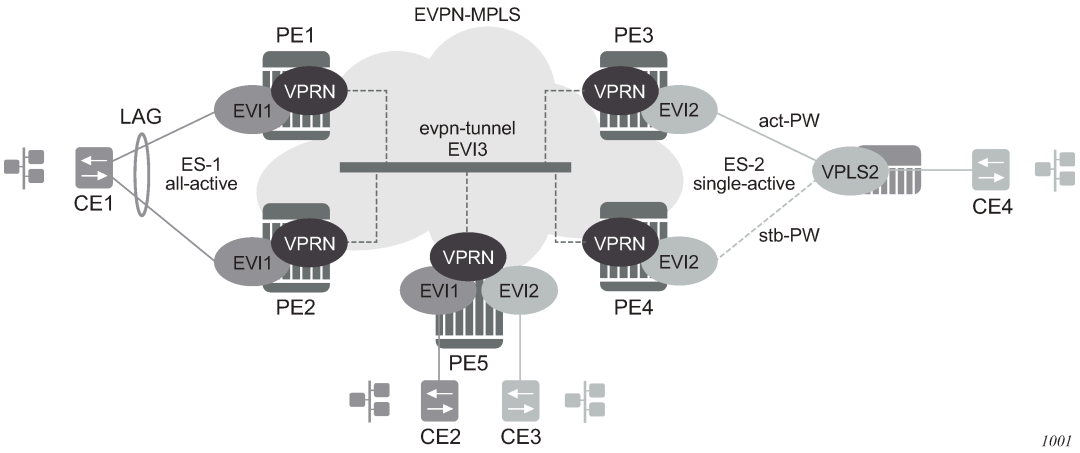
6.3.6.1 EVPN-MPLS multihoming and passive VRRP

SAP and spoke SDP based ESs are supported on R-VPLS services where **bgp-evpn mpls** is enabled.

Figure 181: EVPN-MPLS multihoming in R-VPLS services shows an example of EVPN-MPLS multihoming in R-VPLS services, with the following assumptions:

- There are two subnets for a specific customer (for example, EVI1 and EVI2 in Figure 181: EVPN-MPLS multihoming in R-VPLS services), and a VPRN is instantiated in all the PEs for efficient inter-subnet forwarding.
- A "backhaul" R-VPLS with **evpn-tunnel** mode enabled is used in the core to interconnect all the VPRNs. EVPN IP-prefix routes are used to exchange the prefixes corresponding to the two subnets.
- An all-active ES is configured for EVI1 on PE1 and PE2.
- A single-active ES is configured for EVI2 on PE3 and PE4.

Figure 181: EVPN-MPLS multihoming in R-VPLS services



In the example in Figure 181: EVPN-MPLS multihoming in R-VPLS services, the hosts connected to CE1 and CE4 could use regular VRRP for default gateway redundancy; however, this may not be the most efficient way to provide upstream routing.

For example, if PE1 and PE2 are using regular VRRP, the upstream traffic from CE1 may be hashed to the backup IRB VRRP interface, instead of being hashed to the active interface. The same thing may occur for single-active multihoming and regular VRRP for PE3 and PE4. The traffic from CE4 is sent to PE3, while PE4 may be the active VRRP router. In that case, PE3 has to send the traffic to PE4, instead of route it directly.

In both cases, unnecessary bandwidth between the PEs is used to get to the active IRB interface. In addition, VRRP scaling is limited if aggressive keepalive timers are used.

Because of these issues, passive VRRP is recommended as the best method when EVPN-MPLS multihoming is used in combination with R-VPLS redundant interfaces.

Passive VRRP is a VRRP setting in which the transmission and reception of keepalive messages is completely suppressed, and therefore the VPRN interface always behaves as the active router. Passive VRRP is enabled by adding the **passive** keyword to the VRRP instance at creation, as shown in the following examples:

- **config service vprn 1 interface int-1 vrrp 1 passive**
- **config service vprn 1 interface int-1 ipv6 vrrp 1 passive**

For example, if PE1, PE2, and PE5 in Figure 181: EVPN-MPLS multihoming in R-VPLS services use passive VRRP, even if each individual R-VPLS interface has a different MAC/IP address, because they share the same VRRP instance 1 and the same backup IP, the three PEs own the same virtual MAC and virtual IP address (for example, 00-00-5E-00-00-01 and 10.0.0.254). The virtual MAC is auto-derived from 00-00-5E-00-00-VRID per RFC 3768. The following is the expected behavior when passive VRRP is used in this example:

- All R-VPLS IRB interfaces for EVI1 have their own physical MAC/IP address; they also own the same default gateway virtual MAC and IP address.
- All EVI1 hosts have a unique configured default gateway; for example, 10.0.0.254.
- When CE1 or CE2 send upstream traffic to a remote subnet, the packets are routed by the closest PE because the virtual MAC is always local to the PE.

  For example, the packets from CE1 hashed to PE1 are routed at PE1. The packets from CE1 hashed to PE2 are routed directly at PE2.
- Downstream packets (for example, packets from CE3 to CE1), are routed directly by the PE to CE1, regardless of the PE to which PE5 routed the packets.

  For example, the packets from CE3 sent to PE1 are routed at PE1. The packets from CE3 sent to PE2 are routed at PE2.
- In case of ES failure in one of the PEs, the traffic is forwarded by the available PE.

  For example, if the packets routed by PE5 arrive at PE1 and the link to CE1 is down, then PE1 sends the packets to PE2. PE2 forwards the packets to CE1 even if the MAC source address of the packets matches PE2's virtual MAC address. Virtual MACs bypass the R-VPLS interface MAC protection.

The following list summarizes the advantages of using passive VRRP mode versus regular VRRP for EVPN-MPLS multihoming in R-VPLS services:

- Passive VRRP does not require multiple VRRP instances to achieve default gateway load-balancing. Only one instance per R-VPLS, therefore only one default gateway, is needed for all the hosts.
- The convergence time for link/node failures is not impacted by the VRRP convergence, as all the nodes in the VRRP instance are acting as active routers.
- Passive VRRP scales better than VRRP, as it does not use keepalive or BFD messages to detect failures and allow the backup to take over.

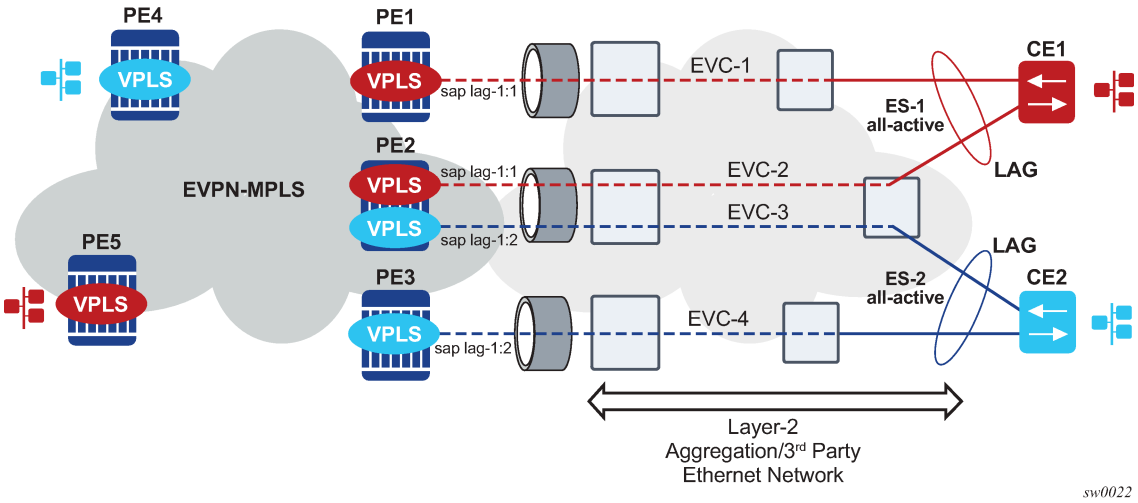
6.3.7 Virtual Ethernet segments

SR OS supports virtual Ethernet Segments (vES) for EVPN multihoming in accordance with *draft-ietf-bess-evpn-virtual-eth-segment*.

Regular ESs can only be associated with ports, LAGs, and SDPs, which satisfies the redundancy requirements for CEs that are directly connected to the ES PEs by a port, LAG, or SDP. However, this implementation does not work when an aggregation network exists between the CEs and the ES PEs, which requires different ESs to be defined for the port or LAG of the SDP.

Figure 182: All-active multihoming on vES shows an example of how CE1 and CE2 use all-active multihoming to the EVPN-MPLS network despite the third-party Ethernet aggregation network to which they are connected.

Figure 182: All-active multihoming on vES



The ES association can be made in a more granular way by creating a vES. A vES can be associated with the following:

- qtag-ranges on dot1q ports or LAGs
- S-tag-ranges on qinq ports or LAGs
- C-tag-ranges per s-tag on qinq ports or LAGs
- VC-ID ranges on SDPs

The following CLI examples show the vES configuration options:

```
config>service>system>bgp-evpn#
...
ethernet-segment vES-1 virtual create
 lag 1
 dot1q
   qtag-range 100 to 200
...
ethernet-segment vES-2 virtual create
 port 1/1/1
 qinq
   s-tag 1 c-tag-range 100 to 200
   s-tag-range 2 to 10
...
```

```
ethernet-segment vES-3 virtual create
 sdp 1
 vc-id-range 1000 to 2000
...
```

Where:

- The **virtual** keyword creates an ES as defined in *draft-sajassi-bess-evpn-virtual-eth-segment*. The configuration of the dot1q or qinq nodes is allowed only when the ES is created as **virtual**.
- On the vES, the user must first create a port, LAG, or SDP before configuring a VLAN or VC-ID association. When added, the port/LAG type and encap-type is checked as follows:
  - If the encap-type is dot1q, only the dot1q context configuration is allowed; the qinq context cannot be configured.
  - If the encap-type is qinq, only the qinq node is allowed; the dot1q context cannot be configured.
  - A dot1q, qinq, or vc-id range is required for the vES to become operationally active.
- The **dot1q qtag-range** <**qtag1**> [**to qtag1**] command determines which VIDs are associated with the vES on a specific dot1q port or LAG. The group of SAPs that match the configured port/LAG and VIDs is part of the vES.
- The **qinq s-tag-range** <**qtag1**> [**to qtag1**] command determines which outer VIDs are associated with the vES on the qinq port or LAG.
- The **qinq s-tag** <**qtag1**> **c-tag-range to** <**qtag2**> [<**qtag2**>] command determines which inner c-tags per s-tag is associated with the vES on the qinq port or LAG.
- The **vc-id range** <**vcid**> [**to vc-id**] command determines which VC ids are associated with the vES on the configured SDP.

Although qtag values 0, * and 1 to 4094 are allowed, the following considerations must be taken in to account when configuring a dot1q or qinq vES:

- Up to 8 dot1q or qinq ranges may be configured in the same vES.
- When configuring a qinq vES, a qtag included in a s-tag-range cannot be included in the s-tag qtag of the **s-tag qtag1 c-tag-range qtag2** [**to qtag2**] command. For example, the following combination is not supported in the same vES:

  ```
  s-tag-range 350 to 500
  s-tag 500 c-tag-range 100 to 200
  ```

  The following example shows a supported combination:

  ```
  *A:PE75>config>service>system>bgp-evpn>eth-seg>qinq# info
  ----------------------------------------------
                  s-tag-range 100 to 200
                  s-tag-range 300 to 400
                  s-tag 500 c-tag-range 100 to 200
                  s-tag 600 c-tag-range 100 to 200
                  s-tag 600 c-tag-range 150 to 200
  ```

- vES associations that contain qtags <0, *, null> are special and treated as follows:
  - When a special qtag value is configured in the **from** value of the range, the **to** value must be the same.
  - qtag values <0, *> are only supported for the **qtag-range** and **c-tag-range**; they are not supported in the **s-tag-range**.

– The qtag "null" value is only supported in the **c-tag-range** if the **s-tag** is configured as "*".

Table 21: Examples of supported qtag values lists examples of the supported qtag values between 1 to 4094.

Table 21: Examples of supported qtag values

vES configuration for port 1/1/1	SAP association
dot1q qtag-range 100	1/1/1:100
dot1q qtag-range 100 to 102	1/1/1:100, 1/1/1:101, 1/1/1:102
qinq s-tag 100 c-tag-range 200	1/1/1:100.200
qinq s-tag-range 100	All the SAPs 1/1/1:100.**x** where: **x** is a value between 1 to 4094, 0, *
qinq s-tag-range 100 to 102	All SAPs 1/1/1:100.x, 1/1/1:101.x, 1/1/1:102.**x** where: **x** is a value between 1 to 4094, 0, *

Table 22: Examples of supported combinations lists all the supported combinations that include qtag values <0, *, null>. Any other combination of these special values is not supported.

Table 22: Examples of supported combinations

vES configuration for port 1/1/1	SAP association
dot1q qtag-range 0	1/1/1:0
dot1q qtag-range *	1/1/1:*
qinq s-tag 0 c-tag-range *	1/1/1:0.*
qinq s-tag * c-tag-range *	1/1/1:*.*
qinq s-tag * c-tag-range null	1/1/1:*.null
qinq s-tag x c-tag-range 0	1/1/1:**x**.0 where: **x** is a value between 1 to 4094
qinq s-tag x c-tag-range *	1/1/1:**x**.* where: **x** is a value between 1 to 4094

On vESs, the single-active and all-active modes are supported for EVPN-MPLS VPLS, Epipe, and PBB-EVPN services. Single-active multihoming is supported on port and SDP-based vESs, and all-active multihoming is only supported on LAG-based vESs.

The following considerations apply if the vES is used with PBB-EVPN services:

- B-MAC allocation procedures are the same as the regular ES procedures.

Note: Two all-active vESs must use different ES B-MACs, even if they are defined in the same LAG.

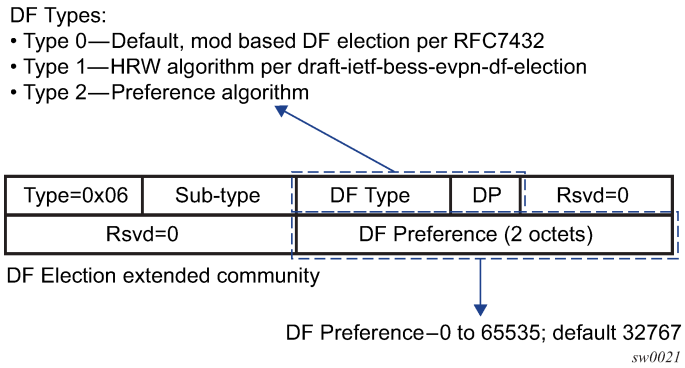
- The vES implements C-MAC flush procedures described in RFC 7623. Optionally, the ISID-based C-MAC flush can be used for cases where the single-active vES does not use ES B-MAC allocation.

6.3.8 Preference-based and non-revertive DF election

In addition to the ES service-carving modes **auto** and **off**, the **manual** mode also supports the preference-based algorithm with the **non-revertive** option, as described in *draft-rabadan-bess-evpn-pref-df*.

When ES is configured to use the preference-based algorithm, the ES route is advertised with the Designated Forwarder (DF) election extended community (sub-type 0x06). Figure 183: DF election extended community shows the DF election extended community.

Figure 183: DF election extended community



In the extended community, a DF type 2 preference algorithm is advertised with a 2-byte preference value (32767 by default) if the preference-based **manual** mode is configured. The Don't Preempt Me (DP) bit is set if the **non-revertive** option is enabled.

The following CLI excerpt shows the relevant commands to enable the preference-based DF election on a specific ES (regular or virtual):

```
config>service>system>bgp-evpn>ethernet-segment#
...
service-carving mode {manual|auto|off}
  service-carving manual
    [no] preference [create] [non-revertive]
      value <value>
    exit
  [no] evi <evi> [to <evi>]
  [no] isid <isid> [to <isid>]
# value 0..65535; default 32767
...
```

Where:

- The preference value can be changed on an active ES without shutting down the ES, and therefore, a new DF can be forced for maintenance or other reasons.
- The **service-carving** mode must be changed to **manual** mode to create the **preference** context.

- The **preference** command is supported on regular or virtual ES, regardless of the multihoming mode (single-active or all-active) or the service type (VPLS, I-VPLS, or Epipe).
- By default, the highest-preference PE in the ES becomes the DF for an EVI or ISID, using the DP bit as the tiebreaker first (DP=1 wins over DP=0) and the lowest PE-IP as the last-resort tiebreaker. All the explicitly configured EVI or ISID ranges select the lowest preference PE as the DF (whereas the non-configured EVI or ISID values select the highest preference PE).

  This selection is displayed as `Cfg Range Type: lowest-pref` in the following **show** command example.

```
*A:PE-2# show service system bgp-evpn ethernet-segment name "vES-23"
===============================================================================
Service Ethernet Segment
===============================================================================
Name                    : vES-23
Eth Seg Type            : Virtual
Admin State             : Enabled            Oper State         : Up
ESI                     : 01:23:23:23:23:23:23:23:23:23
Multi-homing            : allActive          Oper Multi-homing  : allActive
ES SHG Label            : 262141
Source BMAC LSB         : 00-23
ES BMac Tbl Size        : 8                  ES BMac Entries    : 0
Lag Id                  : 1
ES Activation Timer     : 3 secs (default)
Svc Carving             : manual             Oper Svc Carving   : manual
Cfg Range Type          : lowest-pref
-------------------------------------------------------------------------------
DF Pref Election Information
-------------------------------------------------------------------------------
Preference     Preference     Last Admin Change        Oper Pref      Do No
Mode           Value                                   Value          Preempt
-------------------------------------------------------------------------------
non-revertive  100            12/21/2016 15:16:38      100            Enabled
-------------------------------------------------------------------------------
EVI Ranges: <none>
ISID Ranges: <none>
===============================================================================
```

- The EVI and ISID ranges configured on the service-carving context are not required to be consistent with any ranges configured for vESs.
- If the **non-revertive** option is configured, when the former DF comes back up after a failure and checks existing ES routes, it advertises an operational preference and DP bit, which does not cause a DF switchover for the ES EVI/ISID values.
- The **non-revertive** option prevents an ES DF switchover in the following events:
  - ES port recovery after a failure
  - node recovery after a reboot or power-up event
- The **non-revertive** option does not prevent an ES DF switchover when the ES is administratively enabled or when any other event attempts to recover the ES when the ES routes from the ES peers are not present yet. An example of this is when the user executes the **clear card** command on all of the line cards in the router. When the ES is brought up, the BGP session is still recovering and therefore, there are no remote ES routes from the ES peers. Use the following command to prevent this situation for reboots or node power-up events (but not for any other events).

```
configure redundancy bgp-evpn-multi-homing boot-timer
```

The following configuration example shows the use of the preference-based algorithm and non-revertive option in an ES defined in PE1 and PE2.

```
*A:PE-1>config>service>system>bgp-evpn# info
----------------------------------------------
 ethernet-segment "ES1" create
   esi 01:00:00:00:00:12:00:00:00:01
   service-carving manual
       preference non-revertive create
         value 10000
       exit
       evi 2001 to 4000
   exit
   multi-homing single-active
   port 1/1/1
   no shutdown

/* example of vpls 1 - similar config exists for evis 2-4000 */
*A:PE-1>config>service>vpls# info
----------------------------------------------
 vxlan vni 1 create
 exit
 bgp-evpn
   evi 1
   mpls bgp 1
     ecmp 2
     auto-bind-tunnel
       resolution any
     exit
 sap 1/1/1:1 create
 no shutdown
----------------------------------------------
*A:PE-2>config>service>system>bgp-evpn# info
----------------------------------------------
 ethernet-segment "ES1" create
   esi 01:00:00:00:00:12:00:00:00:01
   service-carving manual
       preference non-revertive create
         value 5000
       exit
       evi 2001 to 4000
   exit
   multi-homing single-active
   port 1/1/1
   no shutdown

*A:PE-2>config>service>vpls# info
----------------------------------------------
 vxlan vni 1 create
 exit
 bgp-evpn
   evi 1
   mpls bgp 1
     ecmp 2
     auto-bind-tunnel
       resolution any
     exit
 sap 1/1/1:1 create
 no shutdown
----------------------------------------------
```

Based on the configuration in the preceding example, the PE behavior is as follows:

1. Assuming the ES is **no shutdown** on both PE1 and PE2, the PEs exchange ES routes, including the [Pref, DP-bit] in the DF election extended community.
2. For EVIs 1 to 2000, PE2 is immediately promoted to NDF, whereas PE1 becomes the DF, and (following the **es-activation-timer**) brings up its SAP in EVIs 1 to 2000.

   For EVIs 2001 to 4000, the result is the opposite and PE2 becomes the DF.
3. If port 1/1/1 on PE1 goes down, PE1 withdraws its ES route and PE2 becomes the DF for EVIs 1 to 2000.
4. When port 1/1/1 on PE1 comes back up, PE1 compares its ES1 preference with the preferences in the remote PEs in ES1. PE1 advertises the ES route with an "in-use operational" Pref = 5000 and DP=0. Because PE2's Pref is the same as PE1's operational value, but PE2's DP=1, PE2 continues to be the DF for EVIs 1 to 4000.

   Note: The DP bit is the tiebreaker in case of equal Pref and regardless of the choice of highest or lowest preference algorithm.
5. PE1's "in-use" Pref and DP continue to be [5000,0] until one of the following conditions is true:
   - PE2 withdraws its ES route, in which case PE1 re-advertises its admin Pref and DP [10000,DP=1]
   - The user changes PE1's Pref configuration

6.3.9 EVPN-MPLS routed VPLS multicast routing support

IPv4 multicast routing is supported in an EVPN-MPLS VPRN routed VPLS service through its IP interface, when the source of the multicast stream is on one side of its IP interface and the receivers are on either side of the IP interface. For example, the source for multicast stream G1 could be on the IP side sending to receivers on both other regular IP interfaces and the VPLS of the routed VPLS service, while the source for group G2 could be on the VPLS side sending to receivers on both the VPLS and IP side of the routed VPLS service. See IPv4 and IPv6 multicast routing support for more details.

6.3.10 IGMP snooping in EVPN-MPLS and PBB EVPN services

IGMP snooping is supported in EVPN-MPLS VPLS and PBB-EVPN I-VPLS (where BGP EVPN is running in the associated B-VPLS service) services. It is also supported in EVPN-MPLS VPRN and IES R-VPLS services. It is required in scenarios where the operator does not want to flood all of the IP multicast traffic to the access nodes or CEs, and only wants to deliver IP multicast traffic for which IGMP reports have been received.

The following points apply when IGMP snooping is configured in EVPN-MPLS VPLS or PBB-EVPN I-VPLS services:

- IGMP snooping is enabled using the **configure service vpls igmp-snooping no shutdown** command.
- Queries and reports received on SAP or SDP bindings are snooped and properly handled; they are sent to SAP or SDP bindings as expected.
- Queries and reports on EVPN-MPLS or PBB-EVPN B-VPLS destinations are handled as follows.
  - If received from SAP or SDP bindings, the queries and reports are sent to all EVPN-MPLS and PBB-EVPN B-VPLS destinations, regardless of whether the service is using an ingress replication or mLDP provider tunnel.

- If received on an EVPN-MPLS or PBB-EVPN B-VPLS destination, the queries and reports are processed and propagated to access SAP or SDP bindings, regardless of whether the service is using an ingress replication or mLDP provider tunnel.
- EVPN-MPLS and PBB-EVPN B-VPLS destinations are is treated as a single IGMP snooping interface and is always added as an **mrouter**.
- The debug trace output displays one copy of messages being sent to all EVPN-MPLS and PBB-EVPN B-VPLS destinations (the trace does not show a copy for each destination) and displays messages received from all EVPN-MPLS and PBB-EVPN B-VPLS destinations as coming from a single EVPN-MPLS interface.

Note: When IGMP snooping is enabled with P2MP LSPs, at least one EVPN-MPLS multicast destination must be established to enable the processing of IGMP messages by the system. The use of P2MP LSPs is not supported when sending IPv4 multicast into an EVPN-MPLS R-VPLS service from its IP interface.

In the following show command output, the EVPN-MPLS destinations are shown as part of the MFIB (when **igmp-snooping** is in a **no shutdown** state), and the EVPN-MPLS logical interface is shown as an **mrouter**.

```
*A:PE-2# show service id 2000 mfib
===============================================================================
Multicast FIB, Service 2000
===============================================================================
Source Address  Group Address         SAP or SDP Id                   Svc Id   Fwd
                                                                          Blk
-------------------------------------------------------------------------------
*               *                     eMpls:192.0.2.3:262132       Local    Fwd
                                      eMpls:192.0.2.4:262136       Local    Fwd
                                      eMpls:192.0.2.5:262131       Local    Fwd
-------------------------------------------------------------------------------
Number of entries: 1
===============================================================================
*A:PE-2# show service id 2000 igmp-snooping base
===============================================================================
IGMP Snooping Base info for service 2000
===============================================================================
Admin State : Up
Querier     : 10.0.0.3 on evpn-mpls
-------------------------------------------------------------------------------
SAP or SDP                  Oper MRtr Pim  Send Max   Max  Max   MVR       Num
Id                        Stat Port Port Qrys Grps  Srcs Grp   From-VPLS Grps
                                                        Srcs
-------------------------------------------------------------------------------
sap:1/1/1:2000            Up   No   No   No   None  None None  Local     0
evpn-mpls                 Up   Yes  N/A  N/A  N/A   N/A  N/A   N/A       N/A
===============================================================================


*A:PE-4# show service id 2000 igmp-snooping mrouters

===============================================================================
IGMP Snooping Multicast Routers for service 2000
===============================================================================
MRouter          SAP or SDP Id                 Up Time        Expires   Version
-------------------------------------------------------------------------------
10.0.0.3         evpn-mpls                  0d 00:38:49    175s      3
-------------------------------------------------------------------------------
Number of mrouters: 1
```

```
===============================================================================
```

The equivalent output for PBB-EVPN services is similar to the output above for EVPN-MPLS services, with the exception that the EVPN destinations are named "b-EVPN-MPLS".

6.3.10.1 Data-driven IGMP snooping synchronization with EVPN multihoming

When single-active multihoming is used, the IGMP snooping state is learned on the active multihoming object. If a failover occurs, the system with the newly active multihoming object must wait for IGMP messages to be received to instantiate the IGMP snooping state after the ES activation timer expires; this could result in an increased outage.

The outage can be reduced by using MCS synchronization, which is supported for IGMP snooping in both EVPN-MPLS and PBB-EVPN services (see Multi-chassis synchronization for Layer 2 snooping states). However, MCS only supports synchronization between two PEs, whereas EVPN multihoming is supported between a maximum of four PEs. Also, IGMP snooping state can be synchronized only on a SAP.

An increased outage would also occur when using all-active EVPN multihoming. The IGMP snooping state on an ES LAG SAP or virtual ES to the attached CE must be synchronized between all the ES PEs, as the LAG link used by the DF PE may not be the same as that used by the attached CE. MCS synchronization is not applicable to all-active multihoming as MCS only supports active/standby synchronization.

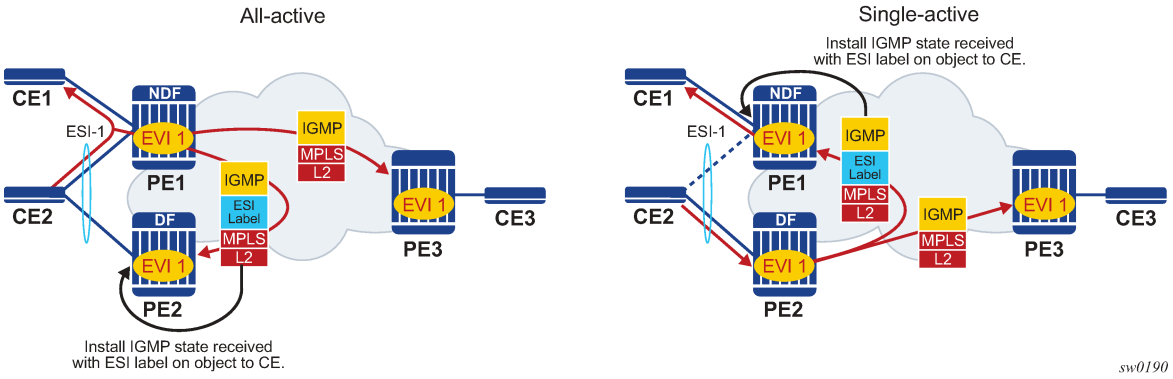
To eliminate any additional outage on a multihoming failover, IGMP snooping messages can be synchronized between the PEs on an ES using data-driven IGMP snooping state synchronization, which is supported in EVPN-MPLS services, PBB-EVPN services, EVPN-MPLS VPRN and IES R-VPLS services. The IGMP messages received on an ES SAP or spoke SDP are sent to the peer ES PEs with an ESI label (for EVPN-MPLS) or ES B-MAC (for PBB-EVPN) and these are used to synchronize the IGMP snooping state on the ES SAP or spoke SDP on the receiving PE.

Data-driven IGMP snooping state synchronization is supported for both all-active multihoming and single-active with an ESI label multihoming in EVPN-MPLS, EVPN-MPLS VPRN and IES R-VPLS services, and for all-active multihoming in PBB-EVPN services. All PEs participating in a multihomed ES must be running an SR OS version supporting this capability. PBB-EVPN with IGMP snooping using single-active multihoming is not supported.

Data-driven IGMP snooping state synchronization is also supported with P2MP mLDP LSPs in both EVPN-MPLS and PBB-EVPN services. When P2MP mLDP LSPs are used in EVPN-MPLS services, all PEs (including the PEs not connected to a multihomed ES) in the EVPN-MPLS service must be running an SR OS version supporting this capability with IGMP snooping enabled and all network interfaces must be configured on FP3 or higher-based line cards.

Figure 184: Data-driven IGMP snooping synchronization with EVPN multihoming shows the processing of an IGMP message for EVPN-MPLS. In PBB-EVPN services, the ES B-MAC is used instead of the ESI label to synchronize the state.

Figure 184: Data-driven IGMP snooping synchronization with EVPN multihoming



Data-driven synchronization is enabled by default when IGMP snooping is enabled within an EVPN-MPLS service using all-active multihoming or single-active with an ESI label multihoming, or in a PBB-EVPN service using all-active multihoming. If IGMP snooping MCS synchronization is enabled on an EVPN-MPLS or PBB-EVPN (I-VPLS) multihoming SAP then MCS synchronization takes precedence over the data-driven synchronization and the MCS information is used. Mixing data-driven and MCS IGMP synchronization within the same ES is not supported.

When using EVPN-MPLS, the ES should be configured as **non-revertive** to avoid an outage when a PE takes over the DF role. The Ethernet A-D per ESI route update is withdrawn when the ES is down which prevents state synchronization to the PE with the ES down, as it does not advertise an ESI label. The lack of state synchronization means that if the ES comes up and that PE becomes DF after the ES activation timer expires, it may not have any IGMP snooping state until the next IGMP messages are received, potentially resulting in an additional outage. Configuring the ES as **non-revertive** can avoid this potential outage. Configuring the ES to be **non-revertive** would also avoid an outage when PBB-EVPN is used, but there is no outage related to the lack of the ESI label as it is not used in PBB-EVPN.

The following steps can be used when enabling IGMP snooping in EVPN-MPLS and PBB-EVPN services:

1. Upgrade SR OS on all ES PEs to a version supporting data-driven IGMP snooping synchronization with EVPN multihoming.
2. Enable IGMP snooping in the required services on all ES PEs. Traffic loss occurs until all ES PEs have IGMP snooping enabled and the first set of join/query messages are processed by the ES PEs.

Note: There is no action required on the non-ES PEs.

If P2MP mLDP LSPs are also configured, the following steps can be used when enabling IGMP snooping in EVPN-MPLS and PBB-EVPN services:

1. Upgrade SR OS on all PEs (both ES and non-ES) to a version supporting data-driven IGMP snooping synchronization with EVPN multihoming.
2. Enable IGMP snooping in EVPN-MPLS and PBB-EVPN services.
   - Perform the following steps for EVPN-MPLS:
     – Enable IGMP snooping on all non-ES PEs. Traffic loss occurs until the first set of join/query messages are processed by the non-ES PEs.
     – Then enable IGMP snooping on all ES PEs. Traffic loss occurs until all PEs have IGMP snooping enabled and the first set of join/query messages are processed by the ES PEs.

- Perform the following steps for PBB-EVPN:
  - Enable IGMP snooping on all ES PEs. Traffic loss occurs until all PEs have IGMP snooping enabled and the first set of join/query messages are processed by the ES PEs.
  - There is no action required on the non-ES PEs.

To aid with troubleshooting, the debug packet output displays the IGMP packets used for the snooping state synchronization. An example of a join sent on ES esi-1 from one ES PE and the same join received on another ES PE follows.

```
6 2017/06/16 18:00:07.819 PDT MINOR: DEBUG #2001 Base IGMP
"IGMP: TX packet on svc 1
  from chaddr 5e:00:00:16:d8:2e
  send towards ES:esi-1
  Port  : evpn-mpls
  SrcIp : 0.0.0.0
  DstIp : 239.0.0.22
  Type  : V3 REPORT
    Num Group Records: 1
        Group Record Type: MODE_IS_EXCL (2), AuxDataLen 0, Num Sources 0
          Group Addr: 239.0.0.1


4 2017/06/16 18:00:07.820 PDT MINOR: DEBUG #2001 Base IGMP
"IGMP: RX packet on svc 1
  from chaddr d8:2e:ff:00:01:41
  received via evpn-mpls on ES:esi-1
  Port  : sap lag-1:1
  SrcIp : 0.0.0.0
  DstIp : 239.0.0.22
  Type  : V3 REPORT
    Num Group Records: 1
        Group Record Type: MODE_IS_EXCL (2), AuxDataLen 0, Num Sources 0
          Group Addr: 239.0.0.1
```

6.3.11 PIM snooping for IPv4 in EVPN-MPLS and PBB-EVPN services

PIM snooping for VPLS allows a VPLS PE router to build multicast states by snooping PIM protocol packets that are sent over the VPLS. The VPLS PE then forwards multicast traffic based on the multicast states. When all receivers in a VPLS are IP multicast routers running PIM, multicast forwarding in the VPLS is efficient when PIM snooping for VPLS is enabled.

PIM snooping for IPv4 is supported in EVPN-MPLS (for VPLS and R-VPLS) and PBB-EVPN I-VPLS (where BGP EVPN is running in the associated B-VPLS service) services. It is enabled using the following command (as IPv4 multicast is enabled by default):

```
configure service vpls <service-id> pim-snooping
```

PIM snooping on SAPs and spoke SDPs operates in the same way as in a plain VPLS service. However, EVPN-MPLS/PBB-EVPN B-VPLS destinations are treated as a single PIM interface, specifically:

- Hellos and join/prune messages from SAPs or SDPs are always sent to all EVPN-MPLS or PBB-EVPN B-VPLS destinations.
- As soon as a hello message is received from one PIM neighbor on an EVPN-MPLS or PBB-EVPN I-VPLS destination, then the single interface representing all EVPN-MPLS or PBB-EVPN I-VPLS destinations has that PIM neighbor.

- The EVPN-MPLS or PBB-EVPN B-VPLS destination split horizon logic ensures that IP multicast traffic and PIM messages received on an EVPN-MPLS or PBB-EVPN B-VPLS destination are not forwarded back to other EVPN-MPLS or PBB-EVPN B-VPLS destinations.
- The debug trace output displays one copy of messages being sent to all EVPN-MPLS or PBB-EVPN B-VPLS destinations (the trace does not show a copy for each destination) and displays messages received from all EVPN-MPLS or PBB-EVPN B-VPLS destinations as coming from a single EVPN-MPLS interface.

PIM snooping for IPv4 is supported in EVPN-MPLS services using P2MP LSPs and PBB-EVPN I-VPLS services with P2MP LSPs in the associated B-VPLS service. When PIM snooping is enabled with P2MP LSPs, at least one EVPN-MPLS multicast destination is required to be established to enable the processing of PIM messages by the system.

Multi-chassis synchronization (MCS) of PIM snooping for IPv4 state is supported for both SAPs and spoke SDPs which can be used with single-active multihoming. Care should be taken when using *.null to define the range for a QinQ virtual ES if the associated SAPs are also being synchronized by MCS, as there is no equivalent MCS sync-tag support to the *.null range.

PBB-EVPN services operate in a similar way to regular PBB services, specifically:

- The multicast flooding between the I-VPLS and the B-VPLS works in a similar way as for PIM snooping for IPv4 with an I-VPLS using a regular B-VPLS. The first PIM join message received over the local B-VPLS from a B-VPLS SAP or SDP or EVPN destination adds all of the B-VPLS SAP or SDP or EVPN components into the related multicast forwarding table associated with that I-VPLS context. The multicast packets are forwarded throughout the B-VPLS on the per ISID single tree.
- When a PIM router is connected to a remote I-VPLS instance over the B-VPLS infrastructure, its location is identified by the B-VPLS SAP, SDP or by the set of all EVPN destinations on which its PIM hellos are received. The location is also identified by the source B-MAC address used in the PBB header for the PIM hello message (this is the B-MAC associated with the B-VPLS instance on the remote PBB PE).

In EVPN-MPLS services, the individual EVPN-MPLS destinations appear in the MFIB but the information for each EVPN-MPLS destination entry is always identical, as shown below:

```
*A:PE# show service id 1 mfib
===============================================================================
Multicast FIB, Service 1
===============================================================================
Source Address  Group Address         Port Id                     Svc Id   Fwd
                                                                        Blk
-------------------------------------------------------------------------------
*               239.252.0.1           sap:1/1/9:1                 Local    Fwd
                                      eMpls:1.1.1.2:262141        Local    Fwd
                                      eMpls:1.1.1.3:262141        Local    Fwd
-------------------------------------------------------------------------------
Number of entries: 1
===============================================================================
*A:PE#
```

Similarly for the PIM neighbors:

```
*A:PE# show service id 1 pim-snooping neighbor
===============================================================================
PIM Snooping Neighbors ipv4
===============================================================================
Port Id                 Nbr DR Prty     Up Time       Expiry Time     Hold Time
  Nbr Address
-------------------------------------------------------------------------------
```

```
SAP:1/1/9:1             1               0d 00:08:17   0d 00:01:29     105
  10.0.0.1
EVPN-MPLS               1               0d 00:27:26   0d 00:01:19     105
  10.0.0.2
EVPN-MPLS               1               0d 00:27:26   0d 00:01:19     105
  10.0.0.3
-------------------------------------------------------------------------------
Neighbors : 3
===============================================================================
*A:PE#
```

A single EVPN-MPLS interface is shown in the outgoing interface, as can be seen in the following output:

```
*A:PE# show service id 1 pim-snooping group detail
===============================================================================
PIM Snooping Source Group ipv4
===============================================================================
Group Address      : 239.252.0.1
Source Address     : *
Up Time            : 0d 00:07:07
Up JP State        : Joined             Up JP Expiry       : 0d 00:00:37
Up JP Rpt          : Not Joined StarG   Up JP Rpt Override : 0d 00:00:00
RPF Neighbor       : 10.0.0.1
Incoming Intf      : SAP:1/1/9:1
Outgoing Intf List : EVPN-MPLS, SAP:1/1/9:1
Forwarded Packets  : 0                  Forwarded Octets   : 0
-------------------------------------------------------------------------------
Groups : 1
===============================================================================
*A:PE#
```

An example of the debug trace output for a join received on an EVPN-MPLS destination is shown below:

```
A:PE1# debug service id 1 pim-snooping packet jp
A:PE1#
32 2016/12/20 14:21:22.68 CET MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: Join/Prune
[000 02:16:02.460] PIM-RX ifId 1071394 ifName EVPN-MPLS 10.0.0.3 -> 224.0.0.13
Length: 34
PIM Version: 2 Msg Type: Join/Prune Checksum: 0xd3eb
Upstream Nbr IP : 10.0.0.1 Resvd: 0x0, Num Groups 1, HoldTime 210
        Group: 239.252.0.1/32 Num Joined Srcs: 1, Num Pruned Srcs: 0
        Joined Srcs:
                10.0.0.1/32 Flag SWR  <*,G>
```

The equivalent output for PBB-EVPN services is similar to that above for EVPN-MPLS services, with the exception that the EVPN destinations are named "b-EVPN-MPLS".

6.3.11.1 Data-driven PIM snooping for IPv4 synchronization with EVPN multihoming

When single-active multihoming is used, PIM snooping for IPv4 state is learned on the active multihoming object. If a failover occurs, the system with the newly active multihoming object must wait for IPv4 PIM messages to be received to instantiate the PIM snooping for IPv4 state after the ES activation timer expires, which could result in an increased outage.

This outage can be reduced by using MCS synchronization, which is supported for PIM snooping for IPv4 in both EVPN-MPLS and PBB-EVPN services (see Multi-chassis synchronization for Layer 2 snooping

states). However, MCS only supports synchronization between two PEs, whereas EVPN multihoming is supported between a maximum of four PEs.

An increased outage would also occur when using all-active EVPN multihoming. The PIM snooping for IPv4 state on an all-active ES LAG SAP or virtual ES to the attached CE must be synchronized between all the ES PEs, as the LAG link used by the DF PE may not be the same as that used by the attached CE. MCS synchronization is not applicable to all-active multihoming as MCS only supports active/standby synchronization.

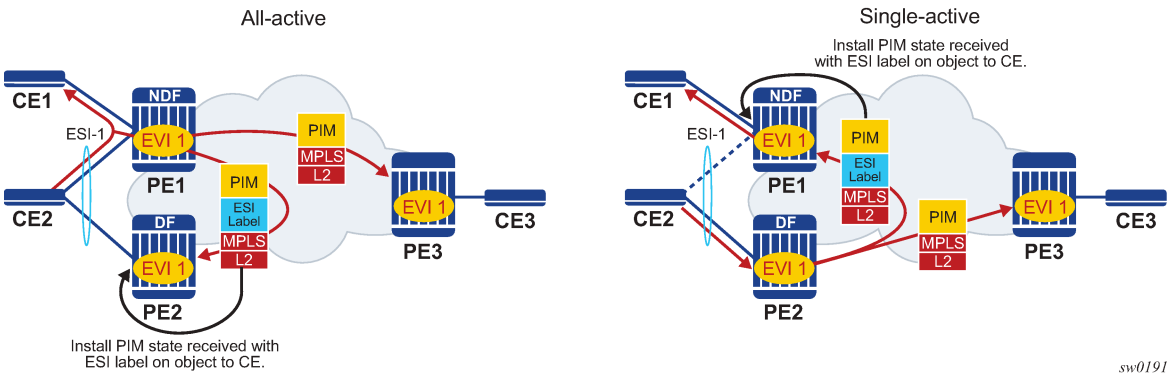
To eliminate any additional outage on a multihoming failover, snooped IPv4 PIM messages should be synchronized between the PEs on an ES using data-driven PIM snooping for IPv4 state synchronization, which is supported in both EVPN-MPLS and PBB-EVPN services. The IPv4 PIM messages received on an ES SAP or spoke SDP are sent to the peer ES PEs with an ESI label (for EVPN-MPLS) or ES B-MAC (for PBB-EVPN) and are used to synchronize the PIM snooping for IPv4 state on the ES SAP or spoke SDP on the receiving PE.

Data-driven PIM snooping state synchronization is supported for all-active multihoming and single-active with an ESI label multihoming in EVPN-MPLS services. All PEs participating in a multihomed ES must be running an SR OS version supporting this capability with PIM snooping for IPv4 enabled. It is also supported with P2MP mLDP LSPs in the EVPN-MPLS services, in which case all PEs (including the PEs not connected to a multihomed ES) must have PIM snooping for IPv4 enabled and all network interfaces must be configured on FP3 or higher-based line cards.

In addition, data-driven PIM snooping state synchronization is supported for all-active multihoming in PBB-EVPN services and with P2MP mLDP LSPs in PBB-EVPN services. All PEs participating in a multihomed ES, and all PEs using PIM proxy mode (including the PEs not connected to a multihomed ES) in the PBB-EVPN service must be running an SR OS version supporting this capability and must have PIM snooping for IPv4 enabled. PBB-EVPN with PIM snooping for IPv4 using single-active multihoming is not supported.

Figure 185: Data-driven PIM snooping for IPv4 synchronization with EVPN multihoming shows the processing of an IPv4 PIM message for EVPN-MPLS. In PBB-EVPN services, the ES B-MAC is used instead of the ESI label to synchronize the state.

Figure 185: Data-driven PIM snooping for IPv4 synchronization with EVPN multihoming



Data-driven synchronization is enabled by default when PIM snooping for IPv4 is enabled within an EVPN-MPLS service using all-active multihoming and single-active with an ESI label multihoming, or in a PBB-EVPN service using all-active multihoming. If PIM snooping for IPv4 MCS synchronization is enabled on an EVPN-MPLS or PBB-EVPN (I-VPLS) multihoming SAP or spoke SDP, then MCS synchronization takes preference over the data-driven synchronization and the MCS information is used. Mixing data-driven and MCS PIM synchronization within the same ES is not supported.

When using EVPN-MPLS, the ES should be configured as **non-revertive** to avoid an outage when a PE takes over the DF role. The Ethernet A-D per ESI route update is withdrawn when the ES is down, which prevents state synchronization to the PE with the ES down as it does not advertise an ESI label. The lack of state synchronization means that if the ES comes up and that PE becomes DF after the ES activation timer expires, it may not have any PIM snooping for IPv4 state until the next PIM messages are received, potentially resulting in an additional outage. Configuring the ES as **non-revertive** can avoid this potential outage. Configuring the ES to be **non-revertive** would also avoid an outage when PBB-EVPN is used, but there is no outage related to the lack of the ESI label as it is not used in PBB-EVPN.

The following steps can be used when enabling PIM snooping for IPv4 (using PIM snooping and PIM proxy modes) in EVPN-MPLS and PBB-EVPN services:

- PIM snooping mode
  1. Upgrade SR OS on all ES PEs to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
  2. Enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by the ES PEs.

     Note: There is no action required on the non-ES PEs.

- PIM proxy mode
  - EVPN-MPLS
    1. Upgrade SR OS on all ES PEs to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
    2. Enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by the ES PEs.

       Note: There is no action required on the non-ES PEs.

  - PBB-EVPN
    1. Upgrade SR OS on all PEs (both ES and non-ES) to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
    2. Enable PIM snooping for IPv4 on all non-ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by each non-ES PE.
    3. Enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by the ES PEs.

If P2MP mLDP LSPs are also configured, the following steps can be used when enabling PIM snooping or IPv4 (using PIM snooping and PIM proxy modes) in EVPN-MPLS and PBB-EVPN services.

- PIM snooping mode
  1. Upgrade SR OS on all PEs (both ES and non-ES) to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
  2. Then enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping enabled and the first set of join/hello messages are processed by the ES PEs.

Note: There is no action required on the non-ES PEs.

- PIM proxy mode
  1. Upgrade SR OS on all PEs (both ES and non-ES) to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
  2. Enable PIM snooping for IPv4 on all non-ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by each non-ES PE.
  3. Enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping enabled and the first set of join/hello messages are processed by the ES PEs.

In the above steps, when PIM snooping for IPv4 is enabled, the traffic loss can be reduced or eliminated by configuring a larger hold-time (up to 300 seconds), during which multicast traffic is flooded.

To aid with troubleshooting, the debug packet output displays the PIM packets used for the snooping state synchronization. An example of a join sent on ES esi-1 from one ES PE and the same join received on another ES PE follows:

```
6 2017/06/16 17:36:37.144 PDT MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: pimVplsFwdJPToEvpn
Forwarding to remote peer on bgp-evpn ethernet-segment esi-1"
7 2017/06/16 17:36:37.144 PDT MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: Join/Prune
[000 00:19:37.040] PIM-TX ifId 1071394 ifName EVPN-MPLS-ES:esi-1 10.0.0.10 -> 22
10.0.0.13 Length: 34
PIM Version: 2 Msg Type: Join/Prune Checksum: 0xd2de
Upstream Nbr IP : 10.0.0.1 Resvd: 0x0, Num Groups 1, HoldTime 210
        Group: 239.0.0.10/32 Num Joined Srcs: 1, Num Pruned Srcs: 0
        Joined Srcs:
                10.0.0.1/32 Flag SWR  <*,G>


4 2017/06/16 17:36:37.144 PDT MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: pimProcessPdu
Received from remote peer on bgp-evpn ethernet-segment esi-1, will be applied on
 lag-1:1
"
5 2017/06/16 17:36:37.144 PDT MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: Join/Prune
[000 00:19:30.740] PIM-RX ifId 1071394 ifName EVPN-MPLS-ES:esi-1 10.0.0.10 -> 22
10.0.0.13 Length: 34
PIM Version: 2 Msg Type: Join/Prune Checksum: 0xd2de
Upstream Nbr IP : 10.0.0.1 Resvd: 0x0, Num Groups 1, HoldTime 210
        Group: 239.0.0.10/32 Num Joined Srcs: 1, Num Pruned Srcs: 0
        Joined Srcs:
                10.0.0.1/32 Flag SWR  <*,G>
```

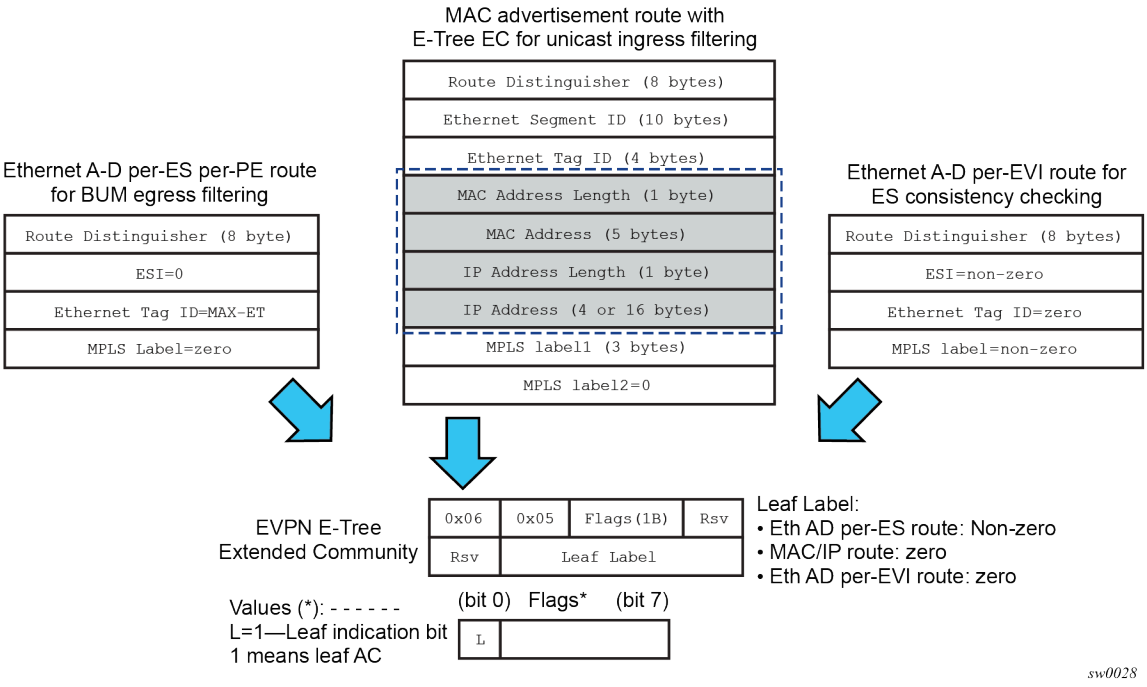
6.3.12 EVPN E-Tree

This section contains information about EVPN E-Tree.

6.3.12.1 BGP EVPN control plane for EVPN E-Tree

BGP EVPN control plane is extended and aligned with IETF RFC 8317 to support EVPN E-Tree services. Figure 186: EVPN E-Tree BGP routes shows the main EVPN extensions for the EVPN E-Tree information model.

Figure 186: EVPN E-Tree BGP routes



The following BGP extensions are implemented for EVPN E-Tree services:

- An EVPN E-Tree extended community (EC) sub-type 0x5 is defined. The following information is included:
  - The lower bit of the Flags field contains the L bit (where L=1 indicates leaf AC).
  - The leaf label contains a 20-bit MPLS label in the high-order 20 bits of the label field. This leaf label is automatically allocated by the system or statically assigned by the **evpn-etree-leaf-label <value>** command.
- The new E-Tree EC is sent with the following routes:
  - AD per-ES per PE route for BUM egress filtering:

    Each EVPN E-Tree capable PE advertises an AD per-ES route with the E-Tree EC, and the following information:

    - Service RD and route-target; if **ad-per-es-route-target evi-rt-set** is configured, then non-zero ESI AD per-ES routes (used for multihoming) are sent per the **evi-rt-set** configuration, but E-Tree zero-ESI routes (used for E-Tree) are sent based on the default **evi-rt** configuration
    - ESI = 0
    - Eth Tag = MAX-ET
    - MPLS label = zero

– AD per-EVI route for root or leaf configuration consistency check as follows:
  - The E-Tree EC is sent with the AD per-EVI routes for a specific ES. In this case, no validation is performed by the implementation, and the leaf indication is only used for troubleshooting on the remote PEs.
  - The MPLS label value is zero.
  - All attachment circuits (ACs) in each EVI for a specific ES must be configured as either a root or leaf AC, but not a combination. In case of a configuration error, for example, where the AC in PE1 is configured as root and in PE2 as leaf AC, the remote PE3 receives the AD per-EVI routes with inconsistent leaf indication. However, the unicast filtering remains unaffected and is still handled by the FDB lookup information.
– MAC/IP routes for known unicast ingress filtering as follows:
  - An egress PE sends all MAC/IP routes learned over a leaf AC SAP or spoke SDP with this E-Tree EC indicating that the MAC/IP belongs to a leaf AC.
  - The MPLS label value in the EC is 0.
  - Upon receiving a route with E-Tree EC, the ingress PE imports the route and installs the MAC in the FDB with a leaf flag (if there is a leaf indication in the route). Any frame coming from a leaf AC for which the MAC destination address (DA) matches a leaf AC MAC is discarded at the ingress.
  - If two PEs send the same MAC with the same ESI but inconsistent root or leaf AC indication, the MAC is installed in the FDB as root.

6.3.12.2 EVPN for MPLS tunnels in E-Tree services

EVPN E-Tree services are modeled as VPLS services configured as E-Trees with the **bgp-evpn mpls** context enabled.

The following example shows a CLI configuration of a VPLS E-Tree service with EVPN E-Tree service enabled.

```
*A:PE1>config>service>system>bgp-evpn#
  evpn-etree-leaf-label
*A:PE1>config>service# vpls 1 customer 1 etree create
*A:PE1>config>service>vpls(etree)# info
----------------------------------------------
  description "ETREE-enabled evpn-mpls-service"
  bgp-evpn
    evi 10
    mpls bgp 1
      no shutdown
      ecmp 2
      auto-bind-tunnel resolution any
      ingress-replication-bum-label
  sap lag-1:1 leaf-ac create
  exit
  sap 2/1/1:1 leaf-ac create
  exit
  sap 2/2/2:1 create
  exit
  spoke-sdp 3:1 leaf-ac create
  exit
```

The following configuration guidelines apply to the EVPN E-Tree service:

- Before configuring an EVPN E-Tree service, the user must first run the **evpn-etree-leaf-label <value>** command. This is relevant for EVPN E-Tree services only. The command allocates an E-Tree leaf label on the system and, when a specific value is configured, the leaf label must match on all other PE nodes attached to the same EVPN service.

  Optionally, the **evpn-etree-leaf-label <value>** command may be configured with a static label value (within the static label range configured in the system using the **config>router>mpls>mpls-label>static-label-range** context). The static label is used when global leaf labels are needed in the network. For example, the case where at least one 7250 IXR Gen 1 router is attached to the EVPN E-Tree service.
- The **configure service vpls create etree** command is compatible with the **bgp-evpn mpls** context.
- As in VPLS E-Tree services, an AC that is not configured as a leaf AC is treated as root AC.
- MAC addresses learned over a leaf AC SAP or SDP binding are advertised as leaf MAC addresses.
- Any PE with one or more **bgp-evpn** enabled VPLS E-Tree service advertises an AD per-ES per-PE route with the leaf indication and leaf label used for BUM egress filtering.
- Any leaf AC SAP or SDP binding defined in an ES triggers the advertisement of an AD per-EVI route with the leaf indication.
- EVPN E-Tree services use the following CLI commands:
  - **sap sap-id leaf-ac create** command using the **configure service vpls** context
  - **mesh-sdp sdp-id:vc-id create leaf-ac** command using the **configure service vpls** context
  - **spoke-sdp sdp-id:vc-id create leaf-ac** command using the **configure service vpls** context
- The **root-leaf-tag** command is blocked in VPLS E-Tree services where **bgp-evpn mpls** context is enabled.

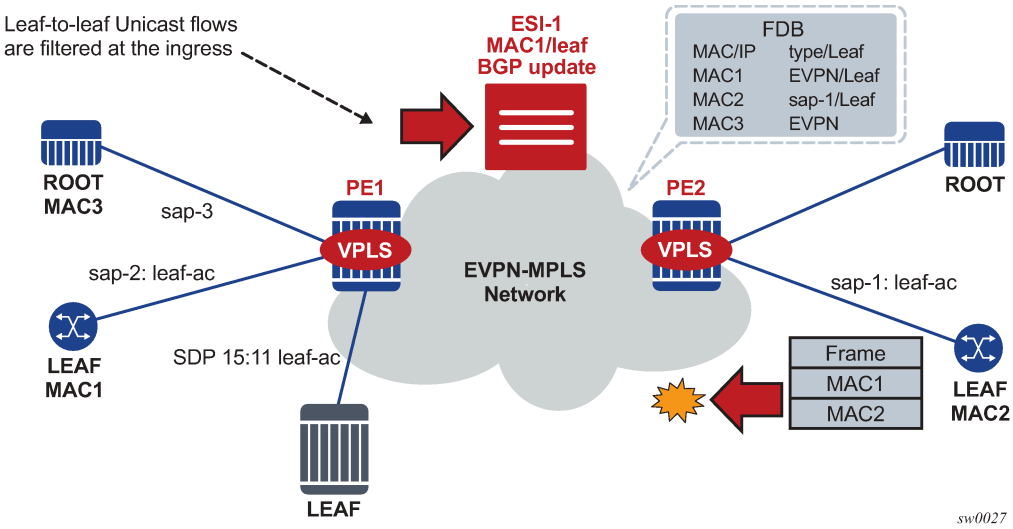
6.3.12.3 EVPN E-Tree operation

EVPN E-Tree supports all operations related to flows among local root AC and leaf AC objects in accordance with IETF RFC 8317. This section describes the extensions required to forward traffic from (or to) root AC and leaf AC objects to (or from) BGP EVPN destinations.

6.3.12.3.1 EVPN E-Tree known unicast ingress filtering

Known unicast traffic forwarding is based on ingress PE filtering. Figure 187: EVPN E-Tree known unicast ingress filtering shows an example of EVPN-E-Tree forwarding behavior for known unicast.

Figure 187: EVPN E-Tree known unicast ingress filtering



MAC addresses learned on **leaf-ac** objects are advertised in EVPN with their corresponding leaf indication.

In Figure 187: EVPN E-Tree known unicast ingress filtering , PE1 advertises MAC1 using the E-Tree EC and leaf indication, and PE2 installs MAC1 with a leaf flag in the FDB.

Assuming MAC DA is present in the local FDB (MAC1 in the FDB of PE2) when PE2 receives a frame, it is handled as follows:

- If the unicast frame enters a root-ac, the frame follows regular data plane procedures; that is, it is sent to the owner of the MAC DA (local SAP or SDP binding or remote BGP EVPN PE) without any filtering.
- If the unicast frame enters a leaf-ac, it is handled as follows:
  1. A MAC DA lookup is performed on the FDB.
  2. If there is a hit and the MAC was learned as an EVPN leaf (or from a leaf-ac), then the frame is dropped at ingress.
  3. The source MAC (MAC2) is learned and marked as a leaf-learned MAC. It is advertised by the EVPN with the corresponding leaf indication.
- A MAC received with a root and leaf indication from different PEs in the same ES is installed as root.

The ingress filtering for E-Tree leaf-to-leaf traffic requires the implementation of an extra leaf EVPN MPLS destination per remote PE (containing leaf objects) per E-Tree service. The ingress filtering for E-Tree leaf-to-leaf traffic is as follows:

- A separate EVPN MPLS bind is created for unicast leaf traffic in the service. The internal EVPN MPLS destination is created for each remote PE that contains a leaf and advertises at least one leaf MAC.
- The creation of the internal EVPN MPLS destination is triggered when a MAC route with L=1 in the E-Tree EC is received. Any EVPN E-Tree service can potentially use one extra EVPN MPLS destination for leaf unicast traffic per remote PE.

  The extra destination in the EVPN E-Tree service is for unicast only and it is not part of the flooding list. It is resource-accounted and displayed in the **tools dump service evpn usage** command, as shown in the following example output.

```
A:PE-4# tools dump service evpn usage
```

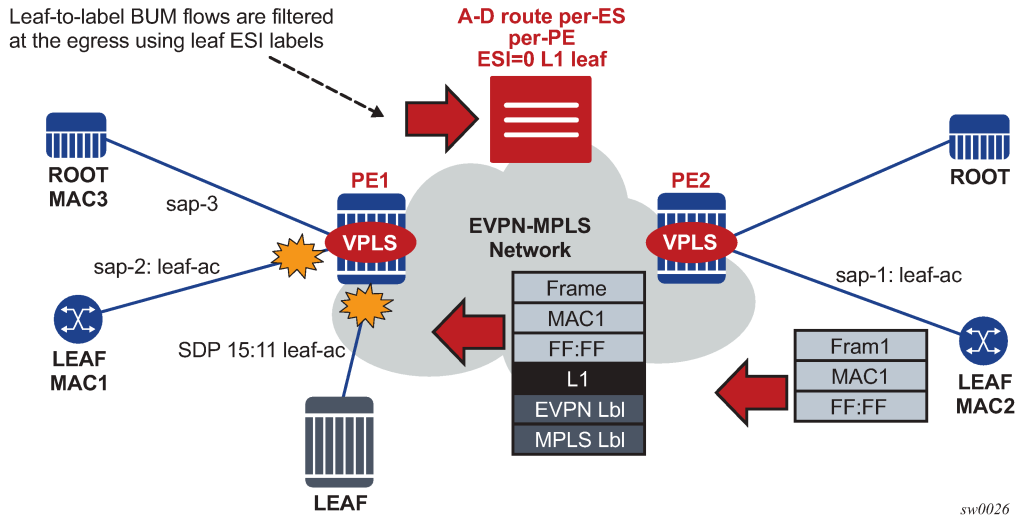
```
vxlan-evpn-mpls usage statistics at 01/23/2017 00:53:14:
MPLS-TEP                                          :             3
VXLAN-TEP                                         :             0
Total-TEP                                         :      3/ 16383
Mpls Dests (TEP, Egress Label + ES + ES-BMAC)     :            10
Mpls Etree Leaf Dests                             :             1
Vxlan Dests (TEP, Egress VNI)                     :             0
Total-Dest                                        :     10/196607
Sdp Bind +  Evpn Dests                            :     13/245759
ES L2/L3 PBR                                      :      0/ 32767
Evpn Etree Remote BUM Leaf Labels                 :             3
```

- MACs received with L=1 point to the EVPN MPLS destination, whereas root MACs point to the "root" destination.

6.3.12.3.2 EVPN E-Tree BUM egress filtering

BUM traffic forwarding is based on egress PE filtering. Figure 188: EVPN E-Tree BUM egress filtering shows an example of EVPN E-Tree forwarding behavior for BUM traffic.

Figure 188: EVPN E-Tree BUM egress filtering



In Figure 188: EVPN E-Tree BUM egress filtering , BUM frames are handled as follows when they ingress PE or PE2:

- If the BUM frame enters a root-ac, the frame follows regular EVPN data plane procedures.
- If the BUM frame enters a leaf-ac, the frame handling is as follows:
  1. The frame is marked as leaf and forwarded or replicated to the egress IOM.
  2. At the egress IOM, the frame is flooded in the default multicast list subject to the following:
     - Leaf entries are skipped when BUM traffic is forwarded. This prevents leaf-to-leaf BUM traffic forwarding.
     - Traffic to remote BGP EVPN PEs is encapsulated with the EVPN label stack. If a leaf ESI label present for the far-end PE (L1 in Figure 188: EVPN E-Tree BUM egress filtering), the leaf ESI label is added at the bottom of the stack; the remaining stack follows (including EVI label). If there is no leaf ESI label for the far-end egress PE, no additional label is added to the stack. This

means that the egress PE does not have any E-Tree enabled service, but it can still work with the VPLS E-Tree service available in PE2.

The BUM-encapsulated packet is received on the network ingress interface at the egress PE or PE1. The packet is processed as follows.

1. A normal ILM lookup is performed for each label (including the EVI label) in the stack.
2. Further label lookups are performed when the EVI label ILM lookup is complete. If the lookup yields a leaf label, all the leaf-acs are skipped when flooding to the default-multicast list at the egress PE.

6.3.12.3.3 EVPN E-Tree egress filtering based on MAC source address

The egress PE checks the MAC Source Address (SA) for traffic received without the leaf MPLS label. This check covers corner cases where the ingress PE sends traffic originating from a leaf-ac but without a leaf indication.

In Figure 188: EVPN E-Tree BUM egress filtering , PE2 receives a frame with MAC DA = MAC3 and MAC SA = MAC2. Because MAC3 is a root MAC, MAC lookup at PE2 allows the system to unicast the packet to PE1 without the leaf label. If MAC3 was no longer in PE1's FDB, PE1 would flood the frame to all the root and leaf-acs, despite the frame having originated from a leaf-ac.

To minimize and prevent leaf traffic from leaking to other leaf-acs (as described in the preceding case), the egress PE always performs a MAC SA check for all types of traffic. The data path performs MAC SA-based egress filtering as follows:

1. An Ethernet frame may be treated as originating from a leaf-ac because of several reasons, which requires the system to set a flag to indicate leaf traffic. The flag is set if one of the following conditions is true:
   - The frames arrive on a leaf SAP.
   - EVPN traffic arrives with a leaf label.
   - A MAC SA is flagged as a leaf SA.
2. After the flag is set, the action taken depends on the type of traffic:
   - **unicast traffic**

     An FDB lookup is performed, and if the MAC DA FDB entry is marked as a leaf type, the frame is dropped to prevent leaf-to-leaf forwarding.
   - **BUM traffic**

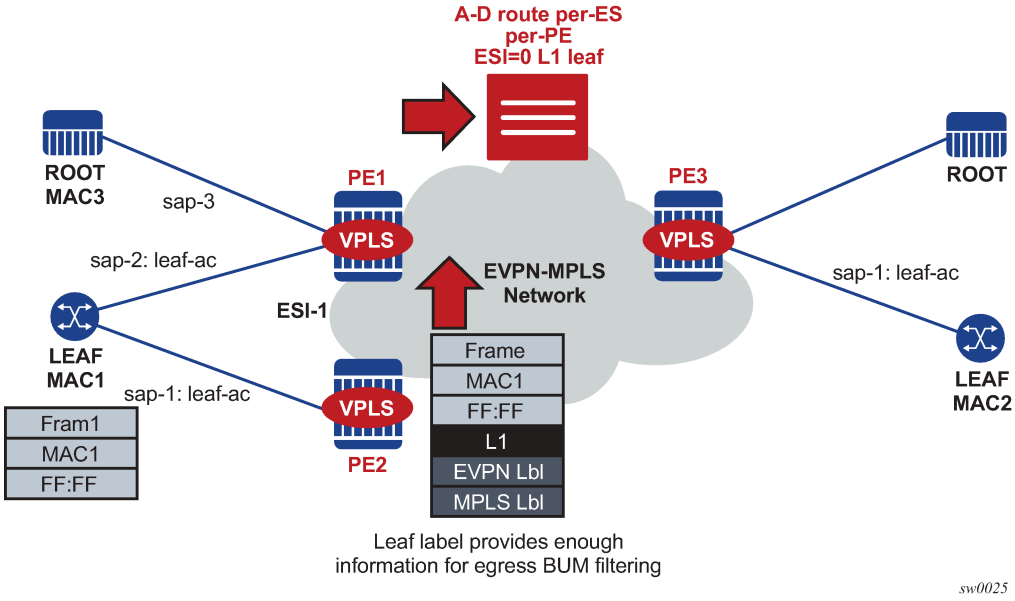
     The flag is considered at the egress IOM and leaf-to-leaf forwarding is suppressed.

6.3.12.4 EVPN E-Tree and EVPN multihoming

EVPN E-Tree procedures support all-active and single-active EVPN multihoming. Ingress filtering can handle MACs learned on ES leaf-ac SAP or SDP bindings. If a MAC associated with an ES leaf-ac is advertised with a different E-Tree indication or if the AD per-EVI routes have inconsistent leaf indications, then the remote PEs performing the aliasing treat the MAC as root.

Figure 189: EVPN E-Tree BUM egress filtering and multihoming shows the expected behavior for multihoming and egress BUM filtering.

Figure 189: EVPN E-Tree BUM egress filtering and multihoming



Multihoming and egress BUM filtering in Figure 189: EVPN E-Tree BUM egress filtering and multihoming is handled as follows:

- BUM frames received on an ES leaf-ac are flooded to the EVPN based on EVPN E-Tree procedures. The leaf ESI label is sent when flooding to other PEs in the same ES, and additional labels are not added to the stack.

  When flooding in the default multicast list, the egress PE skips all the leaf-acs (including the ES leaf-acs) on the assumption that all ACs in a specific ES for a specified EVI have a consistent E-Tree configuration, and they send an AD per-EVI route with a consistent E-Tree indication.
- BUM frames received on an ES root-ac are flooded to the EVPN based on regular EVPN procedures. The regular ES label is sent for split-horizon when packets are sent to the DF or NDF PEs in the same ES. When flooding in the default multicast list, the egress PE skips the ES SAPs based on the ES label lookup.

If the PE receives an ES MAC from a peer that shares the ES and decides to install it against the local ES SAP that is **oper-up**, it checks the E-Tree configuration (root or leaf) of the local ES SAP against the received MAC route. The MAC route is processed as follows:

- If the E-Tree configuration does not match, then the MAC is not installed against any destination until the misconfiguration is resolved.
- If the SAP is **oper-down**, the MAC is installed against the EVPN destination to the peer.

6.3.12.5 PBB-EVPN E-Tree services

SR OS supports PBB-EVPN E-Tree services in accordance with IETF RFC 8317. PBB-EVPN E-Tree services are modeled as PBB-EVPN services where some I-VPLS services are configured as **etree** and some of their SAP or spoke SDPs are configured as leaf-acs.

The procedures for the PBB-EVPN E-Tree are similar to those for the EVPN E-Tree, except that the egress leaf-to-leaf filtering for BUM traffic is based on the B-MAC source address. Also, the leaf label and the EVPN AD routes are not used.

The PBB-EVPN E-Tree operation is as follows:

- When one or more I-VPLS E-Tree services are linked to a B-VPLS, the leaf backbone source MAC address (**leaf-source-bmac** parameter) is used for leaf-originated traffic in addition to the source B-VPLS MAC address (**source-bmac** parameter) that is used for sourcing root traffic.
- The leaf backbone source MAC address for PBB must be configured using the command **config**>**service**>**pbb**>**leaf-source-bmac ieee-address** before the configuration of any I-VPLS E-Tree service.
- The **leaf-source-bmac** address is advertised in a B-MAC route with a leaf indication.
- Known unicast filtering occurs at the ingress PE. When a frame enters an I-VPLS leaf-ac, a MAC lookup is performed. If the C-MAC DA is associated with a leaf B-MAC, the frame is dropped.
- Leaf-to-leaf BUM traffic filtering occurs at the egress PE. When flooding BUM traffic with the B-MAC SA matching a leaf B-MAC, the egress PE skips the I-VPLS leaf-acs.

The following CLI example shows an I-VPLS E-Tree service that uses PBB-EVPN E-Tree. The **leaf-source-bmac** address must be configured before the configuration of the I-VPLS E-Tree. As is the case in regular E-Tree services, SAP and spoke SDPs that are not explicitly configured as leaf-acs are considered root-ac objects.

```
A:PE-2>config>service# info
----------------------------------------------
        pbb
            leaf-source-bmac 00:00:00:00:00:22
        exit
        vpls 1000 customer 1 name "vpls1000" b-vpls create
            service-mtu 2000
            bgp
            exit
            bgp-evpn
                evi 1000
                exit
                mpls bgp 1
                    ingress-replication-bum-label
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
        exit
        vpls 1001 customer 1 i-vpls etree create
            pbb
                backbone-vpls 1000
                exit
            exit
            stp
                shutdown
            exit
            sap 1/1/1:1001 leaf-ac create
                no shutdown
            exit
```

```
            sap 1/1/1:1002 create
                no shutdown
            exit
            no shutdown
        exit
```

The following considerations apply to PBB-EVPN E-Trees and multihoming:

- All-active multihoming is not supported on leaf-ac I-VPLS SAPs.
- Single-active multihoming is supported on leaf-ac I-VPLS SAPs and spoke SDPs.
- ISID- and RFC 7623-based C-MAC flush are supported in addition to PBB-EVPN E-Tree services and single-active multihoming.

6.3.13 MPLS entropy label and hash label

The router supports the MPLS entropy label (RFC 6790) for EVPN VPLS and Epipe services, and the Flow Aware Transport label (known as the hash label) (RFC 6391) on spoke SDPs bound to a VPLS EVPN service. This label allows LSR nodes in a network to load-balance labeled packets in a much more granular fashion than allowed by simply hashing on the standard label stack. The entropy label can be enabled on BGP-EVPN services (VPLS and Epipe).

To configure insertion of the entropy label on a BGP-EVPN VPLS or Epipe, use the **entropy-label** command in the **bgp-evpn**>**mpls** context. Use the **entropy-label** command under the **spoke-sdp** context to configure insertion of the entropy label on spoke SDPs bound to a BGP-EVPN VPLS. Note that the entropy label is only inserted if the far end of the MPLS tunnel is also entropy-label-capable. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide*.

The hash label is configured using the **hash-label** command in the **spoke-sdp** context. Either the hash label or the entropy label can be configured on one object, but not both.

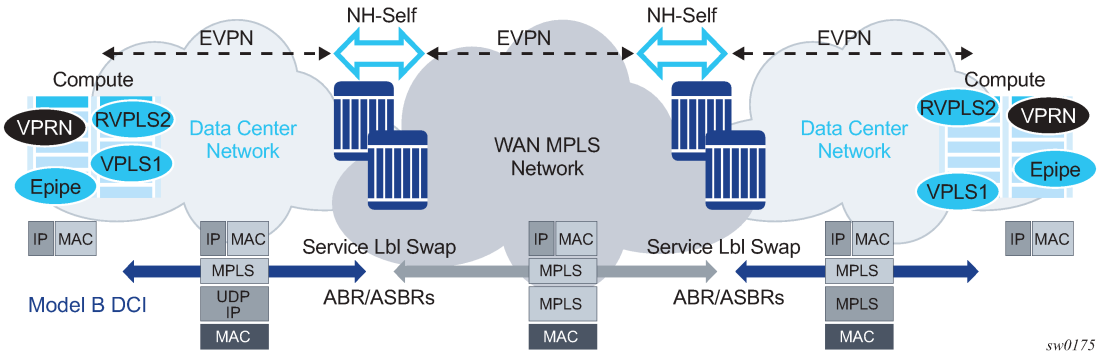
6.3.14 Inter-AS Option B and Next-Hop-Self Route-Reflector for EVPN-MPLS

Inter-AS Option B and Next-Hop-Self Route-Reflector (VPN-NH-RR) functions are supported for the BGP-EVPN family in the same way both functions are supported for IP-VPN families.

A typical use case for EVPN Inter-AS Option B or EVPN VPN-NH-RR is Data Center Interconnect (DCI) networks, where cloud and service providers are looking for efficient ways to extend their Layer 2 and Layer 3 tenant services beyond the data center and provide a tighter DC-WAN integration. While the instantiation of EVPN services in the DGW to provide this DCI connectivity is a common model, some operators use Inter-AS Option B or VPN-NH-RR connectivity to allow the DGW to function as an ASBR or ABR respectively, and the services are only instantiated on the edge devices.

Figure 190: EVPN inter-AS Option B or VPN-NH-RR model shows a DCI example where the EVPN services in two DCs are interconnected without the need for instantiating services on the DC GWs.

Figure 190: EVPN inter-AS Option B or VPN-NH-RR model



The ASBRs or ABRs connect the DC to the WAN at the control plane and data plane levels where the following considerations apply:

- From a control plane perspective, the ASBRs or ABRs perform the following tasks:
  1. accept EVPN-MPLS routes from a BGP peer

     EVPN-VXLAN routes are not supported.
  2. extract the MPLS label from the EVPN NLRI or attribute and program a label swap operation on the IOM
  3. re-advertise the EVPN-MPLS route to the BGP peer in the other Autonomous Systems (ASs) or IGP domains

     The re-advertised route has a Next-Hop-Self and a new label encoded for those routes that came with a label.
- From a data plan perspective, the ASBRs and ABRs terminate the ingress transport tunnel, perform an EVPN label swap operation, and send the packets on to an interface (if E-BGP is used) or a new tunnel (if IBGP is used).
- The ASBR or ABR resolves the EVPN routes based on the existing **bgp next-hop-resolution** command for **family vpn**, where **vpn** refers to EVPN, VPN-IPv4, and VPN-IPv6 families.

```
*A:ABR-1# configure router bgp next-hop-resolution labeled-routes transport-tunnel
family vpn resolution-filter
  - resolution-filter
 [no] bgp             - Use BGP tunnelling for next hop resolution
 [no] ldp             - Use LDP tunnelling for next hop resolution
 [no] rsvp            - Use RSVP tunnelling for next hop resolution
 [no] sr-isis         - Use sr-isis tunnelling for next hop resolution
 [no] sr-ospf         - Use sr-ospf for next hop resolution
 [no] sr-te           - Use sr-te for next hop resolution
 [no] udp             - Use udp for next hop resolution
```

For more information about the next-hop resolution of BGP-labeled routes, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Unicast Routing Protocols Guide*

Inter-AS Option B for EVPN services on ABSRs and VPN-NH-RR on ABRs re-use the existing commands **enable-inter-as-vpn** and **enable-rr-vpn-forwarding** respectively. The two commands enable the ASBR or ABR function for both EVPN and IP-VPN routes. These two features can be used with the following EVPN services:

- EVPN-MPLS Epipe services (EVPN-VPWS)

- EVPN-MPLS VPLS services
- EVPN-MPLS R-VPLS services
- PBB-EVPN and PBB-EVPN E-Tree services
- EVPN-MPLS E-Tree services
- PE and ABR functions (EVPN services and **enable-rr-vpn-forwarding**), which are both supported on the same router
- PE and ASBR functions (EVPN services and **enable-inter-as-vpn**), which are both supported on the same router

The following sub-sections clarify some aspects of EVPN when used in an Inter-AS Option B or VPN-NH-RR network.

6.3.14.1 Inter-AS Option B and VPN-NH-RR procedures on EVPN routes

When **enable-rr-vpn-forwarding** or **enable-inter-as-vpn** is configured, only EVPN-MPLS routes are processed for label swap and the next hop is changed. EVPN-VXLAN routes are re-advertised without a change in the next hop.

The following shows how the router processes and re-advertises the different EVPN route types. For more information about the route fields, see the BGP-EVPN control plane for MPLS tunnels Guide.

- **Auto-discovery (AD) routes (type 1)**

  For AD per EVI routes, the MPLS label is extracted from the route NLRI. The route is re-advertised with Next-Hop-Self (NHS) and a new label. No modifications are made for the remaining attributes.

  For AD per ES routes, the MPLS label in the NLRI is zero. The route is re-advertised with NHS and the MPLS label remains zero. No modifications are made for the remaining attributes.

- **MAC/IP routes (type 2)**

  The MPLS label (Label-1) is extracted from the NLRI. The route is re-advertised with NHS and a new Label-1. No modifications are made for the remaining attributes.

- **Inclusive Multicast Ethernet Tag (IMET) routes (type 3)**

  Because there is no MPLS label present in the NLRI, the MPLS label is extracted from the PMSI Tunnel Attribute (PTA) if needed, and the route is then re-advertised with NHS, with the following considerations:

  – For IMET routes with tunnel-type Ingress Replication, the router extracts the IR label from the PTA. The router programs the label swap and re-advertises the route with a new label in the PTA.

  – For tunnel-type P2MP mLDP, the router re-advertises the route with NHS. No label is extracted; therefore, no swap operation occurs.

  – For tunnel-type Composite, the IR label is extracted from the PTA, the swap operation is programmed and the route re-advertised with NHS. A new label is encoded in the PTA's IR label with no other changes in the remaining fields.

  – For tunnel-type AR, the routes are always considered VXLAN routes and are re-advertised with the next-hop unchanged.

- **Ethernet-Segment (ES) routes (type 4)**

Because ES routes do not contain an MPLS label, the route is re-advertised with NHS and no modifications to the remaining attributes. Although an ASBR or ABR re-advertises ES routes, EVPN multihoming for ES PEs located in different ASs or IGMP domains is not supported.

- **IP-Prefix routes (type 5)**

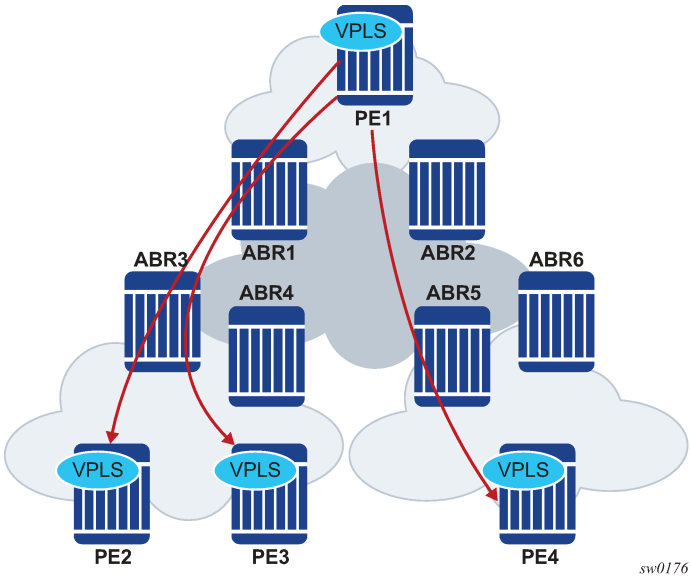
  The MPLS label is extracted from the NLRI and the route is re-advertised with NHS and a new label. No modifications are made to the remaining attributes.

6.3.14.2 BUM traffic in inter-AS Option B and VPN-NH-RR networks

Inter-AS Option B and VPN-NH-RR support the use of non-segmented trees for forwarding BUM traffic in EVPN.

For ingress replication and non-segmented trees, the ASBR or ABR performs an EVPN BUM label swap without any aggregation or further replication. This concept is shown in Figure 191: VPN-NH-RR and ingress replication for BUM traffic.

Figure 191: VPN-NH-RR and ingress replication for BUM traffic



In Figure 191: VPN-NH-RR and ingress replication for BUM traffic, when PE2, PE3, and PE4 advertise their IMET routes, the ABRs re-advertise the routes with NHS and a different label. However, IMET routes are not aggregated; therefore, PE1 sets up three different EVPN multicast destinations and sends three copies of every BUM packet, even if they are sent to the same ABR. This example is also applicable to ASBRs and Inter-AS Option B.

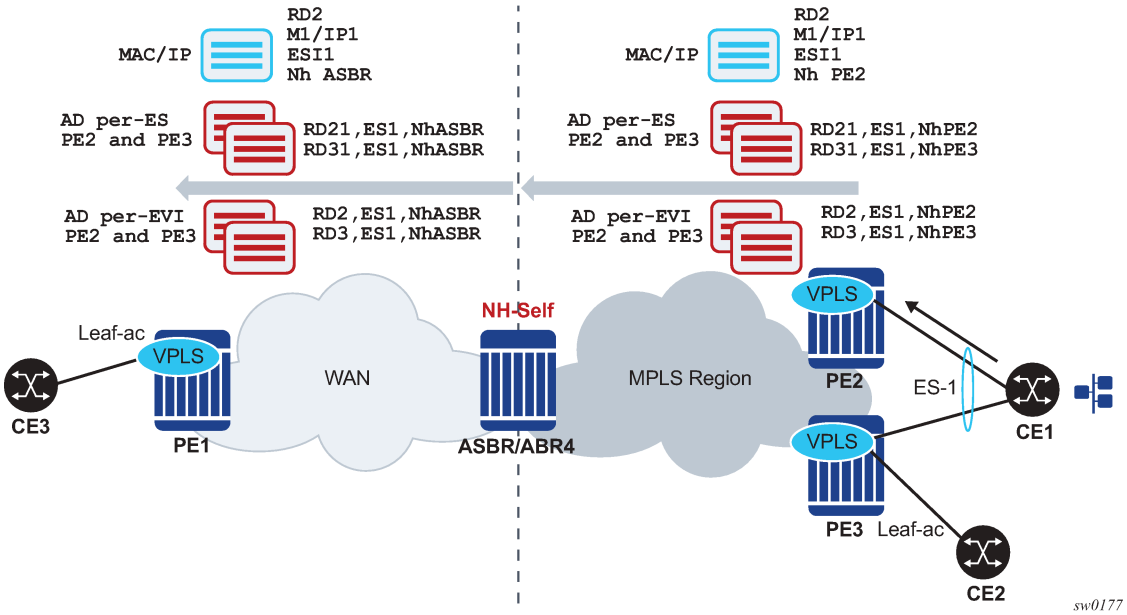
P2MP mLDP may also be used with VPN-NH-RR, but not with Inter-AS Option B. The ABRs, however, do not aggregate or change the mLDP root IP addresses in the IMET routes. The root IP addresses must be leaked across IGP domains. For example, if PE2 advertises an IMET route with mLDP or composite tunnel type, PE1 is able to join the mLDP tree if the root IP is leaked into PE1's IGP domain.

6.3.14.3 EVPN multihoming in inter-AS Option B and VPN-NH-RR networks

In general, EVPN multihoming is supported in Inter-AS Option B or VPN-NH-RR networks with the following limitations:

- An ES PE can only process a remote ES route correctly if the received next hop and origination IP address match. EVPN multihoming is not supported when the ES PEs are in different ASs or IGP domains, or if there is an NH-RR peering the ES PEs and overriding the ES route next hops.
- EVPN multihoming ESs are not supported on EVPN PEs that are also ABRs or ASBRs.
- Mass-withdraw based on the AD per-ES routes is not supported for a PE that is in a different AS or IGP domain that the ES PEs. Figure 192: EVPN multihoming with inter-AS Option B or VPN-NH-RR shows an EVPN multihoming scenario where the ES PEs, PE2 and PE3, and the remote PE, PE1, are in different ASs or IGP domains.

Figure 192: EVPN multihoming with inter-AS Option B or VPN-NH-RR



In Figure 192: EVPN multihoming with inter-AS Option B or VPN-NH-RR, PE1's aliasing and backup functions to the remote ES-1 are supported. However, PE1 cannot identify the originating PE for the received AD per-ES routes because they are both arriving with the same next hop (ASBR/ABR4) and RDs may not help to correlate each AD per-ES route to a specified PE. Therefore, if there is a failure on PE2's ES link, PE1 cannot remove PE2 from the destinations list for ES-1 based on the AD per-ES route. PE1 must wait for the AD per-EVI route withdrawals to remove PE2 from the list. In summary, when the ES PEs and the remote PE are in different ASs or IGP domains, per-service withdrawal based on AD per-EVI routes is supported, but mass-withdrawal based on AD per-ES routes is not supported.

6.3.14.4 EVPN E-Tree in inter-AS Option B and VPN-NH-RR networks

Unicast procedures known to EVPN-MPLS E-Tree are supported in Inter-AS Option B or VPN-NH-RR scenarios, however, the BUM filtering procedures are affected.

As described in EVPN E-Tree, leaf-to-leaf BUM filtering is based on the Leaf Label identification at the egress PE. In a non-Inter-AS or non-VPN-NH-RR scenario, EVPN E-tree AD per-ES (ESI-0) routes carrying the Leaf Label are distinguished by the advertised next hop. In Inter-AS or VPN-NH-RR scenarios, all the AD per-ES routes are received with the ABR or ASBR next hop. Therefore, AD per-ES routes originating from different PEs would all have the same next hop, and the ingress PE would not be able to determine which leaf label to use for a specific EVPN multicast destination.

A simplified EVPN E-Tree solution is supported, where an E-Tree Leaf Label is not installed in the IOM if the PE receives more than one E-Tree AD per-ES route, with different RDs, for the same next hop. In this case, leaf BUM traffic is transmitted without a Leaf Label and the leaf-to-leaf traffic filtering depends on the egress source MAC filtering on the egress PE. See EVPN E-Tree egress filtering based on MAC source address.

PBB-EVPN E-tree services are not affected by Inter-AS or VPN-NH-RR scenarios, as AD per-ES routes are not used.

6.3.15 ECMP for EVPN-MPLS destinations

ECMP is supported for EVPN route next hops that are resolved to EVPN-MPLS destinations as follows:

- **ECMP for Layer 2 unicast traffic on Epipe and VPLS services for EVPN-MPLS destinations**

  This is enabled by the **configure service epipe/vpls bgp-evpn mpls auto-bind-tunnel ecmp number** command and allows the resolution of an EVPN-MPLS next hop to a group of ECMP tunnels of type RSVP-TE, SR-TE or BGP.

- **ECMP for Layer 3 unicast traffic on R-VPLS services with EVPN-MPLS destinations**

  This is enabled by the **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** and **configure service vpls allow-ip-int-bind evpn-mpls-ecmp** commands.

  The VPRN unicast traffic (IPv4 and IPv6) is sprayed among "m" paths, with "m" being the lowest value of (16,n), where "n" is the number of ECMP paths configured in the **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** command.

  CPM originated traffic is not sprayed and picks up the first tunnel in the set.

  This feature is limited to FP3 and above systems.

- **ECMP for Layer 3 multicast traffic on R-VPLS services with EVPN-MPLS destinations**

  This is enabled by the **configure service vpls allow-ip-int-bind ip-multicast-ecmp** and **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** commands. The VPRN multicast traffic (IPv4 and IPv6) are sprayed among up to "m" paths, with "m" being the lowest value of (16,n), and "n" is the number of ECMP paths configured in the **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** command.

In all of these cases, the **configure service epipe/vpls bgp-evpn mpls auto-bind-tunnel ecmp number** command determines the number of Traffic Engineering (TE) tunnels that an EVPN next hop can resolved to. TE tunnels refer to RSVP-TE or SR-TE types. For shortest path tunnels, such as, ldp, sr-isis, sr-ospf, udp, and so on, the number of tunnels in the ECMP group are determined by the **configure router ecmp** command.

In addition, weighted ECMP for Layer 2 unicast traffic on Epipe and VPLS services for EVPN-MPLS destinations is also supported. When the **bgp-evpn>mpls>auto-bind-tunnel>weighted-ecmp** command is configured, packets are sprayed across Traffic Engineering (TE) LSPs in the ECMP set according to the outcome of the hash algorithm and the configured load balancing weight of each LSP.

6.3.16 IPv6 tunnel resolution for EVPN MPLS services

EVPN MPLS services can be deployed in a pure IPv6 network infrastructure, where IPv6 addresses are used as next-hops of the advertised EVPN routes, and EVPN routes received with IPv6 next-hops are resolved to tunnels in the IPv6 tunnel-table.

To change the **system-ipv4** address that is advertised as the next-hop for a local EVPN MPLS service by default, configure the **config**>**service**>**vpls**>**bgp-evpn**>**mpls**>**route-next-hop** {**system-ipv4** | **system-ipv6** | **ip-address**} command or the **config**>**service**>**epipe**>**bgp-evpn**>**mpls**>**route-next-hop** {**system-ipv4** | **system-ipv6** | **ip-address**} command.

The configured IP address is used as a next-hop for the MAC/IP, IMET, and AD per-EVI routes advertised for the service. Note that this configured next-hop can be overridden by a policy with the **next-hop-self** command.

In the case of Inter-AS model B or next-hop-self route-reflector scenarios, at the ASBR/ABR:

- A route received with an IPv4 next-hop can be re-advertised to a neighbor with an IPv6 next-hop. The neighbor must be configured with the **advertise-ipv6-next-hops evpn** command.
- A route received with an IPv6 next-hop can be re-advertised to a neighbor with an IPv4 next-hop. The **no advertise-ipv6-next-hops evpn** command must be configured on that neighbor.

6.3.17 EVPN multihoming support for MPLS tunnels resolved to non-system IPv4/IPv6 addresses

EVPN MPLS multihoming is supported on PEs that use non-system IPv4 or IPv6 addresses for tunnel resolution. Similar to multihoming in EVPN VXLAN networks, (see Non-system IPv4 and IPv6 VXLAN termination for EVPN VXLAN multihoming), additional configuration steps are required.

- The **configure service system bgp-evpn eth-seg es-orig-ip ip-address** command must be configured with the non-system IPv4 or IPv6 address used for the EVPN-MPLS service. This command modifies the originating IP field in the ES routes advertised for the Ethernet Segment, and makes the system use this IP address when adding the local PE as DF candidate.
- The **configure service system bgp-evpn eth-seg route-next-hop ip-address** command must also be configured with the non-system IP address. This command changes the next-hop of the ES and AD per-ES routes to the configured address.
- All the EVPN MPLS services that make use of the Ethernet Segment must be configured with the **configure service vpls|epipe bgp-evpn mpls route-next-hop ip-address** command.

When multihoming is used in the service, the same IP address should be configured in all three of the commands detailed above, so the DF Election candidate list is built correctly.

6.4 EVPN for SRv6 tunnels

EVPN-VPWS, EVPN on VPLS services, and EVPN on VPRN services (EVPN-IFL) are supported with SRv6 tunnels. See the *7750 SR and 7950 XRS Segment Routing and PCE User Guide* for more information about EVPN for SRv6 tunnels.

6.5 General EVPN topics

This section provides information about general topics related to EVPN.

6.5.1 ARP/ND snooping and proxy support

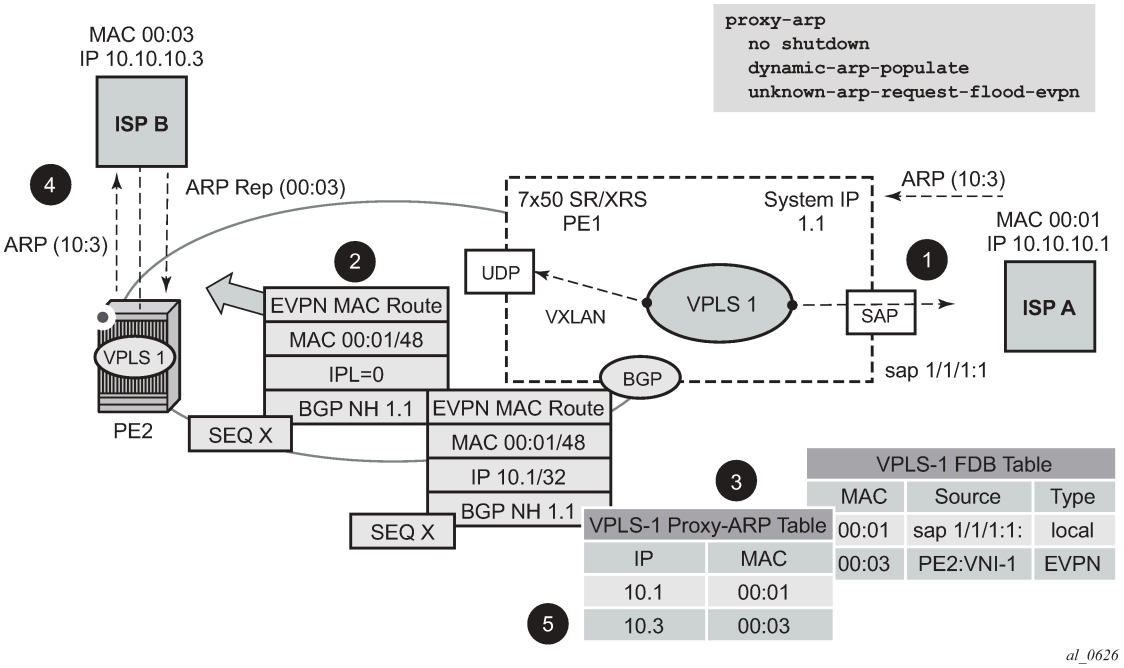
VPLS services support proxy-ARP (Address Resolution Protocol) and proxy-ND (Neighbor Discovery) functions that can be enabled or disabled independently per service. When enabled (proxy-ARP/proxy-ND **no shutdown**), the system populates the corresponding proxy-ARP/proxy-ND table with IP--MAC entries learned from the following sources:

- EVPN-received IP-MAC entries
- User-configured static IP-MAC entries
- Snooped dynamic IP-MAC entries (learned from ARP/GARP/NA messages received on local SAPs/ SDP bindings)

In addition, any ingress ARP or ND frame on a SAP or SDP binding is intercepted and processed. ARP requests and Neighbor Solicitations are answered by the system if the requested IP address is present in the proxy table.

Figure 193: Proxy-ARP example usage in an EVPN network shows an example of how proxy-ARP is used in an EVPN network. Proxy-ND would work in a similar way. The MAC address notation in the diagram is shortened for readability.

Figure 193: Proxy-ARP example usage in an EVPN network



PE1 is configured as follows:

```
*A:PE1>config>service>vpls# info
----------------------------------------------
```

```
    vxlan instance 1 vni 600 create
        exit
        bgp
            route-distinguisher 192.0.2.71:600
            route-target export target:64500:600 import target:64500:600
    exit
    bgp-evpn
        vxlan bgp 1 vxlan-instance 1
            no shutdown
        exit
    exit
    proxy-arp
        age-time 600
        send-refresh 200
        dup-detect window 3 num-moves 3 hold-down max anti-spoof-
mac 00:ca:ca:ca:ca:ca
        dynamic-arp-populate
          no shutdown
            exit
            sap 1/1/1:600 create
            exit
    no shutdown
-----------------------------------------------
```

Figure 193: Proxy-ARP example usage in an EVPN network shows the following steps, assuming proxy-ARP is no shutdown on PE1 and PE2, and the tables are empty:

1. ISP-A sends ARP-request for (10.10.)10.3.
2. PE1 learns the MAC 00:01 in the FDB as usual and advertises it in EVPN without any IP. Optionally, the MAC can be configured as a CStatic mac, in which case it is advertised as protected. If the MAC is learned on a SAP or SDP binding where **auto-learn-mac-protect** is enabled, the MAC is also advertised as protected.
3. The ARP-request is sent to the CPM where:
   - An ARP entry (IP 10.1'MAC 00:01) is populated into the proxy-ARP table.
   - EVPN advertises MAC 00:01 and IP 10.1 in EVPN with the same SEQ number and Protected bit as the previous route-type 2 for MAC 00:01.
   - A GARP is also issued to other SAPs/SDP bindings (assuming they are not in the same split horizon group as the source). If garp-flood-evpn is enabled, the GARP message is also sent to the EVPN network.
   - The original ARP-request can still be flooded to the EVPN or not based on the **unknown-arp-request-flood-evpn** command.
4. Assuming PE1 was configured with **unknown-arp-request-flood-evpn**, the ARP-request is flooded to PE2 and delivered to ISP-B. ISP-B replies with its MAC in the ARP-reply. The ARP-reply is finally delivered to ISP-A.
5. PE2 learns MAC 00:01 in the FDB and the entry 10.1'00:01 in the proxy-ARP table, based on the EVPN advertisements.
6. When ISP-B replies with its MAC in the ARP-reply:
   - MAC 00:03 is learned in FDB at PE2 and advertised in EVPN.
   - MAC 00:03 and IP 10.3 are learned in the proxy-ARP table and advertised in EVPN with the same SEQ number as the previous MAC route.
   - ARP-reply is unicasted to MAC 00:01.

7. EVPN advertisements are used to populate PE1's FDB (MAC 00:03) and proxy-ARP (IP 10.3—>MAC 00:03) tables as mentioned in 5.

From this point onward, the PEs reply to any ARP-request for 00:01 or 00:03, without the need for flooding the message in the EVPN network. By replying to known ARP-requests / Neighbor Solicitations, the PEs help to significantly reduce the flooding in the network.

Use the following commands to customize proxy-ARP/proxy-ND behavior:

- **dynamic-arp-populate** and **dynamic-nd-populate**

  Enables the addition of dynamic entries to the proxy-ARP or proxy-ND table (disabled by default). When executed, the system populates proxy-ARP/proxy-ND entries from snooped GARP/ARP/NA messages on SAPs/SDP bindings in addition to the entries coming from EVPN (if EVPN is enabled). These entries are shown as **dynamic**.

- **static <IPv4-address> <mac-address>** and **static <IPv4-address> <mac-address>** and **static <ipv6-address> <mac-address> {host | router}**

  Configures static entries to be added to the table.

  Note: A static IP-MAC entry requires the addition of the MAC address to the FDB as either learned or CStatic (conditional static mac) to become active (**Status** —> active).

- **age-time <60 to 86400>** (seconds)

  Specifies the aging timer per proxy-ARP/proxy-ND entry. When the aging expires, the entry is flushed. The age is reset when a new ARP/GARP/NA for the same IP MAC is received.

- **send-refresh <120 to 86400>** (seconds)

  If enabled, the system sends ARP-request/Neighbor Solicitation messages at the configured time, so that the owner of the IP can reply and therefore refresh its IP MAC (proxy-ARP entry) and MAC (FDB entry).

- **table-size [1 to 16384]**

  Enables the user to limit the number of entries learned on a specified service. By default, the table-size limit is 250.

  The unknown ARP-requests, NS, or the unsolicited GARPs and NA messages can be configured to be flooded or not in an EVPN network with the following commands:

  – proxy-arp [no] unknown-arp-request-flood-evpn
  – proxy-arp [no] garp-flood-evpn
  – proxy-nd [no] unknown-ns-flood-evpn
  – proxy-nd [no] host-unsolicited-na-flood-evpn
  – proxy-nd [no] router-unsolicited-na-flood-evpn

- **dup-detect** [**anti-spoof-mac** <**mac-address**>] **window** <**minutes**> **num-moves** <**count**> **hold-down** <**minutes** | **max**>

  Enables a mechanism that detects duplicate IPs and ARP/ND spoofing attacks. The working of the **dup-detect** command can be summarized as follows:

  – Attempts (relevant to dynamic and EVPN entry types) to add the same IP (different MAC) are monitored for <**window**> minutes and when <count> is reached within that **window**, the proxy-ARP/proxy-ND entry for the IP is suspected and marked as **duplicate**. An alarm is also triggered.

– The condition is cleared when hold-down time expires (**max** does not expire) or a **clear** command is issued.
– If the **anti-spoof-mac** is configured, the proxy-ARP/proxy-ND offending entry's MAC is replaced by this <mac-address> and advertised in an unsolicited GARP/NA for local SAP or SDP bindings and in EVPN to remote PEs.
– This mechanism assumes that the same **anti-spoof-mac** is configured in all the PEs for the same service and that traffic with destination **anti-spoof-mac** received on SAPs/SDP bindings are dropped. An ingress MAC filter has to be configured to drop traffic to the **anti-spoof-mac**.

Table 23: Proxy-arp entry combinations shows the combinations that produce a **Status = Active** proxy-arp entry in the table. The system replies to proxy-ARP requests for active entries. Any other combination results in a **Status = inActv** entry. If the service is not active, the proxy-arp entries are not active either, regardless of the FDB entries

Note: A static entry is active in the FDB even when the service is down.

Table 23: Proxy-arp entry combinations

Proxy-arp entry type	FDB entry type (for the same MAC)
Dynamic	learned
Static	learned
Dynamic	CStatic/Static
Static	CStatic/Static
EVPN	EVPN, learned/CStatic/Static with matching ESI
Duplicate	—

When proxy-ARP/proxy-ND is enabled on services with all-active multihomed Ethernet Segments, a proxy-arp entry type **evpn** may be associated with learned/CStatic/Static FDB entries (because for example, the CE can send traffic for the same MAC to all the multihomed PEs in the ES). If this is the case, the entry is active if the ESI of the EVPN route and the FDB entry match, or inactive otherwise, as per Table 23: Proxy-arp entry combinations.

6.5.1.1 Proxy-ARP/ND periodic refresh, unsolicited refresh and confirm-messages

When proxy-ARP/proxy-ND is enabled, the system starts populating the proxy table and responding to ARP-requests/NS messages. To keep the active IP-MAC entries alive and ensure that all the host/routers in the service update their ARP/ND caches, the system may generate the following three types of ARP/ND messages for a specified IP-MAC entry:

- **periodic refresh messages (ARP-requests or NS for a specified IP)**

  These messages are activated by the **send-refresh** command and their objective is to keep the existing FDB and Proxy-ARP/ND entries alive to minimize EVPN withdrawals and re-advertisements.
- **unsolicited refresh messages (unsolicited GARP or NA messages)**

These messages are sent by the system when a new entry is learned or updated. Their objective is to update the attached host/router caches.

- **confirm messages (unicast ARP-requests or unicast NS messages)**

  These messages are sent by the system when a new MAC is learned for an existing IP. The objective of the confirm messages is to verify that a specified IP has really moved to a different part of the network and is associated with the new MAC. If the IP has not moved, it forces the owners of the duplicate IP to reply and cause **dup-detect** to kick in.

6.5.1.2 Advertisement of Proxy-ARP/ND flags in EVPN

When a dynamic or static Proxy-ARP/ND entry is learned (or configured), the following property flags are created with it:

- The Router flag (R) is used in IPv6 Neighbor Advertisement messages to indicate if the proxy-ND entry belongs to an IPv6 router or an IPv6 host.
- The Override flag (O) is used in IPv6 Neighbor Advertisement messages to indicate whether the resolved entry should override a potential ND entry that the solicitor may already have for the same IPv6 address.
- The Immutable flag (I) indicates that the proxy-ARP or proxy-ND entry cannot change its binding to a different MAC addresses. This Flag is always set for static proxy-ARP/ND entries or configured dynamic IP addresses that are associated with a **mac-list**.

RFC9047 describes how to convey the flags (R, O and I) in the EVPN ARP/ND extended community that is advertised with the EVPN MAC/IP Advertisement routes. This enables the ingress and egress EVPN PEs to install the proxy-ARP/ND entries with the same property flags. The following figure shows the format of the EVPN ARP/ND extended community.

Figure 194: Format of EVPN ARP/ND extended community

By default, the router does not advertise the ARP/ND extended community. Use the following command to configure the router to advertise all the proxy ARP/ND MAC/IP Advertisement routes with the extended community:

```
configure service vpls bgp-evpn arp-nd-extended-community
```

6.5.1.3 Proxy-ARP/ND and flag processing

Proxy-ND and the Router Flag

RFC 4861 describes the use of the (R) or Router flag in NA messages as follows:

- A node capable of routing IPv6 packets must reply to NS messages with NA messages where the R flag is set (R=1).
- Hosts must reply with NA messages where R=0.

The R flag in NA messages impacts how the hosts select their default gateways when sending packets off-link. The proxy-ND function on the router does one of the following, depending on whether it can provide the appropriate R flag information:

- provides the appropriate R flag information in the proxy-ND NA replies, if possible
- floods the received NA messages, if it cannot provide the appropriate R flag when replying

The use of the R flag (only present in NA messages and not in NS messages) makes the procedure for learning proxy-ND entries and replying to NS messages different from the procedures for proxy-ARP in IPv4. The NA messages snooping determines the router or host flag to add to each entry, and that determines the flag to use when responding to an NS message.

The procedure to add the R flag to a specified entry is as follows:

- Dynamic entries are learned based on received NA messages. The R flag is also learned and added to the proxy-ND entry so that the appropriate R flag is used in response to NS requests for a specified IP.
- Static entries are configured as host or router using the following command.
  - **MD-CLI**

    ```
    configure service vpls proxy-nd static-neighbor ip-address type
    ```

  - **classic CLI**

    ```
    configure service vpls proxy-nd static
    ```

- EVPN entries are learned from BGP and the following command determines the R flag added to them;
  - **MD-CLI**

    ```
    configure service vpls proxy-nd evpn advertise-neighbor-type
    ```

  - **classic CLI**

    ```
    configure service vpls proxy-nd evpn-nd-advertise
    ```

  in case the following command is not configured (if configured, the signaled flag value determines the flag of the entry).

  - **MD-CLI**

    ```
    configure service vpls bgp-evpn routes mac-ip arp-nd-extended-community
    ```

  - **classic CLI**

    ```
    configure service vpls bgp-evpn arp-nd-extended-community-advertisement
    ```

- In addition, the EVPN ND advertisement indicates what static and dynamic IP → MAC entries the system advertises in EVPN.
  - If you specify the router option for EVPN ND advertisement, the system should flood the received unsolicited NA messages for hosts. This is controlled by the following command:
    - **MD-CLI**

      ```
      configure service vpls proxy-nd evpn flood unknown-neighbor-advertise-host
      ```

    - **classic CLI**

      ```
      configure service vpls proxy-nd host-unsolicited-na-flood-evpn
      ```

  - The opposite is also true so that the host option for EVPN ND advertisement is configured with the following command:
    - **MD-CLI**

      ```
      configure service vpls proxy-nd evpn flood unknown-neighbor-advertise-router
      ```

    - **classic CLI**

      ```
      configure service vpls proxy-nd router-unsolicited-na-flood-evpn
      ```

  - The router-host option for EVPN ND advertisement allows the router to advertise both types of entries in EVPN at the same time. That is, static and dynamic entries with the **router** or **host** flag are advertised in EVPN with the corresponding flag in the ARP/ND extended community. This option can be enabled only if the ARP/ND extended community is configured.

EVPN proxy-ND MAC/IP Advertisement routes received without the EVPN ARP/ND extended communities create an entry with type Router (which is the default value). Entries created as duplicate are advertised in EVPN with an R flag value that depends on the configuration of the EVPN ND advertisement command. If the **host** option is configured for the EVPN ND advertisement, the duplicate entry is treated as a host. If the **router** or **router-host** option is configured for the EVPN ND advertisement, the duplicate entry behaves as a router.

Proxy-ARP/ND and the Immutable Flag

The I bit or Immutable flag in the ARP/ND extended community is advertised and used as follows:

- Any static proxy-ARP/ND entry is advertised with I=1 if you enable ARP/ND extended community advertisement.
- Any configured **dynamic** IP address (associated with a mac-list) proxy-ARP/ND entry is advertised with I=1 if you enable ARP/ND extended community
- Duplicate entries are advertised with I=1 as well (in addition to O=1 and R=0 or 1 based on the configuration).
- The setting of the I bit is independent of the static bit associated with the FDB entry, and it is only used with proxy-ARP/ND advertisements.

The I bit in the ARP/ND extended community is processed on reception as follows:

- A PE receiving an EVPN MAC/IP Advertisement route containing an IP-MAC and the I flag set, installs the IP-MAC entry in the ARP/ND or proxy-ARP/ND table as an immutable binding.

- This immutable binding entry overrides an existing non-immutable binding for the same IP-MAC. In general, the ARP/ND extended community command changes the selection of ARP/ND entries when multiple routes with the same IP address exist. This preferred order of ARP/ND entries selection is as follows:

  1. Local immutable ARP/ND entries (static and dynamic)
  2. EVPN immutable ARP/ND entries
  3. Remaining ARP/ND entries

- The absence of the EVPN ARP/ND Extended Community in a MAC/IP Advertisement route indicates that the IP→MAC entry is not an immutable binding.
- Receiving multiple EVPN MAC/IP Advertisement routes with the I flag set to 1 for the same IP but a different MAC address is considered a misconfiguration or a transient error condition. If this happens in the network, a PE receiving multiple routes (with the I flag set to 1 for the same IP and a different MAC address) selects one of them based on the previously described selection rules.

Proxy-ND and the Override Flag

The O bit or Override flag in the ARP/ND extended community is advertised and used as follows:

- The O flag is learned for dynamic entries (being 0 or 1) and added to the proxy-ND table. If the ARP/ND extended community is configured, the O flag associated with the entry is advertised along with the EVPN MAC/IP Advertisement route. Static and duplicate entries are always advertised with O=1.
- Upon receiving an EVPN MAC/IP Advertisement route, the received O flag is stored in the entry created in the proxy-ND table, and used when replying to local NS messages for the IP address.

6.5.1.4 Proxy-ARP/ND mac-List for dynamic entries

SR OS supports the association of configured MAC lists with a configured dynamic proxy-ARP or proxy-ND IP address. The actual proxy-ARP or proxy-ND entry is not created until an ARP or Neighbor Advertisement message is received for the IP and one of the MACs in the associated MAC-list. This is in accordance with IETF RFC 9161, which states that a proxy-ARP or proxy-ND IP entry can be associated with one MAC among a list of allowed MACs.

The following example shows the use of MAC lists for dynamic entries.

```
A:PE-2>config>service#
  proxy-arp-nd
    mac-list ISP-1 create
      mac 00:de:ad:be:ef:01
      mac 00:de:ad:be:ef:02
      mac 00:de:ad:be:ef:03

A:PE-2>config>service>vpls>proxy-arp#
  dynamic 1.1.1.1 create
    mac-list ISP-1
    resolve 30

A:PE-2>config>service>vpls>proxy-nd#
  dynamic 2001:db8:1000::1 create
    mac-list ISP-1
    resolve 30
```

where:

- A dynamic IP (**dynamic ip create**) is configured and associated with a MAC list (**mac-list name**).
- The MAC list is created in the **config**>**service** context and can be reused by multiple configured dynamic IPs as follows:
  - in different services
  - in the same service, for proxy-ARP and proxy-ND entries
- If the MAC list is empty, the proxy-ARP or proxy-ND entry is not created for the configured IP.
- The same MAC list can be applied to multiple configured dynamic entries even within the same service.
- The new proxy-ARP and proxy-ND entries behave as dynamic entries and are displayed as type **dyn** in the **show** commands.

  The following output example displays the entry corresponding to the configured dynamic IP.

```
show service id 1 proxy-arp detail
```

Output example

```
-------------------------------------------------------------------------------
Proxy Arp
-------------------------------------------------------------------------------
Admin State       : enabled
Dyn Populate      : enabled
Age Time          : 900 secs            Send Refresh      : 300 secs
Table Size        : 250                 Total             : 1
Static Count      : 0                   EVPN Count        : 0
Dynamic Count     : 1                   Duplicate Count   : 0
Dup Detect
-------------------------------------------------------------------------------
Detect Window     : 3 mins              Num Moves         : 5
Hold down         : 9 mins
Anti Spoof MAC    : None
EVPN
-------------------------------------------------------------------------------
Garp Flood        : enabled             Req Flood         : enabled
Static Black Hole : disabled
-------------------------------------------------------------------------------
===============================================================================
VPLS Proxy Arp Entries
===============================================================================
IP Address          Mac Address       Type Status    Last Update
-------------------------------------------------------------------------------
1.1.1.1           00:de:ad:be:ef:01  dyn active    02/23/2016 09:05:49
-------------------------------------------------------------------------------
Number of entries : 1
===============================================================================
```

```
show service proxy-arp-nd mac-list "ISP-1" associations
```

Output example

```
===============================================================================
MAC List Associations
===============================================================================
Service Id                    IP Addr
-------------------------------------------------------------------------------
1                             1.1.1.1
1                             2001:db8:1000::1
```

```
-------------------------------------------------------------------------------
Number of Entries: 2
===============================================================================
```

Although no new proxy-ARP or proxy-ND entries are created when a dynamic IP is configured, the router triggers the following resolve procedure:

1. The router sends a resolve message with a configurable frequency of 1 to 60 minutes; the default value is five minutes.

   Note: The resolve message is an ARP-request or NS message flooded to all the non-EVPN endpoints in the service.

2. The router sends resolve messages at the configured frequency until a dynamic entry for the IP is created.

   Note: The dynamic entry is created only if an ARP, GARP, or NA message is received for the configured IP, and the associated MAC belongs to the configured MAC list of the IP. If the MAC list is empty, the proxy-ARP or proxy-ND entry is not created for the configured IP.

After a dynamic entry (with a MAC address included in the list) is successfully created, its behavior (for send-refresh, age-time, and other activities) is the same as a configured dynamic entry with the following exceptions.

- Regular dynamic entries may override configured dynamic entries, but static or EVPN entries cannot override configured dynamic entries.
- If the corresponding MAC is flushed from the FDB after the entry is successfully created, the entry becomes inactive in the proxy-ARP or proxy-ND table and the resolve process is restarted.
- If the MAC list is changed, all the IPs that point to the list delete the proxy entries and the resolve process is restarted.
- If there is an existing configured dynamic entry and the router receives a GARP, ARP, or NA for the IP with a MAC that is not contained in the MAC list, the message is discarded and the proxy-ARP or proxy-ND entry is deleted. The resolve process is restarted.
- If there is an existing configured dynamic entry and the router receives a GARP, ARP, or NA for the IP with a MAC contained in the MAC list, the existing entry is overridden by the IP and new MAC, assuming the confirm procedure passes.
- The dup-detect and confirm procedures work for the configured dynamic entries when the MAC changes are between MACs in the MAC list. Changes to an off-list MAC cause the entry to be deleted and the resolve process is restarted.

Configured **dynamic** entries are advertised as immutable if you enable advertisement of ARP/ND extended community. The following considerations about IP duplication and immutable configured **dynamic** entries apply:

- The CPM drops received dynamic ARP/ND messages without learning them, if they match a dynamic (immutable) entry.
- If there is a local configured dynamic address (irrespective of whether there is an entry for it or not), a received EVPN immutable entry for the same IP address is not installed. Therefore the IP duplication mechanisms do not apply to immutable entries.

6.5.2 BGP-EVPN MAC-mobility

EVPN defines a mechanism to allow the smooth mobility of MAC addresses from an NVE to another NVE. The 7750 SR, 7450 ESS, and 7950 XRS support this procedure as well as the MAC-mobility extended community in MAC advertisement routes as follows:

- The router honors and generates the SEQ (Sequence) number in the MAC mobility extended community for MAC moves.
- When a MAC is EVPN-learned and it is attempted to be learned locally, a BGP update is sent with SEQ number changed to "previous SEQ"+1 (exception: MAC duplication num-moves value is reached).
- SEQ number = zero or no MAC mobility **ext-community** are interpreted as sequence zero.
- In case of mobility, the following MAC selection procedure is followed:
  - If a PE has two or more active remote EVPN routes for the same MAC (VNI can be the same or different), the highest SEQ number is selected. The tie-breaker is the lowest IP (BGP NH IP).
  - If a PE has two or more active EVPN routes and it is the originator of one of them, the highest SEQ number is selected. The tie-breaker is the lowest IP (BGP NH IP of the remote route is compared to the local system address).

Note: When EVPN multihoming is used in EVPN-MPLS, the ESI is compared to determine whether a MAC received from two different PEs has to be processed within the context of MAC mobility or multihoming. Two MAC routes that are associated with the same remote or local ESI but different PEs are considered reachable through all those PEs. Mobility procedures are not triggered as long as the MAC route still belongs to the same ESI.

6.5.3 BGP-EVPN MAC-duplication

EVPN defines a mechanism to protect the EVPN service from control plane churn as a result of loops or accidental duplicated MAC addresses. The 7750 SR, 7450 ESS, and 7950 XRS support an enhanced version of this procedure as described in this section.

A situation may arise where the same MAC address is learned by different PEs in the same VPLS because of two (or more hosts) being misconfigured with the same (duplicate) MAC address. In such situation, the traffic originating from these hosts would trigger continuous MAC moves among the PEs attached to these hosts. It is important to recognize such situation and avoid incrementing the sequence number (in the MAC Mobility attribute) to infinity.

To remedy such situation, a router that detects a MAC mobility event by way of local learning starts a **window <in-minutes>** timer (default value of window = 3) and if it detects **num-moves <num>** before the timer expires (default value of num-moves = 5), it concludes that a duplicate MAC situation has occurred. The router then alerts the operator with a trap message. The offending MAC address can be shown using the **show service id svc-id bgp-evpn** command:

```
10 2014/01/14 01:00:22.91 UTC MINOR: SVCMGR #2331 Base
"VPLS Service 1 has MAC(s) detected as duplicates by EVPN mac-
duplication detection."
# show service id 1 bgp-evpn
===============================================================================
BGP EVPN Table
===============================================================================
MAC Advertisement  : Enabled           Unknown MAC Route  : Disabled
VXLAN Admin Status : Enabled           Creation Origin    : manual
MAC Dup Detn Moves : 5                 MAC Dup Detn Window: 3
```

```
MAC Dup Detn Retry : 9                  Number of Dup MACs : 1
-------------------------------------------------------------------------------
Detected Duplicate MAC Addresses            Time Detected
-------------------------------------------------------------------------------
00:00:00:00:00:12                           01/14/2014 01:00:23
-------------------------------------------------------------------------------
===============================================================================
```

After detecting the duplicate, the router stops sending and processing any BGP MAC advertisement routes for that MAC address until one of the following occurs:

- The MAC is flushed because of a local event (SAP or SDP binding associated with the MAC fails) or the reception of a remote update with better SEQ number (because of a MAC flush at the remote router).
- The retry <**in-minutes**> timer expires, which flushes the MAC and restart the process.

Note: The other routers in the VPLS instance forward the traffic for the duplicate MAC address to the router advertising the best route for the MAC.

The values of **num-moves** and window are configurable to allow for the required flexibility in different environments. In scenarios where BGP rapid-update evpn is configured, the operator may want to configure a shorter window timer than in scenarios where BGP updates are sent every (default) min-route-advertisement interval.

MAC duplication is always enabled in EVPN-VXLAN VPLS services, and the preceding described mac duplication parameters can be configured per VPLS service under the **bgp-evpn mac-duplication** context:

```
*A:DGW1>config>service>vpls>bgp-evpn# info
----------------------------------------------
       mac-advertisement
       unknown-mac-route
       mac-duplication
         detect num-moves num window in_mins
         [no] retry in_mins
       vxlan bgp 1 vxlan-instance 1
         no shutdown
       exit
```

6.5.4 Conditional static MAC and protection

RFC 7432 defines the use of the sticky bit in the MAC mobility extended community to signal static MAC addresses. These addresses must be protected in case there is an attempt to dynamically learn them in a different place in the EVPN-VXLAN VPLS service.

In the 7750 SR, 7450 ESS, and 7950 XRS, any conditional static MAC defined in an EVPN-VXLAN VPLS service is advertised by BGP-EVPN as a static address, that is, with the sticky bit set. An example of the configuration of a conditional static MAC is shown below:

```
*A:PE63>config>service>vpls# info
----------------------------------------------
            description "vxlan-service"
...
            sap 1/1/1:1000 create
            exit
            static-mac
                mac 00:ca:ca:ca:ca:00 create sap 1/1/1:1000 monitor fwd-status
            exit
            no shutdown
```

```
*A:PE64# show router bgp routes evpn mac hunt mac-address 00:ca:ca:ca:ca:00
...
===============================================================================
BGP EVPN Mac Routes
===============================================================================
Network        : 0.0.0.0/0
Nexthop        : 192.0.2.63
From           : 192.0.2.63
Res. Nexthop   : 192.168.19.1
Local Pref.    : 100                    Interface Name : NotAvailable
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : 0
AIGP Metric    : None
Connector      : None
Community      : target:65000:1000     mac-mobility:Seq: 0/Static
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.0.2.63
Flags          : Used  Valid  Best  IGP
Route Source   : Internal
AS-Path        : No As-Path
EVPN type      : MAC
ESI            : 0:0:0:0:0:0:0:0:0:0    Tag            : 1063
IP Address     : ::                     RD             : 65063:1000
Mac Address    : 00:ca:ca:ca:ca:00      Mac Mobility   : Seq:0
Neighbor-AS    : N/A
Source Class   : 0                      Dest Class     : 0
-------------------------------------------------------------------------------
Routes : 1
===============================================================================
```

Local static MACs or remote MACs with sticky bit are considered as "protected". A packet entering a SAP / SDP binding is discarded if its source MAC address matches one of these 'protected' MACs.

6.5.5 Auto-learn MAC protect and restricting protected source MACs

Auto-learn MAC protect, together with the ability to restrict where the protected source MACs are allowed to enter the service, can be enabled within an EVPN-MPLS and EVPN-VXLAN VPLS and routed VPLS services, but not in PBB-EVPN services. The protection, using the **auto-learn-mac-protect** command (described in Auto-learn MAC protect), and the restrictions, using the **restrict-protected-src [discard-frame]** command, operate in the same way as in a non-EVPN VPLS service.

- When **auto-learn-mac-protect** is enabled on an object, source MAC addresses learned on that object are marked as protected within the FDB.
- When **restrict-protected-src** is enabled on an object and a protected source MAC is received on that object, the object is automatically shutdown (requiring the operator to **shutdown** then **no shutdown** the object to make it operational again).
- When **restrict-protected-src discard-frame** is enabled on an object and a frame with a protected source MAC is received on that object, that frame is discarded.

In addition, the following behavioral differences are specific to EVPN services:

- An implicit **restrict-protected-src discard-frame** command is enabled by default on SAPs, mesh-SDPs and spoke SDPs. As this is the default, it is not possible to configure this command in an EVPN service. This default state can be seen in the show output for these objects, for example on a SAP:

```
*A:PE# show service id 1 sap 1/1/9:1 detail
===============================================================================
```

```
Service Access Points(SAP)
===============================================================================
Service Id         : 1
SAP                : 1/1/9:1                 Encap            : q-tag
...
RestMacProtSrc Act : none (oper: Discard-frame)
```

- A **restrict-protected-src discard-frame** can be optionally enabled on EVPN-MPLS/VXLAN destinations within EVPN services. When enabled, frames that have a protected source MAC address are discarded if received on any EVPN-MPLS/VXLAN destination in this service, unless the MAC address is learned and protected on an EVPN-MPLS/VXLAN destination in this service. This is enabled as follows:

```
        configure
           service
              vpls <service id>
                   bgp-evpn
                       mpls bgp <instance>
                           [no] restrict-protected-src discard-frame
                   vxlan instance <instance> vni <vni-id>
                       [no] restrict-protected-src discard-frame
```

- Auto-learned protected MACs are advertised to remote PEs in an EVPN MAC/IP advertisement route with the sticky bit set.
- The source MAC protection action relating to the **restrict-protected-src [discard-frame]** commands also applies to MAC addresses learned by receiving an EVPN MAC/IP advertisement route with the sticky bit set from remote PEs. This causes remotely configured conditional static MACs and auto-learned protected MACs to be protected locally.
- In all-active multihoming scenarios, if **auto-learn-mac-protect** is configured on all-active SAPs and **restrict-protected-src discard-frame** is enabled on EVPN-MPLS/VXLAN destinations, traffic from the CE that enters one multihoming PE and needs to be switched through the other multihoming PE is discarded on the second multihoming PE. Each multihoming PE protects the CE's MAC on its local all-active SAP, which results in any frames with the CE's MAC address as the source MAC being discarded as they are received on the EVPN-MPLS/VXLAN destination from the other multihoming PE.

Conditional static MACs, EVPN static MACs and locally protected MACs are marked as protected within the FDB, as shown in the example output.

```
*A:PE# show service fdb-mac
===============================================================================
Service Forwarding Database
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:00:00:00:00:01 sap:1/1/9:1              LP/30    01/05/16 11:58:22
1         00:00:00:00:00:02 vxlan-1:                 EvpnS:P  01/05/16 11:58:23
                            10.1.1.2:1
1         00:00:00:00:01:01 sap:1/1/9:1              CStatic: 01/04/16 20:05:02
                                                     P
1         00:00:00:00:01:02 vxlan-1:                 EvpnS:P  01/04/16 20:18:02
                            10.1.1.2:1
-------------------------------------------------------------------------------
No. of Entries: 4
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
```

```
===============================================================================
```

In this output:

- the first MAC is locally protected using the **auto-learn-mac-protect** command
- the second MAC has been protected using the **auto-learn-mac-protect** command on a remote PE
- the third MAC is a locally configured conditional static MAC
- the fourth MAC is a remotely configured conditional static MAC

The command **auto-learn-mac-protect** can be optionally extended with an exclude-list by using the following command:

auto-learn-mac-protect [**exclude-list name**]

This list refers to a **mac-list** <**name**> created under the **config**>**service** context and contains a list of MACs and associated masks.

When **auto-learn-mac-protect** [**exclude-list name**] is configured on a service object, dynamically learned MACs are excluded from being learned as protected if they match a MAC entry in the MAC list. Dynamically learned MAC SAs are protected only if they are learned on an object with ALMP configured and one of the following conditions is true:

- there is no exclude list associated with the same object
- there is an exclude-list but the MAC does not match any entry

The MAC lists can be used in multiple objects of the same or different service. When empty, ALMP does not exclude any learned MAC from protection on the object. This extension allows the mobility of specific MACs in objects where MACs are learned as protected.

6.5.6 Blackhole MAC and its application to proxy-ARP/proxy-ND duplicate detection

A blackhole MAC is a local FDB record. It is similar to a conditional static MAC; it is associated with a **black-hole** (similar to a VPRN blackhole static-route in VPRNs) instead of a SAP or SDP binding. A blackhole MAC can be added by using the following command:

```
config>service>vpls# static-mac mac
mac <ieee-address> [create] black-hole
```

The static blackhole MAC can have security applications (for example, replacement of MAC filters) for specific MACs. When used in combination with **restrict-protected-src**, the static blackhole MAC provides a simple and scalable way to filter MAC DA or SA in the data plane, regardless of how the frame arrived at the system (using SAP or SDP bindings or EVPN endpoints).

For example, when a specified **static-mac mac 00:00:ca:fe:ca:fe create black-hole** is added to a service, the following behavior occurs:

1. The configured MAC is created as a static MAC with a **black-hole** source identifier.

```
*A:PE1# show service id 1 fdb detail
===============================================================================
Forwarding Database, Service 1
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
1         00:ca:ca:ba:ca:01 eES:                     Evpn     06/29/15 23:21:34
```

```
                            01:00:00:00:00:71:00:00:00:01
1         00:ca:ca:ba:ca:06 eES:                      Evpn     06/29/15 23:21:34
                            01:74:13:00:74:13:00:00:74:13
1         00:ca:00:00:00:00 sap:1/1/1:2               CStatic:P 06/29/15 23:20:58
1         00:ca:fe:ca:fe:00 black-hole                CStatic:P  06/29/15 23:20:00
1         00:ca:fe:ca:fe:69 eMpls:                    EvpnS:P    06/29/15 20:40:13
                            192.0.2.69:262133
-------------------------------------------------------------------------------
No. of MAC Entries: 5
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
===============================================================================
```

2. After it has been successfully added to the FDB, the blackhole MAC is treated like any other protected MAC, as follows:
   - The blackhole MAC is added as protected (**CStatic:P**) and advertised in EVPN as static.
   - SAP or SDP bindings or EVPN endpoints, where the **restrict-protected-src discard-frame** is enabled, discard frames where MAC SA is equal to blackhole MAC.
   - SAP or SDP bindings, where **restrict-protected-src** (**no discard-frame**) is enabled, go operationally down if a frame with MAC SA is equal to blackhole MAC is received.
3. After the blackhole MAC has been successfully added to the FDB, any frame arriving at any SAP or SDP binding or EVPN endpoint with MAC DA equal to blackhole MAC is discarded.

Blackhole MACs can also be used in services with **proxy-ARP**/**proxy-ND** enabled to filter traffic with destination to **anti-spoof-macs**. The **anti-spoof-mac** provides a way to attract traffic to a specified IP when a duplicate condition is detected for that IP address (see section ARP/ND snooping and proxy support for more information); however, the system still needs to drop the traffic addressed to the **anti-spoof-mac** by using either a MAC filter or a blackhole MAC.

The user does not need to configure MAC filters when configuring a **static-black-hole** MAC address for the **anti-spoof-mac** function. To use a blackhole MAC entry for the **anti-spoof-mac** function in a proxy-ARP/proxy-ND service, the user needs to configure:

- the **static-black-hole** option for the **anti-spoof-mac**

```
*A:PE1# config>service>vpls>proxy-arp#
dup-detect window 3 num-moves 5 hold-down max anti-spoof-
mac 00:66:66:66:66:00 static-black-hole
```

- a static blackhole MAC using the same MAC address used for the **anti-spoof-mac**

```
*A:PE1# config>service>vpls#
static-mac mac 00:66:66:66:66:00 create black-hole
```

When this configuration is complete, the behavior of the **anti-spoof-mac** function changes as follows:

- In the EVPN, the MAC is advertised as static. Locally, the MAC is shown in the FDB as "CStatic" and associated with a **black-hole**.
- The combination of the **anti-spoof-mac** and the **static-black-hole** ensures that any frame that arrives at the system with MAC DA = **anti-spoof-mac** is discarded, regardless of the ingress endpoint type (SAP or SDP binding or EVPN) and without the need for a filter.
- If, instead of discarding traffic, the user wants to redirect it using MAC DA as the **anti-spoof-mac**, then redirect filters should be configured on SAPs or SDP bindings (instead of the **static-black-hole** option).

When the **static-black-hole** option is not configured with the **anti-spoof-mac**, the behavior of the **anti-spoof-mac** function, as described in ARP/ND snooping and proxy support, remains unchanged. In particular:

- the **anti-spoof-mac** is not programmed in the FDB
- any attempt to add a static MAC (or any other MAC) with the **anti-spoof-mac** value is rejected by the system
- a MAC filter is needed to discard traffic with MAC DA = **anti-spoof-mac**.

6.5.7 Blackhole MAC for EVPN loop detection

SR OS can combine a blackhole MAC address concept and the EVPN MAC duplication procedures to provide loop protection in EVPN networks. The feature is compliant with the MAC mobility and multihoming functionality in RFC 7432, and the Loop Protection section in *draft-ietf-bess-rfc7432bis*. The **config**>**service**>**vpls**>**bgp-evpn**>**mac-duplication**>**black-hole-dup-mac** command enables the feature.

If enabled, there are no apparent changes in the MAC duplication; however, if a duplicated MAC is detected (for example, M1), then the router performs the following:

1. adds M1 to the duplicate MAC list
2. programs M1 in the FDB as a "Protected" MAC associated with a blackhole endpoint (where "type" is set to **EvpnD:P** and Source-Identifier is **black-hole**)

While the MAC type value remains **EvpnD:P**, the following additional operational details apply.

- Incoming frames with MAC DA = M1 are discarded by the ingress IOM, regardless of the ingress endpoint type (SAP, SDP, or EVPN), based on an FDB MAC lookup.
- Incoming frames with MAC SA = M1 are discarded by the ingress IOM or cause the router to bring down the SAP or SDP binding, depending on the **restrict-protected-src** setting on the SAP, SDP, or EVPN endpoint.

The following example shows an EVPN-MPLS service where **black-hole-dup-mac** is enabled and MAC duplication programs the duplicate MAC as a blackhole.

```
19 2016/12/20 19:45:59.69 UTC MINOR: SVCMGR #2331 Base
"VPLS Service 30 has MAC(s) detected as duplicates by EVPN mac-duplication
detection."
*A:PE-5# configure service vpls 30
*A:PE-5>config>service>vpls# info
----------------------------------------------
            bgp
            exit
            bgp-evpn
                evi 30
                mac-duplication
                    detect num-moves 3 window 3
                    retry 6
                    black-hole-dup-mac
                exit
                mpls bgp 1
                    ingress-replication-bum-label
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
```

```
            stp
                shutdown
            exit
            sap 1/1/1:30 create
                no shutdown
            exit
            spoke-sdp 56:30 leaf-ac create
                no shutdown
            exit
            no shutdown
-------------------------------------------
*A:PE-5# show service id 30 bgp-evpn
===============================================================================
BGP EVPN Table
===============================================================================
MAC Advertisement  : Enabled           Unknown MAC Route  : Disabled
CFM MAC Advertise  : Disabled
VXLAN Admin Status : Disabled          Creation Origin    : manual
MAC Dup Detn Moves : 3                 MAC Dup Detn Window: 3
MAC Dup Detn Retry : 6                 Number of Dup MACs : 1
MAC Dup Detn BH    : Enabled
IP Route Advert    : Disabled

EVI                : 30
Ing Rep Inc McastAd: Enabled
Accept IVPLS Flush : Disabled
Send EVPN Encap    : Enabled
-------------------------------------------------------------------------------
Detected Duplicate MAC Addresses            Time Detected
-------------------------------------------------------------------------------
00:11:00:00:00:01                           12/20/2016 19:46:00
-------------------------------------------------------------------------------
<snip>
...
*A:PE-5# show service id 30 fdb detail
===============================================================================
Forwarding Database, Service 30
===============================================================================
ServId    MAC               Source-Identifier        Type     Last Change
                                                     Age
-------------------------------------------------------------------------------
30        00:11:00:00:00:01 black-hole               EvpnD:P  12/20/16 19:46:00
-------------------------------------------------------------------------------
No. of MAC Entries: 1
-------------------------------------------------------------------------------
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static Lf=Leaf
===============================================================================
```

If the **retry** time expires, the MAC is flushed from the FDB and the process starts again. The **clear service id 30 evpn mac-dup-detect** {**ieee-address** | **all**} command clears the duplicate blackhole MAC address.

Note: The **clear service id 30 fdb** command clears learned MAC addresses; blackhole MAC addresses are not cleared.

Support for the **black-hole-dup-mac** command and the preceding associated loop detection procedures is as follows:

- not supported on B-VPLS, I-VPLS, or M-VPLS services
- fully supported on EVPN-VXLAN and EVPN-MPLS VPLS services (including EVPN E-Tree)
- fully supported with EVPN MAC mobility and EVPN-MPLS multihoming

6.5.8 CFM interaction with EVPN services

Ethernet Connectivity and Fault Management (ETH-CFM) allows the operator to validate and measure Ethernet Layer 2 services using standard IEEE 802.1ag and ITU-T Y.1731 protocols. Each tool performs a unique function and adheres to that tool's specific PDU and frame format and the associate rules governing the transmission, interception, and process of the PDU. Detailed information describing the ETH-CFM architecture, the tools, and various functions is located in the various OAM and Diagnostics guides and is not repeated here.

EVPN provides powerful solution architectures. ETH-CFM is supported in the various Layer 2 EVPN architectures. Because the destination Layer 2 MAC address, unicast or multicast, is ETH-CFM tool dependent (for example, ETH-CC is sent as an L2 multicast and ETH-DM is sent as an L2 unicast), the ETH-CFM function is allowed to multicast and broadcast to the virtual EVPN connections. The Maintenance Endpoint (MEP) and Maintenance Intermediate Point (MIP) do not populate the local Layer 2 MAC Address forwarding database (FDB) with the MAC related to the MEP and MIP. This means that the 48-bit IEEE MAC address is not exchanged with peers and all ETH-CFM frames are broadcast across all virtual connections. To prevent the flooding of unicast packets and allow the remote forwarding databases to learn the remote MEP and MIP Layer 2 MAC addresses, the command **cfm-mac-advertisement** must be configured under the **config**>**service**>**vpls**>**bgp-evpn** context. This allows the MEP and MIP Layer 2 IEEE MAC addresses to be exchanged with peers. This command tracks configuration changes and send the required updates via the EVPN notification process related to a change.

Up MEP, Down MEP, and MIP creation is supported on the SAP, spoke, and mesh connections within the EVPN service. There is no support for the creation of ETH-CFM Management Points (MPs) on the virtual connection. VirtualMEP (vMEP) is supported with a VPLS context and the applicable EVPN Layer 2 VPLS solution architectures. The vMEP follows the same rules as the general MPs. When a vMEP is configured within the supported EVPN service, the ETH-CFM extraction routines are installed on the SAP, Binding, and EVPN connections within an EVPN VPLS Service. The vMEP extraction within the EVPN-PBB context requires the **vmep-extensions** parameter to install the extraction on the EVPN connections.

When MPs are used in combination with EVPN multihoming, the following must be considered:

- Behavior of operationally down MEPs on SAPs/SDP bindings with EVPN multihoming:
  - **all-active multihoming**

    No ETH-CFM is expected to be used in this case, because the two (or more) SAPs/SDP bindings on the PEs are oper-up and active; however, the CE has a single LAG and responds as though it is connected to a single system. In addition to that, **cfm-mac-advertisement** can lead to traffic loops in all-active multihoming.
  - **single-active multihoming**

    Operationally down MEPs defined on single-active Ethernet-Segment SAPs/SDP bindings do not send any CCMs when the PE is non-DF for the ES and fault-propagation is configured. For single-active multihoming, the behavior is equivalent to MEPs defined on BGP-MH SAPs/binds.
- Behavior for operationally up MEPs on ES SAPs/SDP bindings with EVPN multihoming:
  - **all-active multihoming**

    Operationally up MEPs defined on non-DF ES SAPs can send CFM packets. However, they cannot receive CCMs (the SAP is removed from the default multicast list) or unicast CFM packets (because the MEP MAC is not installed locally in the FDB; unicast CFM packets are treated as unknown, and not sent to the non-DF SAP MEP).
  - **single-active multihoming**

Operationally up MEPs should be able to send or receive CFM packets normally.

– **operationally up MEPs defined on LAG SAPs**

Operationally up MEPs defined on LAG SAPs require the command process_cpm_traffic_on_sap_down so that they can process CFM when the LAG is down and act as regular Ethernet ports.

Because of the above considerations, the use of ETH-CFM in EVPN multihomed SAPs/SDP bindings is only recommended on operationally down MEPs and single-active multihoming. ETH-CFM is used in this case to notify the CE of the DF or non-DF status.

6.5.9 Multi-Instance EVPN: Two instances of different encapsulation in the same VPLS/ R-VPLS service

SR OS supports a maximum of two BGP instances in the same VPLS or R-VPLS, where the two instances can be:

- One EVPN-VXLAN instance and one EVPN-MPLS instance in the same VPLS or R-VPLS service
- Two EVPN-VXLAN instances in the same VPLS or R-VPLS service
- Two EVPN-MPLS instances in the same VPLS or R-VPLS service
- One EVPN-MPLS instance and one EVPN-SRv6 instance in the same VPLS service
- One EVPN-VPLS instance and one EVPN-SRv6 instance in the same VPLS service

6.5.9.1 EVPN-VXLAN to EVPN-MPLS interworking

This section describes the configuration aspects of a VPLS/R-VPLS with EVPN-VXLAN and EVPN-MPLS.

In a service where EVPN-VXLAN and EVPN-MPLS are configured together, the **configure service vpls bgp-evpn vxlan bgp 1** and **configure service vpls bgp-evpn mpls bgp 2** commands allow the user to associate EVPN-MPLS to a different instance from that associated with EVPN-VXLAN, and have both encapsulations simultaneously enabled in the same service. At the control plane level, EVPN MAC/IP advertisement routes received in one instance are consumed and readvertised in the other instance as long as the route is the best route for a specific MAC. Inclusive multicast routes are independently generated for each BGP instance. In the data plane, the EVPN-MPLS and EVPN-VXLAN destinations are instantiated in different implicit Split Horizon Groups (SHGs) so that traffic can be forwarded between them.

The following example shows a VPLS service with two BGP instances and both VXLAN and MPLS encapsulations configured for the same BGP-EVPN service.

```
*A:PE-1>config>service>vpls# info
----------------------------------------------
  description "evpn-mpls and evpn-vxlan in the same service"
  vxlan instance 1 vni 7000 create
  exit
  bgp
    route-distinguisher 10:2
    route-target target:64500:1
  exit
  bgp 2
    route-distinguisher 10:1
    route-target target:64500:1
  exit
```

```
bgp-evpn
  evi 7000
  incl-mcast-orig-ip 10.12.12.12
  vxlan bgp 1 vxlan-instance 1
    no shutdown
  mpls bgp 2
    control-word
    auto-bind-tunnel
      resolution any
    exit
    force-vlan-vc-forwarding
    no shutdown
  exit
exit
no shutdown
```

The following list describes the preceding example:

- **bgp 1** or **bgp** is the default BGP instance
- **bgp 2** is the additional instance required when both **bgp-evpn vxlan** and **bgp-evpn mpls** are enabled in the service
- The commands supported in instance 1 are also available in instance 2 with the following considerations:
  - **pw-template-binding**

    The pw-template-binding can only exist in instance 1; it is not supported in instance 2.
  - **route-distinguisher**

    The operating route-distinguisher in both BGP instances must be different.
  - **route-target**

    The route target in both instances can be the same or different.
  - **vsi-import and vsi-export**

    Import and export policies can also be defined for either BGP instance.
- MPLS and VXLAN can use either BGP instance, and the instance is associated when **bgp-evpn mpls** or **bgp-evpn vxlan** is created. The **bgp-evpn vxlan** command must include not only the association to a BGP instance, but also to a **vxlan-instance** (because the VPLS services support two VXLAN instances).

Note: The **bgp-evpn vxlan no shutdown** command is only allowed if **bgp-evpn mpls shutdown** is configured, or if the BGP instance associated with the MPLS has a different route distinguisher than the VXLAN instance.

The following features are not supported when two BGP instances are enabled on the same VPLS/R-VPLS service:

- SDP bindings
- M-VPLS, I-VPLS, B-VPLS, or E-Tree VPLS
- Proxy-ARP and proxy-ND
- BGP Multihoming
- IGMP, MLD, and PIM snooping
- BGP-VPLS or BGP-AD (SDP bindings are not created)

The **service>vpls>bgp-evpn>ip-route-advertisement** command is not supported on R-VPLS services with two BGP instances.

6.5.9.2 EVPN-SRv6 to EVPN-MPLS or EVPN-VXLAN interworking

EVPN-SRv6 and EVPN-MPLS or EVPN-VXLAN can be simultaneously configured in the same VPLS service (but not R-VPLS), in different instances.

EVPN-SRv6 and EVPN-VXLAN instances in the same VPLS service follow the same configuration rules as described in EVPN-VXLAN to EVPN-MPLS interworking, and the same processing of MAC/IP Advertisement routes and Inclusive Multicast Ethernet Tag routes is applied.

The following example shows a VPLS service with two BGP instances, with both VXLAN and SRv6 encapsulations configured under BGP-EVPN.

Example: MD-CLI

```
A:node-2>config>service>vpls "evpn-srv6-vxlan-1"> info
    admin-state enable
    description "evpn-srv6 and evpn-vxlan in the same service"
    vxlan {
        instance 1 {
            vni 12340
        }
    }
    segment-routing-v6 1 {
        locator "loc-1" {
            function {
                end-dt2u {
                }
                end-dt2m {
                }
            }
        }
    }
    bgp 1 {
        route-distinguisher "12340:1"
        route-target {
            export "target:64500:12340"
            import "target:64500:12340"
        }
    }
    bgp 2 {
        route-distinguisher "12340:2"
        route-target {
            export "target:64500:12341"
            import "target:64500:12341"
        }
    }
    bgp-evpn {
        evi 12340
        incl-mcast-orig-ip 10.12.12.12
        segment-routing-v6 2 {
            admin-state enable
            ecmp 4
            force-vc-forwarding vlan
            srv6 {
                default-locator "loc-1"
            }
        }
        vxlan 1 {
```

```
            admin-state enable
            vxlan-instance 1
        }
    }
```

Example: classic CLI

```
A:node-2>config>service>vpls# info
----------------------------------------------
            description "evpn-srv6 and evpn-vxlan in the same service"
            vxlan instance 1 vni 12340 create
            exit
            segment-routing-v6 1 create
                locator "loc-1"
                    function
                        end-dt2u
                        end-dt2m
                    exit
                exit
            exit
            bgp
                route-distinguisher 12340:1
                route-target export target:64500:12340 import target:64500:12340
            exit
            bgp 2
                route-distinguisher 12340:2
                route-target export target:64500:12341 import target:64500:12341
            exit
            bgp-evpn
                incl-mcast-orig-ip 10.12.12.12
                evi 12340
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
                segment-routing-v6 bgp 2 srv6-instance 1 default-locator "loc-1" create
                    ecmp 4
                    force-vlan-vc-forwarding
                    route-next-hop 2001:db8::76
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
----------------------------------------------
```

When an EVPN-SRv6 instance and an EVPN-MPLS instance are both configured in the same VPLS service, each instance can be configured in a different or the same split horizon group. The former option allows the interconnection of domains of different encapsulation, and the rules of configuration and route processing described in EVPN-VXLAN to EVPN-MPLS interworking apply. The latter option is used for domains where MPLS and SRv6 PEs are attached to the same service, typically for migration purposes.

When the EVPN-SRv6 and the EVPN-MPLS instances are configured in the same split horizon group:

- MAC/IP Advertisement routes are not redistributed between the two instances
- Two BUM EVPN destinations to the same far-end PE (identified by the originator IP of the Inclusive Multicast Ethernet Tag routes) cannot be created. An EVPN-MPLS BUM destination is

removed if there is another BUM destination to the same far end with an SRv6 encapsulation. This is to prevent BUM traffic duplication between multi-instance nodes

- SAPs are supported, but SDP binds are not supported

The following example shows a VPLS service with two BGP instances, with both MPLS and SRv6 encapsulations configured under BGP-EVPN, with the same split horizon group.

Example: MD-CLI

```
configure service vpls "evpn-srv6-mpls-1" >info
admin-state enable
    description "evpn-srv6 and evpn-mpls in the same service"
    segment-routing-v6 1 {
        locator "loc-1" {
            function {
                end-dt2u {
                }
                end-dt2m {
                }
            }
        }
    }
    bgp 1 {
        route-distinguisher "12341:1"
        route-target {
            export "target:64500:12342"
            import "target:64500:12342"
        }
    }
    bgp 2 {
        route-distinguisher "12341:2"
        route-target {
            export "target:64500:12343"
            import "target:64500:12343"
        }
    }
    bgp-evpn {
        evi 12340
        incl-mcast-orig-ip 10.12.12.12
        segment-routing-v6 2 {
            admin-state enable
            ecmp 4
            force-vc-forwarding vlan
            srv6 {
                default-locator "loc-1"
            }
            route-next-hop {
                ip-address 2001:db8::76
            }
        }
        mpls 1 {
            admin-state enable
            force-vc-forwarding vlan
            split-horizon-group "SHG-1"
            ingress-replication-bum-label true
            ecmp 4
            mh-mode access
            auto-bind-tunnel {
                resolution any
            }
        }
    }
    split-horizon-group "SHG-1" {
```

```
    }
```

Example: classic CLI

```
A:node-2>config>service>vpls# info
----------------------------------------------
            description "evpn-srv6 and evpn-mpls in the same service"
            split-horizon-group "SHG-1" create
            exit
            segment-routing-v6 1 create
                locator "loc-1"
                    function
                        end-dt2u
                        end-dt2m
                    exit
                exit
            exit
            bgp
                route-distinguisher 12341:1
                route-target export target:64500:12342 import target:64500:12342
            exit
            bgp 2
                route-distinguisher 12341:2
                route-target export target:64500:12343 import target:64500:12343
            exit
            bgp-evpn
                incl-mcast-orig-ip 10.12.12.12
                evi 12341
                mpls bgp 1
                    mh-mode access
                    force-vlan-vc-forwarding
                    split-horizon-group "SHG-1"
                    ingress-replication-bum-label
                    ecmp 4
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
                segment-routing-v6 bgp 2 srv6-instance 1 default-locator "loc-1" create
                    ecmp 4
                    force-vlan-vc-forwarding
                    route-next-hop 2001:db8::76
                    split-horizon-group "SHG-1"
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
----------------------------------------------
```

6.5.9.3 BGP-EVPN routes in services configured with two BGP instances

From a BGP perspective, the two BGP instances configured in the service are independent of each other. The redistribution of routes between the BGP instances is resolved at the EVPN application layer.

By default, if EVPN-VXLAN and EVPN-MPLS are both enabled in the same service, BGP sends the generated EVPN routes twice: with the RFC 9012 BGP encapsulation extended community set to VXLAN and a second time with the encapsulation type set to MPLS.

Usually, a DCGW peers a pair of Route Reflectors (RRs) in the DC and a pair of RRs in the WAN. For this reason, the user needs to add router policies so that EVPN-MPLS routes are only sent to the WAN RRs and EVPN-VXLAN routes are only sent to the DC RRs. The following examples show how to configure router policies.

```
config>router>bgp#
 vpn-apply-import
 vpn-apply-export
 group "WAN"
  family evpn
  type internal
  export "allow only mpls"
  neighbor 192.0.2.6
 group "DC"
  family evpn
  type internal
  export "allow only vxlan"
  neighbor 192.0.2.2
config>router>policy-options# info
----------------------------------------------
         community "vxlan" members "bgp-tunnel-encap:VXLAN"
         community "mpls" members "bgp-tunnel-encap:MPLS"
            policy-statement "allow only mpls"
                entry 10
                  from
                    family evpn
                    community vxlan
                  action drop
                  exit
                exit
            exit
            policy-statement "allow only vxlan"
                entry 10
             from
              family evpn
              community mpls
                    action drop
                    exit
                exit
            exit
```

In a BGP instance, the EVPN routes are imported based on the route-targets and regular BGP selection procedures, regardless of their encapsulation.

The BGP-EVPN routes are generated and redistributed between BGP instances based on the following rules:

- Auto-discovery (AD) routes (type 1) are not generated by services with two BGP EVPN instances, unless a local Ethernet segment is present on the service. However, AD routes received from the EVPN-MPLS peers are processed for aliasing and backup functions as usual.
- MAC/IP routes (type 2) received in one of the two BGP instances are imported and the MACs added to the FDB according to the existing selection rules. If the MAC is installed in the FDB, it is readvertised in the other BGP instance with the new BGP attributes corresponding to the BGP instance (route target, route distinguisher, and so on). The following considerations apply to these routes:

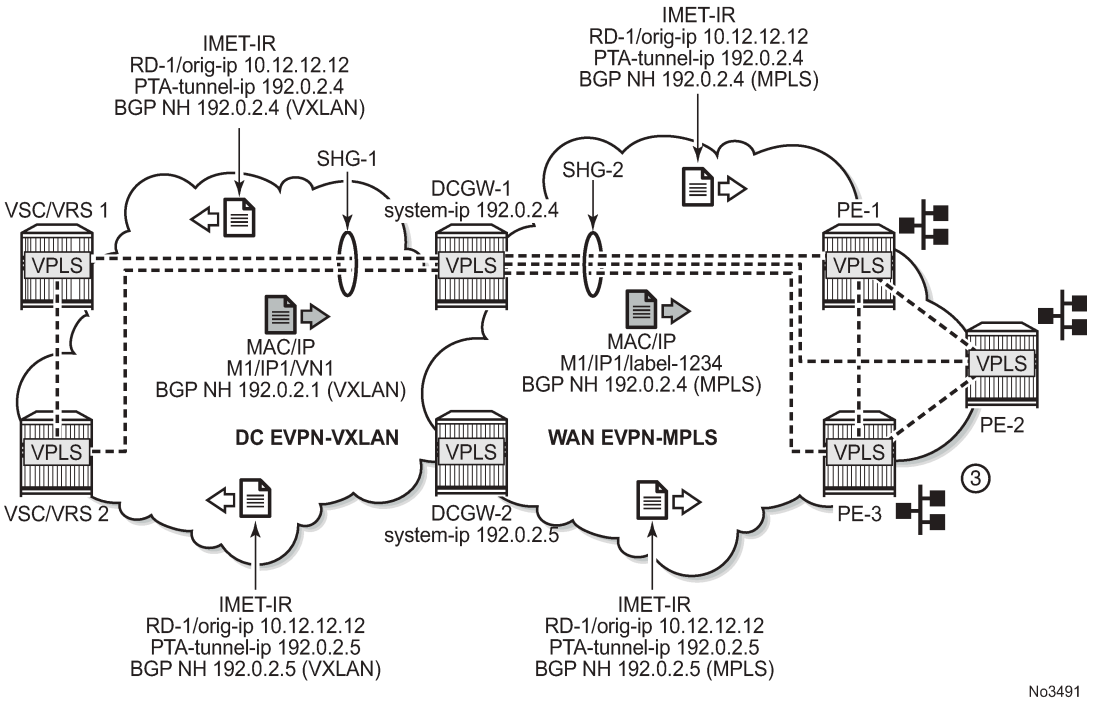
  - The **mac-advertisement** command governs the advertisement of any MACs (even those learned from BGP).
  - A MAC route is redistributed only if it is the best route based on the EVPN selection rules.
  - If a MAC route is the best route and has to be redistributed, the MAC/IP information, along with the MAC mobility extended community, is propagated in the redistribution.
  - The router redistributes any MAC route update for which any attribute has changed. For example, a change in the SEQ or sticky bit in one instance is updated in the other instance for a route that is selected as the best MAC route.
- EVPN inclusive multicast routes are generated independently for each BGP instance with the corresponding BGP encapsulation extended community (VXLAN or MPLS). Also, the following considerations apply to these routes:
  - Ingress Replication (IR) and Assisted Replication (AR) routes are supported in the EVPN-VXLAN instance. If AR is configured, the AR IP address must be a loopback address different from the **system-ip** and the configured **originating-ip** address.
  - IR, P2MP mLDP, and composite inclusive multicast routes are supported in the EVPN-MPLS instance.
  - The modification of the **incl-mcast-orig-ip** command is supported, subject to the following considerations:
    - The configured IP in the **incl-mcast-orig-ip** command is encoded in the **originating-ip** field of the inclusive multicast Routes for IR, P2MP, and composite tunnels.
    - The **originating-ip** field of the AR routes is still derived from the **service>system>vxlan>assisted-replication-ip** configured value.
  - EVPN handles the inclusive multicast routes in a service based on the following rules:
    - For IR routes, the EVPN destination is set up based on the NLRI next hop.
    - For P2MP mLDP routes, the PMSI Tunnel Attribute **tunnel-id** is used to join the mLDP tree.
    - For composite P2MP-IR routes, the PMSI Tunnel Attribute **tunnel-id** is used to join the tree and create the P2MP bind. The NLRI next-hop is used to build the IR destination.
    - For AR routes, the NLRI next-hop is used to build the destination.
    - The following applies if a router receives two inclusive multicast routes in the same instance:
      - If the routes have the same **originating-ip** but different route distinguishers and next-hops, the router processes both routes. In the case of IR routes, it sets up two destinations: one to each next-hop.
      - If the routes have the same **originating-ip**, different route distinguishers, but same next hops, the router sets up only one binding for IR routes.
      - The router ignores inclusive multicast routes received with its own **originating-ip**, regardless of the route distinguisher.
- IP-Prefix routes (type 5) are not generated or imported by a service with two BGP instances.

The rules in this section can be extrapolated to VPLS services where SRv6 and MPLS or VXLAN are configured in different instances of the same VPLS with different split horizon groups.

6.5.9.4 Anycast redundant solution for dual BGP-instance services

Figure 195: Multihomed anycast solution shows the anycast mechanism used to support gateway redundancy for dual BGP-instance services. The example shows two redundant DC gateways (DCGWs) where the VPLS services contain two BGP instances: one each for EVPN-VXLAN and EVPN-MPLS.

Figure 195: Multihomed anycast solution



The example shown in Figure 195: Multihomed anycast solution depends on the ability of the two DCGWs to send the same inclusive multicast route to the remote PE or NVEs, such that:

- The remote PE or NVEs create a single BUM destination to one of the DCGWs (because the BGP selects only the best route to the DCGWs).
- The DCGWs do not create a destination between each other.

This solution avoids loops for BUM traffic, and known unicast traffic can use either DCGW router, depending on the BGP selection. The following CLI example output shows the configuration of each DCGW.

```
/* bgp configuration on DCGW1 and DCGW2 */
config>router>bgp#
 group "WAN"
  family evpn
  type internal
  neighbor 192.0.2.6
 group "DC"
  family evpn
  type internal
  neighbor 192.0.2.2
/* vpls service configuration */
DCGW-1# config>service>vpls(1)#
-----------------------
bgp
```

```
  route-distinguisher 64501:12
  route-target target:64500:1
exit
bgp 2
  route-distinguisher 64502:12
  route-target target:64500:1
exit
vxlan instance 1 vni 1 create
exit
bgp-evpn
  evi 1
  incl-mcast-orig-ip 10.12.12.12
  vxlan bgp 1 vxlan-instance 1
   no shutdown
  mpls bgp 2
   no shutdown
   auto-bind-tunnel
     resolution any
   exit
<snip>
DCGW-2# config>service>vpls(1)#
----------------------
bgp
  route-distinguisher 64501:12
  route-target target:64500:1
exit
bgp 2
  route-distinguisher 64502:12
  route-target target:64500:1
exit
vxlan instance 1 vni 1 create
exit
bgp-evpn
  evi 1
  incl-mcast-orig-ip 10.12.12.12
  vxlan bgp 1 vxlan-instance 1
   no shutdown
  mpls bgp 2
   no shutdown
   auto-bind-tunnel
     resolution any
<snip>
```

Based on the preceding configuration example, the behavior of the DCGWs in this scenario is as follows:

- DCGW-1 and DCGW-2 send inclusive multicast routes to the DC RR and WAN RR with the same route key. For example:
  - DCGW-1 and DCGW-2 both send an IR route to DC RR with RD=64501:12, orig-ip=10.12.12.12, and a different next-hop and tunnel ID
  - DCGW-1 and DCGW-2 both send an IR route to WAN RR with RD=64502:12, orig-ip=10.12.12.12, and different next-hop and tunnel ID
- DCGW-1 and DCGW-2 both receive MAC/IP routes from DC and WAN that are redistributed to the other BGP instances, assuming that the route is selected as best route and the MAC is installed in the FDB.

  As described in section BGP-EVPN routes in services configured with two BGP instances, router peer policies are required so that only VXLAN or MPLS routes are sent or received for a specific peer.
- Configuration of the same **incl-mcast-orig-ip** address in both DCGWs enables the anycast solution for BUM traffic for all the following reasons:

– The configured **originating-ip** is not required to be a reachable IP address and this forces the remote PE or NVEs to select only one of the two DCGWs.
– The BGP next-hops are allowed to be the **system-ip** or even a loopback address. In both cases, the BGP next-hops are not required to be reachable in their respective networks.

In the example shown in Figure 195: Multihomed anycast solution, PE-1 picks up DCGW-1's inclusive multicast route (because of its lower BGP next-hop) and creates a BUM destination to 192.0.2.4. When sending BUM traffic for VPLS-1, it only sends the traffic to DCGW-1. In the same way, the DCGWs do no set up BUM destinations between each other as they use the same **originating-ip** in their inclusive multicast routes.

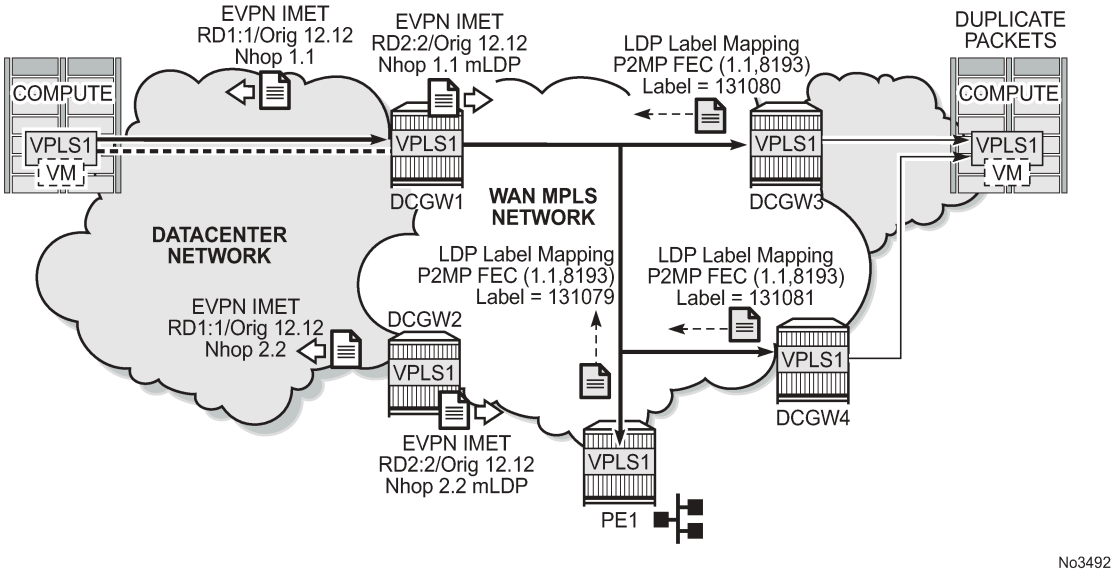
The remote PE or NVEs perform a similar BGP selection for MAC/IP routes, as a specific MAC is sent by the two DCGWs with the same route-key. A PE or NVE sends known unicast traffic for a specific MAC to only one DCGW.

6.5.9.5 Using P2MP mLDP in redundant anycast DCGWs

Figure 196: Anycast multihoming and mLDP shows an example of a common BGP EVPN service configured in redundant anycast DCGWs and mLDP used in the MPLS instance.

Note: Packet duplication may occur if the service configuration is not performed carefully.

Figure 196: Anycast multihoming and mLDP



When mLDP is used with multiple anycast multihoming DCGWs, the same originating IP address must be used by all the DCGWs. Failure to do so may result in packet duplication.

In the example shown in Figure 196: Anycast multihoming and mLDP, each pair of DCGWs (DCGW1/DCGW2 and DCGW3/DCGW4) is configured with a different originating IP address, which causes the following behavior:

1. DCGW3 and DCGW4 receive the inclusive multicast routes with the same route key from DCGW1 and DCGW2.
2. Both DCGWs (DCGW3 and DCGW4) select only one route, which is generally the same, for example, DCGW1's inclusive multicast route.
3. As a result, DCGW3 and DCGW4 join the mLDP tree with root in DCGW1, creating packet duplication when DCGW1 sends BUM traffic.
4. Remote PE nodes with a single BGP-EVPN instance join the mLDP tree without any problem.

To avoid the packet duplication shown in Figure 196: Anycast multihoming and mLDP, Nokia recommends to configure the same originating IP address in all four DCGWs (DCGW1/DCGW2 and DCGW3/DCGW4). However, the route distinguishers can be different per pair.

The following behavior occurs if the same originating IP address is configured on the DCGW pairs shown in Figure 196: Anycast multihoming and mLDP.

Note: This configuration allows the use of mLDP as long as BUM traffic is not required between the two DCs. Ingress replication must be used if BUM traffic between the DCs is required.

- DCGW3 and DCGW4 do not join any mLDP tree sourced from DCGW1 or DCGW2, which prevents any packet duplication. This is because a router ignore inclusive multicast routes received with its own **originating-ip**, regardless of the route-distinguisher.
- PE1 joins the mLDP trees from the two DCs.

6.5.9.6 I-ES solution for dual BGP instance services

SR OS supports Interconnect ESs (I-ES) for VXLAN as per *RFC9014*. An I-ES is a virtual ES that allows DCGWs with two BGP instances to handle VXLAN access networks as any other type of ES. I-ESs support the RFC 7432 multihoming functions, including single-active and all-active, ESI-based split-horizon filtering, DF election, and aliasing and backup on remote EVPN-MPLS PEs.

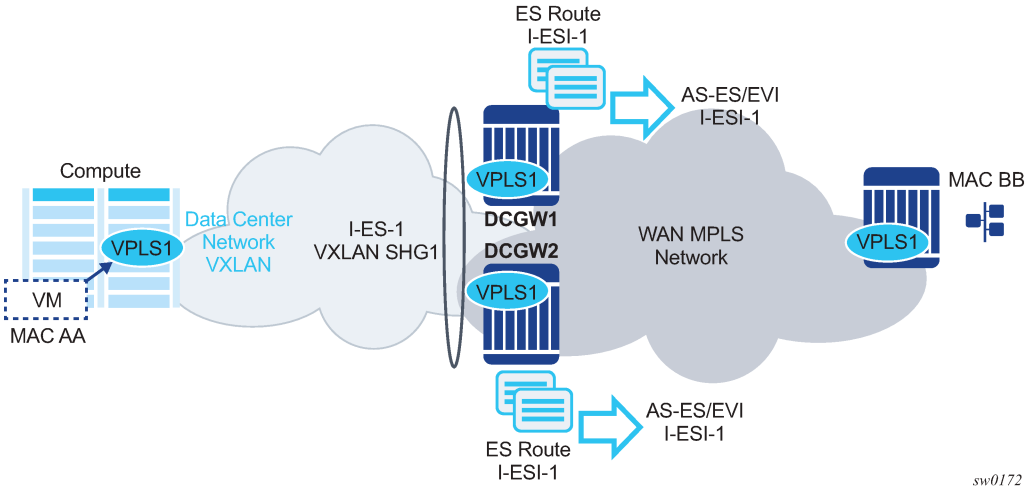
In addition to the EVPN multihoming features, the main advantages of the I-ES redundant solution compared to the redundant solution described in Anycast redundant solution for dual BGP-instance services are as follows:

- The use of I-ES for redundancy in dual BGP-instance services allows local SAPs on the DCGWs.
- P2MP mLDP can be used to transport BUM traffic between DCs that use I-ES without any risk of packet duplication. As described in Using P2MP mLDP in redundant anycast DCGWs, packet duplication may occur in the anycast DCGW solution when mLDP is used in the WAN.

Where EVPN-MPLS networks are interconnected to EVPN-VXLAN networks, the I-ES concept applies only to the access VXLAN network; the EVPN-MPLS network does not modify its existing behavior.

Figure 197: The Interconnect ES concept shows the use of I-ES for Layer 2 EVPN DCI between VXLAN and MPLS networks.

Figure 197: The Interconnect ES concept



The following example shows how I-ES-1 would be provisioned on DCGW1 and the association between I-ES to a specified VPLS service. A similar configuration would occur on DCGW2 in the I-ES.

New I-ES configuration:

```
DCGW1#config>service>system>bgp-evpn#
ethernet-segment I-ES-1 virtual create
 esi 01:00:00:00:12:12:12:12:12:00
 service-carving
  mode auto
 multi-homing all-active
 network-interconnect-vxlan 1
 service-id
   service-range 1 to 1000
 no shutdown
```

Service configuration:

```
DCGW1#config>service>vpls(1)#
 vxlan instance 1 vni 1 instance 1 create
 exit
bgp
 route-distinguisher 1:1
bgp 2
 route-distinguisher 2:2
bgp-evpn
  evi 1
  vxlan bgp 1 vxlan-instance 1
   no shutdown
  exit
  mpls bgp 2
   auto-bind-tunnel resolution any
   no shutdown

...

DCGW1#config>service>vpls(2)#
 vxlan instance 1 vni 2 create
 exit
bgp
```

```
 route-distinguisher 3:3
bgp 2
 route-distinguisher 4:4
 bgp-evpn
  evi 2
  vxlan bgp 1 vxlan-instance 1
   no shutdown
  exit
  mpls bgp 2
   auto-bind-tunnel resolution any
   no shutdown
 sap 1/1/1:1 create
 exit
```

The above configuration associates I-ES-1 to the VXLAN instance in services VPLS1 and VPLS 2. The I-ES is modeled as a virtual ES, with the following considerations:

- The commands **network-interconnect-vxlan** and **service-id service-range svc-id** [**to svc-id**] are required within the ES.
  - The **network-interconnect-vxlan** parameter identifies the VXLAN instance associated with the virtual ES. The value of the parameter must be set to 1. This command is rejected in a non-virtual ES.
  - The **service-range** parameter associates the specific service range to the ES. The ES must be configured as **network-interconnect-vxlan** before any service range can be added.
  - The ES parameters **port**, **lag**, **sdp**, **vc-id-range**, **dot1q**, and **qinq** cannot be configured in the ES when a **network-interconnect-vxlan** instance is configured. The **source-bmac-lsb** option is blocked, as the I-ES cannot be associated with an I-VPLS or PBB-Epipe service. The remaining ES configuration options are supported.
  - All services with two BGP instances associate the VXLAN destinations and ingress VXLAN instances to the ES.
- Multiple services can be associated with the same ES, with the following considerations:
  - In a DC with two DCGWs (as in Figure 197: The Interconnect ES concept), only two I-ESs are needed to load-balance, where one half of the dual BGP-instance services would be associated with one I-ES (for example, I-ES-1, in the above configuration) and one half to another I-ES.
  - Up to eight service ranges per VXLAN instance can be configured. Ranges may overlap within the same ES, but not between different ESs.
  - The service range can be configured before the service.
- After the I-ES is configured using **network-interconnect-vxlan**, the ES operational state depends exclusively on the ES administrative state. Because the I-ES is not associated with a physical port or SDP, when testing the non-revertive service carving manual mode, an ES **shutdown** and **no shutdown** event results in the node sending its own administrative preference and DP bit and taking control if the preference and DP bit are higher than the current DF. This is because the peer ES routes are not present at the EVPN application layer when the ES is configured for **no shutdown**; therefore, the PE sends its own administrative preference and DP values. For I-ESs, the non-revertive mode works only for node failures.
- A VXLAN instance may be placed in MhStandby under any of the following situations:
  - if the PE is single-active NDF for that I-ES
  - if the VXLAN service is added to the I-ES and either the ES or BGP-EVPN MPLS is shut down in all the services included in the ES

The following example shows the change of the MhStandby flag from false to true when BGP-EVPN MPLS is shut down for all the services in the I-ES.

```
A:PE-4# show service id 500 vxlan instance 1 oper-flags
===============================================================================
VPLS VXLAN oper flags
===============================================================================
MhStandby                           : false
===============================================================================
A:PE-4# configure service vpls 500 bgp-evpn vxlan shutdown
*A:PE-4# show service id 500 vxlan instance 1 oper-flags
===============================================================================
VPLS VXLAN oper flags
===============================================================================
MhStandby                           : true
===============================================================================
```

6.5.9.6.1 BGP-EVPN routes on dual BGP-instance services with I-ES

The configuration of an I-ES on DCGWs with two BGP instances has the following impact on the advertisement and processing of BGP-EVPN routes.

- For EVPN MAC/IP routes, the following considerations apply:

  If **bgp-evpn>vxlan>no auto-disc-route-advertisement** and **mh-mode access** are configured on the access instance:

  – MAC/IP routes received in the EVPN-MPLS BGP instance are readvertised in the EVPN-VXLAN BGP instance with the ESI set to zero.

  – EVPN-VXLAN PEs and NVEs in the DC receive the same MAC from two or more different MAC/IP routes from the DCGWs, which perform regular EVPN MAC/IP route selection.

  – MAC/IP routes received in the EVPN-VXLAN BGP instance are readvertised in the EVPN-MPLS BGP instance with the configured non-zero I-ESI value, assuming the VXLAN instance is not in an MhStandby operational state; otherwise the MAC/IP routes are dropped.

  – EVPN-MPLS PEs in the WAN receive the same MAC from two or more DCGWs set with the same ESI. In this case, regular aliasing and backup functions occur as usual.

- If **bgp-evpn>vxlan>auto-disc-route-advertisement** and **mh-mode access** are configured, the following differences apply to the above:

  – MAC/IP routes received in the EVPN-MPLS BGP instance are readvertised in the EVPN-VXLAN BGP instance with the ESI set to the I-ESI.

  – In this case, EVPN-VXLAN PEs and NVEs in the DC receive the same MAC from two or more different MAC/IP routes from the DCGWs, with the same ESI, therefore they can perform aliasing.

- ES routes are exchanged for the I-ES. The routes should be sent only to the MPLS network and not to the VXLAN network. This can be achieved by using router policies.

- AD per-ES and AD per-EVI routes are also advertised for the I-ES, and are sent only to the MPLS network and not to the VXLAN if **bgp-evpn>vxlan>no auto-disc-route-advertisement** is configured. For ES routes, router polices can be used to prevent these routes from being sent to VXLAN peers. If **bgp-evpn>vxlan>auto-disc-route-advertisement** is configured, AD routes must be sent to the VXLAN peers so that they can apply backup or aliasing functions.

In general, when I-ESs are used for redundancy, the use of router policies is needed to avoid control plane loops with MAC/IP routes. Consider the following to avoid control plane loops:

- **loops created by remote MACs**

  Remote EVPN-MPLS MAC/IP routes are readvertised into EVPN-VXLAN routes with an SOO (Site Of Origin) EC added by a BGP peer or VSI export policy identifying the DCGW pair. The other DCGW in the pair drops EVPN-VXLAN MAC/IP routes tagged with the pair SOO. Router policies are needed to add SOO and drop routes received with self SOO.

  When remote EVPN-VXLAN MAC/IP routes are readvertised into EVPN-MPLS, the DCGWs automatically drop EVPN-MPLS MAC/IP routes received with their own non-zero I-ESI.

- **loops created by local SAP MACs**

  Local SAP MACs are learned and MAC/IP routes are advertised into both BGP instances. The MAC/IP routes advertised in the EVPN-VXLAN instance are dropped by the peer based on the SOO router policies as described above for loops created by remote MACs. The DCGW local MACs are always learned over the EVPN-MPLS destinations between the DCGWs.

The following describes the considerations for BGP peer policies on DCGW1 to avoid control plane loops. Similar policies would be configured on DCGW2.

- Avoid sending service VXLAN routes to MPLS peers and service MPLS routes to VXLAN peers.
- Avoid sending AD and ES routes to VXLAN peers. If **bgp-evpn>vxlan>auto-disc-route-advertisement** is configured AD routes must be sent to the VXLAN peers.
- Add SOO to VXLAN routes sent to the ES peer.
- Drop VXLAN routes received from the ES peer.

The following shows the CLI configuration:

```
A:DCGW1# configure router bgp
A:DCGW1>config>router>bgp# info
----------------------------------------------
            family vpn-ipv4 evpn
            vpn-apply-import
            vpn-apply-export
            rapid-withdrawal
            rapid-update vpn-ipv4 evpn
            group "wan"
                type internal
                export "allow only mpls"
                neighbor 192.0.2.4
                exit
                neighbor 192.0.2.5
                exit
            exit
            group "internal"
                type internal
                neighbor 192.0.2.1
                    export "allow only vxlan"
                exit
                neighbor 192.0.2.3
                    import "drop SOO-DCGW-23"
                    export "add SOO to vxlan routes"
                exit
            exit
            no shutdown
----------------------------------------------
A:DCGW1>config>router>bgp# /configure router policy-options
A:DCGW1>config>router>policy-options# info
```

```
-----------------------------------------
            community "mpls" members "bgp-tunnel-encap:MPLS"
            community "vxlan" members "bgp-tunnel-encap:VXLAN"
            community "SOO-DCGW-23" members "origin:64500:23"
```

```
// This policy prevents the router from sending service VXLAN routes to MPLS peers. //
```

```
  policy-statement "allow only mpls"
               entry 10
                   from
                       community "vxlan"
                       family evpn
                   exit
                   action drop
                   exit
               exit
           exit
```

This policy ensures the router only exports routes that include the VXLAN encapsulation.

```
   policy-statement "allow only vxlan"
               entry 10
                   from
                       community "vxlan"
                       family evpn
                   exit
                   action accept
                   exit
               exit
               default-action drop
               exit
           exit
```

This import policy avoids importing routes with a self SOO.

```
  policy-statement "drop SOO-DCGW-23"
               entry 10
                   from
                       community "SOO-DCGW-23"
                       family evpn
                   exit
                   action drop
                   exit
               exit
           exit
```

This import policy adds SOO only to VXLAN routes. This allows the peer to drop routes based on the SOO, without affecting the MPLS routes.

```
  policy-statement "add SOO to vxlan routes"
               entry 10
                   from
                       community "vxlan"
                       family evpn
                   exit
                   action accept
                       community add "SOO-DCGW-23"
                   exit
               exit
```

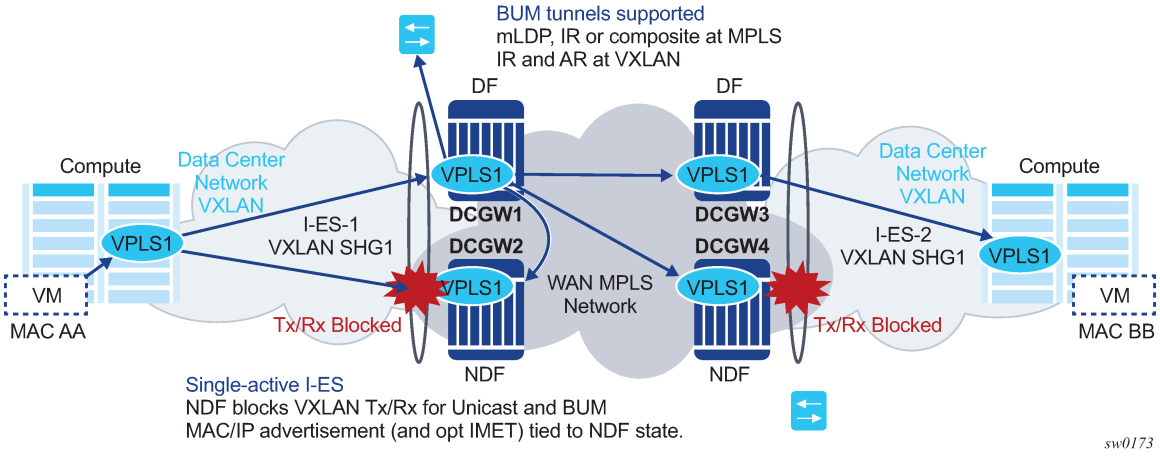
```
                default-action accept
                exit
            exit
-----------------------------------------------
```

6.5.9.6.2 Single-active multihoming on I-ES

When an I-ES is configured as single-active and configured as **no shutdown** with at least one associated service, the DCGWs send ES and AD routes as for any ES. It also runs DF election as normal, based on the ES routes, with the candidate list being pruned by the AD routes.

Figure 198: I-ES — single-active shows the expected behavior for a single-active I-ES.

Figure 198: I-ES — single-active



As shown in Figure 198: I-ES — single-active, the Non-Designated Forwarder (NDF) for a specified service carries out the following tasks:

- From a data path perspective, the VXLAN instance on the NDF goes into an MhStandby operational state and blocks ingress and egress traffic on the VXLAN destinations associated with the I-ES.
- The MAC/IP routes and the FDB process
  - MAC/IP routes associated with the VXLAN instance and readvertised to EVPN-MPLS peers are withdrawn.
  - MAC/IP routes corresponding to local SAP MACs or EVPN-MPLS binding MACs are withdrawn if they were advertised to the EVPN-VXLAN instance.
  - Received MAC/IP routes associated with the VXLAN instance are not installed in the FDB. MAC/IP routes show as "used" in BGP; however, only the MAC/IP route received from MPLS (from the ES peer) is programmed.
- The Inclusive Multicast Ethernet Tag (IMET) routes process
  - IMET-AR-R routes (IMET-AR with replicator role) must be withdrawn if the VXLAN instance goes into an MhStandby operational state. Only the DF advertises the IMET-AR-R routes.
  - IMET-IR advertisements in the case of the NDF (or MhStandby) are controlled by the command **config>service>vpls>bgp-evpn>vxlan** [**no**] **send-imet-ir-on-ndf**.

By default, the command is enabled and the router advertises IMET-IR routes, even if the PE is NDF (MhStandby). This attracts BUM traffic, but also speeds up convergence in the case of a DF switchover. The command is supported for single-active and all-active.

If the command is disabled, the router withdraws the IMET-IR routes when the PE is NDF and do not attract BUM traffic.

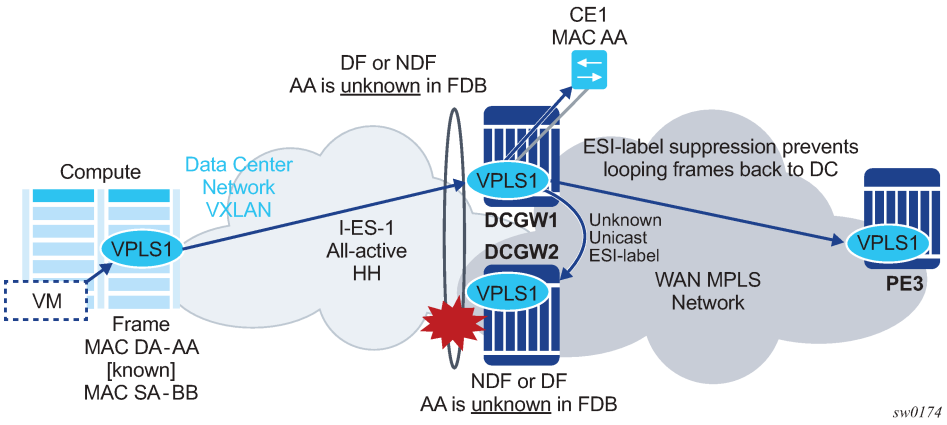
The I-ES DF PE for the service continues advertising IMET and MAC/IP routes for the associated VXLAN instance as usual, as well as forwarding on the DF VXLAN bindings. When the DF DCGW receives BUM traffic, it sends the traffic with the egress ESI label if needed.

6.5.9.6.3 All-active multihoming on I-ES

The same considerations for ES and AD routes, and DF election apply for all-active multihoming as for single-active multihoming; the difference is in the behavior on the NDF DCGW. The NDF for a specified service performs the following tasks:

- From a data path perspective, the NDF blocks ingress and egress paths for broadcast and multicast traffic on the VXLAN instance bindings associated with the I-ES, while unknown and known unicast traffic is still allowed. The unknown unicast traffic is transmitted on the NDF if there is no risk of duplication. For example, unknown unicast packets are transmitted on the NDF if they do not have an ESI label, do not have an EVPN BUM label, and they pass a MAC SA suppression. In the example in Figure 199: All-active multihoming and unknown unicast on the NDF, the NDF transmits unknown unicast traffic. Regardless of whether DCGW1 is a DF or NDF, it accepts the unknown unicast packets and floods to local SAPs and EVPN destinations. When sending to DGW2, the router sends the ESI label identifying the I-ES. Because of the ESI-label suppression, DCGW2 does not send unknown traffic back to the DC.

Figure 199: All-active multihoming and unknown unicast on the NDF



- The MAC/IP routes and the FDB process
  - MAC/IP routes associated with the VXLAN instance are advertised normally.
  - MACs are installed as normal in the FDB for received MAC/IP routes associated with the VXLAN instance.
- The IMET routes process

- As with single-active multihoming, IMET-AR-R routes must be withdrawn on the NDF (MhStandby state). Only the DF advertises the IMET-AR-R routes.
- The IMET-IR advertisements in the case of the NDF (or MhStandby) are controlled by the command **config**>**service**>**vpls**>**bgp-evpn**>**vxlan** [**no**] **send-imet-ir-on-ndf**, as in single-active multihoming.

The behavior on the non-DF for BUM traffic can also be controlled by the command **config>service>vpls>vxlan>rx-discard-on-ndf {bm | bum | none}**, where the default option is **bm**. However, the user can change this option to discard all BUM traffic, or forward all BUM traffic (none).

The I-ES DF PE for the service continues advertising IMET and MAC/IP routes for the associated VXLAN instance as usual. When the DF DCGW receives BUM traffic, it sends the traffic with the egress ESI label if needed.

6.5.10 Multi-instance EVPN: Two instances of the same encapsulation in the same VPLS/R-VPLS service

As described in Multi-Instance EVPN: Two instances of different encapsulation in the same VPLS/R-VPLS service, two BGP instances are supported in VPLS services, where one instance can be associated with the EVPN-VXLAN and the other instance with the EVPN-MPLS. In addition, both BGP instances in a VPLS/R-VPLS service can also be associated with EVPN-VXLAN, or both instances can be associated with EVPN-MPLS.

For example, a VPLS service can be configured with two VXLAN instances that use VNI 500 and 501 respectively, and those instances can be associated with different BGP instances:

```
*A:PE-2# configure service vpls 500
*A:PE-2>config>service>vpls# info
----------------------------------------------
vxlan instance 1 vni 500 create
exit
vxlan instance 2 vni 501 create
exit
bgp
 route-distinguisher 192.0.2.2:500
 vsi-export "vsi-500-export"
 vsi-import "vsi-500-import"
exit
bgp 2
 route-distinguisher 192.0.2.2:501
 vsi-export "vsi-501-export"
 vsi-import "vsi-501-import"
exit
bgp-evpn
 incl-mcast-orig-ip 23.23.23.23
 evi 500
 vxlan bgp 1 vxlan-instance 1
 no shutdown
 exit
 vxlan bgp 2 vxlan-instance 2
 no shutdown
 exit
exit
stp
shutdown
exit
no shutdown
----------------------------------------------
```

From a data plane perspective, each VXLAN instance is instantiated in a different implicit SHG, so that traffic can be forwarded between them.

In addition, multi-instance EVPN-VXLAN services support:

- assisted-replication for IPv4 VTEPs in both VXLAN instances, where a single assisted-replication IPv4 address can be used for both instances
- non-system IP and IPv6 termination, where a single **vxlan-src-vtep ip-address** can be configured for each service, and therefore used for the two instances

For example, a VPLS service can be configured with two EVPN-MPLS instances that are associated with two BGP instances as follows.

```
*A:PE-2# configure service vpls 700
*A:PE-2>config>service>vpls# info
----------------------------------------------
description "two bgp-evpn mpls instances"
bgp
  route-distinguisher auto-rd
  vsi-export "vsi-700-export"
  vsi-import "vsi-700-import"
exit
bgp 2
  route-distinguisher auto-rd
  vsi-export "vsi-701-export"
  vsi-import "vsi-701-import"
exit
bgp-evpn
  evi 700
  mpls bgp 1
    mh-mode access
    ingress-replication-bum-label
    auto-bind-tunnel
      resolution any
    exit
    no shutdown
  exit
  mpls bgp 2
    mh-mode network
    ingress-replication-bum-label
    auto-bind-tunnel
      resolution any
    exit
    no shutdown
  exit
exit
stp
  shutdown
exit
no shutdown
----------------------------------------------
```

Multi-instance EVPN-MPLS VPLS/R-VPLS services have the same limitations as any multi-instance service, as described in Multi-Instance EVPN: EVPN-VXLAN and EVPN-MPLS in the same VPLS/R-VPLS service. In addition, services with two EVPN-MPLS instances do not support SAPs.

The **mh-mode {network|access}** command in the **vpls>bgp-evpn>mpls** context determines which instance is considered access and which instance is considered network.

- The default form of the **bgp-evpn>mpls** command is **mh-mode network** and only one instance can be configured. The other instance must be configured as **mh-mode access**.

- The use of **provider-tunnel** is supported if there is one instance configured as **network**, and the P2MP tunnel is implicitly associated with the network instance.

Multi-instance EVPN-MPLS VPLS/R-VPLS services support:

- all of the **auto-bind-tunnel resolution** options in each of the two instances. This includes resolution of IPv4 next-hops to TTMv4 entries and resolution of IPv6 next-hops to TTMv6 entries.
- different address families in different instances. For instance, **mpls bgp 1** may resolve routes to TTMv4 and **mpls bgp 2** to TTMv6, or the reverse. In a VPLS service with two EVPN-VXLAN instances, it is not possible to have an instance with routes resolved to IPv4 VXLAN tunnels and the other instance with routes resolved to IPv6 VXLAN tunnels.
- an explicit **split-horizon-group** in each instance; however, the same split-horizon-group cannot be configured on the two instances of the same VPLS service
- a **restrict-protected-src discard-frame** per instance. If a MAC is protected in one instance and a frame arrives at the other instance with the protected MAC as source MAC, the frame is discarded if **restrict-protected-src discard-frame** is configured.

At the control plane level for two EVPN-VXLAN or two EVPN-MPLS instances, the processing of MAC/IP routes and inclusive multicast routes is described in BGP-EVPN routes in services configured with two BGP instances with the differences between the two scenarios described in BGP-EVPN routes in multi-instance EVPN services with the same encapsulation.

6.5.10.1 BGP-EVPN routes in multi-instance EVPN services with the same encapsulation

If two BGP instances with the same encapsulation (VXLAN or MPLS) are configured in the same VPLS/R-VPLS service, different import route targets in each BGP instance are mandatory (although this is not enforced).

BGP-EVPN routes in services configured with two BGP instances describes the use of policies to avoid sending WAN routes (routes meant to be redistributed from DC to WAN) to the DC again and DC routes (routes meant to be redistributed from WAN to DC) to the WAN again. Those policies are based on export policy statements that match on the RFC 9012 BGP encapsulation extended community (MPLS and VXLAN respectively).

When the two BGP instances are of the same encapsulation (VXLAN or MPLS), the policies matching on different BGP encapsulation extended community are not feasible because both instances advertise routes with the same encapsulation value. Because the export route targets in the two BGP instances must be different, the policies, to avoid sending WAN routes back to the WAN and DC routes back to the DC, can be based on export policies that prevent routes with a DC route target from being sent to the WAN peers (and opposite for routes with a WAN route target).

In scaled scenarios, matching based on route targets, does not scale well. An alternative and preferred solution is to configure a **default-route-tag** that identifies all the EVPN instances connected to the DC (or one domain), and a different **default-route-tag** in all the EVPN instances connected to the WAN (or the other domain). Anycast redundant solution for multi-instance EVPN services with the same encapsulation shows an example that demonstrates the use of **default-route-tags**.

Other than the specifications described in this section, the processing of MAC/IP routes and inclusive multicast Ethernet tag routes in multi-instance EVPN services of the same encapsulation follow the rules described in BGP-EVPN routes in services configured with two BGP instances.

6.5.10.2 Anycast redundant solution for multi-instance EVPN services with the same encapsulation

The solution described in Anycast redundant solution for dual BGP-instance services is also supported in multi-instance EVPN VPLS/R-VPLS services with the same encapsulation.

The following CLI example output shows the configuration of DCGW-1 and DCGW-2 in Figure 195: Multihomed anycast solution where VPLS 500 is a multi-instance EVPN-VXLAN service and BGP instance 2 is associated with VXLAN instead of MPLS.

Different default-route-tags are used in BGP instance 1 and instance 2, so that in the export route policies, DC routes are not advertised to the WAN, and WAN routes are not advertised to the DC, respectively.

```
*A:DCGW-1(and DCGW-2)>config>service>vpls(500)# info
----------------------------------------------
vxlan instance 1 vni 500 create
exit
vxlan instance 2 vni 501 create
exit
bgp
 route-distinguisher 192.0.2.2:500
 route-target target:64500:500
exit
bgp 2
 route-distinguisher 192.0.2.2:501
 route-target target:64500:501
exit
bgp-evpn
 incl-mcast-orig-ip 23.23.23.23
 evi 500
 vxlan bgp 1 vxlan-instance 1
 default-route-tag 500
 no shutdown
 exit
 vxlan bgp 2 vxlan-instance 2
   default-route-tag 501
   no shutdown
 exit
exit
stp
shutdown
exit
no shutdown
----------------------------------------------
config>router>bgp#
 vpn-apply-import
 vpn-apply-export
 group "WAN"
  family evpn
  type internal
  export "allow only mpls"
  neighbor 192.0.2.6
 group "DC"
  family evpn
  type internal
export "allow only vxlan"
  neighbor 192.0.2.2

config>router>policy-options# info
----------------------------------------------
            policy-statement "allow only mpls"
                entry 10
                  from
```

```
                  family evpn
                  tag 500
                action drop
                exit
              exit
          exit
          policy-statement "allow only vxlan"
              entry 10
                from
                  family evpn
                  tag 501
                action drop
exit
              exit
          exit
```

The same Anycast redundant solution can be applied to VPLS/R-VPLS with two instances of EVPN-MPLS encapsulation. The configuration would be identical, other than replacing the VXLAN aspects with the EVPN-MPLS-specific parameters.

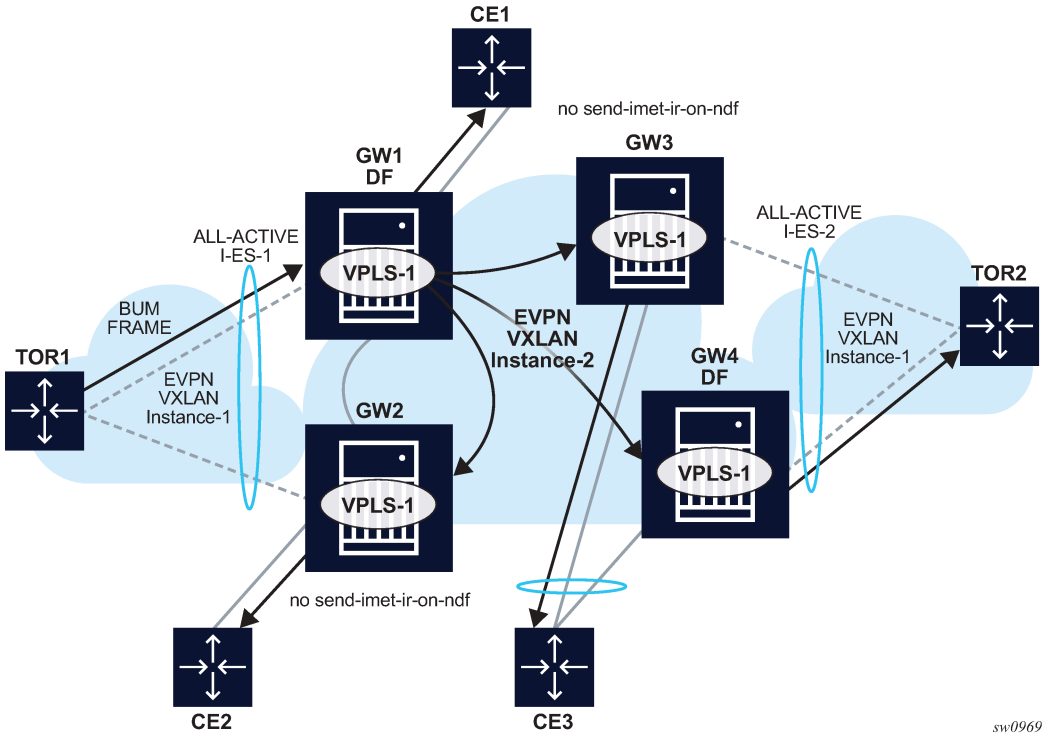
For a full description of this solution, see the Anycast redundant solution for dual BGP-instance services

6.5.10.3 I-ES solution for dual BGP EVPN instance services with the same encapsulation

The I-ES of network-interconnect VXLAN Ethernet segment is described in I-ES solution for dual BGP instance services. I-ES's are also supported on VPLS and R-VPLS services with two EVPN-VXLAN instances.

Figure 200: I-ES in dual EVPN-VXLAN services shows the use of an I-ES in a dual EVPN-VXLAN instance service.

Figure 200: I-ES in dual EVPN-VXLAN services



Similar to (single-instance) EVPN-VXLAN all-active multihoming, the BUM forwarding procedures follow the "Local Bias" behavior.

At the ingress PE, the forwarding rules for EVPN-VXLAN services are as follows:

- The **no send-imet-ir-on-ndf** or **rx-discard-on-ndf bum** command must be enabled so that the NDF does not forward any BUM traffic.
- BUM frames received on any SAP or I-ES VXLAN binding are flooded to:
  - local non-ES and single-active DF ES SAPs
  - local all-active ES SAPs (DF and NDF)
  - EVPN-VXLAN destinations

    BUM received on an I-ES VXLAN binding follows SHG rules, for example, it can only be forwarded to EVPN-VXLAN destinations that belong to the other VXLAN instance (instance 2), which is a different SHG.
- As an example, in Figure 200: I-ES in dual EVPN-VXLAN services:
  - GW1 and GW2 are configured with **no send-imet-ir-on-ndf**.
  - TOR1 generates BUM traffic that only reaches GW1 (DF).
  - GW1 forwards to CE1 and EVPN-VXLAN destinations.

The forwarding rules at the egress PE are as follows:

- The source VTEP is looked up for BUM frames received on EVPN-VXLAN.

- If the source VTEP matches one of the PEs with which the local PE shares an ES _AND_ a VXLAN service:
  - Then the local PE does not forward to the shared local ES'es (this includes port, lag, or network-interconnect-vxlan ES'es). It forwards though to non-shared ES SAPs unless they are in NDF state.
  - Else, the local PE forwards normally to local ES'es unless they are in NDF state.
- Because there is no multicast label or multicast B-MAC in VXLAN, the only way the egress PE can identify BUM traffic is by looking at the customer MAC DA. Therefore, BM or unknown MAC DAs identify BUM traffic.
- As an example, in Figure 200: I-ES in dual EVPN-VXLAN services:
  - GW2 receives BUM on EVPN-VXLAN. GW2 identifies the source VTEP as a PE with which the I-ES-1 is shared, therefore it does not forward the BUM frames to the local I-ES. It forwards to the non-shared ES and local SAPs though (CE2).
  - GW3 receives BUM on EVPN-VXLAN, however the source VTEP does not match any PE with which GW3 shares an ES. Hence GW3 forwards to all local ES'es that are DF, in other words, CE3.

The following configuration example shows how I-ES-1 would be provisioned on DCGW1 and the association between I-ES to a specified VPLS service. A similar configuration would occur on DCGW2 in the I-ES.

I-ES configuration:

```
*A:GW1>config>service>system>bgp-evpn>eth-seg# info
----------------------------------------------
                esi 00:23:23:23:23:23:23:00:00:01
                service-carving
                    mode manual
                    manual
                        preference non-revertive create
                            value 150
                        exit
                        evi 101 to 200
                    exit
                exit
                multi-homing all-active
                network-interconnect-vxlan 1
                service-id
                    service-range 1
                    service-range 1000 to 1002
                    service-range 2000
                exit
                no shutdown
```

Service configuration:

```
*A:GW1>config>service>vpls# info
----------------------------------------------
            vxlan instance 1 vni 1000 create
                rx-discard-on-ndf bum
            exit
            vxlan instance 2 vni 1002 create
            exit
            bgp
                route-target export target:64500:1000 import target:64500:1000
            exit
            bgp 2
                route-distinguisher auto-rd
```

```
            route-target export target:64500:1002 import target:64500:1002
        exit
        bgp-evpn
            evi 1000
            vxlan bgp 1 vxlan-instance 1
                ecmp 2
                default-route-tag 100
                auto-disc-route-advertisement
                no shutdown
            exit
            vxlan bgp 2 vxlan-instance 2
                ecmp 2
                default-route-tag 200
                auto-disc-route-advertisement
                mh-mode network
                no shutdown
            exit
        exit
        no shutdown
```

Multi-instance EVPN VPLS/R-VPLS services with two EVPN-MPLS instances do not support I-ESs.

For information about how the EVPN routes are processed and advertised in an I-ES, see the I-ES solution for dual BGP instance services.

6.5.11 Configuring static VXLAN and EVPN in the same VPLS/R-VPLS service

In some DCGW use cases, static VXLAN must be used to connect DC switches that do not support EVPN to the WAN so that a tenant subnet can be extended to the WAN. For those cases, the DC Gateway is configured with VPLS services that include a static VXLAN instance and a BGP-EVPN instance in the same service. The following combinations are supported in the same VPLS/R-VPLS service:

- two static VXLAN instances
- one static VXLAN instance and one EVPN-VXLAN instance
- one static VXLAN instance and one EVPN-MPLS instance

When a static VXLAN instance coexists with EVPN-MPLS in the same VPLS/R-VPLS service, the VXLAN instance can be associated with a **network-interconnect-vxlan** ES if VXLAN uses instance 1. Both single-active and all-active multihoming modes are supported as follows:

- In single-active mode, the following behavior is for a VXLAN binding associated with the ES on the NDF:
  - **TX (transmission to VXLAN)**

    No MACs are learned against the binding, and the binding is removed from the default multicast list.
  - **RX (reception from VXLAN)**

    The RX state is down for the binding.
- In all-active mode, the following behavior is for the NDF:
  - **on TX**

    The binding is kept in the default multicast list, but only forwards the unknown-unicast traffic.
  - **on RX**

    The behavior is determined by the command **rx-discard-on-ndf {bm | bum | none}** where:

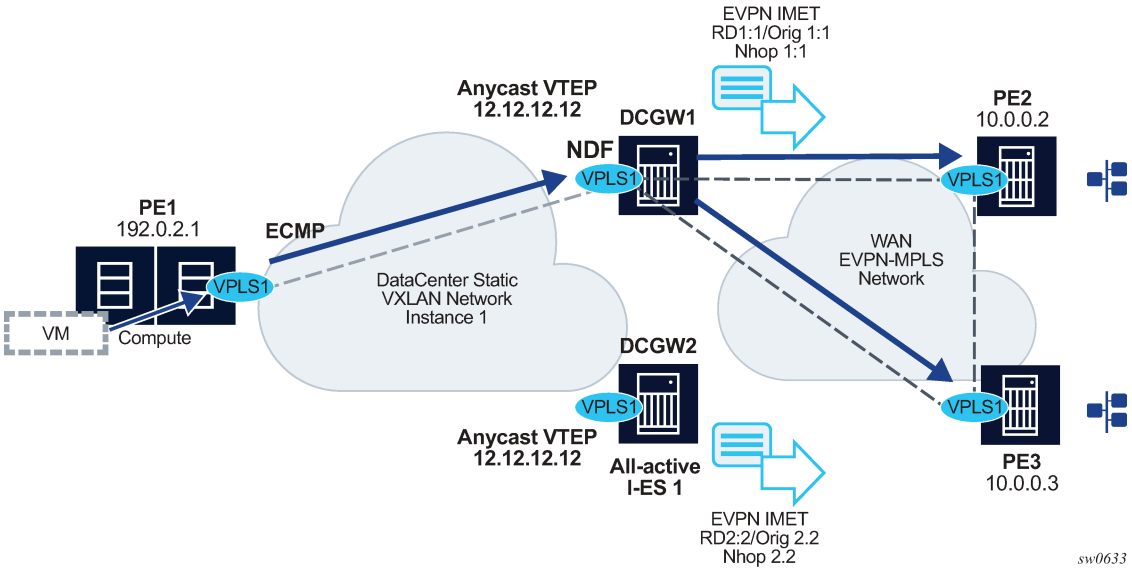
- The option **bm** is the default option, discards broadcast and multicast traffic, and allows unicast (known and unknown).
- The option **bum** discards any BUM frame on the NDF reception.
- The option **none** does not discard any BUM frame on the NDF reception.

The use of the **rx-discard-on-ndf** options is shown in the following cases.

Use case 1: Static VXLAN with anycast VTEPs and all-active ES

This use case, which is illustrated in Figure 201: I-ES multihoming – static VXLAN with anycast VTEPs, works only for all-active I-ESs.

Figure 201: I-ES multihoming – static VXLAN with anycast VTEPs



In this use case, the DCGWs use anycast VTEPs, that is, PE1 has a single egress VTEP configured to the DCGWs, for example, 12.12.12.12. Normally, PE1 finds ECMP paths to send the traffic to both DCGWs. However, because a specified BUM flow can be sent to either the DF or the NDF (but not to both at the same time), the DCGWs must be configured with the following option so that BUM is not discarded on the NDF:

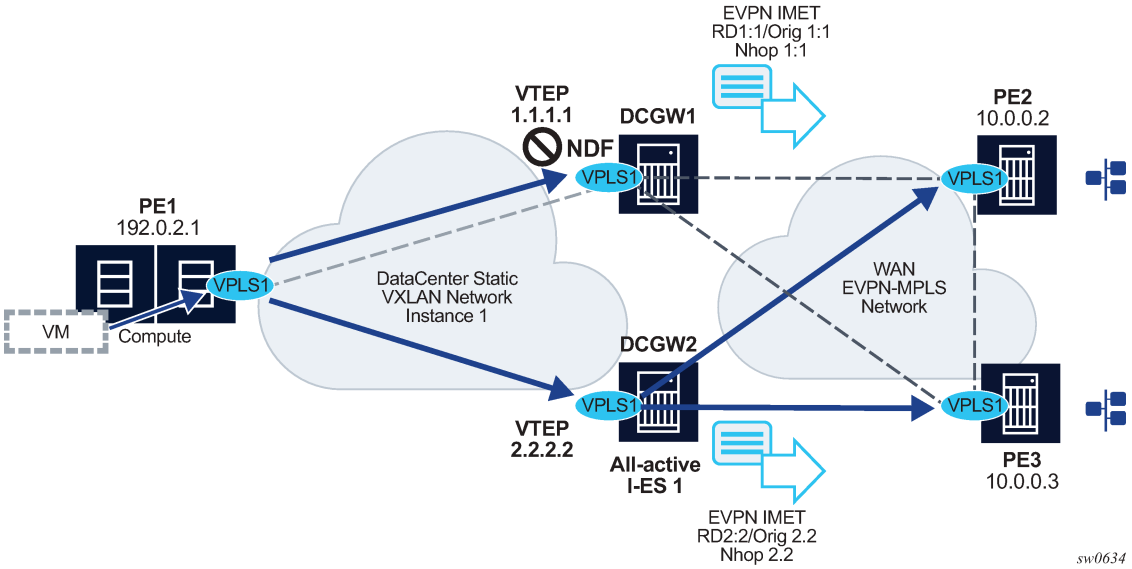
rx-discard-on-ndf none

Similar to any LAG-like scenario at the access, the access CE load balances the traffic to the multihomed PEs, but a specific flow is only sent to one of these PEs. With the option **none**, the BUM traffic on RX is accepted, and there are no duplicate packets or black-holed packets.

Use case 2: Static VXLAN with non-anycast VTEPs

This use case, which is shown in the following figure, works with single or all-active multihoming.

Figure 202: I-ES multihoming - static VXLAN with non-anycast VTEPs



In this case, the DCGWs use different VTEPs, for example 1.1.1.1 and 2.2.2.2 respectively. PE1 has two separate egress VTEPs to the DCGWs. Therefore, PE1 sends BUM flows to both DCGWs at the same time. Concerning all-active multihoming, if the default option for **rx-discard-on-ndf** is configured, PE2 and PE3 receive duplicate unknown unicast packets from PE1 (because the default option accepts unknown unicast on the RX of the NDF). So, the DCGWs must be configured with **rx-discard-on-ndf bum** .

Any use case in which the access PE sends BUM flows to all multihomed PEs, including the NDF, is similar to Figure 202: I-ES multihoming - static VXLAN with non-anycast VTEPs. BUM traffic must be blocked on the NDF's RX to avoid duplicate unicast packets.

For single-active multihoming, the **rx-discard-on-ndf** is irrelevant because BUM and known unicast are always discarded on the NDF.

Also, when non-anycast VTEPs are used on DCGWs, the following can be stated:

- MAC addresses learned on one DCGW and advertised in EVPN, are not learned on the redundant DCGW through EVPN, based on the presence of a local ES in the route. Figure 202: I-ES multihoming - static VXLAN with non-anycast VTEPs, shows a scenario in which the MAC of VM can be advertised by DCGW1, but not learned by DCGW2.
- As a result of the above behavior and because PE2 known unicast to M1 can be aliased to DGW2, when traffic to M1 gets to DCGW2, it is flooded because M1 is unknown. DCGW2 floods to all the static bindings, as well as local SAPs.
- ESI-label filtering, and no VXLAN binding between DCGWs, avoid loops for BUM traffic sent from the DF.

When a static VXLAN instance coexists with EVPN-VXLAN in the same VPLS or R-VPLS service, no VXLAN instance should be associated with an all-active network-interconnect-vxlan ES. This is because when multihoming is used with an EVPN-VXLAN core, the non-DF PE always discards unknown unicast traffic to the static VXLAN instance (this is not the case with EVPN-MPLS if the unknown traffic has a BUM label) and traffic blackholes may occur. This is discussed in the following example:

- Consider the example in Figure 202: I-ES multihoming - static VXLAN with non-anycast VTEPs I-ES multihoming – static VXLAN with non-anycast VTEPs, only replacing EVPN-MPLS by EVPN-VXLAN in the WAN network.
- Consider the PE2 has learned VM's MAC via ES-1 EVPN destination. Because of the regular aliasing procedures, PE2 may send unicast traffic with destination VM to DCGW1, which is the non-DF for I-ES 1.
- Because EVPN-VXLAN is used in the WAN instead of EVPN-MPLS, when the traffic gets to DCGW1, it is dropped if the VM's MAC is not learned on DCGW1, creating a blackhole for the flow. If the I-ES had used EVPN-MPLS in the WAN, DCGW1 would have flooded to the static VXLAN binds and no blackhole would have occurred.

Because of the behavior illustrated above, when a static VXLAN instance coexists with an EVPN-VXLAN instance in the same VPLS/R-VPLS service, redundancy based on all-active I-ES is not recommended and single-active or an anycast solution without I-ES should be used instead. Anycast solutions are discussed in Anycast redundant solution for multi-instance EVPN services with the same encapsulation, only with a static VXLAN instance in instance 1 instead of EVPN-VXLAN in this case.

6.5.12 EVPN IP-prefix route interoperability

SR OS supports the three IP-VRF-to-IP-VRF models defined in *draft-ietf-bess-evpn-prefix-advertisement* for EVPN-VXLAN and EVPN-MPLS R-VPLS services. Those three models are known as:

- interface-less IP-VRF-to-IP-VRF
- interface-ful IP-VRF-to-IP-VRF with SBD IRB (Supplementary Bridge Domain Integrated Routing Bridging)
- interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB

SR OS supports all three models for IPv4 and IPv6 prefixes. The three models have pros and cons, and different vendors have chosen different models depending on the use cases that they intend to address. When a third-party vendor is connected to an SR OS router, it is important to know which of the three models the third-party vendor implements. The following sections describe the models and the required configuration in each of them.

6.5.12.1 Interface-ful IP-VRF-to-IP-VRF with SBD IRB model

The SBD is equivalent to an R-VPLS that connects all the PEs that are attached to the same tenant VPRN. Interface-ful refers to the fact that there is a full IRB interface between the VPRN and the SBD (an interface object with MAC and IP addresses, over which interface parameters can be configured).

Figure 203: Interface-ful IP-VRF-to-IP-VRF with SBD IRB model illustrates this model.

Figure 203: Interface-ful IP-VRF-to-IP-VRF with SBD IRB model

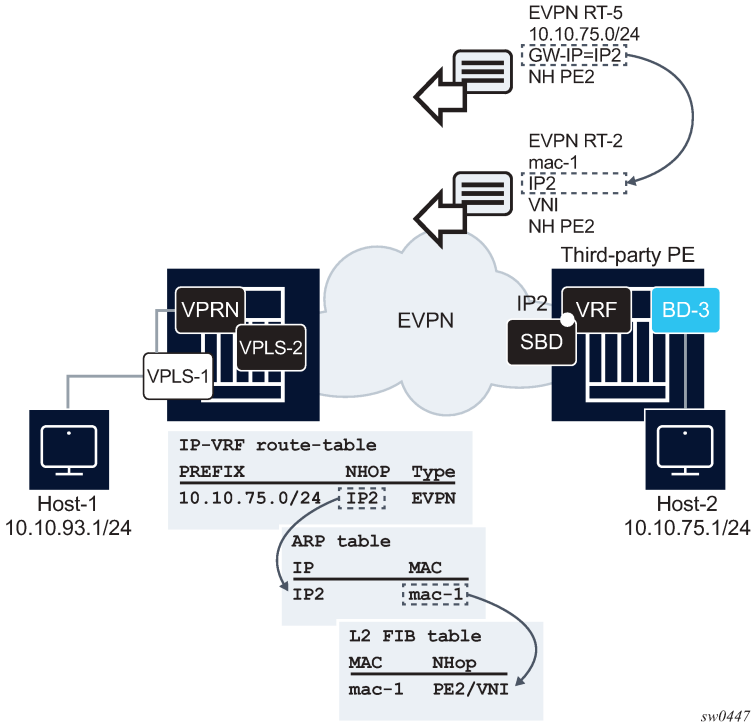


Figure 203: Interface-ful IP-VRF-to-IP-VRF with SBD IRB model shows a 7750 SR and a third-party router using interface-ful IP-VRF-to-IP-VRF with SBD IRB model. The two routers are attached to a VPRN for the same tenant, and those VPRNs are connected by R-VPLS-2, or SBD. Both routers exchange IP prefix routes with a non-zero gateway IP (this is the IP address of the SBD IRB). The SBD IRB MAC and IP are advertised in a MAC/IP route. On reception, the IP prefix route creates a route-table entry in the VPRN, where the gateway IP must be recursively resolved to the information provided by the MAC/IP route and installed in the ARP and FDB tables.

This model is described in detail in EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes. As an example, and based on Figure 203: Interface-ful IP-VRF-to-IP-VRF with SBD IRB model above, the following CLI output shows the configuration of a 7750 SR SBD and VPRN, using on this interface-ful with SBD IRB mode:

```
7750SR#config>service#

vpls 2 customer 1 name "sbd" create
 allow-ip-int-bind
 exit
 bgp
 exit
 bgp-evpn
 evi 2
 ip-route-advertisement
 mpls bgp 1
 auto-bind-tunnel resolution any
 no shutdown

vprn 1 customer 1 name "vprn1" create
 route-distinguisher auto-rd
 interface "sbd" create
```

```
address 192.168.0.1/16
ipv6
  30::3/64
exit
vpls "sbd"
```

The model is, also, supported for IPv6 prefixes. There are no configuration differences except the ability to configure an IPv6 address and interface.

6.5.12.2 Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model

Interface-ful refers to the fact that there is a full IRB interface between the VPRN and the SBD. However, the SBD IRB is unnumbered in this model, which means no IP address is configured on it. In SR OS, an unnumbered SBD IRB is equivalent to an R-VPLS linked to a VPRN interface through an EVPN tunnel. See EVPN for VXLAN in EVPN tunnel R-VPLS services for more information.

Figure 204: Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model illustrates this model.

Figure 204: Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model

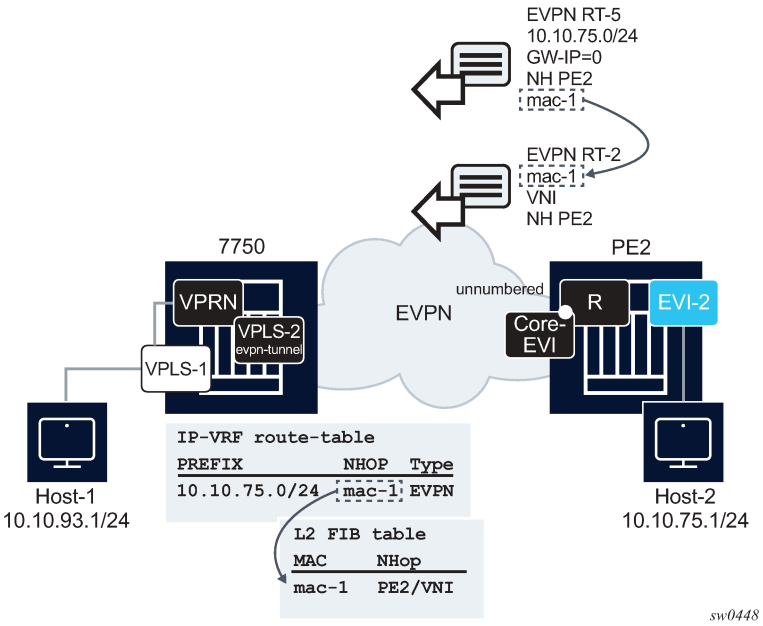


Figure 204: Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model shows a 7750 SR and a third-party router running interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model. The IP prefix routes are now expected to have a zero gateway IP and the MAC in the router's MAC extended community used for the recursive resolution to a MAC/IP route.

The corresponding configuration of the 7750 SR VPRN and SBD in the example could be:

```
7750SR#config>service#

vpls 2 customer 1 name "sbd" create
 allow-ip-int-bind
 exit
 bgp
 exit
```

```
 bgp-evpn
evi 2
ip-route-advertisement
mpls bgp 1
auto-bind-tunnel resolution any
no shutdown

vprn 1 customer 1 create
 route-distinguisher auto-rd
 interface "sbd" create
  ipv6
  exit
 vpls "sbd"
  evpn-tunnel ipv6-gateway-address mac
```

Note that the **evpn-tunnel** command controls the use of the Router's MAC extended community and the zero gateway IP in the IPv4-prefix route. For IPv6, the **ipv6-gateway-address mac** option makes the router advertise the IPv6-prefix routes with a Router's MAC extended community and zero gateway IP.

6.5.12.3 Interoperable interface-less IP-VRF-to-IP-VRF model (Ethernet encapsulation)

This model is interface-less because no Supplementary Broadcast Domain (SBD) is required to connect the VPRNs of the tenant, and no recursive resolution is required upon receiving an IP prefix route. In other words, the next-hop of the IP prefix route is directly resolved to an EVPN tunnel, without the need for any other route. The standard specification *draft-ietf-bess-evpn-ip-prefix* supports two variants of this model that are not interoperable with each other:

- **EVPN IFL for Ethernet NVO (Network Virtualization Overlay) tunnels**

  Ethernet NVO indicates that the EVPN packets contain an inner Ethernet header. This is the case for tunnels such as VXLAN.

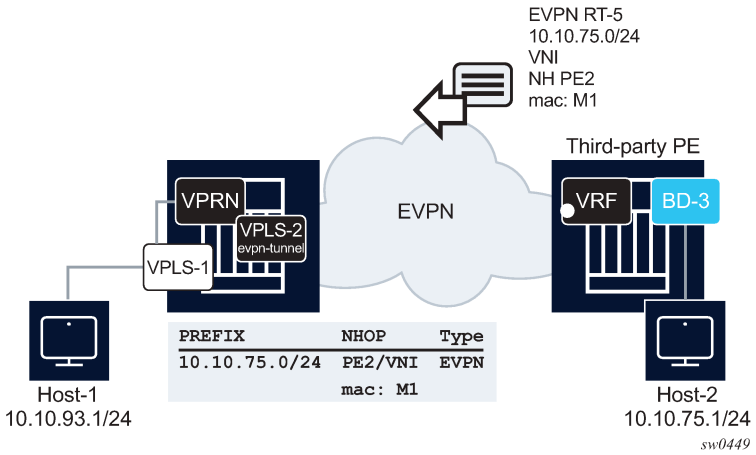
  In the Ethernet NVO option, the ingress PE uses the received router's MAC extended community address (received along with the route type 5) as the inner destination MAC address for the EVPN packets sent to the prefix

- **EVPN IFL for IP NVO tunnels**

  IP NVO indicates that the EVPN packets contain an inner IP packet, but without Ethernet header. This is similar to the IPVPN packets exchanged between PEs.

Figure 205: Interface-less IP-VRF-to-IP-VRF model illustrates the Interface-less IP-VRF-to-IP-VRF model.

Figure 205: Interface-less IP-VRF-to-IP-VRF model



SR OS supports the interoperable Interface-less IP-VRF-to-IP-VRF Model for Ethernet NVO tunnels. In Figure 205: Interface-less IP-VRF-to-IP-VRF model this interoperable model is shown on the left side PE router. The following is the model implementation:

- There is no data path difference between this model and the existing R-VPLS EVPN tunnel model or the model described in Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model.
- This model is enabled by configuring **config>service>vprn>if>vpls>evpn-tunnel** (with **ipv6-gateway-address mac** for IPv6), and **bgp-evpn>ip-route-advertisement**. In addition, because the SBD IRB MAC/IP route is no longer needed, the **bgp-evpn no mac-advertisement** command prevents the advertisement of the MAC/IP route.
- The IP prefix routes are processed as follows:
  - On transmission, there is no change in the IP prefix route processing compared to the configuration of the Interface-ful IP-VRF-to-IP-VRF with Unnumbered SBD IRB Model.
    - IPv4/IPv6 prefix routes are advertised based on the information in the route-table for IPv4 and IPv6, with GW-IP=0 and the corresponding MAC extended community.
    - If **bgp-evpn no mac-advertisement** is configured, no MAC/IP route is sent for the R-VPLS.
  - The received IPv4/IPv6 prefix routes are processed as follows:
    1. Upon receiving an IPv4/IPv6 prefix route with a MAC extended community for the router, an internal MAC/IP route is generated with the encoded MAC and the RD, Ethernet tag, ESI, Label/VNI and next hop derived from the IP prefix route itself.
    2. If no competing received MAC/IP routes exist for the same MAC, this IP prefix-derived MAC/IP route is selected and the MAC is installed in the R-VPLS FDB with type "Evpn".
    3. After the MAC is installed in FDB, there are no differences between this interoperable interface-less model and the interface-ful with unnumbered SBD IRB model. Therefore, SR OS is compatible with the received IP prefix routes for both models.

The following is an example of a typical configuration of a PE's SBD and VPRN that work in interface-less model for IPv4 and IPv6:

```
7750SR#config>service#

vpls 2 customer 1 name "sbd" create
```

```
        allow-ip-int-bind
        exit
        bgp
        exit
        bgp-evpn
             evi 2
             no mac-advertisement
             ip-route-advertisement
             mpls bgp 1
                  auto-bind-tunnel resolution any
                  no shutdown
vprn 1 customer 1 create
        route-distinguisher auto-rd
        interface "sbd" create
             ipv6
             exit
             vpls "sbd"
                  evpn-tunnel ipv6-gateway-address mac
```

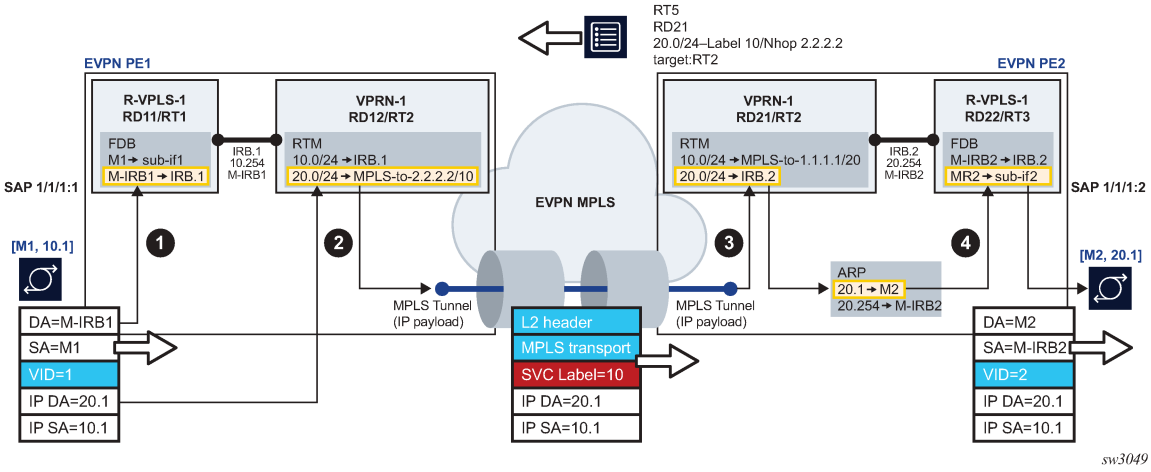
6.5.12.4 Interface-less IP-VRF-to-IP-VRF model (IP encapsulation) for MPLS tunnels

In addition to the Interface-ful and interoperable Interface-less models described in the previous sections, SR OS also supports Interface-less Model (EVPN IFL) with IP encapsulation for MPLS tunnels. In the standard specification - *draft-ietf-bess-evpn-ip-prefix* - this refers to the EVPN IFL model for IP NVO tunnels.

Compared to the Ethernet NVO option, the ingress PE no longer pushes an inner Ethernet header, but the IP packet is directly encapsulated with an EVPN service label and the transport labels.

Figure 206: Interface-less IP-VRF-to-IP-VRF model for IP encapsulation in MPLS tunnels illustrates the Interface-less Model (EVPN IFL) with IP encapsulation for MPLS tunnels.

Figure 206: Interface-less IP-VRF-to-IP-VRF model for IP encapsulation in MPLS tunnels



EVPN IFL uses EVPN IP Prefix routes to exchange prefixes between PEs without the need for an R-VPLS service, termed Supplementary Broadcast Domain (SBD) in the standards, and any destination MAC lookup. The data path used in EVPN IFL is the same as that is used for IP-VPN services in the VPRN.

In the example of Figure 206: Interface-less IP-VRF-to-IP-VRF model for IP encapsulation in MPLS tunnels:

1. PE2 advertises IP Prefix 20.0/24 (shorthand for 20.0.0.0/24) in an EVPN IP Prefix route that does not contain a Router's MAC extended community anymore. As usual, and depicted in step 1, arriving frames with IP destination of 20.0.0.1 on PE1's R-VPLS-1 are processed for a route lookup on VPRN-1.
2. However, in step 2 and as opposed to the previous models, the lookup yields a route-table entry that does not point at an SBD R-VPLS, but rather to an MPLS tunnel terminated on PE2. PE1 then pushes the EVPN service label that was received on the IP Prefix route at the top of the IP packet, and the packet is sent on the wire without any inner Ethernet header.
3. In step 3, the MPLS tunnel is terminated on PE2 and the EVPN label identifies the VPRN-1 service for a route lookup.
4. Step 4 corresponds to the regular R-VPLS forwarding that happens in the other EVPN L3 models.

A new **vprn**>**bgp-evpn**>**mpls** context has been added to configure a VPRN service for EVPN IFL. This context is like the one existing in VPLS and Epipe services and enables the use of EVPN IFL in the VPRN service. When configured, no R-VPLS with evpn-tunnel should be added to the VPRN, that is, no SBD is configured. As an example, in Figure 206: Interface-less IP-VRF-to-IP-VRF model for IP encapsulation in MPLS tunnels PE1 and PE2 VPRN-1 service are configured as follows:

```
[ex:configure service vprn "vprn-1"]
A:admin@PE1# info
    admin-state enable
    ecmp 2
    bgp-evpn {
        mpls 1 {
            admin-state enable
            route-distinguisher "192.0.2.1:12"
            vrf-target {
                community "target:64500:2"
            }
            auto-bind-tunnel {
                resolution any
            }
        }
    }
    interface "irb-1" {
        ipv4 {
            primary {
                address 10.0.0.254
                prefix-length 24
            }
        }
        vpls "r-vpls-1" {
        }
    }
```

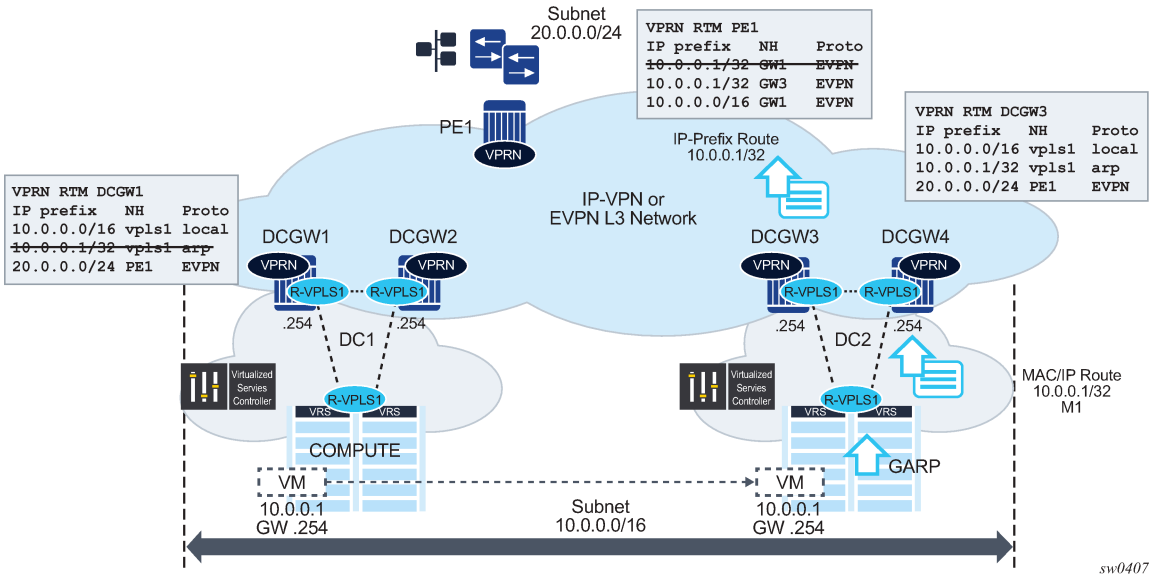
```
[ex:configure service vprn "vprn-1"]
A:admin@PE2# info
    admin-state enable
    ecmp 2
    bgp-evpn {
        mpls 1 {
            admin-state enable
            route-distinguisher "192.0.2.2:21"
            vrf-target {
                community "target:64500:2"
            }
            auto-bind-tunnel {
                resolution any
            }
```

```
        }
    }
    interface "irb-2" {
        ipv4 {
            primary {
                address 20.0.0.254
                prefix-length 24
            }
        }
        vpls "r-vpls-1" {
        }
    }
```

6.5.13 ARP-ND host routes for extended Layer 2 Data Centers

SR OS supports the creation of host routes for IP addresses that are present in the ARP or neighbor tables of a routing context. These host routes are referred to as ARP-ND routes and can be advertised using EVPN or IP-VPN families. A typical use case where ARP-ND routes are needed is the extension of Layer 2 Data Centers (DCs). Figure 207: Extended Layer-2 Data Centers illustrates this use case.

Figure 207: Extended Layer-2 Data Centers



Subnet 10.0.0.0/16 in Figure 207: Extended Layer-2 Data Centers is extended throughout two DCs. The DC gateways are connected to the users of subnet 20.0.0.0/24 on PE1 using IP-VPN (or EVPN). If the virtual machine VM 10.0.0.1 is connected to DC1, when PE1 needs to send traffic to host 10.0.0.1, it performs a Longest Prefix Match (LPM) lookup on the VPRN's route table. If the only IP prefix advertised by the four DC GWs was 10.0.0.0/16, PE1 could send the packets to the DC where the VM is not present.

To provide efficient downstream routing to the DC where the VM is located, DGW1 and DGW2 must generate host routes for the VMs to which they connect. When the VM moves to the other DC, DGW3 and DGW4 must be able to learn the VM's host route and advertise it to PE1. DGW1 and DGW2 must withdraw the route for 10.0.0.1, because the VM is no longer in the local DC.

In this case, the SR OS is able to learn the VM's host route from the generated ARP or ND messages when the VM boots or when the VM moves.

A route owner type called "ARP-ND" is supported in the base or VPRN route table. The ARP-ND host routes have a preference of 1 in the route table and are automatically created out of the ARP or ND neighbor entries in the router instance.

The following commands enable ARP-ND host routes to be created in the applicable route tables:

- **configure service vprn/ies interface arp-host-route populate** {**evpn** | **dynamic** | **static**}
- **configure service vprn/ies interface ipv6 nd-host-route populate** {**evpn** | **dynamic** | **static**}

When the command is enabled, the EVPN, dynamic and static ARP entries of the routing context create ARP-ND host routes in the route table. Similarly, ARP-ND host routes are created in the IPv6 route table out of static, dynamic, and EVPN neighbor entries if the command is enabled.

The **arp** and **nd-host-route populate** commands are used with the following features:

- **adding ARP-ND hosts**

  A route tag can be added to ARP-ND hosts using the **route-tag** command. This tag can be matched on BGP VRF export and peer export policies.

- **keeping entries active**

  The ARP-ND host routes are kept in the route table as long as the corresponding ARP or neighbor entry is active. Even if there is no traffic destined for them, the **arp-proactive-refresh** and **nd-proactive-refresh** commands configure the node to keep the entries active by sending an ARP refresh message 30 seconds before the **arp-timeout** or starting NUD when the stale time expires.

- **speeding up learning**

  To speed up the learning of the ARP-ND host routes, the **arp-learn-unsolicited** and **nd-learn-unsolicited** commands can be configured. When **arp-learn-unsolicited** is enabled, received unsolicited ARP messages (typically GARPs) create an ARP entry, and consequently, an ARP-ND route if **arp-populate-host-route** is enabled. Similarly, unsolicited Neighbor Advertisement messages create a stale neighbor. If **nd-populate-host-route** is enabled, a confirmation message (NUD) is sent for all the neighbor entries created as stale, and if confirmed, the corresponding ARP-ND routes are added to the route table.

Note: The ARP-ND host routes are created in the route table but not in the routing context FIB. This helps preserve the FIB scale in the router.

In Figure 207: Extended Layer-2 Data Centers, enabling **arp-host-route-populate** on the DCGWs allows them to learn or advertise the ARP-ND host route 10.0.0.1/32 when the VM is locally connected and to remove or withdraw the host routes when the VM is no longer present in the local DC.

ARP-ND host routes installed in the route table can be exported to VPN IPv4, VPN IPv6, or EVPN routes. No other BGP families or routing protocols are supported.

6.5.14 EVPN host mobility procedures within the same R-VPLS service

EVPN host mobility is supported in SR OS as in Section 4 of *draft-ietf-bess-evpn-inter-subnet-forwarding*. When a host moves from a source PE to a target PE, it can behave in one of the following ways:

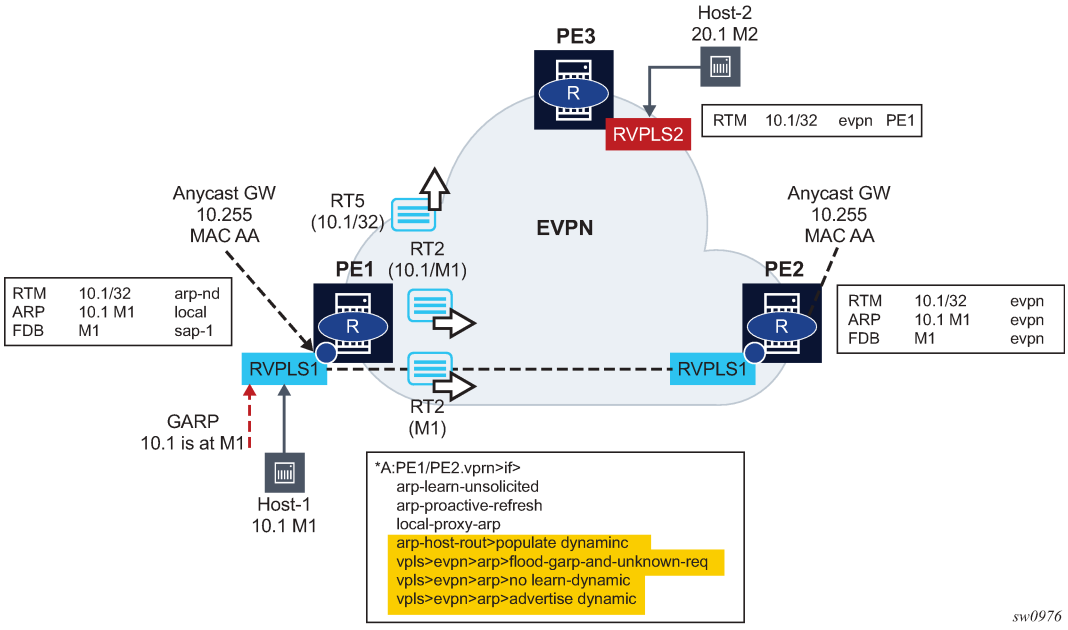
- The host initiates an ARP request or GARP upon moving to the target PE.
- The host sends a data packet without first initiating an ARP request of GARP.
- The host is silent.

The SR OS supports the above scenarios as follows.

6.5.14.1 EVPN host mobility configuration

Figure 208: Host mobility within the same R-VPLS – initial phase shows an example of a host connected to a source PE, PE1, that moved to the target, PE2. The figure shows the expected configuration on the VPRN interface, where R-VPLS 1 is attached (for both PE1 and PE2). PE1 and PE2 are configured with an "anycast gateway", that is, a VRRP passive instance with the same backup MAC and IP in both PEs.

Figure 208: Host mobility within the same R-VPLS – initial phase



In this initial phase:

1. PE1 learns Host-1 IP to MAC (10.1-M1) in the ARP table and generates a host route (RT5) for 10.1/32, because Host-1 is locally connected to PE1. In particular:
   - **arp-learn-unsolicited** triggers the learning of 10.1-M1 upon receiving a GARP from Host-1 or any other ARP
   - **arp-proactive-refresh** triggers the refresh of host-1's ARP entry 30 seconds before the entry ages out
   - **local-proxy-arp** makes sure PE1 replies to any received ARP request on behalf of other hosts in the R-VPLS
   - **arp-host-route populate dynamic** ensures that only the dynamically learned ARP entries create a host route, for example, 10.1
   - **no flood-garp-and-unknown-req** suppresses ARP flooding (from CPM) within the R-VPLS1 context and reduces significantly the unnecessary ARP flooding because the ARP entries are synchronized through EVPN
   - **advertise dynamic** triggers the advertisement of MAC/IP routes for the dynamic ARP entries, including the IP and MAC addresses, for example, 10.1-M1; a MAC/IP route for M1-only that has been previously advertised as M1 is learned on the FDB as local or dynamic

2. PE2 learns Host-1 10.1-M1 in the ARP and FDB tables as EVPN type. PE2 must not learn 10.1-M1 as dynamic, so that PE2 is prevented from advertising an RT5 for 10.1/32. If PE2 advertises 10.1/32, then PE3 could select PE2 as the next-hop to reach Host-1, creating an unwanted hair-pinning forwarding behavior. PE2 is expected to have the same configuration as PE1, including the following commands, as well as those described for PE1:
   - **no learn-dynamic** prevents PE2 from learning ARP entries from ARP traffic received on an EVPN tunnel.
   - **populate dynamic**, as in PE1, makes sure PE2 only creates route-table ARP-ND host routes for dynamic entries. Hence, 10.1-M1 does not create a host route as long as it is learned via EVPN only.

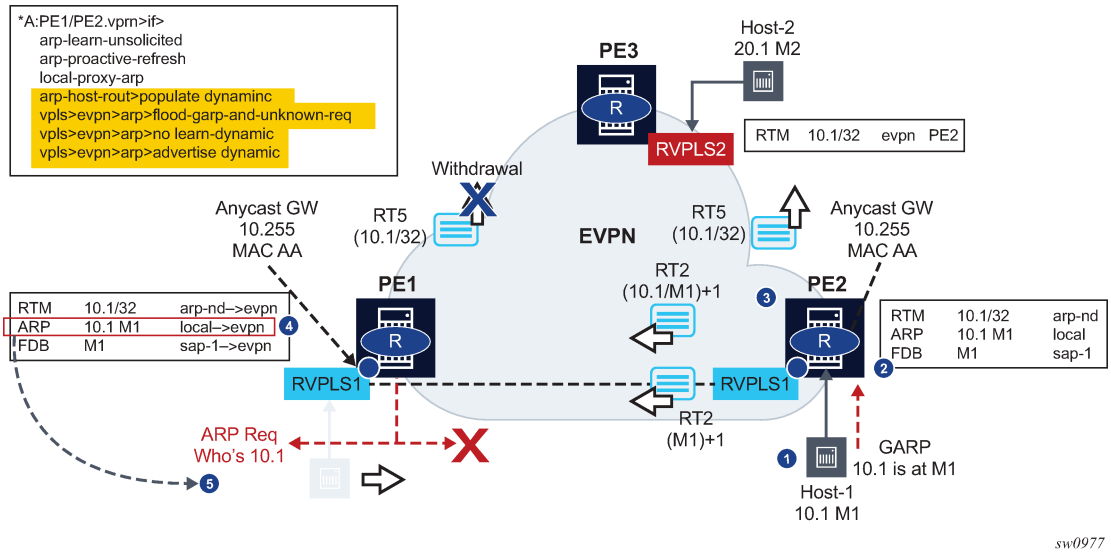
The configuration described in this section and the cases in the following sections are for IPv4 hosts, however, the functionality is also supported for IPv6 hosts. The IPv6 configuration requires equivalent commands, that use the prefix "nd-" instead of "arp-". The only exception is the **flood-garp-and-unknown-req** command, which does not have an equivalent command for ND.

6.5.14.1.1 Host initiates an ARP/GARP upon moving to the target PE

An example is illustrated in Figure 209: Host mobility within the same R-VPLS – move with GARP. This is the expected behavior based on the configuration described in EVPN host mobility configuration .

1. Host-1 moves from PE1 to PE2 and issues a GARP with 10.1-M1.
2. Upon receiving the GARP, PE2 updates its FDB and ARP table.
3. The route-table entry for 10.1/32 changes from EVPN to type **arp-nd** (based on populate dynamic), therefore, PE2 advertises a RT5 with 10.1/32. Also, M1 is now learned in FDB and ARP as local, therefore, MAC/IP routes with a higher sequence number are advertised (one MAC/IP route with M1 only and another one with 10.1-M1).
4. Upon receiving the routes, PE1:
   a. Updates its FDB and withdraws its RT2(M1) based on the higher SEQ number.
   b. Updates its ARP entry 10.1-M1 from dynamic to type **evpn**.
   c. Removes its **arp-nd** host from the route-table and withdraws its RT5 for 10.1/32 (based on **populate dynamic**).
5. The move of 10.1-M1 from **dynamic** to **evpn** triggers an ARP request from PE1 asking for 10.1. The **no flood-garp-and-unknown-req** command prevents PE1 from flooding the ARP request to PE2.

Figure 209: Host mobility within the same R-VPLS – move with GARP



After step 5, no one replies to PE1's ARP request and the procedure is over. If a host replied to the ARP for 10.1, the process starts again.

6.5.14.1.2 Host sends a data packet upon a move to target PE

In this case, the host does not send a GARP/ARP packet when moving to the target PE. Only regular data packets are sent. The steps are illustrated in Figure 210: Host mobility within the same R-VPLS – move with data packet.

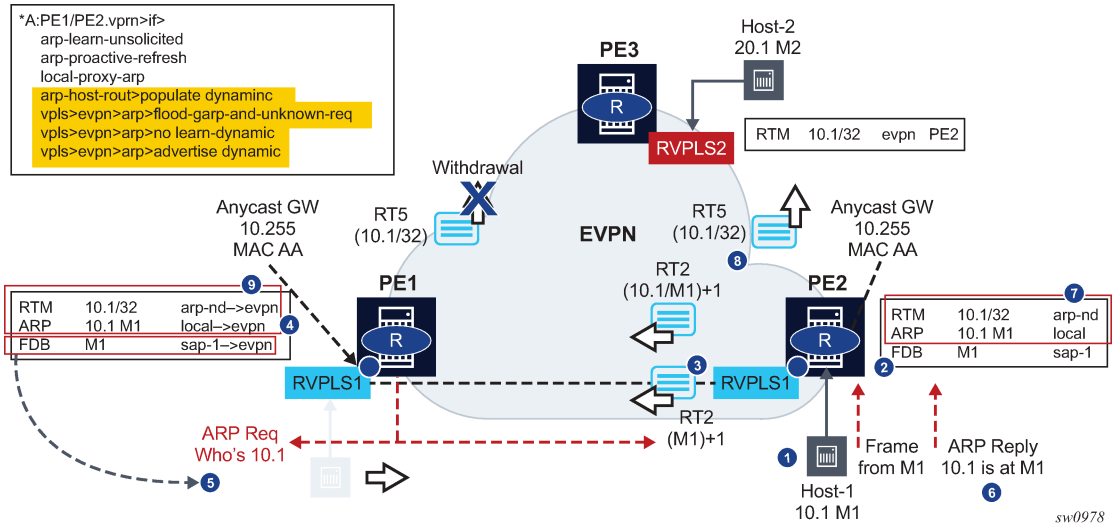
1. Host-1 moves from PE1 to PE2 and issues a (non-ARP) frame with MAC SA=M1.
2. When receiving the frame, PE2 updates its FDB and starts the mobility procedures for M1 (because it was previously learned from EVPN). At the same time, PE2 also creates a short-lived dynamic ARP entry for the host, and triggers an ARP request for it.
3. PE2 advertises a RT2 with M1 only, and a higher sequence number.
4. PE1 receives the RT2, updates its FDB and withdraws its RT2s for M1 (this includes the RT2 with M1-only and the RT2 with 10.1-M1).
5. PE1 issues an ARP request for 10.1, triggered by the update on M1.

   In this case, the PEs are configured with **flood-garp-and-unknown-req** and therefore, the generated ARP request is flooded to local SAP and SDP-binds and EVPN destinations. When the ARP request gets to PE2, it is flooded to PE2's SAP and SDP-binds and received by Host-1.
6. Host-1 sends an ARP reply that is snooped by PE2 and triggers a similar process described in Host initiates an ARP/GARP upon moving to the target PE (this is illustrated in the following).

   Because passive VRRP is used in this scenario, the ARP reply uses the anycast backup MAC that is consumed by PE2.
7. Upon receiving the ARP reply, PE2 updates its ARP table to **dynamic**.

8. Because the route-table entry for 10.1/32 now changes from EVPN to type **arp-nd** (based on **populate dynamic**), PE2 advertises a RT5 with 10.1/32. Also, M1 is now learned in ARP as local, therefore a RT2 for 10.1-M1 is sent (the sequence number follows the RT2 with M1 only).
9. Upon receiving the route, PE1:
   a. Updates the ARP entry 10.1-M1, from type **local** to type **evpn**.
   b. Removes its **arp-nd** host from the route-table and withdraws its RT5 for 10.1/32 (based on **populate dynamic**).

Figure 210: Host mobility within the same R-VPLS – move with data packet



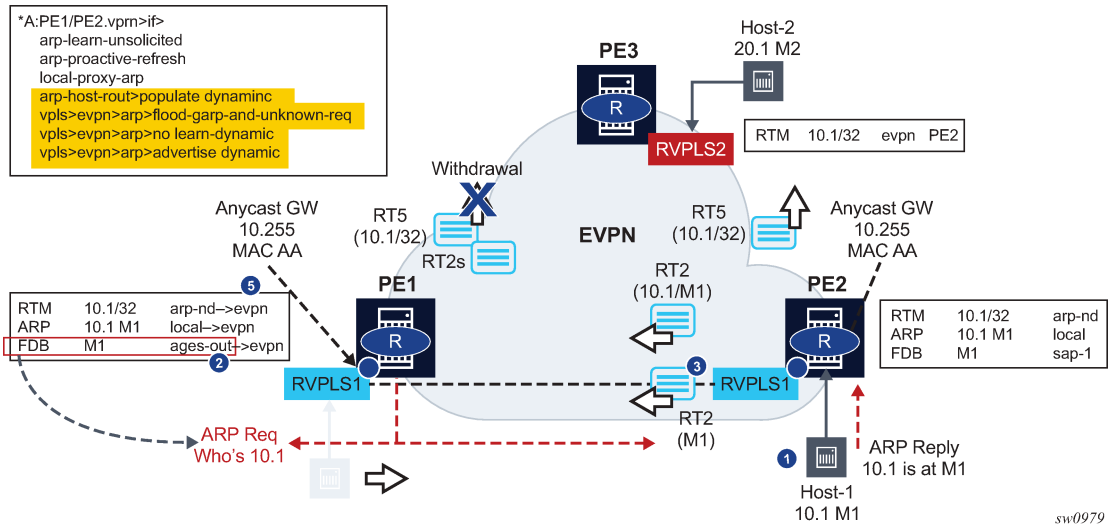
6.5.14.1.3 Silent host upon a move to the target PE

This case assumes the host moves but it stays silent after the move. The steps are illustrated in Figure 211: Host mobility within the same R-VPLS – silent host.

1. Host-1 moves from PE1 to PE2 but remains silent.
2. Eventually M1 ages out in PE1's FDB and the RT2s for M1 are withdrawn. This update on M1 triggers PE1 to issue an ARP request for 10.1.

   The **flood-garp-and-unknown-req** is configured. The ARP request makes it to PE2 and Host-1.
3. Host-1 sends an ARP reply that is consumed by PE2. FDB and ARP tables are updated.
4. The FDB and ARP updates trigger RT2s with M1-only and with 10.1-M1. Because an **arp-nd dynamic** host route is also created in the route-table, an RT5 with 10.1/32 is triggered.
5. Upon receiving the routes, PE1 updates FDB and ARP tables. The update on the ARP table from **dynamic** to **evpn** removes the host route from the route-table and withdraws the RT5 route.

Figure 211: Host mobility within the same R-VPLS – silent host



6.5.15 BGP and EVPN route selection for EVPN routes

When two or more EVPN routes are received at a PE, BGP route selection typically takes place when the route key or the routes are equal. When the route key is different, but the PE has to make a selection (for instance, the same MAC is advertised in two routes with different RDs), BGP hands over the routes to EVPN and the EVPN application performs the selection.

EVPN and BGP selection criteria are described below:

- **EVPN route selection for MAC routes**

  When two or more routes are received with the same **mac-length**/**mac** but different route key, BGP hands the routes over to EVPN. EVPN selects the route based on the following tiebreaking order:

  1. Conditional static MACs (local protected MACs)
  2. Auto-learned protected MACs (locally learned MACs on SAPs or mesh or spoke SDPs because of the configuration of **auto-learn-mac-protect**)
  3. EVPN ES PBR MACs (see ES PBR MAC routes below)
  4. EVPN static MACs (remote protected MACs)
  5. Data plane learned MACs (regular learning on SAPs or SDP bindings) and EVPN MACs with higher SEQ numbers. Learned MACs and EVPN MACs are considered equal if they have the same SEQ number.
  6. EVPN MACs with higher SEQ number
  7. EVPN E-tree root MACs
  8. EVPN non-RT-5 MACs (this tie-breaking rule is only observed if the selection algorithm is comparing received MAC routes and internal MAC routes derived from the MACs in IP-Prefix routes, for example, RT-5 MACs)
  9. Lowest IP (next-hop IP of the EVPN NLRI)
  10. Lowest Ethernet tag (that is zero for MPLS and may be different from zero for VXLAN)

11. Lowest RD
12. Lowest BGP instance (this tie-breaking rule is only considered if the above rules fail to select a unique MAC and the service has two BGP instances of the same encapsulation)

- **ES PBR MAC routes**

  When a PBR filter with a forward action to an ESI and SF-IP (Service Function IP) exists, a MAC route is created by the system. This MAC route is compared to other MAC routes received from BGP.

  – When ARP resolves (it can be static, EVPN, or dynamic) for a SF-IP and the system has an AD EVI route for the ESI, a "MAC route" is created by ES PBR with the <MAC Address = ARPed MAC Address, VTEP = AD EVI VTEP, VNI = AD EVI VNI, RD = ES PBR RD (special RD), Static = 1> and installed in EVPN.

  – This MAC route does not add anything (back) to ARP; however, it goes through the MAC route selection in EVPN and triggers the FDB addition if it is the best route.

  – In terms of priority, this route's priority is lower than local static but higher than remote EVPN static (number 2 in the tiebreaking order above).

  – If there are two competing ES PBR MAC routes, then the selection goes through the rest of checks (Lowest IP > Lowest RD).

- **EVPN route selection for IP-prefix and IPv6-prefix routes**

  See Route selection across EVPN-IFL and other owners in the VPRN service.

- **BGP route selection**

  The BGP route selection for MAC routes with the same route-key follows the following priority order:

  1. EVPN static MACs (remote protected MACs).
  2. EVPN MACs with higher sequence number.
  3. Regular BGP selection (local-pref, aigp metric, shortest as-path, lowest IP).

  Regular BGP selection is followed for the rest of the EVPN routes.

Note: In case BGP has to run an actual selection and a specified (otherwise valid) EVPN route 'loses' to another EVPN route, the non-selected route is displayed by the **show router BGP routes evpn x detail** command with a tie-breaker reason.

Note: Protected MACs do not overwrite EVPN static MACs; in other words, if a MAC is in the FDB and protected because being received with the sticky/static bit set in a BGP EVPN update and a frame is received with the source MAC on an object configured with **auto-learn-mac-protect**, that frame is dropped because of the implicit **restrict-protected-src discard-frame**. The reverse is not true; when a MAC is learned and protected using **auto-learn-mac-protect**, its information is not overwritten with the contents of a BGP update containing the same MAC address.

6.5.16 LSP tagging for BGP next-hops or prefixes and BGP-LU

It is possible to constrain the tunnels used by the system for resolution of BGP next-hops or prefixes and BGP labeled unicast routes using LSP administrative tags. For more information, see the *7450 ESS, 7750 SR, 7950 XRS, and VSR MPLS Guide*, "LSP Tagging and Auto-Bind Using Tag Information".

6.5.17 Oper-groups interaction with EVPN services

Operational groups, also referred to as oper-groups, are supported in EVPN services. In addition to supporting SAP and SDP-binds, oper-groups can also be configured under the following objects:

- EVPN-VXLAN instances (except on Epipe services)
- EVPN-MPLS instances
- Ethernet segments

These oper-groups can be monitored in LAGs or service objects. Oper-groups are particularly useful for the following applications:

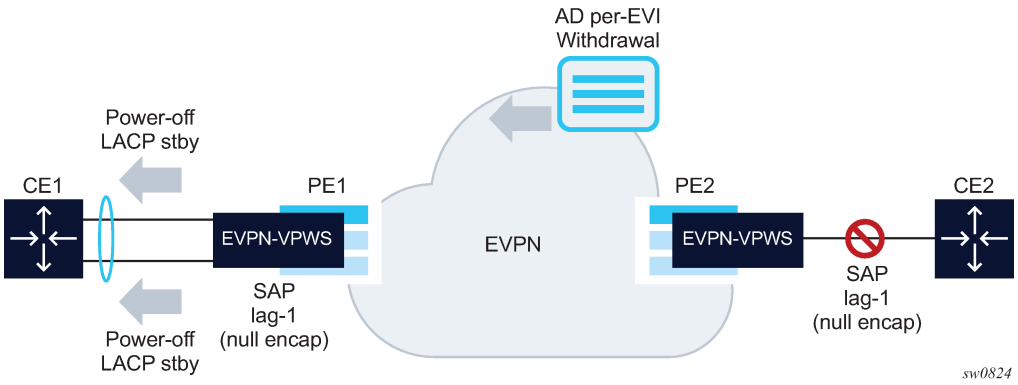
- Link Loss Forwarding (LLF) for EVPN VPWS services
- core isolation blackhole avoidance
- LAG standby signaling to CE on non-DF EVPN PEs (single-active)

6.5.17.1 LAG-based LLF for EVPN-VPWS services

SR OS uses Eth-CFM fault-propagation to support CE-to-CE fault propagation in EVPN-VPWS services. That is, upon detecting a CE failure, an EVPN-VPWS PE withdraws the corresponding Auto-Discovery per-EVI route, which then triggers a down MEP on the remote PE that signals the fault to the connected CE. In cases where the CE connected to EVPN-VPWS services does not support Eth-CFM, the fault can be propagated to the remote CE by using LAG standby-signaling, which can be LACP-based or simply power-off.

Figure 212: Link loss forwarding for EVPN-VPWS shows an example of link loss forwarding for EVPN-VPWS.

Figure 212: Link loss forwarding for EVPN-VPWS



In this example, PE1 is configured as follows:

```
A:PE1>config>lag(1)# info
----------------------------------------------
mode access
encap-type null
port 1/1/1
port 1/1/2
standby-signaling power-off
monitor-oper-group "llf-1"
no shutdown
```

```
-----------------------------------------------
*A:PE1>config>service>epipe# info
-----------------------------------------------
bgp
exit
bgp-evpn
    evi 1
    local-attachment-circuit ac-1
        eth-tag 1
        exit
    remote-attachment-circuit ac-2
        eth-tag 2
        exit
    mpls bgp 1
        oper-group "llf-1"
        auto-bind-tunnel
            resolution any
        exit
        no shutdown
    exit
sap lag-1 create
no shutdown
exit
no shutdown
```

The following applies to the PE1 configuration:

- The EVPN Epipe service is configured on PE1 with a null LAG SAP and the oper-group "llf-1" under **bgp-evpn**>**mpls**. This is the only member of oper-group "llf-1".

  Note: Do not configure the oper-group under **config**>**service**>**epipe**, because circular dependencies are created when the access SAPs go down because of the LAG **monitor-oper-group** command.

- The operational group monitors the status of the BGP-EVPN instance in the Epipe service. The status of the BGP-EVPN instance is determined by the existence of an EVPN destination at the Epipe.
- The LAG, in **access** mode and encap-type **null**, is configured with the command **monitor-oper-group** "llf-1".

  Note: The **configure**>**lag**>**monitor-oper-group name** command is only supported in **access** mode. Any **encap-type** can be used.

As shown in Figure 212: Link loss forwarding for EVPN-VPWS, upon failure on CE2, the following events occur:

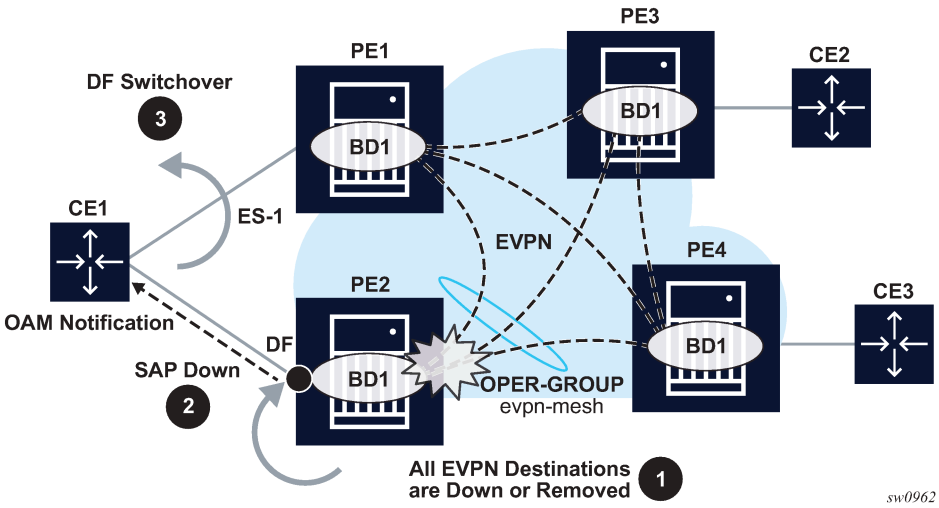
1. PE2 withdraws the EVPN route.
2. The EVPN destination is removed in PE1 and oper-group "llf-1" also goes down.
3. Because lag-1 is monitoring "llf-1", the oper-group that is becoming inactive triggers standby signaling on the LAG; that is, power-off or LACP out-of-sync signaling to the CE1.

   When the SAP or port is down because of the LAG monitoring of the oper-group, PE1 does not trigger an AD per-EVI route withdrawal, even if the SAP is brought operationally down.
4. After CE2 recovers and PE2 re-advertises the AD per-EVI route, PE1 creates the EVPN destination and oper-group "llf-1" comes up. As a result, the monitoring LAG stops signaling standby and the LAG is brought up.

6.5.17.2 Core isolation blackhole avoidance

Figure 213: Core isolation blackhole avoidance shows how blackholes can be avoided when a PE becomes isolated from the core.

Figure 213: Core isolation blackhole avoidance



In this example, consider that PE2 and PE1 are single-active multihomed to CE1. If PE2 loses all its core links, PE2 must somehow notify CE1 so that PE2 does not continue attracting traffic and so that PE1 can take over. This notification is achieved by using oper-groups under the BGP-EVPN instance in the service. The following is an example output of the PE2 configuration.

```
*[ex:configure service vpls ”evi1"]
A:admin@PE-2# info
    admin-state enable
    bgp-evpn {
        evi 1
        mpls 1 {
            admin-state enable
            oper-group “evpn-mesh”
            auto-bind-tunnel {
                resolution any
            }
        }
    }
    sap lag-1:351 {
        monitor-oper-group “evpn-mesh”
            }
*[ex:configure service oper-group ”evpn-mesh"]
A:admin@PE-2# info detail
    hold-time {
        up 4
    }
```

With the PE2 configuration and Figure 213: Core isolation blackhole avoidance example, the following steps occur:

1. PE2 loses all its core links, therefore, it removes its EVPN-MPLS destinations. This causes oper-group "evpn-mesh” to go down.

2. Because PE2 is the DF in the Ethernet Segment (ES) ES-1 and sap lag-1:351 is monitoring the oper-group, the SAP becomes operationally down. If ETH-CFM fault propagation is enabled on a down MEP configured on the SAP, CE1 is notified of the failure.
3. PE1 takes over as the DF based on the withdrawal of the ES (and AD) routes from PE2, and CE1 begins sending traffic immediately to PE1 only, therefore, avoiding a traffic blackhole.

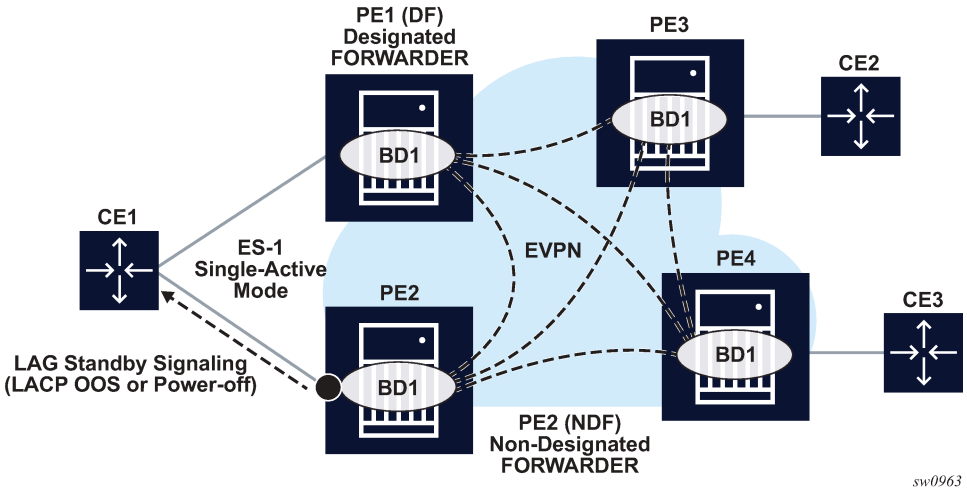
Generally, when oper-groups are associated with EVPN instances:

- The oper-group state is determined by the existence of at least one EVPN destination in the EVPN instance.
- The oper-group that is configured under a BGP EVPN instance cannot be configured under any other object (for example, SAP, SDP binding, and so on) of the same or different service.
- The status of an oper-group associated with an EVPN instance does not go down if all the EVPN destinations are operationally down due to a control-word or MTU mismatch.
- The status of an oper-group associated with an EVPN instance goes down in the following cases:
  - the service admin-state is disabled (only for VPLS services, not for Epipes)
  - the BGP EVPN VXLAN or MPLS admin-state are disabled
  - there are no EVPN destinations associated with the instance

6.5.17.3 LAG or port standby signaling to the CE on non-DF EVPN PEs (single-active)

As described in EVPN for MPLS tunnels, EVPN single-active multihoming PEs that are elected as non-DF must notify their attached CEs so the CE does not send traffic to the non-DF PE. This can be performed on a per-service basis that is based on the ETH-CFM and fault-propagation. However, sometimes ETH-CFM is not supported in multihomed CEs and other notification mechanisms are needed, such as LACP standby or power-off. This scenario is shown in the following figure.

Figure 214: LACP standby signaling from the non-DF



As shown in the preceding figure, the multihomed PEs are configured with multiple EVPN services that use ES-1. ES-1 and its associated LAG is configured as follows:

```
*[ex:configure lag 1]
```

```
A:admin@PE-2# info
    admin-state enable
    standby-signaling {power-off|lacp}
    monitor-oper-group ”DF-signal-1"
    mode access
    port 1/1/c2/1 {
    }
<snip>
ex:configure service system bgp evpn]
A:admin@PE-2# info
    ethernet-segment "ES-1" {
        admin-state enable
        esi 0x01010000000000000000
        multi-homing-mode single-active
        oper-group “DF-signal-1”
        association {
            lag 1 {
            }
<snip>
```

When the operational group is configured on the ES and monitored on the associated LAG:

- The operational group status is driven by the ES DF status (defined by the number of DF SAPs or oper-up SAPs owned by the ES).
- The operational group goes down if all the SAPs in the ES go down (this happens in PE2 in Figure 214: LACP standby signaling from the non-DF). The ES operational group goes up when at least one SAP in the ES goes up.

  As a result, if PE2 becomes non-DF on all the SAPs in the ES, they all go operationally down, including the ES-1 operational group.
- Because LAG-1 is monitoring the operational group, when its status goes down, LAG-1 signals LAG standby state to the CE. The standby signaling can be configured as LACP or power-off.
- The ES and AD routes for the ES are not withdrawn because the router recognizes that the LAG becomes standby for the ES operational group.

If the Single-Active ES is associated with a port instead of a LAG, the **config**>**port**> **monitor-oper-group** *DF-signal-1* command can be configured. In this case, the port monitors the ES operational group and the following rules apply:

- As in the case of the LAG, if the ES goes non-DF, its operational group also goes down.
- The port that is monitoring the ES operational group signals standby state by powering off the port itself.
- As in the case of the LAG, the ES and AD routes for the ES are not withdrawn because the router recognizes that the port is in standby state because of the ES operational group.

Operational groups cannot be assigned to ESs that are configured as **virtual**, **all-active** or **service-carving mode auto**.

6.5.17.4 AC-Influenced DF Election Capability on an ES with oper-group

The Attachment Circuit Influenced (AC-Influenced) Designated Forwarder Election Capability (AC-DF), as described in RFC8584, is supported in SR OS. By default, the **ac-df-capability** command is set to the **include** option. This configuration addresses the need to consider EVPN Auto-discovery per EVI/ES (AD per EVI/ES) routes for a specific PE, which ensures that the PE is included on the candidate DF list.

Configuring **ac-df-capability** to **exclude** disables the AC-DF capability. When **ac-df-capability exclude** is configured on a specific ES, the presence or absence of the AD per EVI/ES routes from the ES peers does

not modify the DF Election candidate list for the ES. The **exclude** option is recommended in ESs that use an **oper-group**, that is monitored by the access LAG, to signal standby **lacp** or **power-off**, as described in LAG or port standby signaling to the CE on non-DF EVPN PEs (single-active). All PE routers attached to the same ES must be configured consistently for the specific **ac-df-capability**.

6.5.18 EVPN Layer 3 OISM

Optimized Inter-Subnet Multicast (OISM) is an EVPN-based solution that optimizes the forwarding of IP multicast across R-VPLS of the same or a different subnet. EVPN OISM is supported for EVPN-MPLS and EVPN-VXLAN services, IPv4 and IPv6 multicast groups, and is described in this section.

6.5.18.1 Introduction and terminology

EVPN OISM is similar to Multicast VPNs (MVPN) in some aspects, because it does IP multicast routing in VPNs, uses MP-BGP to signal the interest of a PE in a specified multicast group and uses Provider Multicast Service Interface (PMSI) trees among the PEs to send and receive the IP multicast traffic.

However, OISM is simpler than MVPN and allows efficient multicast in networks that integrate Layer 2 and Layer 3; that is, networks where PEs may be attached to different subnets, but could also be attached to the same subnet.

OISM is simpler than MVPN in some aspects:

- it does not need to setup shared trees (that need to switchover to shortest path trees)
- it does not require of the MVPN Any Source Multicast (ASM) complex procedures or the Rendezvous Point (RP) function
- it does not require Upstream Multicast Hop (UMH) selection and therefore does not have the UMH potential issues and limitations described in RFC6513 and RFC6514
- multiple PEs can be attached to the same Receiver subnet or Source subnet, which provides full flexibility when designing the multicast network

EVPN OISM is defined by *draft-ietf-bess-evpn-irb-mcast* and uses the following terminology that is also used in the rest of this section:

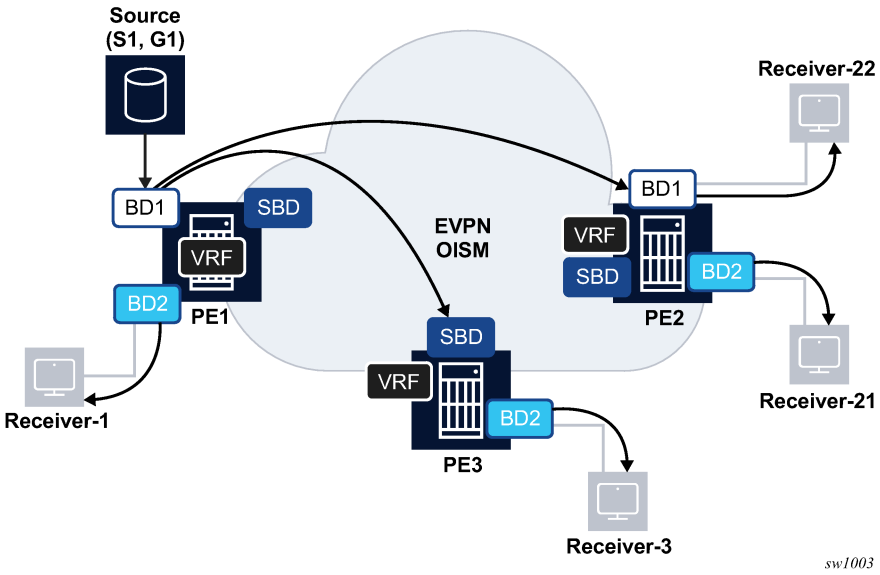
BD with IRB	Broadcast Domain with an Integrated Routing and Bridging interface. It is an R-VPLS service in SR OS.
Ordinary BD	refers to an R-VPLS where sources or receivers, or both, are connected
SBD	Supplementary Broadcast Domain. It is a backhaul R-VPLS that connects the PEs' VPRN services and is configured as an evpn-tunnel interface in the VPRN services. The SBD is mandatory in OISM and is needed to receive multicast traffic on the PEs that are not attached to the source ordinary BD.
EVPN Tenant Domain	refers to the group of BDs and IP-VRFs (VPRNs) of the same tenant
SMET route or EVPN route type 6	the EVPN route that the PEs use to signal interest for a specific multicast group (S ,G) or (*,G)
IIF and OIF	refers to Incoming Interface and Outgoing Interface. A multicast enabled VPRN has Layer 3 IIF and OIFs. A multicast enabled R-VPLS have Layer 2 OIFs.

Upstream and Downstream PEs	refers to the PEs that are connected to sources and receivers respectively
I-PMSIs and S-PMSIs	refers to Inclusive and Selective (Provider Multicast Service Interface) trees. The inclusive trees are signaled via IMET routes and include all the PEs attached to the service. The selective trees are signaled via S-PMSI A-D routes, and only the downstream PEs with receivers for the group signaled by the S-PMSI A-D route join the tree.
S-PMSI A-D route or EVPN route type 10	Selective Provider Multicast Service Interface (S-PMSI) Auto-Discovery route, the EVPN route that the root PEs use to signal S-PMSI trees, when the root PE decides that setting up a specific tree for a specific (S,G) or (*G) is needed.

6.5.18.2 OISM forwarding plane

In an EVPN OISM network, it is assumed that the sources and receivers are connected to ordinary BDs and EVPN is the only multicast control plane protocol used among the PEs. Also, the subnets (and optionally hosts) are advertised normally by the EVPN IP Prefix routes. The IP-Prefix routes are installed in the PEs' VPRN route tables and are used for multicast RPF checks when routing multicast packets. Figure 215: EVPN OISM forwarding plane illustrates a simple EVPN OISM network.

Figure 215: EVPN OISM forwarding plane



In Figure 215: EVPN OISM forwarding plane, and from the perspective of the multicast flow (S1,G1), PE1 is considered an upstream PE, whereas PE2 and PE3 are downstream PEs. The OISM forwarding rules are as follows.

- On the upstream PE (PE1), the multicast traffic is sent to local receivers irrespective of the receivers being attached to the source BD (BD1) or not (BD2).

Note: OISM does not use any multicast Designated Router (DR) concept, therefore the upstream PE always routes locally as long as it has local receivers.

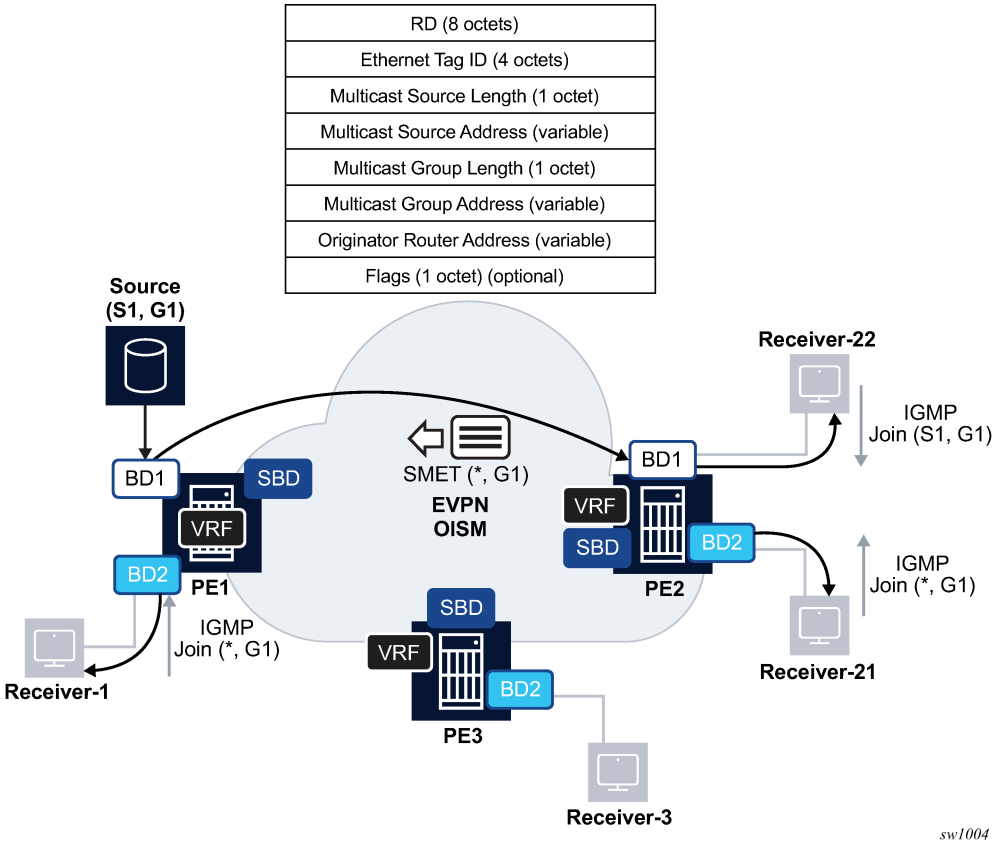
- On downstream PEs that are attached to the source BD (PE2), the multicast traffic is always received on the source BD (BD1) and forwarded locally to receivers in the same or different ordinary BD (as in the case of Receiver-22 or Receiver-21). Multicast traffic received on this PE is never sent back to the SBD or remote EVPN PEs.
- On downstream PEs that are not attached to the source BD (PE3), the multicast traffic is always received on the SBD and sent to local receivers. Multicast received on this PE is never sent to remote EVPN PEs.

Note: In order for PE3 to receive the multicast traffic on the SBD, the source PE, PE1, forms an EVPN destination from BD1 to PE3's SBD. This EVPN destination on PE1 is referred to as an SBD destination.

6.5.18.3 OISM control plane

OISM uses the Selective Multicast Ethernet Tag (SMET) route or route type 6 to signal interest on a specific (S,G) or (*,G). Figure 216: Use of the SMET route provides an example.

Figure 216: Use of the SMET route



As shown in Figure 216: Use of the SMET route, a PE with local receivers interested in a multicast group G1 issues an SMET route encoding the source and group information (upon receiving local IGMP join messages for that group). EVPN OISM uses the SMET route in the following way:

- A route type-6 (SMET) can carry information for IPv4 or IPv6 multicast groups, for (S,G) or (*,G) or even wildcard groups (*,*).

  Note: MVPN uses different route types or even families to address the different multicast group types.

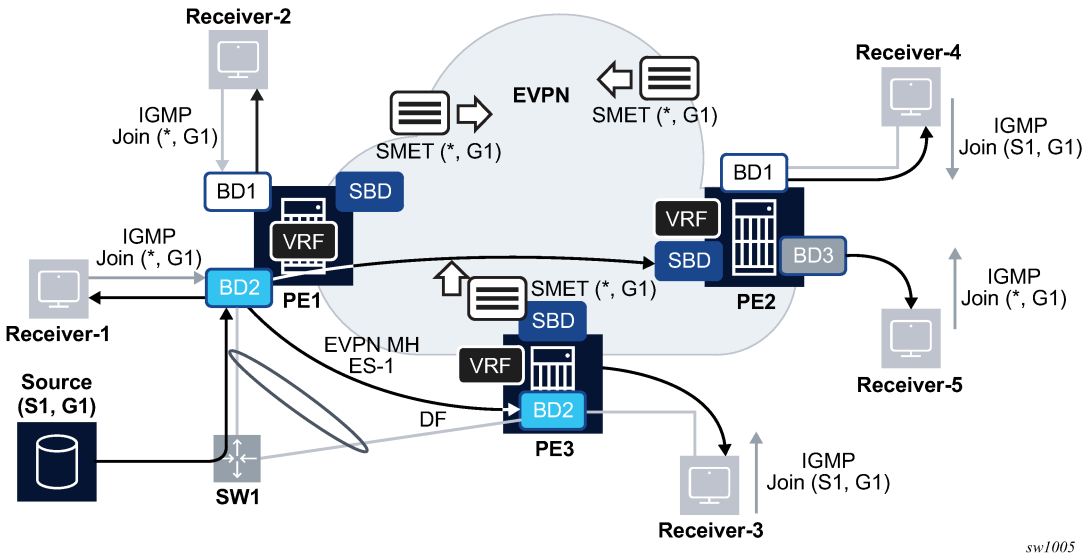
- The SMET routes are advertised with the route-target of the SBD, that guarantees that the SMET routes are imported by all the PEs of the tenant.
- The SMET routes also help minimize the control plane overhead because they aggregate the multicast state created on the downstream PEs. This is illustrated in Figure 216: Use of the SMET route, where PE2 sends the minimum number of SMET routes to pull multicast traffic for G1. That is, if PE2 has state for (S1,G1) and (*,G1), the SMET route for (*,G1) is enough to attract the multicast traffic required by the local receivers. There is no need to send an SMET route for (S1,G1) and a different route for (*,G1). Only (*,G1) SMET route is advertised.
- The SMET routes also provide an implicit S-PMSI (Selective Provider Multicast Service Interface) tree in case Ingress Replication is used to transport IP multicast. That is, PE1 sends the multicast traffic only to the PEs requesting it, for example, PE2 and not to PE3. In MVPN, even for Ingress Replication, a separate S-PMSI tree is setup to avoid PE1 from sending multicast to PE3.

6.5.18.4 EVPN OISM and multihoming

EVPN OISM supports multihomed multicast sources and receivers.

While MVPN requires complex UMH (Upstream Multicast Hop) selection procedures to provide multihoming for sources, EVPN simply reuses the existing EVPN multihoming procedures. Figure 217: EVPN OISM and multihomed sources illustrates an example of a multihomed source that makes use of EVPN all-active multihoming.

Figure 217: EVPN OISM and multihomed sources



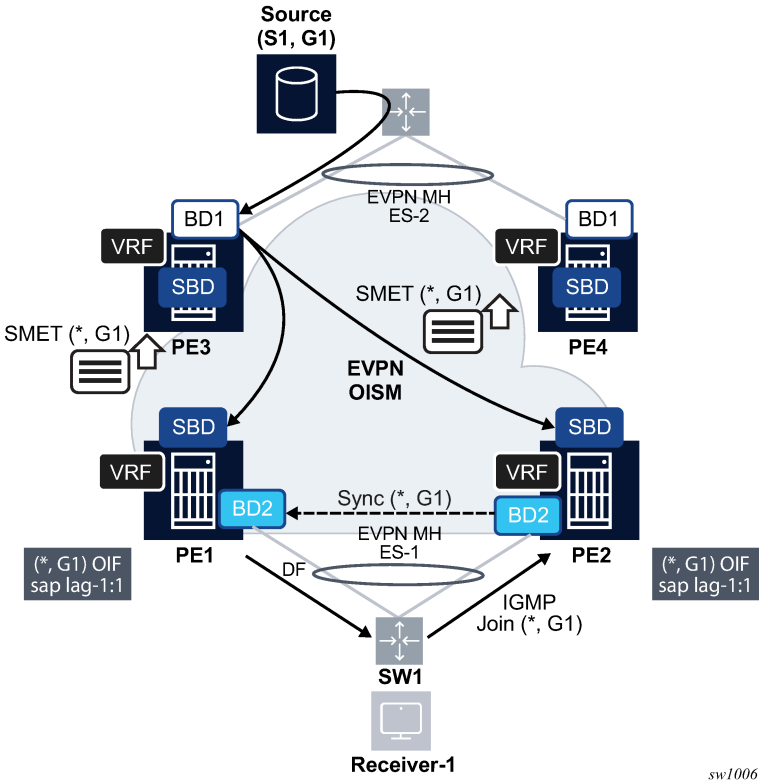
The source S1 is attached to a switch SW1 that is connected via single LAG to PE1 and PE2, a pair of EVPN OISM PEs. PE1 and PE2 define Ethernet Segment ES-1 for SW1, where ES-1 is all-active in this

case (single-active multihoming being supported too). Even in case of all-active, the multicast flow for (S1,G1) is only sent to one OISM PE, and the regular all-active multihoming procedures (Split-Horizon) make sure that PE3 does not send the multicast traffic back to SW1. This is true for EVPN-MPLS and EVPN-VXLAN BDs.

Convergence, in case of failure, is very fast because the downstream PEs, for example, PE2, advertise the SMET route for (*,G1) with the SBD route target and it is imported by both PE1 and PE3. In case of failure on PE2, PE3 already has state for (*,G1) and can forward the multicast traffic immediately.

EVPN OISM also supports multihomed receivers. Figure 218: EVPN OISM and multihomed receivers illustrates an example of multihomed receivers.

Figure 218: EVPN OISM and multihomed receivers



Multi-homed receivers as depicted in Figure 218: EVPN OISM and multihomed receivers, require the support of multicast state synchronization on the multihoming PEs to avoid blackholes. As an example, consider that SW1 hashes an IGMP join (*,G1) to PE2, and PE2 adds the ES-1 SAP to the OIF list for (*,G1). Consider PE1 is the ES-1 DF. Unless the (*,G1) state is synchronized on PE1, the multicast traffic is pulled to PE2 only and then discarded. The state synchronization on PE1 pulls the multicast traffic to PE1 too, and PE1 forwards to the receiver using its DF SAP.

In SR OS, the IGMP/MLD-snooping state is synchronized across ES peers using EVPN Multicast Synch routes, as specified in RFC 9251.

The same mechanism must be used in all the PEs attached to the same Ethernet Segment. MCS takes precedence when both mechanisms are simultaneously used.

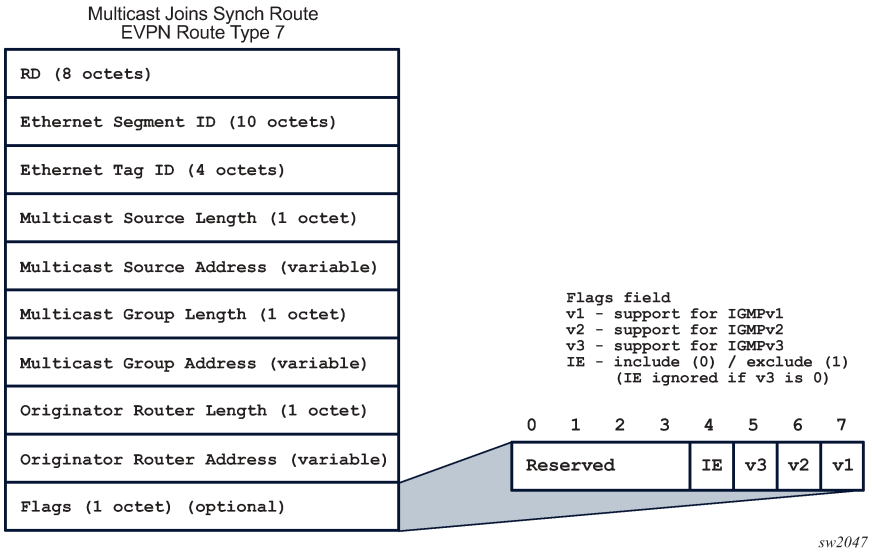
Note: The use of Multi-Chassis Synchronization (MCS) protocol is not supported in VPLS services in OISM mode or evpn-proxy mode.

EVPN Multicast Synch routes are supported as specified in RFC 9251 for OISM services too. They use EVPN route types 7 and 8, and are known as the Multicast Join Synch and Multicast Leave Synch routes, respectively.

When a PE that is attached to an EVPN Ethernet Segment receives an IGMP or MLD join, it creates multicast state and advertises a Multicast Join Synch route so that the peer ES PEs can synchronize the state. Similarly, when a PE in the Ethernet Segment receives a leave message, it advertises a Multicast Leave Synch route so that all the PEs in the Ethernet Segment can synchronize the Last Member Query procedures.

The Multicast Join Synch route or EVPN route type 7 is similar to the SMET route, but also includes the ESI. The Multicast Join Synch route indicates the multicast group that must be synchronized in all objects of the Ethernet Segment. Figure 219: Multicast join synch route depicts the format of the Multicast Join Synch route.

Figure 219: Multicast join synch route



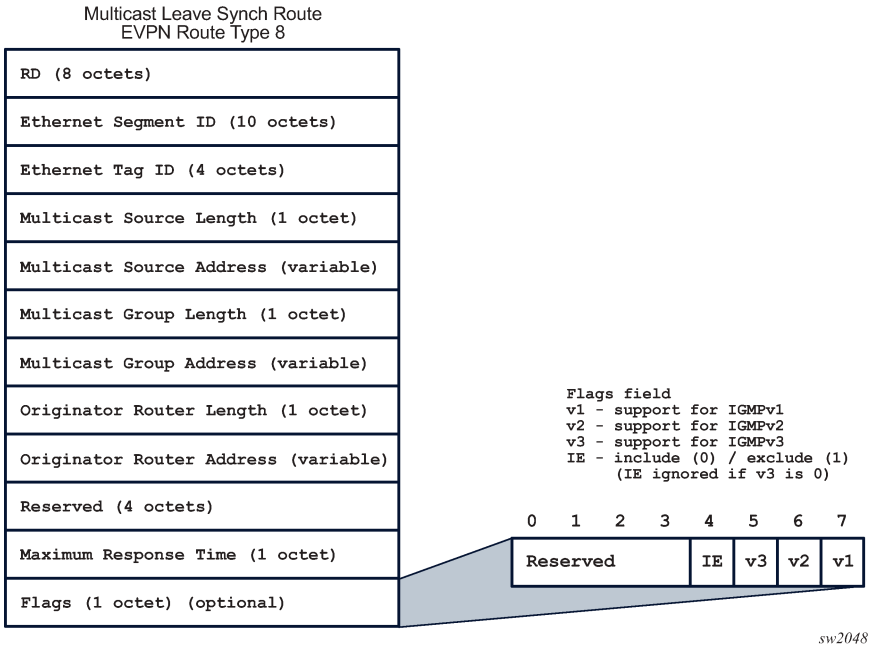
In accordance with RFC 9251, the following rules pertain:

- All fields except for the Flags are part of the route key for BGP processing purposes.
- Synch routes are resolved by BGP auto-bind resolution, as any other service route.
- The Flags are advertised and processed based on the received IGMP or MLD report that triggered the advertisement of the route (this includes the versions for IGMP or MLD and Include/Exclude bit for IGMPv3).
- The Route Distinguisher (RD) is the service RD.
- This route is only distributed to the ES peers - it is advertised with the ES-import route target, which limits its distribution to ES peers only.
- In addition, the route is sent with one EVI-RT extended community. The EVI-RT EC does not use a route target type/sub-type, therefore, it does not affect the distribution of the route, for example, it is not

considered for route target constraint filtering; only the ES-import route target is. However, its value is still taken from the configured service route target or EVI auto-derived route target.

The Multicast Leave Synch route or EVPN route type 8 indicates the multicast group Leave states that must be synchronized in all objects of the Ethernet Segment. Figure 220: Multicast leave synch route depicts the format of the Multicast Leave Synch route.

Figure 220: Multicast leave synch route



In accordance with RFC 9251, the following rules pertain:

- All fields except for the Flags, the Maximum Response Time and "reserved" field are part of the route key for BGP processing purposes.
- Synch routes are resolved by BGP auto-bind resolution, as any other service route.
- The Flags are generated based on the version of the leave message that triggered the advertisement of the route.
- As with the Multicast Join Synch route, this is a service level route sent with one ES-import route target and one EVI-RT EC. RD, Flags, ES-import and EVI-RT EC are advertised and processed in the same way as for the Multicast Join Synch route.

The EVI-RT is automatically added to the routes type 7 and 8, depending on the type of route target being configured on the service.

- If the service is configured with **target:2byte-asnumber:ext-comm-val** as route target, an EVI-RT type 0 is automatically added to routes type 7 and 8. No route target (other than the ES-import route target) is added to the route.
- If the service is configured with **target:ip-addr:comm-val** as route target, an EVI-RT type 1 is automatically added to routes type 7 and 8. No route target (other than the ES-import route target) is added to the route.

- If the service is configured with **target:4byte-asnumber:comm-val** as route target, an EVI-RT type 2 is automatically added to routes type 7 and 8. No route target (other than the ES-import route target) is added to the route.
- If auto-derived service RTs are used in the service, the corresponding operating route target is used as the EVI-RT.
- EVI-RT type 3 is not supported (type 3 is specified in RFC 9251).
- In general, **vsi-import** and **vsi-export** must not be used in OISM mode services or when the Multicast Synch routes are used. Using **vsi-import** or **vsi-export** policies instead of the **route target** command or the EVI-derived route target leads to issues when advertising and processing the Multicast Synch routes.

The following are additional considerations about the Multicast Synch routes:

- The routes are advertised without the need to configure any command as long as **igmp-snooping** or **mld-snooping** are enabled on an R-VPLS in OISM mode attached to a regular or virtual Ethernet Segment.
- The reception of Multicast Join or Leave Synch routes triggers the synchronization of states and the associated procedures in RFC 9251.
- Upon receiving a Leave message, the triggered Multicast Synch route encodes the configured Last Member Query interval times robust-count (LMQ × **robust-count**) in the Maximum Response Time field. The local PE expires the multicast state after the usual time plus an additional time that accounts for the BGP propagation to the remote ES peers and can be configured with the following command.

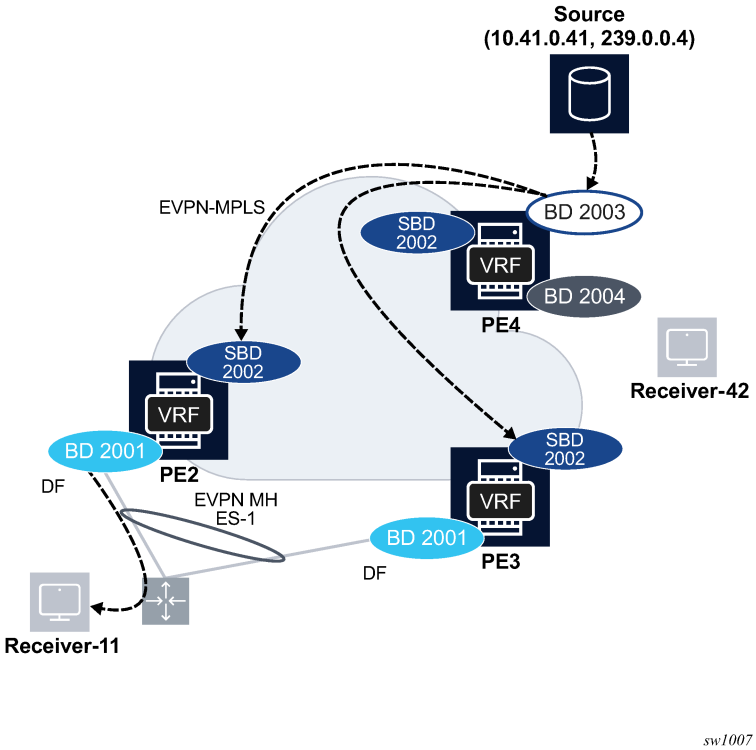
```
configure service system bgp-evpn multicast-leave-sync-propagation
```

This timer value should be configured the same in all the PEs attached to the same ES.

6.5.18.5 EVPN OISM configuration guidelines

This section shows a configuration example for the network illustrated in Figure 221: EVPN OISM example.

Figure 221: EVPN OISM example



The following CLI excerpt shows the configuration required on PE4 for services 2000 (VPRN), BD-2003 and BD-2004 (ordinary BDs) and BD-2002 (SBD).

```
vprn 2000 name "tenant-2k" customer 1 create
    route-distinguisher auto-rd
    interface "bd-2003" create
        address 10.41.0.1/24
        vpls "bd-2003"
        exit
    exit
    interface "bd-2004" create
        address 10.42.0.1/24
        vpls "bd-2004"
        exit
    exit
    interface "bd-2002" create
        vpls "bd-2002"
            evpn-tunnel supplementary-broadcast-domain  <------
        exit
    exit
    igmp                        <------
        interface "bd-2003"     <------
            no shutdown
        exit
        interface "bd-2004"     <------
            no shutdown
        exit
        no shutdown
    exit
    pim                         <------
```

```
        rpf-table both           <------
        interface "bd-2002"      <------
            multicast-senders always     <------
        exit
        apply-to all             <------
        no shutdown
    exit
    no shutdown
exit
```

As shown in the previous configuration commands, the VPRN must be configured as follows:

- The SBD interface in the VPRN must be configured as using the following command so that the OISM forwarding mode is enabled.

  ```
  configure service vprn interface vpls evpn-tunnel supplementary-broadcast-domain
  ```

- IGMP must be enabled on the ordinary BD (R-VPLS) interfaces so that the PEs can process the received IGMP messages from the receivers.
- Even though the protocol itself is not used, PIM is enabled in the VPRN on all the IRB interfaces so that the multicast source addresses can be resolved. Also, the following command must be enabled on the SBD interface;

  ```
  configure service vprn pim interface multicast-senders always
  ```

  this is because the SBD interface is unnumbered (it does not have an IP address associated) and the multicast traffic source RPF-check would discard the multicast traffic arriving at the SBD interface unless the system is informed that legal multicast traffic may be expected on the SBD. The **multicast-senders always** command allows the system to process multicast on the unnumbered SBD interface. The following command is needed in case sources are added to the VPRN route-table as ARP-ND host routes (which is typical in Data Centers).

  – **MD-CLI**

  ```
  configure service vprn pim ipv4 rpf-table both
  configure service vprn pim ipv6 rpf-table both
  ```

  – **classic CLI**

  ```
  configure service vprn pim rpf-table both
  ```

Besides the VPRN, BD-2003, BD-2004 and BD-2002 (SBD) must be configured as follows.

```
vpls 2003 name "bd-2003" customer 1 create
    allow-ip-int-bind
        forward-ipv4-multicast-to-ip-int    <------
    exit
    bgp
    exit
    bgp-evpn
        evi 2003
        mpls bgp 1
            ingress-replication-bum-label
            auto-bind-tunnel
                resolution any
            exit
            no shutdown
        exit
```

```
    exit
    igmp-snooping                             <------
        no shutdown                           <------
    exit
    sap 1/1/1:2003 create
        igmp-snooping
            mrouter-port
        exit
        no shutdown
    exit
    no shutdown
exit
vpls 2004 name "bd-2004" customer 1 create
    allow-ip-int-bind
        forward-ipv4-multicast-to-ip-int      <------
    exit
    bgp
    exit
    bgp-evpn
        evi 2004
        mpls bgp 1
            ingress-replication-bum-label
            auto-bind-tunnel
                resolution any
            exit
            no shutdown
        exit
    exit
    igmp-snooping                             <------
        no shutdown                           <------
    exit
    sap 1/1/1:2004 create
        igmp-snooping
            fast-leave
        exit
        no shutdown
    exit
    no shutdown
exit
vpls 2002 name "bd-2002" customer 1 create
    allow-ip-int-bind
        forward-ipv4-multicast-to-ip-int      <------
    exit
    bgp
    exit
    bgp-evpn
        no mac-advertisement
        ip-route-advertisement
        sel-mcast-advertisement               <------
        evi 2002
        mpls bgp 1
            auto-bind-tunnel
                resolution any
            exit
            no shutdown
        exit
    exit
    igmp-snooping                             <------
        no shutdown                           <------
    exit
    no shutdown
exit
```

As shown in the previous configuration commands, the following command must be configured in ordinary and SBD R-VPLS services so that IGMP messages or SMET routes are processed by the IGMP module.

- **MD-CLI**

```
configure service vpls igmp-snooping admin-state enable
```

- **classic CLI**

```
configure service vpls no igmp-snooping
```

Also, the following command allows the system to forward multicast traffic from the R-VPLS to the VPRN interface.

- **MD-CLI**

```
configure service vpls routed-vpls multicast ipv4 forward-to-ip-interface
```

- **classic CLI**

```
configure service vpls allow-ip-int-bind forward-ipv4-multicast-to-ip-int
```

Finally, the following command must be enabled on the SBD R-VPLS, so that the SBD aggregates the multicast state for all the ordinary BDs and advertises the corresponding SMET routes with the SBD route-target.

- **MD-CLI**

```
configure service vpls bgp-evpn routes sel-mcast advertise true
```

- **classic CLI**

```
configure service vpls bgp-evpn sel-mcast-advertisement
```

In OISM mode, the SMET route is only advertised from the SBD R-VPLS and not from the other ordinary BD R-VPLSes.

PE2 and PE3 are configured with the VPRN (2000), ordinary BD (BD-2001) and SBD (BD-2002) as above. In addition, PE2 and PE3 are attached to ES-1 where a receiver is connected. Multicast state synchronization through BGP Multicast Synch routes is automatically enabled in R-VPLS services in OISM mode and no additional configuration is needed:

```
/* Example of ES-1 configuration and MCS on PE3. Similar configuration is needed in
PE2.
bgp-evpn
    ethernet-segment "ES-1" virtual create
        esi 01:00:00:00:00:00:01:00:00:00
        service-carving
            mode manual
            manual
                preference non-revertive create
                    value 30
                exit
            exit
        exit
        multi-homing single-active
        lag 1
        dot1q
            q-tag-range 2001
```

```
        exit
        no shutdown
    exit
```

When the previous configuration is executed in the three nodes, the EVPN routes are exchanged. BD2003 in PE4 receives IMET routes from the remote SBD PEs and creates "SBD" destinations to PE2 and PE3. Those SBD destinations are used to forward multicast traffic to PE2 and PE3, following the OISM forwarding procedures described in OISM forwarding plane. The following command shows an example of IMET route (flagged as SBD route working on OISM mode) and SMET route received on PE4 from PE2.

IMET route received from PE2 on PE4.

```
show router bgp routes evpn incl-mcast community target:64500:2002 hunt
```

Example

```
<snip>
-------------------------------------------------------------------------------
RIB In Entries
-------------------------------------------------------------------------------
Nexthop        : 192.0.2.2
From           : 192.0.2.2
Res. Nexthop   : 192.168.24.1
Local Pref.    : 100                    Interface Name : int-PE-4-PE-2
<snip>
Community      : target:64500:2002
                 mcast-flags:SBD/NO-MEG/NO-PEG/OISM/NO-MLD-Proxy/NO-IGMP-Proxy <---
                 bgp-tunnel-encap:MPLS
<snip>
EVPN type      : INCL-MCAST
Tag            : 0
Originator IP  : 192.0.2.2                  <------
Route Dist.    : 192.0.2.2:2002
<snip>
-------------------------------------------------------------------------------
PMSI Tunnel Attributes :
Tunnel-type    : Ingress Replication
Flags          : Type: RNVE(0) BM: 0 U: 0 Leaf: not required
MPLS Label     : LABEL 524241
Tunnel-Endpoint: 192.0.2.2
-------------------------------------------------------------------------------
```

SMET route from PE2 received on PE4.

```
show router bgp routes evpn smet community target:64500:2002 hunt
```

Output example

```
<snip>
-------------------------------------------------------------------------------
RIB In Entries
-------------------------------------------------------------------------------
Nexthop        : 192.0.2.2
From           : 192.0.2.2
Res. Nexthop   : 192.168.24.1
Local Pref.    : 100                    Interface Name : int-PE-4-PE-2
<snip>
Community      : target:64500:2002 bgp-tunnel-encap:MPLS
```

```
<snip>
EVPN type      : SMET
Tag            : 0
Src IP         : 0.0.0.0                <------
Grp IP         : 239.0.0.4              <------
Originator IP  : 192.0.2.2              <------
Route Dist.    : 192.0.2.2:2002
<snip>
```

When PE4 receives the IMET routes from PE2 and PE3 SBDs, it identifies the routes as SBD routes in OISM mode, and PE4 creates special EVPN destinations on the BD-2003 service that are used to forward the multicast traffic. The SBD destinations are shown as **Sup BCast Domain** in the show commands output.

```
show service id 2003 evpn-mpls
```

Output example

```
===============================================================================
BGP EVPN-MPLS Dest (Instance 1)
===============================================================================
TEP Address             Transport:Tnl   Egr Label     Oper     Mcast      Num
                                                      State               MACs
-------------------------------------------------------------------------------
192.0.2.2               ldp:65551       524266        Up       m          0
192.0.2.3               ldp:65537       524266        Up       m          0
-------------------------------------------------------------------------------
Number of entries : 2
===============================================================================

*A:PE-4#
```

```
show service id 2003 evpn-mpls detail
```

Output example

```
===============================================================================
BGP EVPN-MPLS Dest (Instance 1)
===============================================================================
TEP Address             Transport:Tnl   Egr Label     Oper     Mcast     Num
                                                      State              MACs
-------------------------------------------------------------------------------
192.0.2.2               ldp:65551       524266        Up       m         0
  Oper Flags       : None
  Sup BCast Domain : Yes
  Last Update      : 02/07/2023 14:59:03
192.0.2.3               ldp:65537       524266        Up       m         0
  Oper Flags       : None
  Sup BCast Domain : Yes
  Last Update      : 02/07/2023 13:21:09
-------------------------------------------------------------------------------
Number of entries : 2
===============================================================================
```

Based on the reception of the SMET routes from PE2 and PE3, PE4 adds the SBD EVPN destinations to its MFIB on BD-2003.

```
show service id 2003 igmp-snooping base
```

Output example

```
===============================================================================
IGMP Snooping Base info for service 2003
===============================================================================
Admin State : Up
Querier     : 10.41.0.1 on rvpls bd-2003
SBD service : 2002
-------------------------------------------------------------------------------
Port                      Oper MRtr Pim  Send Max   Max  Max   MVR       Num
Id                        Stat Port Port Qrys Grps  Srcs Grp   From-VPLS Grps
                                                      Srcs
-------------------------------------------------------------------------------
sap:1/1/1:2003            Up   Yes  No   No   None  None None  Local     0
rvpls                     Up   Yes  No   N/A  N/A   N/A  N/A   N/A       N/A
sbd-mpls:192.0.2.2:524241 Up   No   No   N/A  N/A   N/A  N/A   N/A       1 <------
sbd-mpls:192.0.2.3:524253 Up   No   No   N/A  N/A   N/A  N/A   N/A       1 <------
===============================================================================
*A:PE-4#
```

```
show service id 2003 igmp-snooping statistics
```

Output example

```
===============================================================================
IGMP Snooping Statistics for service 2003
===============================================================================
Message Type            Received      Transmitted   Forwarded
-------------------------------------------------------------------------------
<snip>
EVPN SMET Routes        2             0             N/A    <------
-------------------------------------------------------------------------------
*A:PE-4# show service id 2003 mfib
<snip>
-------------------------------------------------------------------------------
*               *                     sap:1/1/1:2003               Local    Fwd
*               239.0.0.4             sap:1/1/1:2003               Local    Fwd
                                      sbd-eMpls:192.0.2.2:524241   Local    Fwd
                                      sbd-eMpls:192.0.2.3:524253   Local    Fwd
```

PE2 and PE3 also creates regular destinations and SBD destinations based on the reception of IMET routes. As an example, the following command shows the destinations created by PE3 in the ordinary BD-2001.

```
show service id 2001 evpn-mpls
```

Output example

```
===============================================================================
BGP EVPN-MPLS Dest (Instance 1)
===============================================================================
TEP Address               Transport:Tnl     Egr Label   Oper   Mcast   Num
```

```
                                                        State          MACs
-------------------------------------------------------------------------------
192.0.2.2                ldp:65551         524266      Up     m       0
192.0.2.2                ldp:65551         524267      Up     bum     0
192.0.2.2                ldp:65551         524268      Up     none    1
192.0.2.4                ldp:65539         524269      Up     m       0
-------------------------------------------------------------------------------
Number of entries : 4
===============================================================================
```

```
show service id 2001 evpn-mpls detail
```

Output example

```
===============================================================================
BGP EVPN-MPLS Dest (Instance 1)
===============================================================================
TEP Address              Transport:Tnl     Egr Label   Oper   Mcast   Num
                                                       State          MACs
-------------------------------------------------------------------------------
192.0.2.2                ldp:65551         524266       Up     m       0
  Oper Flags       : None
  Sup BCast Domain : Yes
  Last Update      : 02/07/2023 14:59:04
192.0.2.2                ldp:65551         524267       Up     bum     0
  Oper Flags       : None
  Sup BCast Domain : No
  Last Update      : 02/07/2023 14:59:04
192.0.2.2                ldp:65551         524268       Up     none     1
  Oper Flags       : None
  Sup BCast Domain : No
  Last Update      : 02/07/2023 14:59:04
192.0.2.4                ldp:65539         524269       Up     m        0
  Oper Flags       : None
  Sup BCast Domain : Yes
  Last Update      : 02/07/2023 13:21:10
-------------------------------------------------------------------------------
Number of entries : 4
===============================================================================
```

In case of an SBD destination and a non-SBD destination to the same PE (PE2), IGMP only uses the non-SBD one in the MFIB. The non-SBD destination always has priority over the SBD destination. This can be seen in the following command in PE3, where the SBD destination to PE2 is down as long as the non-SBD destination is up.

```
show service id 2001 igmp-snooping base
```

Output example

```
===============================================================================
IGMP Snooping Base info for service 2001
===============================================================================
Admin State : Up
Querier     : 10.0.0.3 on rvpls bd-2001
SBD service : 2002
-------------------------------------------------------------------------------
Port                     Oper MRtr Pim  Send Max   Max  Max   MVR       Num
Id                       Stat Port Port Qrys Grps  Srcs Grp   From-VPLS Grps
                                                     Srcs
-------------------------------------------------------------------------------
```

```
sap:lag-1:2001            Down No   No   No   None  None None  Local     1
rvpls                     Up   Yes  No   N/A  N/A   N/A  N/A   N/A       N/A
sbd-mpls:192.0.2.2:524241 Down No   No   N/A  N/A   N/A  N/A   N/A       0 <------
mpls:192.0.2.2:524242     Up   No   No   N/A  N/A   N/A  N/A   N/A       1 <------
sbd-mpls:192.0.2.4:524245 Up   No   No   N/A  N/A   N/A  N/A   N/A       0
===============================================================================
```

```
show service id 2001 mfib
```

Output example

```
===============================================================================
Multicast FIB, Service 2001
===============================================================================
Source Address  Group Address         Port Id                     Svc Id   Fwd
                                                                        Blk
-------------------------------------------------------------------------------
*               239.0.0.4             sap:lag-1:2001              Local    Fwd
                                      eMpls:192.0.2.2:524242      Local    Fwd <---
```

Finally, to check the Layer 3 IIF and OIF entries on the VPRN services, enter the following command. As an example, the command is executed in PE2:

```
show router 2000 pim group detail
```

Output example

```
===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.0.0.4
Source Address     : *
<snip>
===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.0.0.4
Source Address     : 10.41.0.41
<snip>
Up Time            : 0d 00:13:20       Resolved By        : rtable-u
Up JP State        : Joined            Up JP Expiry       : 0d 00:00:00
Up JP Rpt          : Pruned            Up JP Rpt Override : 0d 00:00:00

Rpf Neighbor       : 10.41.0.41
Incoming Intf      : bd-2002
Outgoing Intf List : bd-2001

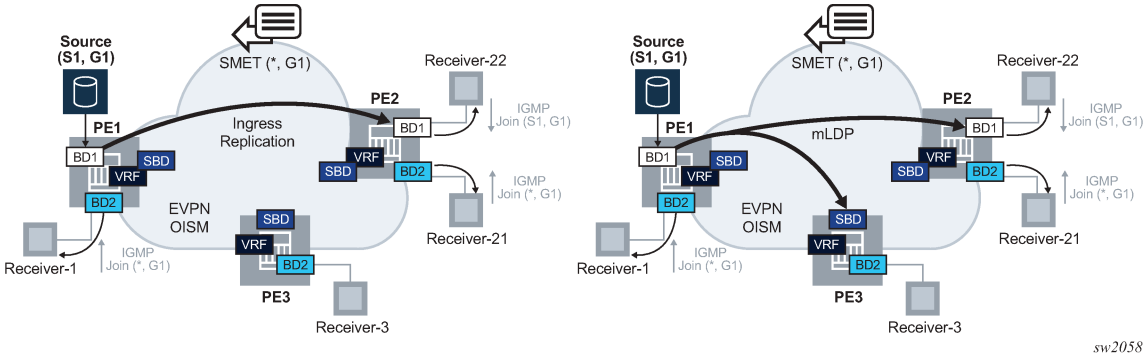
Curr Fwding Rate   : 0.000 kbps
Forwarded Packets  : 1000              Discarded Packets  : 0
Forwarded Octets   : 84000             RPF Mismatches     : 0
Spt threshold      : 0 kbps            ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
-------------------------------------------------------------------------------
Groups : 2
===============================================================================
```

6.5.18.6 Inclusive Provider mLDP Tunnels in OISM

Inclusive provider tunnels of type mLDP are supported in OISM PEs. These tunnels can be used to transport multicast flows from root PEs to leaf PEs while preventing multiple copies of the same multicast packet on the same link.

Figure 222: OISM with IR versus inclusive mLDP illustrates the difference between using Ingress Replication (IR) and inclusive mLDP provider tunnels in OISM. With a source S1 connected to BD1 and sending a flow to G1, if IR is used, the multicast traffic is only sent to PEs with receivers for (S1,G1). However, if an inclusive mLDP tunnel on PE1 is used (right side of Figure 222: OISM with IR versus inclusive mLDP) the multicast flow is sent to all the PEs in the tenant domain. For example, PE3 receives the flow only to drop it because there are no local receivers.

Figure 222: OISM with IR versus inclusive mLDP



mLDP tunnels are referred to as Inclusive BUM tunnels, because, although IP multicast traffic uses these tunnels, any BUM frame is also distributed to all PEs in the tenant. For example, in Figure 222: OISM with IR versus inclusive mLDP (right hand side), any BUM frame generated by any host connected to BD1 in PE1 uses the mLDP tunnel and is also sent to PE3.

The use of mLDP-inclusive provider tunnels in OISM requires the following configuration and procedures to be enabled on the PEs:

- All the PEs in the OISM tenant domain that need to transmit or receive multicast traffic on an mLDP tree in a BD, are configured with the following commands:

```
configure service vpls provider-tunnel inclusive owner bgp-evpn-mpls
configure service vpls provider-tunnel inclusive mldp
```

- The PEs attached to the sources (root PEs) should be configured with the following command on the ordinary BDs, and the PEs attached to the receivers should be configured as **root-and-leaf** or leaf-only.

```
configure service vpls provider-tunnel inclusive root-and-leaf
```

- The PEs attached to the receivers (leaf PEs) need to be configured using the following command on the BDs or SBDs.
  - **MD-CLI**

```
configure service vpls bgp-evpn routes incl-mcast advertise-ingress-replication
```

– **classic CLI**

```
configure service vpls bgp-evpn ingress-repl-inc-mcast-advertisement
```

This ensures the leaf PEs advertise a label in the IMET routes so that the root PEs can create EVPN-MPLS destinations to the leaf PEs and add them to their MFIB. Having EVPN-MPLS destinations in the MFIB is required on the root PE to use the mLDP tunnel for the multicast traffic.

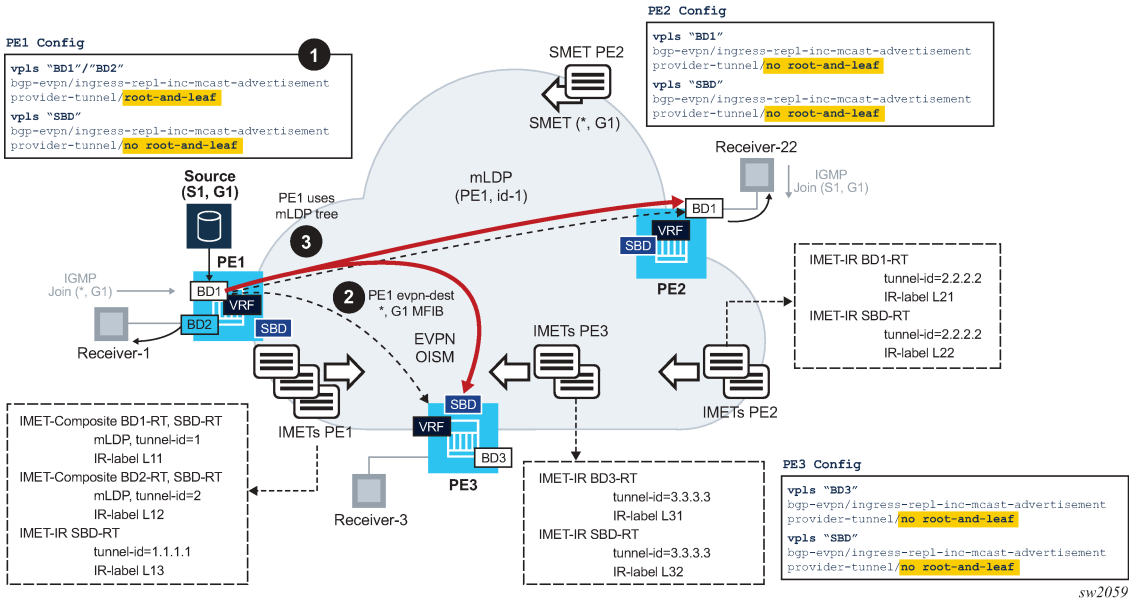
- The SBD must always be configured as leaf-only in all PEs, because the SBD mLDP tree is not used to transmit IP multicast.
- For the IMET and SMET routes to be exported and imported with the correct route targets, no **vsi-import** or **vsi-export** policies should be configured on the ordinary BDs and the SBDs.

Assuming the above guidelines are followed, and as illustrated in Figure 222: OISM with IR versus inclusive mLDP (right side), the root PE (PE1) that is attached to the source in BD1 sends the multicast traffic in an mLDP tree that is joined by leaf PEs either on BD1 (if BD1 is exists) or on the SBD (if BD1 does not exist on the leaf PE).

6.5.18.7 Example of Inclusive Provider Tunnels in OISM

Figure 223: OISM with inclusive mLDP example illustrates an example of the OISM procedures with mLDP trees.

Figure 223: OISM with inclusive mLDP example



Consider three PEs, PE1, PE2, and PE3, attached to BD1/BD2, BD1, and BD3 respectively, as in Figure 223: OISM with inclusive mLDP example. Assume that the source S1 is connected to BD1 in PE1. PE2 and PE3 are leaf PEs, because they have receivers but no sources. In this example:

- BD and SBD services must be configured for provider tunnel as follows:

– To have PE1 sending multicast traffic in P2MP mLDP tunnels on BD1 and BD2, both BDs are configured using the following command.

```
configure service vpls provider-tunnel inclusive root-and-leaf
```

They are also configured with the following command.

- **MD-CLI**

```
configure service vpls bgp-evpn routes incl-mcast advertise-ingress-replication
```

- **classic CLI**

```
configure service vpls bgp-evpn ingress-repl-inc-mcast-advertisement
```

The following is an example configuration of BD1 in PE1.

```
*A:PE-1>config>service>vpls# info
----------------------------------------------
            allow-ip-int-bind
            exit
            bgp
            exit
            bgp-evpn
                evi 1
                ingress-repl-inc-mcast-advertisement // default value
                mpls bgp 1
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            provider-tunnel
                inclusive
                    owner bgp-evpn-mpls
                    root-and-leaf
                    data-delay-interval 10
                    mldp
                    no shutdown
                exit
            exit
            igmp-snooping / mld-snooping
              no shutdown
            exit
<snip>
```

– PE2 and PE3 BDs are configured as leaf as they must be able to join mLDP trees but not set up an mLDP tree themselves.

- **MD-CLI**
  Do not configure **root-and-leaf**. An unconfigured **root-and-leaf** command functions as a leaf-only node. If configured, use the following command to delete the configuration.

```
configure groups group service vpls provider-tunnel inclusive delete root-and-leaf
```

- **classic CLI**

```
configure service vpls provider-tunnel inclusive no root-and-leaf
```

It is important that these BDs are configured with the following command that allows upstream PEs to create destinations to them.

- **MD-CLI**

```
configure service vpls bgp-evpn routes incl-mcast advertise-ingress-replication
```

- **classic CLI**

```
configure service vpls bgp-evpn ingress-repl-inc-mcast-advertisement
```

Multicast traffic cannot use the mLDP tree unless there is an EVPN-MPLS destination in the MFIB for the multicast stream.

– The SBDs in all PEs must be configured as follows:

```
configure service vpls provider-tunnel inclusive root-and-leaf
```

and with the following command.

- **MD-CLI**

```
configure service vpls bgp-evpn routes incl-mcast advertise-ingress-replication
```

- **classic CLI**

```
configure service vpls bgp-evpn ingress-repl-inc-mcast-advertisement
```

- When the configuration is added, the PEs create EVPN-MPLS destinations as follows, where a destination is represented as {pe, label} with "pe" being the IP address of the remote PE and "label" being the EVPN label advertised by the remote PE.
  – PE1 creates the following EVPN-MPLS destinations:
    - On BD1: {pe2,bd1-L21}, {pe2,sbd-L22}, {pe3,sbd-L32}
    - On BD2: {pe2,sbd-L22}, {pe3,sbd-L32}
    - On SBD: {pe2,sbd-L22}, {pe3,sbd-L32}
  – PE2 creates destinations as follows:
    - On BD1: {pe1,bd1-L11}, {pe1,sbd-L13}, {pe3,sbd-L32}
    - On SBD: {pe1,sbd-L13}, {pe3,sbd-L32}
  – PE3 creates destinations as follows:
    - On BD3: {pe1,sbd-L13}, {pe2,sbd-L22}
    - On SBD: {pe1,sbd-L13}, {pe2,sbd-L22}
  – PE2's BD1 and PE3's BD3 does not create an EVPN-MPLS destination to PE1's BD2. Also, PE3's BD3 does not create a destination to PE1's BD1. This is in spite of receiving IMET-Composite routes for those BDs with the SBD-RT, which is imported in PE2/PE3 ordinary BDs.

- As an example, on BD1, PE1's IGMP process adds the EVPN-MPLS destinations {pe2,bd1-L21}, {pe3,sbd-L32} to the MFIB. The third destination {pe2,sbd-L22} is kept down because the EVPN-MPLS destination in BD1 has higher priority.
  1. Upon receiving the SMET route from PE2, PE1 adds {pe2,bd1-L21} as OIF for the MFIB (*,G1).
  2. In the meantime, PE2 and PE3 have joined the mLDP tree with tunnel-id 1.
  3. When multicast to G1 is received from S1, because there is an MFIB EVPN OIF entry, the multicast traffic is forwarded. At the IOM level, PE1 replaces the MFIB EVPN destination with the P2MP tunnel with tunnel-id 1, as long as the P2MP tree is operationally up.
  4. The multicast traffic is sent along the mLDP tree and arrives at PE2/BD1 and PE3/SBD. Then local forwarding or routing is performed in PE2 and PE3, as normally in OISM.

6.5.18.8 OISM interworking with MVPN and PIM for MEG or PEG gateways

For EVPN OISM to successfully interwork with MVPN and PIM, it is important to ensure that the MVPN/PIM procedures in the IPVPN network are not modified. In this interworking scenario, two (or more) OISM PEs act as the gateway between the EVPN and the MVPN/PIM network to ensure the OISM procedures are transparent to MVPN/PIM, and vice versa.

SR OS supports the MVPN-to-EVPN Gateway (MEG) and PIM-to-EVPN Gateway (PEG)functions in accordance with *draft-ietf-bess-evpn-irb-mcast*. Both, Ingress Replication (IR) and mLDP trees are supported on the SBD so that multicast traffic can be received from or transmitted to OISM PEs.

When more than one MEG or PEG is present per EVPN tenant (that is, per SBD), one of the MEG or PEGs acts as the MEG or PEG designated router (DR). The following are the special functions of MEG/PEGs DRs.

- The DRs behave as a First Hop Router (FHR) from the MVPN/PIM perspective network and register sources in the OISM domain with the RP in the MVPN/PIM domain.
- The DRs behave as Last Hop Router (LHR) from the MVPN/PIM network perspective, and join the shared or source tree. The non-DR PEs remove the SBD R-VPLS interface from the VPRN's Layer 3 multicast OIF list, which prevents the PEs from sending multicast traffic to the OISM receivers.

The MEG or PEG DR election occurs in each PE attached to the SBD configured as MEG or PEG. Each PE builds a DR candidate list based on the reception of the Inclusive Multicast Ethernet Tag (IMET) routes for the SBD that include the MEG and/or PEG flag. After the **evpn-mcast-gateway**>**dr-activation-timer** expires, the PE runs the DR election based on the default algorithm used for EVPN DF election (modulo function of the EVI and number of PEs).

Note: A single DR election is run for MEGs and PEGs of the same SBD. The advertisement of an IMET route with MEG, PEG, or both flags set is controlled by the **evpn-mcast-gateway**>**advertise** {**mvpn-pim** | **mvpn-only** | **pim-only**} command.

Note: This section describes the MVPN and PIM procedures for MEG PEs, using MVPN examples. These procedures also apply to PEG PEs, using PIM messages instead of C-MCAST shared or source join routes.

Procedures for sources in MVPN and PIM and receivers in OISM, Procedures for ASM sources in OISM and receivers in MVPN, and Procedures for SSM sources in OISM and receivers in MVPN describe the MVPN and PIM procedures depending on whether the sources and receivers are attached to the OISM or MVPN network.

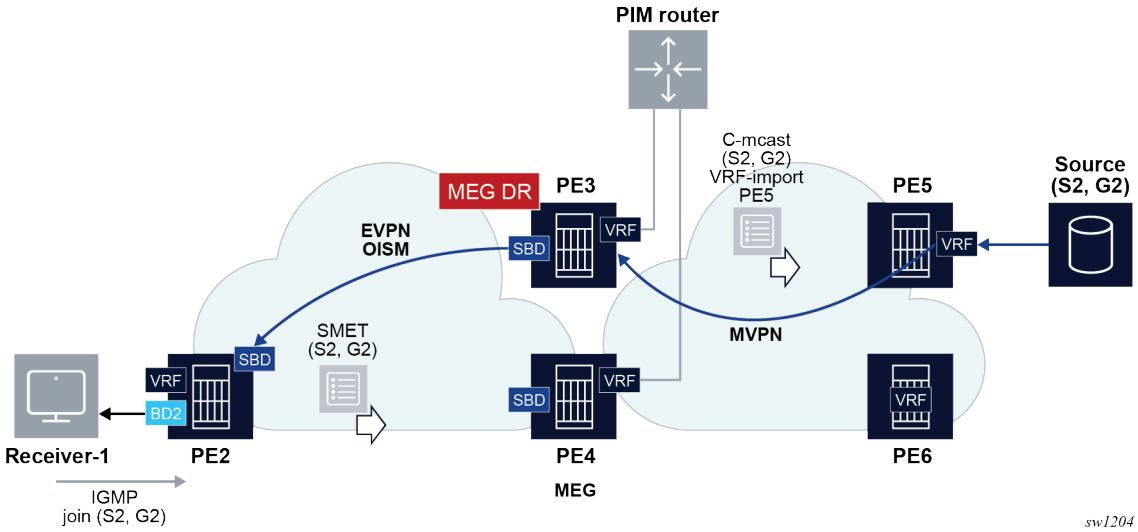
6.5.18.8.1 Procedures for sources in MVPN and PIM and receivers in OISM

The MEG DR for the SBD generates C-multicast source/shared tree join routes for receivers in the OISM domain. The following information applies to this procedure:

- This is similar to a PIM DR and its Last Hop Router (LHR) function.
- For directly connected receivers, to the MEG DR, the MEG DR creates a Layer 3 multicast state upon receiving an IGMP or MLD message and generates the corresponding C-multicast routes. This handling applies to MEG and PEG PEs.
- For receivers not directly connected, the MEG DR creates a Layer 3 multicast state upon receiving an SMET route from the PE connected to the receiver. Based on this newly created state, the MEG generates the corresponding C-multicast routes. This scenario is shown in the following figure.
- Use one of the following commands to trigger the non-DR MEG to create the Layer 3 multicast state too and advertises the C-multicast routes to attract the multicast traffic. The attracted multicast traffic is dropped at the non-DR MEG; however, by configuring this command the convergence is faster in case of a MEG DR failure.
  - **Classic CLI**
    evpn-mcast-gateway>non-dr-attract-traffic from-pim-mvpn
  - **MD-CLI**
    evpn-gateway non-dr-attract-traffic from-pim-mvpn

  .

Figure 224: Sources in the MVPN and PIM network



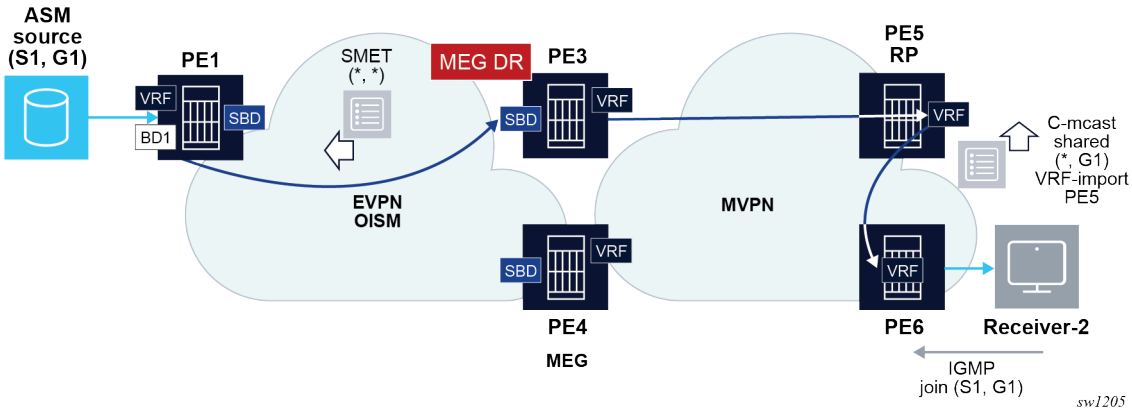
6.5.18.8.2 Procedures for ASM sources in OISM and receivers in MVPN

When Any-Source Multicast (ASM) group sources in the EVPN OISM domain, the MEG DR for the SBD needs to attract the ASM traffic from the EVPN sources and initiate the MVPN register and source discovery procedure. This is homologous to a PIM DR and its First Hop Router (FHR) function, and the scenario is shown in the following figure.

To attract ASM source traffic and act as the FHR, the MEG DR performs the following steps:

- The MEG DR generates a wildcard SMET route.
  - The wildcard SMET route is automatically generated as soon as the MEG is elected as DR. The wildcard SMET route is formatted in accordance with RFC 6625, with address and length equal to zero.
  - In addition, to attract multicast traffic from ASM sources on the non-DR routers, the user can configure the **config>service>vpls(sbd)>allow-ip-int-bind>evpn-mcast-gateway>non-dr-attract-traffic** command in the SBD. The command triggers the advertisement of the wildcard SMET route from the non-DR routers.
- When the MEG DR (for example, PE3 in Figure 225: ASM sources in the EVPN network) receives the ASM multicast traffic, it is handled as follows:
  - Assuming the MEG DR does not have the Layer 3 multicast state for it, the multicast traffic, (S1,G1) in the example shown in Figure 225: ASM sources in the EVPN network, is sent to the CPM.
  - The CPM encapsulates the multicast traffic into unicast register messages to the RP. For example, PE5 decapsulates the traffic and sends the multicast traffic down the shared tree.
  - In MVPN, PE5 triggers Source A-D routes and a C-multicast route for (S1,G1), and the SPT switchover occurs.
- If the MEG non-DR (for example, PE4) receives the ASM multicast traffic, it is handled as follows:
  - Assuming the MEG non-DR does not have the Layer 3 multicast state for it, the multicast traffic, (S1,G1) in the example shown in Figure 225: ASM sources in the EVPN network, is sent to the CPM and discarded.
  - The multicast traffic is heavily rate limited in the CPM.

Figure 225: ASM sources in the EVPN network

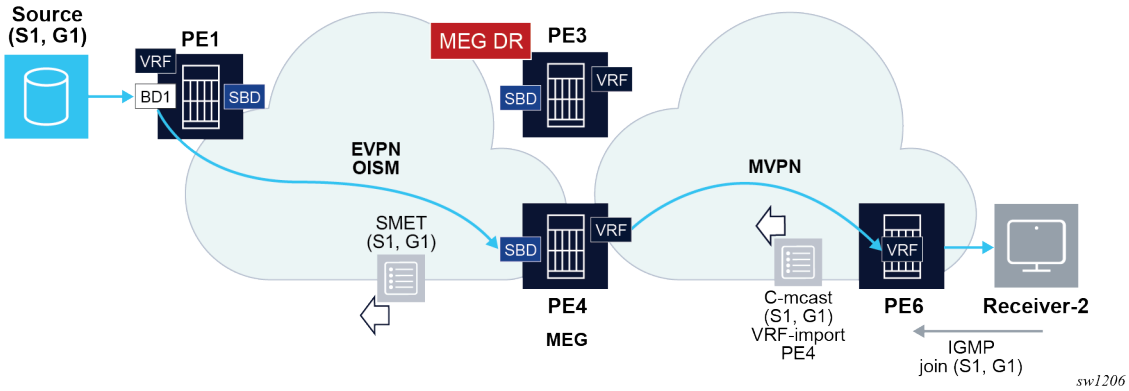


- On the remote OISM PEs attached to the ASM source (for example, PE1), the PE creates an MFIB for (*,*) with OIFs for the MEGs that sent the wildcard SMET route. For example:
  - (*,*) OIF: evpn-dest-PE3 (assuming **non-dr-attract-traffic false**)
  - (*,*) OIF: evpn-dest-PE3, evpn-dest-PE4 (assuming **non-dr-attract-traffic true**)
- Any multicast traffic is forwarded based on the MFIB for (*,*).
- The preceding handling also applies to ASM sources attached to the non-DR MEG or PEG. The non-DR creates an EVPN destination (on the BD attached to the source) to the DR as OIF for (*,*).

6.5.18.8.3 Procedures for SSM sources in OISM and receivers in MVPN

Irrespective of the DR election and source discovery process, when a MEG receives an MVPN C-multicast join route, it creates the Layer 3 multicast state and generates an SMET route for the S,G. This is shown in the following figure.

Figure 226: SSM sources in the EVPN network



- PE6 may pick PE4 as the Upstream Multicast Hop (UMH) PE for S1,G1 following regular MVPN procedures. In this case, PE4 adds the SBD interface as IIF, the MVPN tunnel as OIF member, and generates an SMET (S1,G1) route to draw the multicast traffic.
- After PE4 creates the state for (S1,G1), traffic to (S1,G1) is no longer sent to the CPM to be discarded, but it is forwarded in the datapath based on the Layer 3 MFIB state.
- PE1 creates an MFIB for (S1,G1) and starts sending traffic to PE4. The following are two potential scenarios in this case.
  - If PE4 is configured in the classic CLI as **no non-dr-attract-traffic** (or in the MD-CLI as **non-dr-attract-traffic none**), it does not send the wildcard SMET. PE1 creates the following entries in the MFIB and sends traffic to both MEGs:
    - (*,*) oif: evpn-dest-PE3
    - (S1,G1) oif: evpn-dest-PE3, evpn-dest-PE4
  - If PE4 is configured in the classic and MD-CLI as **non-dr-attract-traffic from-evpn** , PE4 and PE3 both send the wildcard SMET. PE1, ignores any SMET (S/*,G) routes from a PE when a SMET (*,*) is received from the same PE. If the (*,*) route is removed, PE1 reverts to handling (S/*,G) entries. For this reason, PE1 in this case creates only (*,*) OIFs and sends the traffic to both MEGs. The following OIF entry is created: (*,*) oif: evpn-dest-PE3, evpn-dest-PE4.

Note: To avoid duplication between MEG or PEG nodes, MEG or PEG PEs do not create EVPN destinations to each other in the SBD.

6.5.18.9 MEG or PEG gateways and local receivers or sources

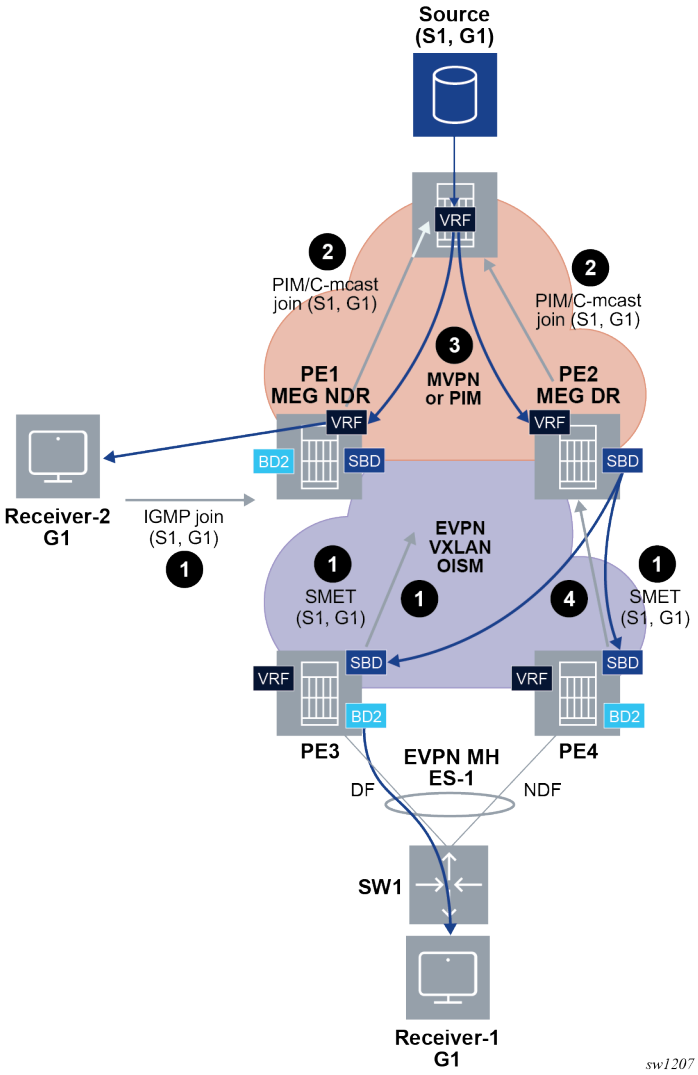
This section uses examples to describe the applicable considerations for local receivers and sources on MEG and PEG PEs.

6.5.18.9.1 Local singlehomed Receiver-2 on a MEG or PEG PE1, BD2

Figure 227: Local singlehomed receivers shows an initial situation where PE1 and PE2 are MEG/PEGs and PE2 is elected as the MEG or PEG DR. PE1 and PE2 do not have an EVPN destination between their SBDs.

The following shows a local singlehomed receiver.

Figure 227: Local singlehomed receivers



The following workflow applies to the example shown in the preceding graphic:

1. PE1 learns via IGMP/MLD that Receiver-2 is interested in (S1,G1).

   As shown in Figure 227: Local singlehomed receivers, Receiver-1, which is connected to a remote OISM non-MEG PE, issues an IGMP join for the same group. This triggers the corresponding SMET route from PE3 and PE4.

2. PE1 determines from its route table that there is a route to S1 via IP-VPN.

PE1 originates an MVPN C-multicast source tree join (S,G) route or a PIM (S,G) join, via normal MVPN or PIM procedures.

a. PE1 adds the MVPN tunnel or PIM interface as the Layer 3 IIF. The BD2 IRB is added to the Layer 3 OIF list.

b. PE1 also issues an SMET route as usual.

c. Since PE2 is the SBD's MEG DR, PE2 also sends a PIM/C-multicast join route upon receiving the SMET route from PE3 and PE4.

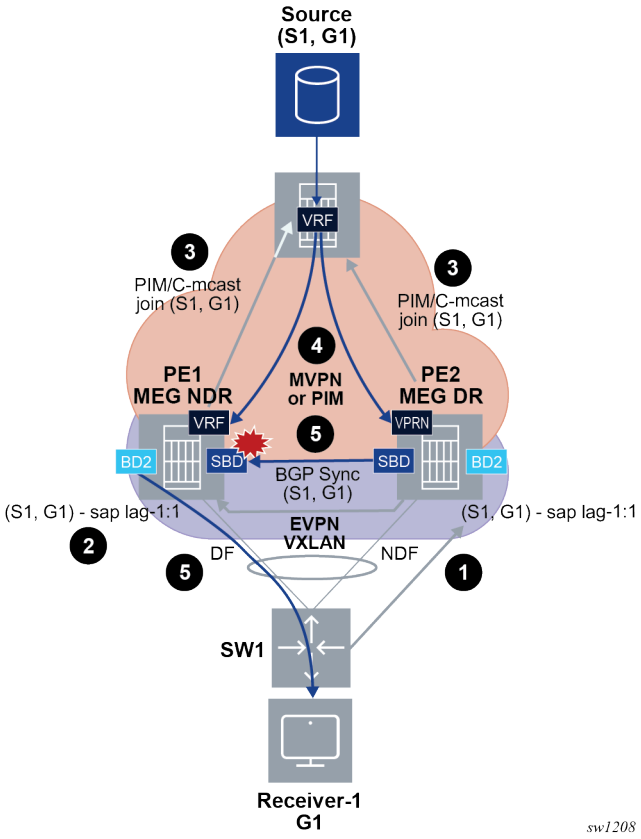
3. PE1 or PE2 receives the multicast traffic from the appropriate tunnel or interface, and passes RPF check. PE1 sends multicast down the BD2 IRB to the receiver. Since PE1 is non-DR for the SBD, the SBD IRB is not in the Layer 3 OIF list.
   PE2's SBD does not send the multicast flow to PE1, because there are no EVPN multicast destinations between MEG or PEG PEs of the same SBD.
4. Only PE2, the SBD's MEG or PEG DR, sends the multicast down the SBD's IRB to the remote OISM PEs and regular OISM forwarding follows on PE3 and PE4.

6.5.18.9.2 Local multihomed Receiver-1 on a pair of MEG or PEG PE1 and PE2, BD2

Figure 228: Local multihomed receivers, shows an initial situation where MEG or PEG routers PE1 and PE2 are multihomed to a local receiver in BD2. PE2 is the DR for the SBD. As both PEs are MEG or PEG for the same SBD, no EVPN multicast destination exists between the PEs for the SBD.

The following figure shows local multihomed receivers.

Figure 228: Local multihomed receivers



The following workflow applies to the example shown in the preceding graphic.

1. PE2 learns, via IGMP/MLD, that Receiver-1 is interested in (S1,G1) and adds the ES SAP to the OIF list.
2. PE2 synchronizes the (S1,G1) state with PE1 via BGP multicast synch routes and adds the ES SAP to the OIF list.
3. Both PE1 and PE2 originate an SMET (S1,G1) following normal OISM procedures.
   a. Also, both PEs generate the corresponding MVPN/PIM join route for (S1,G1) because the MEG or PEG DR election only occurs in the SBD. In this case, the state is created in BD2 and both PEs send the MVPN/PIM join route.
   b. Also, both PEs generate the corresponding MVPN/PIM join route for (S1,G1). This is because the MEG or PEG DR election occurs only in the SBD and the state is created in BD2. Consequently, both PEs send the MVPN/PIM join route in this case.
4. Step 3 causes traffic from the source to flow to both the DF and NDF, although only the DF forwards the traffic.
   a. The MEG or PEG DR and non-DR states only impact the addition of the SBD interface to the Layer 3 OIF.
   b. The datapath extensions prevent MVPN traffic from being sent to EVPN destinations other than an SBD EVPN destination.

5. PE2's SBD is added to the Layer 3 OIF list. However, since there is no EVPN multicast destination between the MEG/PEGs of the same SBD, multicast is not sent from PE2 to PE1.

Note: In OISM, all BDs are assumed to be EVPN enabled.

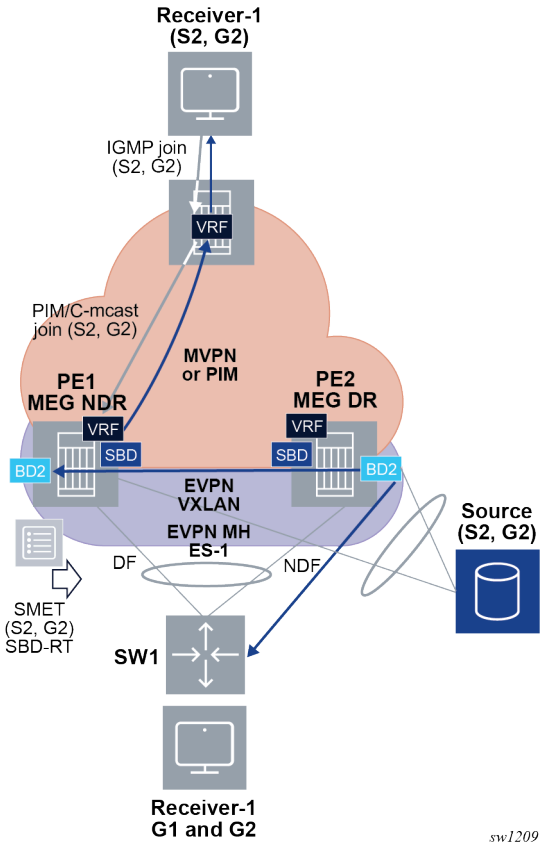
Local-bias behavior only applies to Layer 2 multicast (BUM in general) and not to Layer 3 multicast. That is, in Figure 228: Local multihomed receivers, the following applies to Layer 3 multicast traffic arriving at PE1 and PE2:

- can be forwarded to single-homed and DF SAPs in BD2
- cannot be forwarded to non-DF SAPs in BD2
- cannot be forwarded to EVPN destinations in BD2, in accordance with the OISM rules

6.5.18.9.3 Local multihomed source S2 on a pair of MEG or PEG PE1 and PE2, BD2

Figure 229: Local multihomed sources shows a scenario where PE1 and PE2 are multihomed to a local source. A local receiver is also using multihoming to the same MEG or PEG pair. PE2 is the SBD-DR.

Figure 229: Local multihomed sources



When the source sends multicast traffic for S2,G2, VXLAN local-bias or regular ESI-label filtering ensures the multihomed local receiver does not get duplicate traffic. The following applies in this scenario:

- The MEG SBD-DR (PE2) still performs the FHR functionality in this case (sends register/Source A-D routes), even if the source was singlehomed to the non-DR.
- If S2 was singlehomed to PE2 only, to avoid tromboning, the source S2 would be learned via ARP/ND as a host route and advertised in a VPN-IP route to attract the join route on PE2.
- If S2 is multihomed, as shown in Figure 229: Local multihomed sources, tromboning may occur, but traffic still flows correctly. For example:
  1. the remote PE performs UMH selection and picks up PE1
  2. PE1 generates a SMET route in the SBD as usual
  3. the SMET is imported and the state added to BD2 in PE2
  4. traffic is received by PE2, forwarded to PE1 via BD2, and then forwarded to the remote MVPN/PIM PE

6.5.18.10 MEG or PEG configuration example for Ingress Replication on the SBD

This section shows a configuration example for a pair of redundant MEGs. For a PEG example, replace the MVPN configuration for PIM interfaces in the VPRN service.

Each MEG in the pair is configured with a VPRN that contains the MVPN configuration and an SBD R-VPLS. It is assumed that there are no local sources or receivers in this example. The use of **domain-id** in the VPRN and the SBD R-VPLS prevents control plane loops for unicast routes reinjected from the IP-VPN domain into the EVPN domain, and the other way around. Preventing these loops guarantees the correct installation of unicast routes in the MEGs' route tables, and therefore ensures the C-multicast routes are correctly advertised and processed. See BGP D-PATH attribute for Layer 3 loop protection for more information about the configuration of domain ID. The following CLI shows the configuration in MEG1.

```
// MEG1’s VPRN service

*A:MEG1# configure service vprn 6000
*A:MEG1>config>service>vprn# info
----------------------------------------------
            interface "SBD-6002" create
                vpls "SBD-6002"
                    evpn-tunnel supplementary-broadcast-domain
                exit
            exit
            bgp-ipvpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    domain-id 64500:6000
                    route-distinguisher 192.0.2.2:6000
                    vrf-target target:64500:6000
                    no shutdown
                exit
            exit
            igmp
                interface "SBD-6002"
                    no shutdown
                exit
                no shutdown
            exit
            pim
```

```
                interface "SBD-6002"
                    multicast-senders always
                exit
                apply-to all
                rp
                    static
                        address 2.2.2.2
                            group-prefix 239.0.0.0/8
                        exit
                    exit
                    bsr-candidate
                        shutdown
                    exit
                    rp-candidate
                        shutdown
                    exit
                exit
                no shutdown
            exit
            mvpn
                auto-discovery default
                c-mcast-signaling bgp
                intersite-shared persistent-type5-adv
                provider-tunnel
                    inclusive
                        mldp
                            no shutdown
                        exit
                    exit
                exit
                vrf-target unicast
                exit
            exit
            no shutdown
----------------------------------------------

// MEG1’s SBD service

*A:MEG1>config>service>vprn# /configure service vpls 6002
*A:MEG1>config>service>vpls# info
----------------------------------------------
            allow-ip-int-bind
                forward-ipv4-multicast-to-ip-int
                forward-ipv6-multicast-to-ip-int
                evpn-mcast-gateway create
                    non-dr-attract-traffic from-evpn from-pim-mvpn
                    no shutdown
                exit
            exit
            bgp
            exit
            bgp-evpn
                no mac-advertisement
                ip-route-advertisement domain-id 64500:6002
                sel-mcast-advertisement
                evi 6002
                mpls bgp 1
                    ingress-replication-bum-label
                    ecmp 2
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
```

```
            exit
            igmp-snooping
                no shutdown
            exit
            mld-snooping
                no shutdown
            exit
            no shutdown
-----------------------------------------------
```

The configuration of the redundant MEG2 is as follows:

```
// MEG2’s VPRN configuration

*A:MEG2# configure service vprn 6000
*A:MEG2>config>service>vprn# info
-----------------------------------------------
            interface "SBD-6002" create
                vpls "SBD-6002"
                    evpn-tunnel supplementary-broadcast-domain
                exit
            exit
            bgp-ipvpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    domain-id 64500:6000
                    route-distinguisher 192.0.2.3:6000
                    vrf-target target:64500:6000
                    no shutdown
                exit
            exit
            igmp
                interface "SBD-6002"
                    no shutdown
                exit
                no shutdown
            exit
            pim
                interface "SBD-6002"
                    multicast-senders always
                exit
                apply-to all
                rp
                    static
                        address 3.3.3.3
                            group-prefix 239.0.0.0/8
                        exit
                    exit
                    bsr-candidate
                        shutdown
                    exit
                    rp-candidate
                        shutdown
                    exit
                exit
                no shutdown
            exit
            mvpn
                auto-discovery default
                c-mcast-signaling bgp
                intersite-shared persistent-type5-adv
```

```
                provider-tunnel
                    inclusive
                        mldp
                            no shutdown
                        exit
                    exit
                exit
                vrf-target unicast
                exit
            exit
            no shutdown
----------------------------------------------

// MEG2 SBD configuration

*A:MEG2>config>service>vprn# /configure service vpls 6002
*A:MEG2>config>service>vpls# info
----------------------------------------------
            allow-ip-int-bind
                forward-ipv4-multicast-to-ip-int
                forward-ipv6-multicast-to-ip-int
                evpn-mcast-gateway create
                    non-dr-attract-traffic from-evpn from-pim-mvpn
                    no shutdown
                exit
            exit
            bgp
            exit
            bgp-evpn
                no mac-advertisement
                ip-route-advertisement domain-id 64500:6002
                sel-mcast-advertisement
                evi 6002
                mpls bgp 1
                    ingress-replication-bum-label
                    ecmp 2
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            igmp-snooping
                no shutdown
            exit
            mld-snooping
                no shutdown
            exit
            no shutdown
----------------------------------------------
```

After the preceding configuration is added, MEG1 and MEG2 run the DR election. In the following example, which displays a sample DR election result, MEG1 is the DR:

```
*A:MEG1# show service id "SBD-6002" evpn-mcast-gateway all

===============================================================================
Service Evpn Multicast Gateway
===============================================================================
Type                        : mvpn-pim
Admin State                 : Enabled
DR Activation Timer         : 3 secs
Mvpn Evpn Gateway DR        : Yes
```

```
Pim  Evpn Gateway DR         : Yes
===============================================================================

===============================================================================
Mvpn Evpn Gateway
===============================================================================
DR Activation Timer Remaining: 3 secs
DR                           : Yes
DR Last Change               : 09/27/2021 08:50:32
===============================================================================

===============================================================================
Candidate list
===============================================================================
Orig-Ip                                 Time Added
-------------------------------------------------------------------------------
192.0.2.2                               09/27/2021 08:50:29
192.0.2.3                               09/27/2021 08:51:20
-------------------------------------------------------------------------------
Number of Entries: 2
===============================================================================

===============================================================================
Pim Evpn Gateway
===============================================================================
DR Activation Timer Remaining: 3 secs
DR                           : Yes
DR Last Change               : 09/27/2021 08:50:32
===============================================================================

===============================================================================
Candidate list
===============================================================================
Orig-Ip                                 Time Added
-------------------------------------------------------------------------------
192.0.2.2                               09/27/2021 08:50:29
192.0.2.3                               09/27/2021 08:51:20
-------------------------------------------------------------------------------
Number of Entries: 2
===============================================================================


*A:MEG2# show service id “SBD-6002” evpn-mcast-gateway all

===============================================================================
Service Evpn Multicast Gateway
===============================================================================
Type                         : mvpn-pim
Admin State                  : Enabled
DR Activation Timer          : 3 secs
Mvpn Evpn Gateway DR         : No
Pim  Evpn Gateway DR         : No
===============================================================================

===============================================================================
Mvpn Evpn Gateway
===============================================================================
DR Activation Timer Remaining: 3 secs
DR                           : No
DR Last Change               : 09/27/2021 08:51:24
===============================================================================

===============================================================================
Candidate list
```

```
===============================================================================
Orig-Ip                                 Time Added
-------------------------------------------------------------------------------
192.0.2.2                               09/27/2021 08:51:21
192.0.2.3                               09/27/2021 08:50:37
-------------------------------------------------------------------------------
Number of Entries: 2
===============================================================================

===============================================================================
Pim Evpn Gateway
===============================================================================
DR Activation Timer Remaining: 3 secs
DR                           : No
DR Last Change               : 09/27/2021 08:51:24
===============================================================================

===============================================================================
Candidate list
===============================================================================
Orig-Ip                                 Time Added
-------------------------------------------------------------------------------
192.0.2.2                               09/27/2021 08:51:21
192.0.2.3                               09/27/2021 08:50:37
-------------------------------------------------------------------------------
Number of Entries: 2
===============================================================================
```

If a source 40.0.0.1 is located in a remote PE of the MVPN network, and it is streaming group 239.0.0.44, the DR (for example, MEG1) attracts the traffic (by sending a C-multicast source join route) and forwards it to the SBD. The non-DR MEG2 does not add the SBD to the OIF list, and therefore it does not forward the multicast traffic to the OISM domain. The following is a sample output for this scenario.

```
// On the DR, MEG1, the SBD-6002 is added to the OIF list

*A:MEG1# show router 6000 pim group 239.0.0.44 detail

===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.0.0.44
Source Address     : 40.0.0.1
RP Address         : 2.2.2.2
Advt Router        : 192.0.2.4
Flags              : spt                Type               : (S,G)
Mode               : sparse
MRIB Next Hop      : 192.0.2.4
MRIB Src Flags     : remote
Keepalive Timer    : Not Running
Up Time            : 2d 04:42:53        Resolved By        : rtable-u

Up JP State        : Joined             Up JP Expiry       : 0d 00:00:06
Up JP Rpt          : Not Joined StarG   Up JP Rpt Override : 0d 00:00:00

Register State     : No Info
Reg From Anycast RP: No

Rpf Neighbor       : 192.0.2.4
Incoming Intf      : mpls-if-73731
Outgoing Intf List : SBD-6002

Curr Fwding Rate   : 0.000 kbps
Forwarded Packets  : 9999               Discarded Packets  : 0
```

```
Forwarded Octets   : 839916             RPF Mismatches     : 0
Spt threshold      : 0 kbps             ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
-------------------------------------------------------------------------------
Groups : 1
===============================================================================

// SBD-6002 is not added to the OIF list on the non-DR MEG2

*A:PE-3# show router 6000 pim group 239.0.0.44 detail

===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.0.0.44
Source Address     : 40.0.0.1
RP Address         : 3.3.3.3
Advt Router        : 192.0.2.4
Flags              : spt                Type               : (S,G)
Mode               : sparse
MRIB Next Hop      : 192.0.2.4
MRIB Src Flags     : remote
Keepalive Timer    : Not Running
Up Time            : 2d 04:43:02        Resolved By        : rtable-u

Up JP State        : Joined             Up JP Expiry       : 0d 00:00:58
Up JP Rpt          : Not Joined StarG   Up JP Rpt Override : 0d 00:00:00

Register State     : No Info
Reg From Anycast RP: No

Rpf Neighbor       : 192.0.2.4
Incoming Intf      : mpls-if-73733
Outgoing Intf List :

Curr Fwding Rate   : 0.000 kbps
Forwarded Packets  : 0                  Discarded Packets  : 0
Forwarded Octets   : 0                  RPF Mismatches     : 0
Spt threshold      : 0 kbps             ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
-------------------------------------------------------------------------------
Groups : 1
===============================================================================
```

6.5.18.11 MEG or PEG configuration example for mLDP on the SBD

This section shows a configuration example for a pair of redundant MEGs that use mLDP in the SBD to transmit and receive multicast traffic.

As in the previous example, each MEG in the pair is configured with a VPRN that contains the MVPN configuration and an SBD R-VPLS. Local sources/receivers are supported in this example and they are attached to local BDs or local interfaces in the VPRN. A receiver connected to a local BD (BD-6023) is multihomed to MEG1 and MEG2. Also, as in the previous example, the use of **domain-id** in the VPRN and the SBD R-VPLS prevents control plane loops for unicast routes. The following CLI shows the configuration in MEG1.

```
// MEG1’s VPRN service

*A:MEG1# configure service vprn 6000
*A:MEG1>config>service>vprn# info
```

```
-----------------------------------------------
            local-routes-domain-id 64500:2 // avoids loops for local routes
            interface "BD-6023" create  // local BD
                address 11.0.0.2/24
                vrrp 1 passive
                    backup 11.0.0.254
                exit
                vpls "BD-6023"
                    evpn
                        arp
                            no learn-dynamic
                            advertise dynamic
                        exit
                    exit
                exit
            exit
            interface "SBD-6002" create
                vpls "SBD-6002"
                    evpn-tunnel supplementary-broadcast-domain
                exit
            exit
            interface "local" create // local interface
                address 20.0.0.254/24
                sap pxc-6.a:600 create
                exit
            exit
            bgp-ipvpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    domain-id 64500:6000
                    route-distinguisher 192.0.2.2:6000
                    vrf-target target:64500:6000
                    no shutdown
                exit
            exit
            igmp
                interface "BD-6023"
                    no shutdown
                exit
                interface "SBD-6002"
                    no shutdown
                exit
                interface "local"
                    no shutdown
                exit
            exit
            pim
                interface "SBD-6002"
                    multicast-senders always
                exit
                apply-to all
                rp
                    static
                        address 4.4.4.4
                            group-prefix 224.0.0.0/4
                        exit
                    exit
                    bsr-candidate
                        shutdown
                    exit
                    rp-candidate
                        shutdown
```

```
                    exit
                exit
                no shutdown
            exit
            mvpn
                auto-discovery default
                c-mcast-signaling bgp
                intersite-shared persistent-type5-adv
                provider-tunnel
                    inclusive
                        mldp
                            no shutdown
                        exit
                    exit
                exit
                vrf-target unicast
                exit
            exit
            no shutdown
-----------------------------------------------

// MEG1’s SBD service

*A:MEG1>config>service>vprn# /configure service vpls 6002
*A:MEG1>config>service>vpls# info
-----------------------------------------------
            allow-ip-int-bind
                forward-ipv4-multicast-to-ip-int
                forward-ipv6-multicast-to-ip-int
                evpn-mcast-gateway create
                    non-dr-attract-traffic from-evpn from-pim-mvpn
                    no shutdown
                exit
            exit
            bgp
            exit
            bgp-evpn
                no mac-advertisement
                ip-route-advertisement domain-id 64500:6002
                sel-mcast-advertisement
                evi 6002
                mpls bgp 1
                    ingress-replication-bum-label
                    ecmp 2
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            provider-tunnel // mldp is enabled on the SBD
                inclusive
                    owner bgp-evpn-mpls
                    data-delay-interval 10
                    root-and-leaf
                    mldp
                    no shutdown
                exit
            exit
            igmp-snooping
                no shutdown
            exit
            mld-snooping
                no shutdown
```

```
            exit
            no shutdown
-----------------------------------------------

// MEG1’s local BD-6023 service

*A:MEG1>config>service>vprn# /configure service vpls 6023
*A:MEG1>config>service>vpls# info
-----------------------------------------------
            allow-ip-int-bind
                forward-ipv4-multicast-to-ip-int
                forward-ipv6-multicast-to-ip-int
                igmp-snooping
                    mrouter-port
                exit
                mld-snooping
                    mrouter-port
                exit
            exit
            bgp
            exit
            bgp-evpn
                evi 623
                mpls bgp 1
                    ingress-replication-bum-label
                    auto-bind-tunnel
                        resolution any
                        exit
                    no shutdown
                exit
             exit
             provider-tunnel
                 inclusive
                     owner bgp-evpn-mpls
                     data-delay-interval 10
                     root-and-leaf
                     mldp
                     no shutdown
                exit
              exit
              stp
                  shutdown
              exit
              igmp-snooping
                  no shutdown
              exit
              mld-snooping
                  no shutdown
              exit
              sap lag-1:623 create
                  igmp-snooping
                      send-queries
                  exit
                  no shutdown
              exit
              no shutdown
```

The configuration of the redundant MEG2 is as follows:

```
// MEG2’s VPRN configuration

*A:MEG2# configure service vprn 6000
*A:MEG2>config>service>vprn# info
```

```
------------------------------------------
            local-routes-domain-id 64500:3
            interface "BD-6023" create
                address 11.0.0.3/24
                vrrp 1 passive
                    backup 11.0.0.254
                exit
                vpls "BD-6023"
                    evpn
                        arp
                            no learn-dynamic
                            advertise dynamic
                        exit
                    exit
                 exit
            exit
            interface "SBD-6002" create
                vpls "SBD-6002"
                    evpn-tunnel supplementary-broadcast-domain
                exit
            exit
            interface "local" create
                address 30.0.0.254/24
                sap pxc-6.a:600 create
                exit
            exit
            bgp-ipvpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    domain-id 64500:6000
                    route-distinguisher 192.0.2.3:6000
                    vrf-target target:64500:6000
                    no shutdown
                exit
            exit
            igmp
                interface "BD-6023"
                    no shutdown
                exit
                interface "SBD-6002"
                    no shutdown
                exit
                interface "local"
                    no shutdown
                exit
                no shutdown
            exit
            pim
                interface "SBD-6002"
                    multicast-senders always
                exit
                apply-to all
                rp
                    static
                        address 4.4.4.4
                            group-prefix 224.0.0.0/4
                        exit
                    exit
                    bsr-candidate
                        shutdown
                    exit
                    rp-candidate
```

```
                        shutdown
                    exit
                exit
                no shutdown
            exit
            mvpn
                auto-discovery default
                c-mcast-signaling bgp
                intersite-shared persistent-type5-adv
                provider-tunnel
                    inclusive
                        mldp
                            no shutdown
                        exit
                    exit
                exit
                vrf-target unicast
                exit
            exit
            no shutdown
-----------------------------------------------

// MEG2 SBD configuration

*A:MEG2>config>service>vprn# /configure service vpls 6002
*A:MEG2>config>service>vpls# info
-----------------------------------------------
            allow-ip-int-bind
                forward-ipv4-multicast-to-ip-int
                forward-ipv6-multicast-to-ip-int
                evpn-mcast-gateway create
                    non-dr-attract-traffic from-evpn from-pim-mvpn
                    no shutdown
                exit
            exit
            bgp
            exit
            bgp-evpn
                no mac-advertisement
                ip-route-advertisement domain-id 64500:6002
                sel-mcast-advertisement
                evi 6002
                mpls bgp 1
                    ingress-replication-bum-label
                    ecmp 2
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            provider-tunnel
                inclusive
                    owner bgp-evpn-mpls
                    data-delay-interval 10
                    root-and-leaf
                    mldp
                    no shutdown
                exit
            exit
            igmp-snooping
                no shutdown
            exit
            mld-snooping
```

```
                no shutdown
            exit
            no shutdown
-----------------------------------------------

// MEG2's local BD-6023 service

A:MEG2>config>service>vpls# /configure service vpls 6023
A:MEG2>config>service>vpls# info
-----------------------------------------------
            allow-ip-int-bind
                forward-ipv4-multicast-to-ip-int
                forward-ipv6-multicast-to-ip-int
                igmp-snooping
                    mrouter-port
                exit
                mld-snooping
                    mrouter-port
                exit
            exit
            bgp
            exit
            bgp-evpn
                evi 623
                mpls bgp 1
                    ingress-replication-bum-label
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            provider-tunnel
                inclusive
                    owner bgp-evpn-mpls
                    data-delay-interval 10
                    root-and-leaf
                    mldp
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            igmp-snooping
                no shutdown
            exit
            mld-snooping
                no shutdown
            exit
            sap lag-1:623 create
                igmp-snooping
                    send-queries
                exit
                no shutdown
            exit
            no shutdown
-----------------------------------------------
```

After the preceding configuration is added, MEG1 and MEG2 run the DR election. As in the previous example, MEG1 is elected as DR and MEG2 as non-DR. In this example, the SBD is using mLDP instead

of ingress replication to transmit and receive multicast traffic. The following sample output shows the status of the provider-tunnel in MEG1 and MEG2.

```
A:MEG1# show service id "SBD-6002" provider-tunnel

===============================================================================
Service Provider Tunnel Information
===============================================================================
Type               : inclusive          Root and Leaf      : enabled
Admin State        : enabled            Data Delay Intvl   : 10 secs
PMSI Type          : ldp                LSP Template       :
Remain Delay Intvl : 0 secs             LSP Name used      : 8195
PMSI Owner         : bgpEvpnMpls
Oper State         : up                 Root Bind Id       : 32767
===============================================================================

A:MEG1# tools dump service id "SBD-6002" provider-tunnels

===============================================================================
VPLS 6002 Inclusive Provider Tunnels Originating
===============================================================================
ipmsi (LDP)                                     P2MP-ID  Root-Addr
-------------------------------------------------------------------------------
8195                                            8195    192.0.2.2

-------------------------------------------------------------------------------

===============================================================================
VPLS 6002 Inclusive Provider Tunnels Terminating
===============================================================================
ipmsi (LDP)                                     P2MP-ID  Root-Addr
-------------------------------------------------------------------------------
                                                8193    192.0.2.1

-------------------------------------------------------------------------------

A:MEG2# show service id "SBD-6002" provider-tunnel

===============================================================================
Service Provider Tunnel Information
===============================================================================
Type               : inclusive          Root and Leaf      : enabled
Admin State        : enabled            Data Delay Intvl   : 10 secs
PMSI Type          : ldp                LSP Template       :
Remain Delay Intvl : 0 secs             LSP Name used      : 8195
PMSI Owner         : bgpEvpnMpls
Oper State         : up                 Root Bind Id       : 32767
===============================================================================

A:MEG2# tools dump service id "SBD-6002" provider-tunnels

===============================================================================
VPLS 6002 Inclusive Provider Tunnels Originating
===============================================================================
ipmsi (LDP)                                     P2MP-ID  Root-Addr
-------------------------------------------------------------------------------
8195                                            8195    192.0.2.3

-------------------------------------------------------------------------------

===============================================================================
VPLS 6002 Inclusive Provider Tunnels Terminating
===============================================================================
```

```
ipmsi (LDP)                                          P2MP-ID  Root-Addr
-------------------------------------------------------------------------------
                                                     8193    192.0.2.1

-------------------------------------------------------------------------------
```

Also the example in MEG or PEG configuration example for Ingress Replication on the SBD showed the IIF and OIF lists on the MEGs for a source 40.0.0.1 that was connected to a remote MVPN PE and was streaming group 239.0.0.44. In this example, there is a source 10.0.0.1 connected to a remote OISM PE and it is streaming group 239.0.0.1. The group has local receivers on the local SBD-6023, which is multihomed to MEG1 and MEG2, and has receivers on a remote MVPN PE. The remote MVPN PE is configured with the default **mvpn umh-selection highest-ip** and therefore a local join triggers a C-multicast source-join route that is imported only by MEG2 (given that it has higher IP address than MEG1).

The following shows a sample output for this scenario.

```
// The source-join route for 239.0.0.1 is only imported by MEG2

A:MEG1# show router bgp routes mvpn-ipv4 type source-join group-ip 239.0.0.1 source-ip
 10.0.0.1
===============================================================================
 BGP Router ID:192.0.2.2        AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked, x - stale, > - best, b - backup, p - purge
 Origin codes  : i - IGP, e - EGP, ? - incomplete

===============================================================================
BGP MVPN-IPv4 Routes
===============================================================================
Flag  RouteType                  OriginatorIP           LocalPref   MED
      RD                         SourceAS               Path-Id     IGP Cost
      Nexthop                    SourceIP                           Label
      As-Path                    GroupIP
-------------------------------------------------------------------------------
No Matching Entries Found.
===============================================================================

A:PE-3# show router bgp routes mvpn-ipv4 type source-join group-ip 239.0.0.1 source-ip
 10.0.0.1
===============================================================================
 BGP Router ID:192.0.2.3        AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked, x - stale, > - best, b - backup, p - purge
 Origin codes  : i - IGP, e - EGP, ? - incomplete

===============================================================================
BGP MVPN-IPv4 Routes
===============================================================================
Flag  RouteType                  OriginatorIP           LocalPref   MED
      RD                         SourceAS               Path-Id     IGP Cost
      Nexthop                    SourceIP                           Label
      As-Path                    GroupIP
-------------------------------------------------------------------------------
u*>i  Source-Join                -                      100         0
      192.0.2.3:6000             64500                  None        -
      192.0.2.4                  10.0.0.1
      No As-Path                 239.0.0.1
-------------------------------------------------------------------------------
```

```
Routes : 1
===============================================================================

// Therefore, only MEG2 will add the MVPN tunnel to the OIF list for the group
// MEG1 only adds the local BD-6023 to the OIF list

A:MEG1# show router 6000 pim group 239.0.0.1 detail

===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.0.0.1
Source Address     : 10.0.0.1
RP Address         : 4.4.4.4
Advt Router        :
Flags              : spt               Type              : (S,G)
Mode               : sparse
MRIB Next Hop      : 10.0.0.1
MRIB Src Flags     : direct
Keepalive Timer Exp: 0d 00:02:43
Up Time            : 1d 17:27:02       Resolved By       : rtable-u

Up JP State        : Joined            Up JP Expiry      : 0d 00:00:00
Up JP Rpt          : Not Joined StarG  Up JP Rpt Override : 0d 00:00:00

Register State     : Pruned            Register Stop Exp  : 0d 00:00:09
Reg From Anycast RP: No

Rpf Neighbor       : 10.0.0.1
Incoming Intf      : SBD-6002
Outgoing Intf List : BD-6023, SBD-6002

Curr Fwding Rate   : 67.200 kbps
Forwarded Packets  : 15258             Discarded Packets  : 0
Forwarded Octets   : 1281672           RPF Mismatches     : 0
Spt threshold      : 0 kbps            ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
-------------------------------------------------------------------------------
Groups : 1
===============================================================================

// on MEG1's local BD-6023 there is a receiver on sap:lag-1:623

A:MEG1# show service id "BD-6023" mfib

===============================================================================
Multicast FIB, Service 6023
===============================================================================
Source Address  Group Address         Port Id                     Svc Id   Fwd
                                                                         Blk
-------------------------------------------------------------------------------
*               *                     mpls:192.0.2.3:524258       Local    Fwd
10.0.0.1        239.0.0.1             sap:lag-1:623               Local    Fwd
                                      mpls:192.0.2.3:524258       Local    Fwd
*               * (mac)               mpls:192.0.2.3:524258       Local    Fwd
-------------------------------------------------------------------------------
Number of entries: 3
===============================================================================

// MEG2 adds the local BD-6023 and the MVPN tunnel to the OIF list

A:PE-3# show router 6000 pim tunnel-interface

===============================================================================
```

```
PIM Interfaces ipv4
===============================================================================
Interface                       Originator Address   Adm  Opr  Transport Type
-------------------------------------------------------------------------------
mpls-if-73729                   192.0.2.3            Up   Up   Tx-IPMSI
mpls-if-73733                   192.0.2.4            Up   Up   Rx-IPMSI
mpls-if-73736                   192.0.2.2            Up   Up   Rx-IPMSI
-------------------------------------------------------------------------------
Interfaces : 3
===============================================================================

A:PE-3# show router 6000 pim group 239.0.0.1 detail

===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address      : 239.0.0.1
Source Address     : 10.0.0.1
RP Address         : 4.4.4.4
Advt Router        :
Flags              : spt               Type              : (S,G)
Mode               : sparse
MRIB Next Hop      : 10.0.0.1
MRIB Src Flags     : direct
Keepalive Timer    : Not Running
Up Time            : 1d 17:32:43       Resolved By       : rtable-u

Up JP State        : Joined            Up JP Expiry      : 0d 00:00:00
Up JP Rpt          : Not Joined StarG  Up JP Rpt Override : 0d 00:00:00

Register State     : No Info
Reg From Anycast RP: No

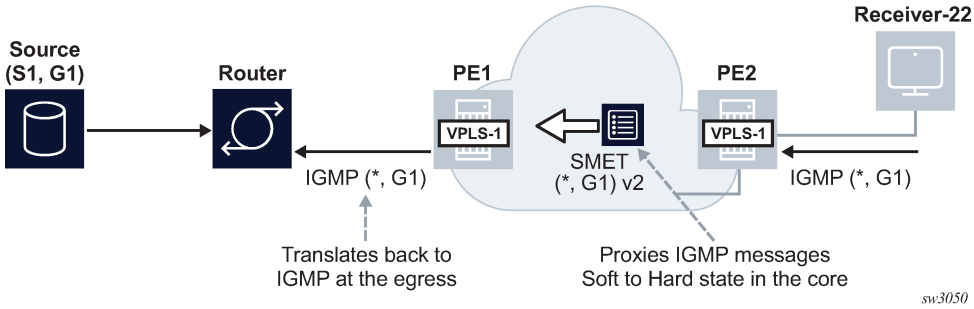
Rpf Neighbor       : 10.0.0.1
Incoming Intf      : SBD-6002
Outgoing Intf List : BD-6023, mpls-if-73729

Curr Fwding Rate   : 66.864 kbps
Forwarded Packets  : 27221             Discarded Packets  : 0
Forwarded Octets   : 2286564           RPF Mismatches     : 0
Spt threshold      : 0 kbps            ECMP opt threshold : 7
Admin bandwidth    : 1 kbps
-------------------------------------------------------------------------------
Groups : 1
===============================================================================
```

6.5.19 EVPN Layer-2 multicast (IGMP/MLD proxy)

SR OS supports EVPN Layer-2 multicast as described in the EVPN IGMP/MLD Proxy specification RFC9251. When this is enabled in a VPLS service with active IGMP or MLD snooping, IGMP or MLD messages are no longer sent to EVPN destinations. SMET routes (EVPN routes type 6) are advertised instead, so that the interest in a specific (S,G) can be signaled to the rest of the PEs attached to the same VPLS (also known as a Broadcast Domain (BD)). See Figure 230: SMET routes replace IGMP/MLD reports.

Figure 230: SMET routes replace IGMP/MLD reports



A VPLS service supporting EVPN-based proxy-IGMP/MLD functionality is configured as follows:

```
vpls 1 name "evi-1" customer 1 create
            bgp
            exit
            bgp-evpn
                evi 1
                sel-mcast-advertisement
                vxlan
                    shutdown
                exit
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            igmp/mld-snooping
                evpn-proxy
                    no shutdown
                exit
            sap lag-1:101 create
                igmp-snooping
                    send-queries
                exit
                no shutdown
            exit
```

Where:

- The **sel-mcast-advertisement** command allows the advertisement of SMET routes.

  The received SMET routes are processed regardless of the command.
- The **evpn-proxy** command in either the **igmp-snooping** or **mld-snooping** contexts:
  - triggers an IMET route update with the multicast flags EC and the proxy bits set. The multicast flags extended community carries a flag for IGMP proxy, that is set if **igmp-snooping**>**evpn-proxy no shutdown** is configured. Similarly, the MLD proxy flag is set if **mld-snooping**>**evpn-proxy no shutdown** is configured.
  - no longer turns EVPN MPLS into an Mrouter port, when used in EVPN MPLS service
  - enables EVPN proxy (IGMP or MLD snooping must be shutdown)

When the VPLS service is configured as an EVPN proxy service, IGMP or MLD queries or reports are no longer forwarded to EVPN destinations of PEs that support EVPN proxy. The reports are also no longer processed when received from PEs that support EVPN proxy.

The IGMP or MLD snooping function works in the following manner when the **evpn-proxy** command is enabled:

- IGMP or MLD works in proxy mode despite its configuration as IGMP or MLD snooping.
- Received IGMP or MLD join or leave messages on SAP or SDP bindings are processed by the proxy database to summarize the IGMP or MLD state in the service based on the group joined (each join for a group lists all sources to join). The proxy database can be displayed as follows.

```
# show service id 4000 igmp-snooping proxy-db

===============================================================================
IGMP Snooping Proxy-reporting DB for service 4000
===============================================================================
Group Address    Mode     Up Time        Num Sources
-------------------------------------------------------------------------------
239.0.0.1        exclude  0d 00:53:00    0
239.0.0.2        include  0d 00:53:01    1
-------------------------------------------------------------------------------
Number of groups: 2
===============================================================================
```

- When **evpn-proxy** is enabled, an additional EVPN proxy database is created to hand the version flags over to BGP and generate the SMET routes with the proper IGMP or MLD version flags. This EVPN proxy database is populated with local reports received on SAP or SDP binds but not with received SMET routes (the regular proxy database includes reports from SMETs too, without the version). The EVPN proxy database can be displayed as follows:

```
# show service id 4000 igmp-snooping evpn-proxy-db

===============================================================================
IGMP Snooping Evpn-Proxy-reporting DB for service 4000
===============================================================================
Group Address    Mode     Up Time        Num Sources    V1   V2   V3
-------------------------------------------------------------------------------
239.0.0.1        exclude  0d 00:53:55    0                       V3
239.0.0.2        include  0d 00:53:55    1                       V3
-------------------------------------------------------------------------------
Number of groups: 2
===============================================================================
```

- The EVPN proxy database or proxy database process IGMP or MLD reports as follows:
  - The EVPN proxy database result is communicated to the EVPN layer so that the corresponding SMET routes and flags are sent to the BGP peers. If multiple versions exist on the EVPN proxy database, multiple flags are set in the SMET routes.
  - The regular proxy database result is conveyed to the local Mrouter ports on SAP or SDP binds by IGMP or MLD reports and they are never sent to EVPN destinations of PEs with **evpn-proxy** configured.
- IGMP or MLD messages received on local SAP or SDP bind Mrouter ports (which have a default *.* entry) and queries are not processed by the proxy database. Instead, they are forwarded to local SAP or SDP binds but never to EVPN destinations of PEs with **evpn-proxy** configured (they are, however, still sent to non-EVPN proxy PEs).

- IGMP or MLD reports or queries are not received from EVPN PEs with **evpn-proxy** configured, but they are received and processed from EVPN PEs with **no evpn-proxy** configured. A PE determines if a specified remote PE, in the same BD, supports EVPN proxy based on the received igmp-proxy and mld-proxy flags along with the IMET routes.
- The Layer-2 MFIB OIF list for an (S,G) is built out of the local IGMP or MLD reports and remote SMET routes.
  - For backwards compatibility, PEs that advertise IMET routes without the multicast flags EC or with the EC but without the proxy bit set, are considered as Mrouters. For example, its EVPN binds are added to all OIF lists and reports are sent to them.
  - Even if MLD snooping is shut down and only IGMP snooping is enabled, the MFIB shows the EVPN binds added to *,* for MAC scope. If MLD snooping is enabled, the EVPN binds are not added as Mrouter ports for MAC scope.
- When SMET routes are received for a specific (S,G), the corresponding reports are sent to local SAP or SDP binds connected to queriers. The report version is set based on the local version of the querier.

The IGMP or MLD EVPN proxy functionality is supported in VPLS services with EVPN-VXLAN or EVPN-MPLS, and along with ingress replication or mLDP provider-tunnel trees.

In addition, EVPN proxy VPLS services support EVPN multihoming with multicast state synchronization using EVPN routes type 7 and 8. No additional command is needed to trigger the advertisement and processing of the multicast synch routes. In VPLS services, BGP sync routes are advertised or processed whenever the **evpn-proxy** command is enabled and there is a local Ethernet segment in the service. See EVPN OISM and multihoming for more information about the EVPN multicast synchronization routes and state synchronization in Ethernet segments.

6.5.20 Selective Provider Tunnels in OISM and EVPN-proxy services

Selective Provider Tunnels (S-PMSI)

Selective Provider Tunnels or Selective Provider Multicast Service Interface (S-PMSI) tunnels are supported in R-VPLS services configured in Optimized Inter-Subnet Multicast (OISM) mode or VPLS services configured in **evpn-proxy** mode.

Selective Provider Tunnels are signaled using the EVPN Selective Provider Multicast Service Interface Auto-Discovery (S-PMSI A-D) route, or EVPN route type 10. SR OS supports two types of Selective Provider Tunnels:

- mLDP wildcard S-PMSI trees, which are used to optimize the delivery of multicast and forward it to only PEs with IP Multicast sources or receivers. Wildcard S-PMSIs are enabled by the following command.

```
configure service vpls provider-tunnel selective wildcard-spmsi
```

- mLDP specific S-PMSI trees for (S,G) and/or (*,G) groups, which are used to optimize the delivery of some multicast groups that have receivers only in a limited number of PEs. (S,G) and (*,G) S-PMSIs are enabled by configuring the following command.

```
configure service vpls provider-tunnel selective data-threshold
```

The configuration of mLDP S-PMSIs for EVPN is similar to the mLDP S-PMSIs for MVPN. A **data-threshold** for a group-address and mask is configured. When the threshold (configured in kbps) for a group contained in the group-address and mask is exceeded, the router sets up a selective provider tunnel and the PEs with receivers for that group will join the mLDP selective tree. Options to setup the S-

PMSIs based on the number of interested PEs are also supported, as well as a **maximum-p2mp-spmsi** parameter that limits the number of S-PMSI trees created per service.

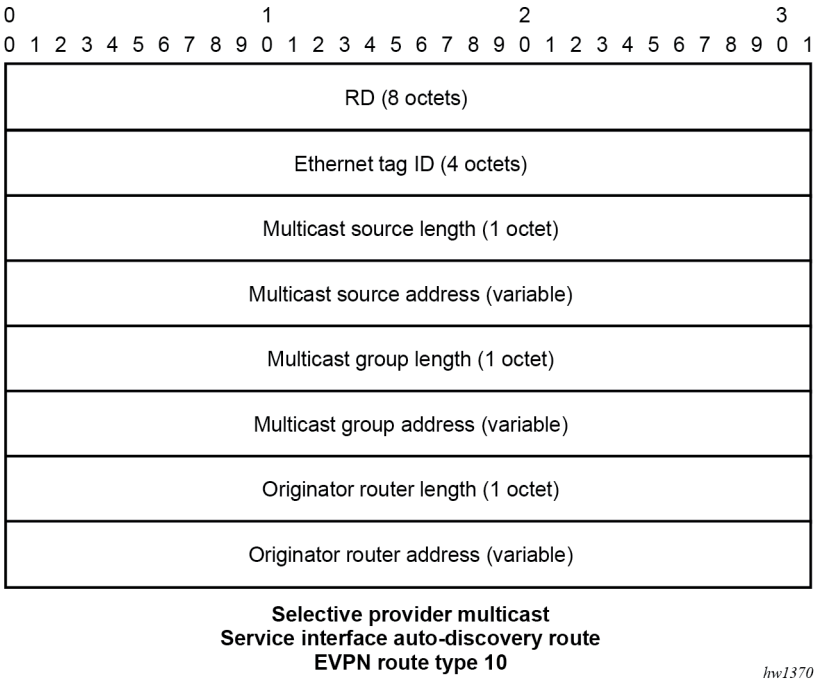
BGP-EVPN S-PMSI A-D route

The EVPN Selective Provider Multicast Service Interface Auto-Discovery route or simply S-PMSI A-D route or route type 10 is required to advertise:

- Wildcard PMSI routes to setup mLDP IP multicast trees
- Selective S-PMSI routes to setup mLDP selective IP multicast trees.

The S-PMSI A-D route is specified in *draft-ietf-bess-evpn-bum-procedure-updates* and the format is depicted in Figure 231: S-PMSI A-D route format:

Figure 231: S-PMSI A-D route format



Where:

- All the fields are considered part of the route key for BGP processing.
- When a service is configured to advertise wildcard spmsi routes, a route type 10 is advertised with Source and Group being all zeros. Otherwise the Source and Groups are populated as in the case of the other multicast routes.
- The S-PMSI A-D route above is only supported along with tunnel type mLDP (in the Provider Tunnel Attribute). No other tunnel types are supported.
- While in VPLS evpn-proxy services the S-PMSI AD routes is advertised using the route distinguisher and the route target of the service, in OISM mode, the S-PMSI A-D routes are advertised from the SBD or from an ordinary BD:
  - When advertised from an ordinary BD, the route includes the BD route-target (and route distinguisher) where the selective tree is configured plus the SBD route target

– When advertised from the SBD, the route includes the SBD route target only. This is only required in cases where the PE is a MEG/PEG.

Wildcard Selective Provider tunnels

As per *draft-ietf-bess-evpn-irb-mcast*, wildcard S-PMSI tunnels allow PEs decouple trees used for BUM traffic from trees used for IP multicast. If wildcard S-PMSIs are enabled, BUM I-PMSI tunnels are not used to send IP Multicast traffic.

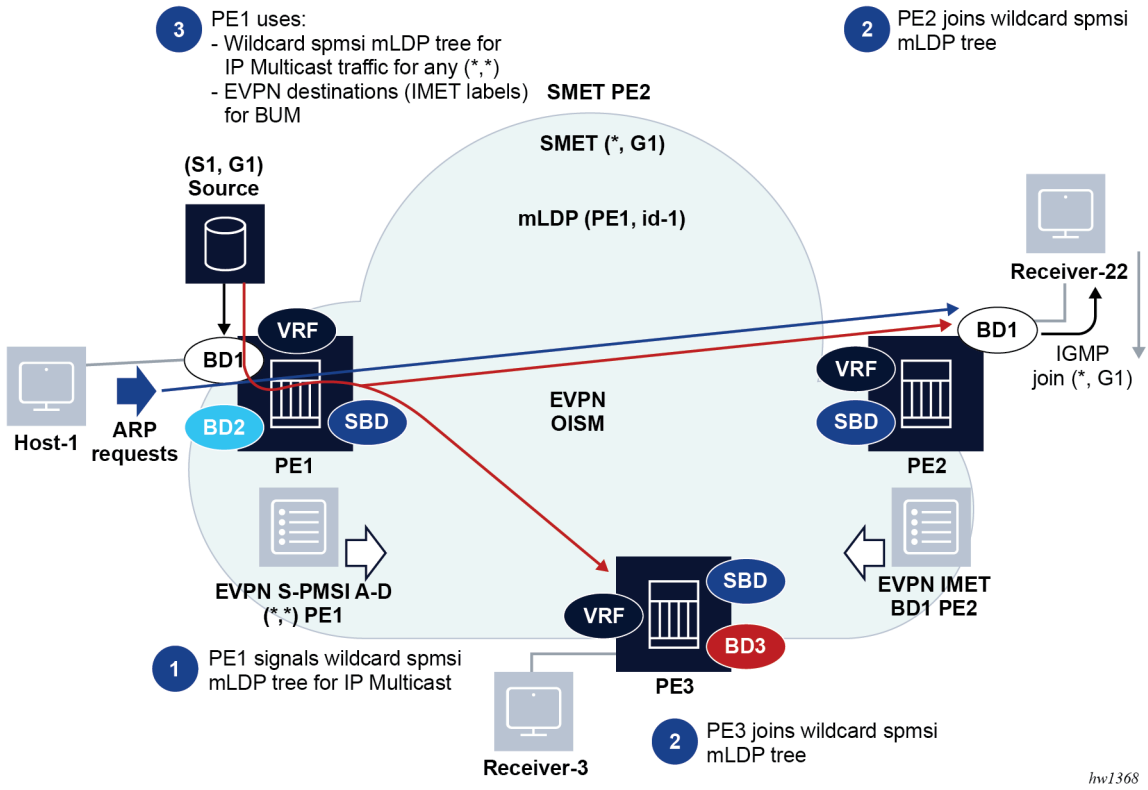
Note: Per *draft-ietf-bess-evpn-irb-mcast*, *Note that this will cause all the BUM traffic from a given BD in a Tenant Domain to be sent to all PEs that attach to that Tenant Domain, even the PEs that don't attach to the given BD. To avoid this, it is RECOMMENDED that the BUM tunnels not be used as IP Multicast inclusive tunnels, ...*

The wildcard S-PMSI A-D route is supported in OISM and VPLS **evpn-proxy** modes.

- An mLDP tree can be configured as selective wildcard-spmsi in all the R-VPLS services of the tenant, including the SBD.
- An mLDP tree can be configured as selective wildcard-spmsi in VPLS services as long as evpn-proxy is enabled.
- The selective provider-tunnel configuration is blocked on services where evpn-proxy or OISM are not enabled.

Figure 232: Wildcard S-PMSI A-D route in OISM illustrates an example for OISM:

Figure 232: Wildcard S-PMSI A-D route in OISM



- Based on the configuration of the following command PE1 signals a wildcard S-PMSI A-D route for BD1 (in addition to the IMET routes as in the regular OISM case or the EVPN proxy case). The route contains the SDB-RT (SBD's route target) in addition to the BD1-RT (BD1's route target).

  ```
  configure service vpls provider-tunnel selective wildcard-spmsi
  ```

- PE2 and PE3 import the route as they would do for BD1 IMET in OISM mode. PE2 and PE3 join the wildcard S-PMSI mLDP tree if they have been enabled using the following command and they have any local receivers that issued an IGMP/MLD join. A PE will not join the wildcard S-PMSI if no local receivers are joined.
  - **MD-CLI**

    ```
    configure service vpls provider-tunnel selective admin-state enable
    ```

  - **classic CLI**

    ```
    configure service vpls provider-tunnel selective no shutdown
    ```

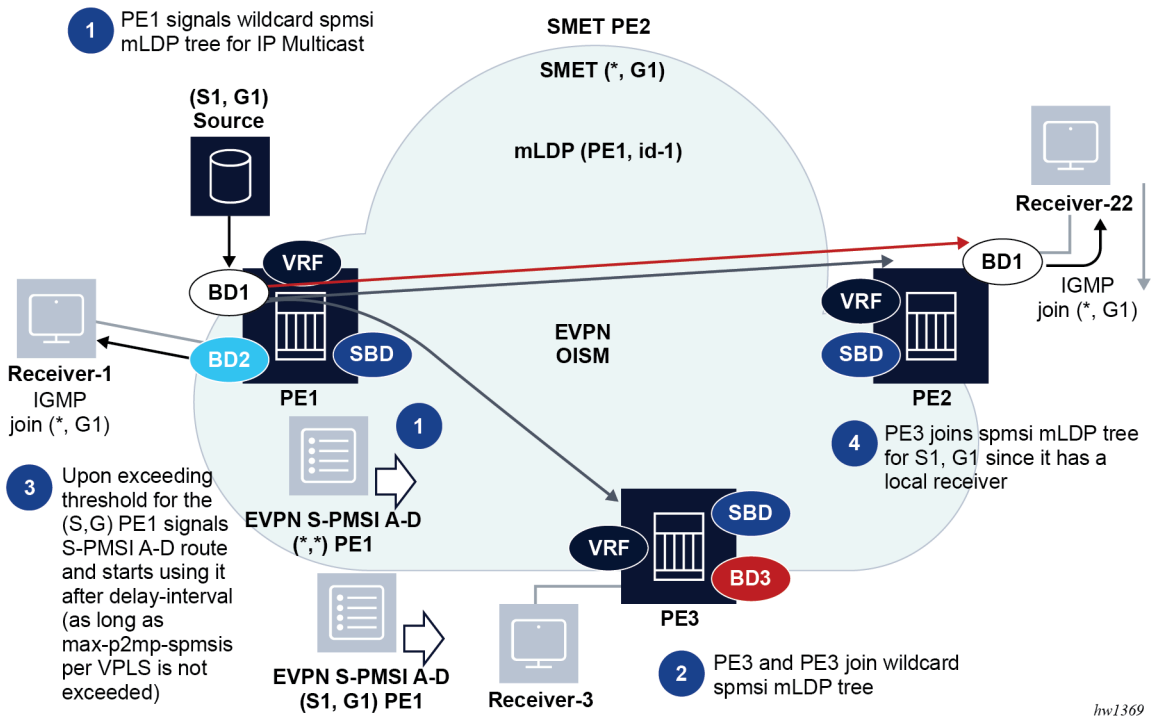
- The impact of this procedure is twofold:
  - PE1 now uses the wildcard S-PMSI mLDP tree for IP Multicast traffic. The IP multicast traffic is delivered to only those downstream PEs that joined the wildcard S-PMSI tree, and not the rest of the PEs of the tenant. Note that, in MVPN, the wildcard-spmsi does not carry traffic (the route does not even contain PTA). In EVPN, the wildcard-spmsi carries IP multicast and the route is advertised with an MLDP PTA.
  - PE1 now sends BUM traffic to only the PEs attached to the source BD, for example, PE2, and not to PE-3, while still using MLDP for multicast traffic. Without wildcard-spmsis, if we wanted to use mLDP for multicast, it had to be used for BUM traffic too, which would mean BUM was attracted by PE3 as well (in the example above).
- The wildcard-spmsi is used for multicast and the BUM EVPN destinations can be used for BUM. Note that the PE1’s EVPN SBD destination bind to PE3 is of type multicast (‘m’), so it is not used for BUM.

Inclusive and Selective mLDP Provider Tunnels are not simultaneously supported in the same service.

(S,G) and (*,G) Selective Provider Tunnels

SR OS supports mLDP Selective Provider Tunnels for specific (S,G) or (*,G) trees. Figure 233: Selective Provider Tunnels for OSIM illustrates an example for OISM services.

Figure 233: Selective Provider Tunnels for OSIM



- PE1 may use **wildcard-spmsi** or regular inclusive forwarding for IP multicast traffic. In the example, PE1 uses wildcard-spmsi.
- PE2 and PE3 are configured with the following command and therefore join the wildcard-spmsi tree.
  - **MD-CLI**

    ```
    configure service vpls provider-tunnel selective admin-state enable
    ```

  - **classic CLI**

    ```
    configure service vpls provider-tunnel selective no shutdown
    ```

- Since PE2 receives a local IGMP join (*,G1), PE2 triggers an SMET (*,G1) that creates an MFIB entry for (*,G1) on PE1.
- PE1 is configured with a threshold in ‘kbps’ units for G1 and starts polling stats for all MFIB entries that include G1. When the configured kbps threshold (and optionally, the number of PEs for a S,G) is exceeded, PE1 signals an S-PMSI A-D route for the (*,G), and after the **delay-interval** it starts using the new tree for S,G.
- If PE1 receives an SMET (S,G), then it generates a S-PMSI A-D route for (S,G) instead.
- If both SMETs are received, for example, (*,G1) and (S,G1), both S-PMSI types are generated, with the different mLDP tree information (in that way, a receiver only interested in (S,G) would not attract (*,G) traffic).
- Interested PEs with local receivers for the (S,G) join the new tree. In the example, only PE2 joins the spmsi tree, because it is the only PE with a local MFIB entry for (*,G1).

While the example uses a (*,G) SPMSI, (S,G) SPMSIs are possible too. The use of spmsis is configured as follows:

```
*A:PE-2>config>service>vpls>provider-tunnel# tree detail
selective
|
+---data-delay-interval <seconds>
| no data-delay-interval
|
+---mldp
| no mldp
|
+---wildcard-spmsi
| no wildcard-spmsi
|
+---data-threshold {<c-grp-ip-addr/mask>|<c-grp-ip-addr> <netmask>} <s-pmsi-threshold> [pe-
threshold-add <pe-threshold-add>] [pe-threshold-delete <pe-threshold-delete>]
data-threshold <c-grp-ipv6-addr/prefix-length> <s-pmsi-threshold> [pe-threshold-add <pe-
threshold-add>] [pe-threshold-delete <pe-threshold-delete>]
no data-threshold {<c-grp-ip-addr/mask>|<c-grp-ip-addr> <netmask>}
no data-threshold <c-grp-ipv6-addr/prefix-length>|
|
+---maximum-p2mp-spmsi <range>
| no maximum-p2mp-spmsi
|
+---no shutdown
| shutdown
```

Where:

- The selective container and the commands above are supported in VPLS services in **evpn-proxy** mode and R-VPLS services in OISM mode, in particular, in all ordinary BDs and the SBD of MEG/PEG nodes.
- **group-address/mask** — specifies an IPv4 or IPv6 multicast group address and netmask length. Multiple group-address/masks can be specified. In case of overlapping ranges, for a aowxudux group, only the longest prefix match is used. For instance, if the following two overlapping ranges are configured, and an SMET route for (*,232.0.1.1) is received, the S-PMSI tree for (*,232.0.1.1) is created only when the BW threshold exceeds 10kbps.

  ```
  *A:PE-4>config>service>vpls>pt>selective$ info
  ----------------------------------------------
  data-threshold 232.0.0.0/16 0
  data-threshold 232.0.1.0/24 10
  ----------------------------------------------
  ```

- **s-pmsi-threshold -** rate in kbps. If the rate for a given (S,G) or (*,G) within the specified group range exceeds the threshold, traffic for the (S,G) or (*,G) included in the group range is switched to the selective provider tunnel. Threshold 0 is also supported for mLDP. When threshold 0 is configured, the (S,G) or (*,G) switches to S-PMSI as soon as it is learned in the SBD/BD.
- **pe-threshold-add** — specifies the number of receiver PEs for creating S-PMSI. When the number of receiver PEs for a given multicast group configuration is non-zero and below this value, and the bandwidth threshold is satisfied, the S-PMSI is created. The number of receiver PEs is derived out of the SMET count (of routes included in the group range) for the SBD/BD. The originator-IP of the SMET route is checked so that the same PE is not counted multiple times. For example, for a (*,G1) SPMSI setup by PE1, if PE2 has a local receiver for (S1,G1) and another one for (S2,G1), PE2 issues two SMET routes. However, those are received by PE1 with the same originator-IP and therefore they count as one PE. The command **pe-threshold-add** dictates when to bring back the spmsi-tunnels after the number of receiver PEs counter has hit the **pe-threshold-delete**, in which case we have deleted the

spmsi-tunnel for this group. It has no implication on when to setup the spmsi-tunnel, since the router always waits for the **s-pmsi-threshold** to be exceeded.

- **pe-threshold-delete** — specifies the number of receiver PEs needed to delete the S-PMSI. When the number of receiver PEs for a given multicast group configuration is above the threshold, the S-PMSI is deleted and the multicast group is moved to ingress replication EVPN destinations or a wildcard-spmsi if configured, or potentially to a (*,G) P2MP if the MFIB was previously using a (S,G) PMSI. It is recommended that the delete threshold is significantly larger than the add threshold, to avoid re-signaling of S-PMSI as the receiver PE count fluctuates.
- Note that the threshold add/delete commands are based on SMET route counts, which not always match the number of receivers in the network for a specific (*,G) or (S,G). For instance:
  - SMETs may be received from non-spmsi enabled PEs. These routes are counted, however the receivers on these PEs do not get the traffic because they do not support spmsi trees.
  - SMETs from a PE can be aggregated, for example, for local (*,G), (S1..Sn,G) state, SMETs are aggregated into a single (*,G) SMETs. That does not provide a clear indication of the amount of receivers for a specific group on the root PE.
- Examples of how these thresholds work are shown in the following tables, assuming **pe-threshold-add 2 pe-threshold-delete 5:**

Table 24: Receiver PE count rising thresholds

Receiver PE count (rising) (based on SMET routes)	PMSI used by the root PE	Effect
0→1	Selective	PE count < pe-threshold-add S-PMSI used to carry traffic
1→2	Selective	PE count < pe-threshold-delete Traffic remains on S-PMSI
2→3	Selective	PE count < pe-threshold-delete Traffic remains on S-PMSI
3→4	Selective	PE count < pe-threshold-delete Traffic remains on S-PMSI
4→5	wildcard-spmsi if exists or EVPN destinations (Ingress Replication)	PE count = pe-threshold-delete Traffic switched back to wildcard-spmsi if exists or IR otherwise. Or potentially a (*,G) SPMSI if the MFIB was previously using it before moving to (S,G) SPMSI

Table 25: Receiver PE count falling thresholds

Receiver PE count (falling) (based on SMET routes)	PMSI used by root PE	Effect
5	wildcard-spmsi if exists or EVPN destinations (Ingress Replication)	Traffic flows on wildcard-spmsi if exists or IR. Or potentially a (*,G) SPMSI if the SMETs are for (S,G)
5→4	wildcard-spmsi if exists or EVPN destinations (Ingress Replication)	PE count > pe-threshold-add Traffic remains on wildcard-spmsi if exists or IR. Or potentially a (*,G) SPMSI if the SMETs are for (S,G)
4→3	wildcard-spmsi if exists or EVPN destinations (Ingress Replication)	PE count > pe-threshold-add Traffic remains on wildcard-spmsi if exists or IR. Or potentially a (*,G) SPMSI if the SMETs are for (S,G)
3→2	Selective	PE count = pe-threshold-add S-PMSI re-signaled. Traffic switched to S-PMSI.
2→1	Selective	Traffic flows on S-PMSI

- **maximum-p2mp-spmsi** - determines the maximum number of originating spmsi tunnels in the service (including the wildcard-spmsi). This limit is not validated against the total number of p2mp tunnels supported in the system.

Other parameters are configured as in the **provider-tunnel inclusive** context.

Selective Provider Tunnels are also supported in MEG/PEG gateways. In a MEG/PEG scenario, when the source is attached to an OISM PE, the PE may not use the S-PMSI tree for a given (x,G) if the only OIF is the MEG/PEG Designated Router (DR). This is because the way the implementation handles the wildcard SMET versus specific SMET routes in the MFIB (a specific SMET does not create an entry if there is a wildcard SMET from the same PE). Suppose MEG1 and MEG2 are the two MEGs between an OISM and an MVPN network, where the receivers are in the MVPN network and the source in the OISM domain. In that case:

- An OISM PE would create only a (*,*) entry if it received a wildcard SMET and a (S,G) SMET from the MEG DR. Therefore even if the threshold for (S,G) is exceeded, the OISM PE still uses the wildcard S-PMSI as opposed to the more specific S-PMSI.
- In addition, the same OISM PE would create an OIF to the non-DR MEG and a (S,G) entry (with OIFs to the two MEGs) if it received a (S,G) SMET from the non-DR MEG.

6.5.20.1 Configuration examples for selective provider tunnels

The following sections provide example configurations for selective provider tunnels in EVPN proxy and OISM services.

6.5.20.1.1 Use of Selective Provider Tunnels in EVPN proxy services

Suppose PE2, PE3 and PE4 are attached to the same VPLS service (named "evpn-proxy-bd-10k") that is configured in EVPN proxy mode. At the same time, PE2 and PE3 are multihomed to the same CE1. The PEs are configured as follows:

```
// PE2

[ex:/configure service vpls "evpn-proxy-bd-10k"]
A:admin@PE-2# info
admin-state enable
service-id 10000
customer "1"
bgp 1 {
}
igmp-snooping {
  admin-state enable
  evpn-proxy {
    admin-state enable
  }
}
bgp-evpn {
  evi 10000
  routes {
    sel-mcast {
      advertise true
    }
  }
  mpls 1 {
    admin-state enable
    ingress-replication-bum-label true
    ecmp 2
    auto-bind-tunnel {
      resolution any
    }
  }
}
sap lag-1:100 {
  igmp-snooping {
    send-queries true
  }
}
provider-tunnel {
  selective {
    admin-state enable
    owner bgp-evpn-mpls
    wildcard-spmsi true
    mldp true
    data-threshold {
      group-prefix 224.0.0.0/4 {
        threshold 0
      }
    }
  }
}

// PE3

[ex:/configure service vpls "evpn-proxy-bd-10k"]
A:admin@PE-3# info
admin-state enable
service-id 10000
customer "1"
```

```
bgp 1 {
}
igmp-snooping {
  admin-state enable
  evpn-proxy {
    admin-state enable
  }
}
bgp-evpn {
  evi 10000
  routes {
    sel-mcast {
      advertise true
    }
  }
  mpls 1 {
    admin-state enable
    ingress-replication-bum-label true
    ecmp 2
    auto-bind-tunnel {
      resolution any
    }
  }
}
sap lag-1:100 {
  igmp-snooping {
    send-queries true
  }
}
provider-tunnel {
  selective {
    admin-state enable
    owner bgp-evpn-mpls
    wildcard-spmsi true
    mldp true
    data-threshold {
      group-prefix 224.0.0.0/4 {
        threshold 0
      }
    }
  }
}

// PE4


[ex:/configure service vpls "evpn-proxy-bd-10k"]
A:admin@PE-4# info
admin-state enable
service-id 10000
customer "1"
bgp 1 {
}
igmp-snooping {
  admin-state enable
  evpn-proxy {
    admin-state enable
  }
}
bgp-evpn {
  evi 10000
  routes {
    sel-mcast {
      advertise true
```

```
    }
  }
  mpls 1 {
    admin-state enable
    ingress-replication-bum-label true
    ecmp 2
    auto-bind-tunnel {
      resolution any
    }
  }
}
sap pxc-6.a:100 {
}
provider-tunnel {
  selective {
    admin-state enable
    data-delay-interval 5
    owner bgp-evpn-mpls
    wildcard-spmsi true
    mldp true
    data-threshold {
      group-prefix 224.0.0.0/4 {
        threshold 0
      }
      group-prefix 239.0.0.0/8 {
        threshold 1
      }
    }
  }
}
```

Assuming a source with IP address 10.0.0.4 connected to PE4 starts sending multicast traffic to 239.0.0.4, PE4 detects the stream and as soon as it exceeds the configured threshold (1 kbps), PE4 advertises an S-PMSI A-D route. Since PE2 and PE3 receive an IGMP join for (10.0.0.4,239.0.0.4), they advertise the corresponding SMET routes and the S-PMSI trees are setup:

```
// PE4 advertises the S-PMSI A-D route since the received stream exceeds 1kbps:

A:PE-4#
439 2023/02/07 19:25:13.370 UTC MINOR: DEBUG #2001 Base Peer 1: 192.0.2.6
"Peer 1: 192.0.2.6: UPDATE
Peer 1: 192.0.2.6 - Send BGP UPDATE:
Withdrawn Length = 0
Total Path Attr Length = 100
Flag: 0x90 Type: 14 Len: 38 Multiprotocol Reachable NLRI:
Address Family EVPN
NextHop len 4 NextHop 192.0.2.4
Type: EVPN-SPMSI-AD Len: 27 RD: 192.0.2.4:10000, tag: 0, Mcast-Src-Len:
32, Mcast-Src-Addr: 10.0.0.4, Mcast-Grp-Len: 32, Mcast-Grp-Addr: 239.0.0.4, Orig Addr:
 192.0.2.4/32
Flag: 0x40 Type: 1 Len: 1 Origin: 0
Flag: 0x40 Type: 2 Len: 0 AS Path:
Flag: 0x40 Type: 5 Len: 4 Local Preference: 100
Flag: 0xc0 Type: 16 Len: 16 Extended Community:
target:64500:10000
bgp-tunnel-encap:MPLS
Flag: 0xc0 Type: 22 Len: 22 PMSI:
Tunnel-type LDP P2MP LSP (2)
Flags: (0x0)[Type: None BM: 0 U: 0 Leaf: not required]
MPLS Label 0
Root-Node 192.0.2.4, LSP-ID 0x2008
"
3 2023/02/07 19:25:13.368 UTC MAJOR: SVCMGR #2320 Base
```

```
"Service Id 10000, Dynamic vplsPmsi SDP Bind Id 32767:4294967285 was created."
```

Output of the MFIB and S-PMSIs in PE2 and PE4

```
show service id "10000" mfib statistics
```

```
===============================================================================
Multicast FIB Statistics, Service 10000
===============================================================================
Source Address   Group Address   Matched Pkts   Matched Octets
                                                Forwarding Rate
-------------------------------------------------------------------------------
10.0.0.4         239.0.0.4       11190           1096620
                                                 77.616 kbps
*                * (mac)         0               0
                                                 0.000 kbps
-------------------------------------------------------------------------------
Number of entries: 2
===============================================================================
```

```
show service id "10000" provider-tunnel spmsi-tunnels
```

```
===============================================================================
LDP Spmsi Tunnels
===============================================================================
LSP ID : 8199
Root Address : 192.0.2.2
S-PMSI If Index : 73750
Num. Leaf PEs : 1
Uptime : 0d 04:32:59
Group Address : 239.0.0.4
Source Address : 10.0.0.4
Origin IP Address : 192.0.2.2
State : TX Joined
Remain Delay Intvl : 0
-------------------------------------------------------------------------------
LSP ID : 8200
Root Address : 192.0.2.3
S-PMSI If Index : 73748
Num. Leaf PEs : 1
Uptime : 0d 04:33:02
Group Address : 239.0.0.4
Source Address : 10.0.0.4
Origin IP Address : 192.0.2.3
State : RX Joined
Remain Delay Intvl : 0
-------------------------------------------------------------------------------
LSP ID : 8200
Root Address : 192.0.2.4
S-PMSI If Index : 73754
Num. Leaf PEs : 1
Uptime : 0d 00:00:32
Group Address : 239.0.0.4
Source Address : 10.0.0.4
Origin IP Address : 192.0.2.4
State : RX Joined
Remain Delay Intvl : 0
-------------------------------------------------------------------------------
LSP ID : 8197
```

```
Root Address : 192.0.2.2
S-PMSI If Index : 73733
Uptime : 0d 04:32:59
Group Address : * (wildcard)
Source Address : *
Origin IP Address : 192.0.2.2
State : TX Joined
Remain Delay Intvl : 0
-------------------------------------------------------------------------------
LSP ID : 8198
Root Address : 192.0.2.3
S-PMSI If Index : 73747
Uptime : 0d 04:33:02
Group Address : * (wildcard)
Source Address : *
Origin IP Address : 192.0.2.3
State : RX Joined
Remain Delay Intvl : 0
-------------------------------------------------------------------------------
LSP ID : 8197
Root Address : 192.0.2.4
S-PMSI If Index : 73746
Uptime : 0d 04:33:02
Group Address : * (wildcard)
Source Address : *
Origin IP Address : 192.0.2.4
State : RX Joined
Remain Delay Intvl : 0
-------------------------------------------------------------------------------
===============================================================================
```

```
tools dump service id "10000" provider-tunnels type terminating
```

```
===============================================================================
VPLS 10000 Inclusive Provider Tunnels Terminating
===============================================================================
ipmsi (LDP) P2MP-ID Root-Addr
-------------------------------------------------------------------------------
               8197 192.0.2.4
               8198 192.0.2.3
               8200 192.0.2.3
               8200 192.0.2.4
-------------------------------------------------------------------------------
===============================================================================
VPLS 10000 Selective Provider Tunnels Terminating
===============================================================================
spmsi (LDP)   Source-Addr   Group-Addr   Root-Addr   LSP-ID   Lsp-Name
-------------------------------------------------------------------------------
              10.0.0.4       239.0.0.4   192.0.2.3   8200
              10.0.0.4       239.0.0.4   192.0.2.4   8200
              *              *           192.0.2.3   8198
              *              *           192.0.2.4   8197
-------------------------------------------------------------------------------
```

Outputs in PE4

```
===============================================================================
Multicast FIB Statistics, Service 10000
===============================================================================
Source Address     Group Address     Matched Pkts     Matched Octets
                                                      Forwarding Rate
-------------------------------------------------------------------------------
```

```
*                   *                 0               0
                                                      0.000 kbps
10.0.0.4            239.0.0.4         31484           3211368
                                                      57.691 kbps
*                   * (mac)           0               0
                                                      0.000 kbps
-------------------------------------------------------------------------------
Number of entries: 3
===============================================================================
```

```
tools dump service id 10000 provider-tunnels type originating
```

```
===============================================================================
VPLS 10000 Inclusive Provider Tunnels Originating
===============================================================================
No Tunnels Found
-------------------------------------------------------------------------------
===============================================================================
VPLS 10000 Selective Provider Tunnels Originating
===============================================================================
spmsi (LDP) Source-Addr     Group-Addr   Root-Addr   LSP-ID   Lsp-Name
-------------------------------------------------------------------------------
            10.0.0.4        239.0.0.4    192.0.2.4   8200     8200
            *               *            192.0.2.4   8197     8197
-------------------------------------------------------------------------------
```

6.5.20.1.2 Use of Selective Provider Tunnel in OISM service

Suppose an OISM network exists in PE2, PE3 and PE4. The three PEs are configured with VPRN "oism-vprn-20000" and use the SBD "SB20001", however, PE2 and PE3 are attached to the same ordinary BD "BD20023" whereas PE4 is attached to the ordinary BD "BD20004". In this example, a source with address 40.0.0.4 is connected to PE4's ordinary BD (and its stream triggers the setup of an S-PMSI) and wildcard-spmsis and S-PMSI thresholds are configured appropriately. Configurations are shown as follows. Since the configuration in PE2 and PE3 are equivalent, only the configuration of PE2 and PE4 are shown:

```
// PE2's relevant configuration for OISM

[ex:/configure service]
A:admin@PE-2# info
vpls "BD20023" {
  admin-state enable
  service-id 20023
  customer "1"
  routed-vpls {
    multicast {
      ipv4 {
        forward-to-ip-interface true
      }
    }
  }
  bgp 1 {
  }
  igmp-snooping {
    admin-state enable
  }
  bgp-evpn {
    evi 20023
```

```
    mpls 1 {
      admin-state enable
      ingress-replication-bum-label true
      auto-bind-tunnel {
        resolution any
      }
    }
  }
  sap lag-1:200 {
  }
  provider-tunnel {
    selective {
      admin-state enable
      owner bgp-evpn-mpls
      wildcard-spmsi true
      mldp true
    }
  }
}
vpls "SBD20001" {
  admin-state enable
  service-id 20001
  customer "1"
  routed-vpls {
    multicast {
      ipv4 {
        forward-to-ip-interface true
      }
    }
  }
  bgp 1 {
  }
  igmp-snooping {
    admin-state enable
  }
  bgp-evpn {
    evi 20001
    routes {
      ip-prefix {
        advertise true
      }
      sel-mcast {
        advertise true
      }
    }
    mpls 1 {
      admin-state enable
      auto-bind-tunnel {
        resolution any
      }
    }
  }
  provider-tunnel {
    selective {
      admin-state enable
      owner bgp-evpn-mpls
      mldp true
    }
  }
}
vprn "oism-vprn-20000" {
  admin-state enable
  service-id 20000
  customer "1"
```

```
  ecmp 2
  igmp {
    interface "BD20023" {
    }
  }
  pim {
    apply-to all
    ipv4 {
      rpf-table both
    }
    interface "SBD20001" {
      multicast-senders always
    }
  }
  interface "BD20023" {
    ipv4 {
      primary {
        address 10.0.0.2
        prefix-length 24
      }
      neighbor-discovery {
        learn-unsolicited true
        proactive-refresh true
        host-route {
          populate dynamic {
          }
        }
      }
      vrrp 1 {
        backup [10.0.0.254]
        passive true
      }
    }
  }
}

// PE4 relevant configuration for OISM


[ex:/configure service]
A:admin@PE-4# info
vpls "BD20004" {
  admin-state enable
  service-id 20004
  customer "1"
  routed-vpls {
    multicast {
      ipv4 {
        forward-to-ip-interface true
      }
    }
  }
  bgp 1 {
  }
  igmp-snooping {
    admin-state enable
  }
  bgp-evpn {
    evi 20004
    mpls 1 {
      admin-state enable
      ingress-replication-bum-label true
      auto-bind-tunnel {
        resolution any
```

```
      }
    }
  }
  sap pxc-6.a:200 {
  }
  provider-tunnel {
    selective {
      admin-state enable
      owner bgp-evpn-mpls
      wildcard-spmsi true
      mldp true
      data-threshold {
        group-prefix 239.0.0.0/8 {
          threshold 0
        }
      }
    }
  }
}
vpls "SBD20001" {
  admin-state enable
  service-id 20001
  customer "1"
  routed-vpls {
    multicast {
      ipv4 {
        forward-to-ip-interface true
      }
    }
  }
  bgp 1 {
  }
  igmp-snooping {
    admin-state enable
  }
  bgp-evpn {
    evi 20001
    routes {
      ip-prefix {
        advertise true
      }
      sel-mcast {
        advertise true
      }
    }
    mpls 1 {
      admin-state enable
      auto-bind-tunnel {
        resolution any
      }
    }
  }
  provider-tunnel {
    selective {
      admin-state enable
      owner bgp-evpn-mpls
      mldp true
    }
  }
}
vprn "oism-vprn-20000" {
  admin-state enable
  service-id 20000
  customer "1"
```

```
  ecmp 2
  igmp {
    interface "BD20004" {
    }
  }
  pim {
    apply-to all
    ipv4 {
      rpf-table both
    }
    interface "SBD20001" {
      multicast-senders always
    }
  }
  interface "BD20004" {
    ipv4 {
      primary {
        address 40.0.0.1
        prefix-length 24
      }
      neighbor-discovery {
        learn-unsolicited true
        proactive-refresh true
        host-route {
          populate dynamic {
          }
        }
      }
    }
    vpls "BD20004" {
      evpn {
        arp {
          learn-dynamic false
          advertise dynamic {
          }
        }
      }
    }
  }
  interface "SBD20001" {
    mac 00:00:00:00:00:04
    vpls "SBD20001" {
      evpn-tunnel {
        supplementary-broadcast-domain true
      }
    }
  }
}
```

With the above configuration on PE2, PE3 and PE4, PE4 advertises an S-PMSI A-D route for group (40.0.0.4,239.0.0.4), in addition to the wildcard-spmsi route:

```
show router bgp routes evpn spmsi-ad rd 192.0.2.4:20004
```

Output example

```
===============================================================================
BGP Router ID:192.0.2.2 AS:64500 Local AS:64500
===============================================================================
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
l - leaked, x - stale, > - best, b - backup, p - purge
```

```
Origin codes : i - IGP, e - EGP, ? - incomplete


=======================================================
BGP EVPN SPMSI AD Routes
=======================================================
Flag   Route Dist.       Src Address
       Tag               Grp Address
                         Orig Address
-------------------------------------------------------
u*>i   192.0.2.4:20004   0.0.0.0
       0                 0.0.0.0
                         192.0.2.4
u*>i   192.0.2.4:20004   40.0.0.4
       0                 239.0.0.4
                         192.0.2.4
-------------------------------------------------------
Routes : 2
=======================================================
```

The S-PMSI A-D route for 239.0.0.4 makes PE2 and PE3 join the Selective mLDP tree setup by PE4. The multicast group is delivered over the S-PMSI tree. The following commands help show the established S-PMSI trees (which are modeled as SDP-binds at the service level and therefore consume SDP-bind resources). PE2 and PE3 join the S-PMSI tree for 239.0.0.4 on the SBD because they are not attached to the source ordinary BD. The traffic is received at Layer 3, therefore the statistics are seeing at the VPRN level and not at the MFIB level (as in the case of EVPN proxy):

```
show service id "20001" provider-tunnel spmsi-tunnels detail
```

Output example

```
===============================================================================
LDP Spmsi Tunnels
===============================================================================
LSP ID : 8199
Root Address : 192.0.2.4
S-PMSI If Index : 73752
Num. Leaf PEs : 1
Uptime : 0d 14:45:11
Group Address : 239.0.0.4
Source Address : 40.0.0.4
Origin IP Address : 192.0.2.4
State : RX Joined
Remain Delay Intvl : 0
-------------------------------------------------------------------------------
LSP ID : 8198
Root Address : 192.0.2.4
S-PMSI If Index : 73751
Uptime : 0d 14:45:11
Group Address : * (wildcard)
Source Address : *
Origin IP Address : 192.0.2.4
State : RX Joined
Remain Delay Intvl : 0
-------------------------------------------------------------------------------
===============================================================================
```

```
tools dump service id "20001" provider-tunnels type terminating
```

Output example

```
===============================================================================
VPLS 20001 Inclusive Provider Tunnels Terminating
===============================================================================
No Tunnels Found
-------------------------------------------------------------------------------
===============================================================================
VPLS 20001 Selective Provider Tunnels Terminating
===============================================================================
spmsi (LDP)   Source-Addr    Group-Addr     Root-Addr   LSP-ID    Lsp-Name
-------------------------------------------------------------------------------
              40.0.0.4       239.0.0.4      192.0.2.4   8199
              *              *              192.0.2.4   8198
-------------------------------------------------------------------------------
```

```
show router "20000" pim group detail
```

Output example

```
===============================================================================
PIM Source Group ipv4
===============================================================================
Group Address : 239.0.0.4
Source Address : 40.0.0.4
RP Address : 0
Advt Router :
Flags : Type : (S,G)
Mode : sparse
MRIB Next Hop : 40.0.0.4
MRIB Src Flags : direct
Keepalive Timer : Not Running
Up Time : 0d 15:42:23 Resolved By : rtable-u

Up JP State : Joined                Up JP Expiry : 0d 00:00:14
Up JP Rpt : Not Joined StarG        Up JP Rpt Override : 0d 00:00:00

Register State : No Info
Reg From Anycast RP: No

Rpf Neighbor : 40.0.0.4
Incoming Intf : SBD20001
Outgoing Intf List : BD20023

Curr Fwding Rate : 67.200 kbps
Forwarded Packets : 29945            Discarded Packets : 0
Forwarded Octets : 2515380           RPF Mismatches : 0
Spt threshold : 0 kbps               ECMP opt threshold : 7
Admin bandwidth : 1 kbps
-------------------------------------------------------------------------------
Groups : 1
===============================================================================
```

6.5.21 EVPN-VPWS PW headend functionality

EVPN-VPWS is often used as an aggregation technology to connect access devices to the residential or business PE in the service provider network. The PE receives tagged traffic inside EVPN-VPWS circuits

and maps each tag to a different service in the core, such as ESM services, Epipe services, or VPRN services.

SR OS implements this PW headend functionality by using PW ports that use multihomed Ethernet Segments (ESs) for redundancy. ESs can be associated with PW ports in two different modes of operation.

- PW port-based ESes with multihoming procedures on PW SAPs
- PW port-based ESes with multihoming procedures on stitching Epipe

PW port-based ESes with multihoming procedures on PW SAPs

PW ports in ESs and virtual ESs (vESs) are supported for EVPN-VPWS MPLS services. In addition to LAG, port, and SDP objects, PW port ID can be configured in an Ethernet Segment. In this mode of operation, PW port-based ESs only support **all-active** configuration mode, and not **single-active** configuration mode.

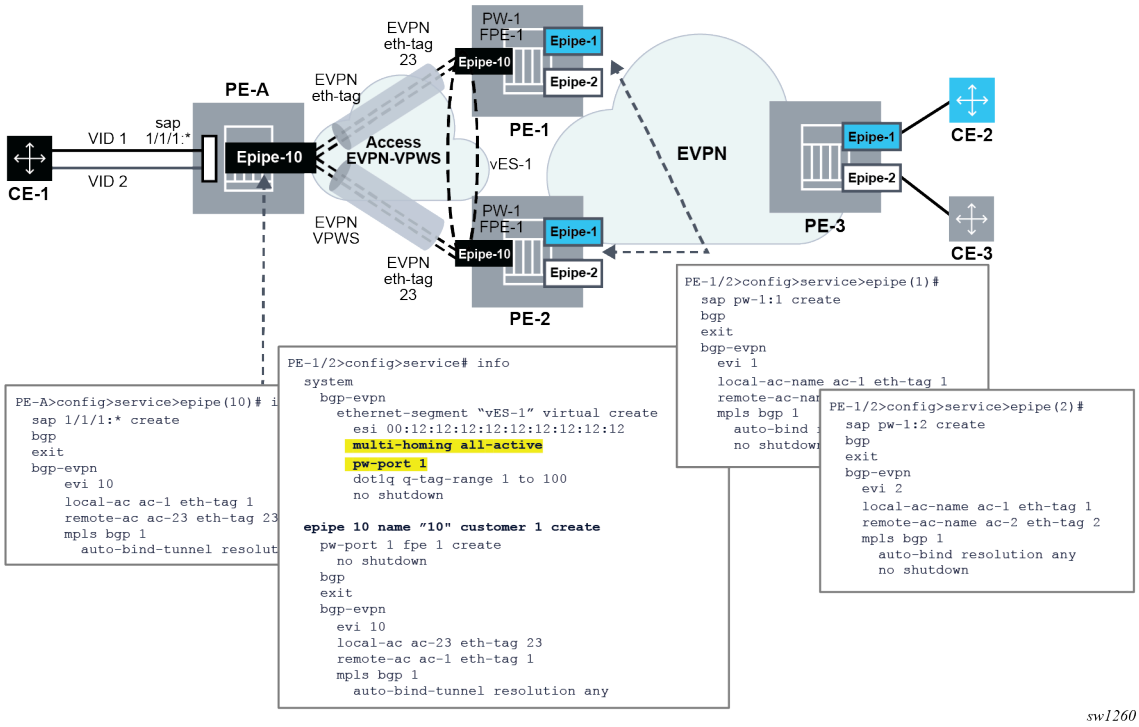
The following requirements apply:

- Port-based or FPE-based PW ports can be used in ESs
- PW port scenarios supported along with ESs are as follows:
  - port-based PW port
  - FPE-based PW port, where the stitching service uses a spoke SDP to the access CE
  - FPE-based PW port, where the stitching service uses EVPN-VPWS (MPLS) to the access CE

For all the preceding scenarios, fault-propagation to the access CE only works in the case of physical failures. Administrative shutdown of individual Epipes, PW SAPs, ESs or BGP-EVPN may result in traffic black holes.

The following figure shows the use of PW ports in ESs. In this example, an FPE-based PW port is associated with the ES, where the stitching service itself also uses EVPN-VPWS.

Figure 234: ES FPE-based PW port access using EVPN-VPWS



In this example, the following conditions apply:

- Redundancy is driven by EVPN all-active multihoming. ES-1 is a virtual ES configured on the FPE-based PW port on PE-1 and PE-2.
- The access network between the access PE (PE-A) and the network PEs (PE-1 and PE-2), uses EVPN-VPWS to backhaul the traffic. Therefore, PE-1 and PE-2 use EVPN-VPWS in the PW port stitching service, where:
  - PE-1 and PE-2 apply the same Ethernet tag configuration on the stitching service (Epipe 10)
  - Optionally PE-1 and PE-2 can use the same RD on the stitching service
  - AD per-EVI routes for the stitching service Ethernet tags are advertised with ESI=0
- Forwarding in the CE-1 to CE-2 or CE-3 direction, works as follows:
  - PE-A forwards traffic based on the selection of the best AD per-EVI route advertised by PE-1 and PE-2 for the stitching Epipe 10. This selection can be either BGP-based if PE-2 and PE-3 use the same RD in the stitching service, or EVPN-based if different RD is used.
  - When the PE-1 route is selected, PE-1 receives the traffic on the local PW-SAP for Epipe 1 or Epipe 2, and forwards it based on the customer EVPN-VPWS rules in the core.
- Forwarding in the CE-2 or CE-3 to CE-1 direction, works as follows:
  - PE-3 forwards the traffic based on the configuration of ECMP and aliasing rules for Epipe 1 and Epipe 2.
  - PE-3 can send the traffic to PE-2 and PE-2 to PE-A, following different directions.

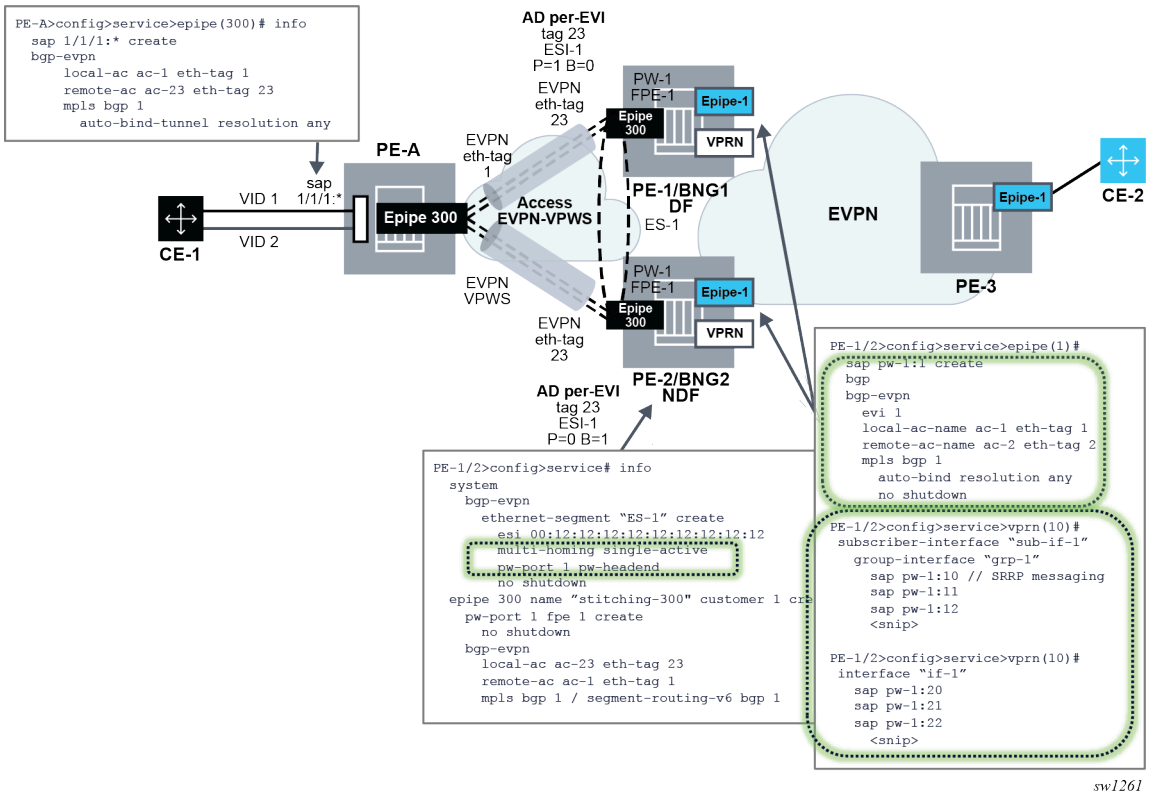
- If the user needs the traffic to follow a symmetric path in both directions, then the AD per-EVI route selection on PE-A and PE-3 can be handled so that the same PE (PE-1 or PE-2) is selected for both directions.
- For this example, the solution provides redundancy in case of node failures in PE-1 or PE-2. However, the administrative shutdowns, configured in some objects, are not propagated to PE-A, leading to traffic blackholing. As a result, black holes may be caused by the following events in PE-1 or PE-2:
  - Epipe 1 or Epipe 2 service shutdown
  - Epipe 1 or Epipe 2 BGP-EVPN MPLS shutdown
  - vES-1 shutdown
  - BGP shutdown

PW port-based ESes with multihoming on stitching Epipe

The solution described in PW port-based ESes with multihoming procedures on PW SAPs provides PW-headend redundancy where the access PE selects one of the PW-headend PE devices based on BGP best path selection, and the traffic from the core to the access may follow an asymmetric path. This is because the multihoming procedures are actually run on the PW SAPs of the core services, and the AD per-EVI routes advertised in the context of the stitching Epipe use an ESI=0.

SR OS also supports a different mode of operation called **pw-port headend** which allows running the multihoming procedures in the stitching Epipe and, therefore, use regular EVPN-VPWS primary or backup signaling to the access PE. The mode of operation is supported in a single-active mode shown in the following figure.

Figure 235: ES FPE-based pw-port headend



The following configuration triggers the needed behavior:

```
// ES and stitching Epipe config

PE-1/2>config>service# info
  system
    bgp-evpn
      ethernet-segment “ES-1” create
        esi 00:12:12:12:12:12:12:12:12:12
        multi-homing single-active
        pw-port 1 pw-headend
        no shutdown

  epipe 300 name ”stitching-300" customer 1 create
    pw-port 1 fpe 1 create
      no shutdown
    bgp-evpn
      local-attachment-circuit ac-23 eth-tag 23
      remote-attachment-circuit ac-1 eth-tag 1
      mpls bgp 1
        auto-bind-tunnel resolution any


// Services config

epipe 10
  sap pw-1:10 create
  bgp-evpn
    mpls bgp 1
```

```
epipe 11
  sap pw-1:10 create
  bgp-evpn
    mpls bgp 1
```

The configuration and functionality are divided in four aspects.

Configuration of single-active multihoming on ESs associated with PW ports of type pw-headend

In this mode, PW Ports are associated with single-active non-virtual Ethernet Segments. The **pw-headend** keyword is needed when associating the PW port.

```
PE-1/2>config>service# info
  system
    bgp-evpn
      ethernet-segment “ES-1” create
        esi 00:12:12:12:12:12:12:12:12:12
        multi-homing single-active
        pw-port 1 pw-headend
        no shutdown
```

The **pw-port** *id* **pw-headend** command indicates to the system that the multihoming procedures are run in the PW port stitching Epipe and the routes advertised in the context of the stitching Epipe contains the ESI of the ES.

Configuration of the PW port stitching Epipe

A configuration example of the stitching Epipe follows.

```
epipe 300 name ”stitching-300" customer 1 create
    pw-port 1 fpe 1 create
      no shutdown
    bgp-evpn
      local-attachment-circuit ac-23 eth-tag 23
      remote-attachment-circuit ac-1 eth-tag 1
      mpls bgp 1
        auto-bind-tunnel resolution any
```

The preceding example shows the configuration of a stitching EVPN VPWS Epipe with MPLS transport, however SRv6 transport is also supported.

When the ES is configured with a PW port in **pw-headend** mode, the stitching Epipe associated with the PW port is now running the ES and DF election procedures. Therefore, the following actions apply:

- an AD per-ES route is advertised with:
  - the RD or RT of the stitching Epipe
  - the configured ESI of the ES associated with the PW port
  - the ESI-label extended community with the multihomed mode indication and ESI label
- an AD per EVI route is advertised with:
  - the RD or RT of the stitching Epipe
  - the configured ESI where the PW port resides
  - the P/B bits according to the DF election procedures

- the non-DF drives the PW port operationally down with a flag MHStandby. As a result, all the PW SAPs contained in the PW port are brought operationally down. Optionally, the **config**>**service**>**epipe**>**pw-port**>**oper-up-on-mhstandby** command can be configured so that the PW port stays operationally up even if it is in MHStandby state (that is, the PE is non-DF). This command may speed up convergence in case a significant number of PW SAPs are configured in the same PW port.

Configuration of the PW port-contained PW SAPs and edge services

The edge services that contain the PW SAPs of the **pw-headend pw-port** command are configured without any other additional commands. These PW SAPs can be configured on Epipes, VPRN interfaces, or subscriber interfaces, VPLS (capture SAPs). As an example, if the PW SAP is configured on an Epipe EVPN-VPWS service:

```
epipe 10
  sap pw-1:10 create
  bgp-evpn
    mpls bgp 1
```

The behavior of the PW SAPs when the PW port is configured with the **pw-headend** keyword follows:

- The PW SAP is brought operationally down if the PW port is down. The PW port goes down with the reason MHStandby if the PE is a non-DF, or with reason stitching-svc-down if the EVPN destination is removed from the stitching Epipe.
- If the PW SAP is configured in an EVPN-VPWS edge service as in the preceding example, the following actions are performed:
  - An AD per ES route is advertised for the EVPN-VPWS service with the RD or RT of the service Epipe, the configured ESI of the ES associated with the PW port, and the ESI-label extended community with the multihomed mode indication of the ES and ESI label (label is the same value as in the AD per ES for the stitching Epipe). If the PW port is only down because of the MHStandby flag, the AD per ES route for the Epipe service is still advertised.
  - In addition, an AD per EVI route is advertised with the RD or RT of the service Epipe, the configured ESI of the ES associated with the PW port, and the P/B flags of the ES:
    - P=1/B=0 on the DF
    - P=0/B=1 on backup
    - P=0/B=0 on non-DFs and non-backup
  - If the PW port is down only because of MHStandby, the AD per EVI route for the service Epipe is still advertised.

Some considerations and dependencies between the PW port and the service Epipe PW SAPs

- If all the PW SAPs associated with the FPE PW port are brought down, the following rules apply:
  - state of the PW port does not change
  - does not trigger any AD per-ES/EVI or ES route withdraw toward the CE from the stitching Epipe
- Any event that brings down the PW port (except for MHStandby) triggers:
  - an AD per-EVI/ES route withdrawal within the context of the stitching Epipe
  - an ES route withdrawal
  - an AD per-EVI/ES routes withdrawal within the context of the service Epipes

  - the **pw-port**>**monitoring-oper-group** command can also modify the state of the PW port driven by the state of the operational group
- An individual PW SAP going administrative or operationally down while the PW port is still operationally up, the following actions may be performed:
  - may create black holes for that particular service
  - triggers the withdrawal of the AD per-EVI routes for the service Epipe (not the AD per-ES route, which is kept advertised if the PW port is up)
  - if the PW SAP is administratively not shutdown, the service Epipe AD per-ES/EVI routes mirror the AD per-ES/EVI routes of the stitching service and they are advertised if the routes for the stitching Epipe are advertised

The PW SAP can also be configured on VPRN services (under regular interfaces or subscriber interfaces) and works without any special consideration, other than that a PW port in non-DF state brings down the PW SAP and, therefore, the interface. Similarly, VPLS services with capture PW SAPs support this mode of operation too.

6.5.22 Interaction of EVPN and other features

This section contains information about EVPN and how it interacts with other features.

6.5.22.1 Interaction of EVPN-VXLAN and EVPN-MPLS with existing VPLS features

When enabling existing VPLS features in an EVPN-VXLAN or an EVPN-MPLS enabled service, the following must be considered:

- EVPN-VXLAN services are not supported on I-VPLS/B-VPLS. VXLAN cannot be enabled on those services. EVPN-MPLS is only supported in regular VPLS and B-VPLS. Other VPLS types, such as **m-vpls**, are not supported with either EVPN-VXLAN or EVPN-MPLS VPLS **etree** services are supported with EVPN-MPLS.
- In general, no router-generated control packets are sent to the EVPN destination bindings, except for ARP, VRRP, ping, BFD and Eth-CFM for EVPN-VXLAN, and proxy-ARP/proxy-ND confirm messages and Eth-CFM for EVPN-MPLS.
- The following rules apply to xSTP and M-VPLS services:
  - xSTP can be configured in BGP-EVPN services. BPDUs are not sent over the EVPN bindings.
  - **bgp-evpn** is blocked in m-vpls services; however, a different m-vpls service can manage a SAP or spoke SDP in a **bgp-evpn**-enabled service.
- In **bgp-evpn** enabled VPLS services, **mac-move** can be used in SAPs/SDP bindings; however, the MACs being learned through BGP-EVPN are considered.

  Note: The MAC duplication already provides a protection against mac-moves between EVPN and SAPs/SDP bindings.

- **disable-learning** and other fdb-related tools only work for data plane learned MAC addresses.
- **mac-protect** cannot be used in conjunction with EVPN.

Note: EVPN provides its own protection mechanism for static MAC addresses.

- MAC OAM tools are not supported for **bgp-evpn** services, that is: **mac-ping**, **mac-trace**, **mac-populate**, **mac-purge,** and **cpe-ping**.
- EVPN multihoming and BGP-MH can be enabled in the same VPLS service, as long as they are not enabled in the same SAP-SDP or spoke SDP. There is no limitation on the number of BGP-MH sites supported per EVPN-MPLS service.

  Note: The number of BGP-MH sites per EVPN-VXLAN service is limited to 1.

- SAPs/SDP bindings that belong to a specified ES but are configured on non-BGP-EVPN-MPLS-enabled VPLS or Epipe services are kept down with the **StandByForMHProtocol** flag.
- CPE-ping is not supported on EVPN services but it is in PBB-EVPN services (including I-VPLS and PBB-Epipe). CPE-ping packets are not sent over EVPN destinations. CPE-ping only works on local active SAP or SDP bindings in I-VPLS and PBB-Epipe services.
- Other commands not supported in conjunction with **bgp-evpn** are:
  - Subscriber management commands under service, SAP, and SDP binding interfaces
  - BPDU translation
  - L2PT termination
  - MAC-pinning
- Other commands not supported in conjunction with **bgp-evpn mpls** are:

  SPB configuration and attributes

6.5.22.2 Interaction of PBB-EVPN with existing VPLS features

In addition to the B-VPLS considerations described in section Interaction of EVPN-VXLAN and EVPN-MPLS with existing VPLS features, the following specific interactions for PBB-EVPN should also be considered:

- When **bgp-evpn mpls** is enabled in a **b-vpls** service, an **i-vpls** service linked to that **b-vpls** cannot be an R-VPLS (the **allow-ip-int-bind** command is not supported).
- The ISID value of 0 is not allowed for PBB-EVPN services (I-VPLS and Epipes).
- The **ethernet-segments** can be associated with **b-vpls** SAPs/SDP bindings and **i-vpls/epipe** SAPs/SDP bindings,; however, the same ES cannot be associated with **b-vpls** and **i-vpls/epipe** SAP or SDP bindings at the same time.
- When PBB-Epipes are used with PBB-EVPN multihoming, spoke SDPs are not supported on **ethernet-segments**.
- When **bgp-evpn mpls** is enabled, eth-tunnels are not supported in the b-vpls instance.

6.5.22.3 Interaction of VXLAN, EVPN-VXLAN and EVPN-MPLS with existing VPRN or IES features

When enabling existing VPRN features on interfaces linked to VXLAN R-VPLS (static or BGP-EVPN based), or EVPN-MPLS R-VPLS interfaces, consider that the following are not supported:

- the commands **arp-populate** and **authentication-policy**
- dynamic routing protocols such as IS-IS, RIP, and OSPF
- BFD on EVPN tunnel interfaces

When enabling existing IES features on interfaces linked to VXLAN R-VPLS or EVPN-MPLS R-VPLS interfaces, the following commands are not supported:

- **if>vpls>evpn-tunnel**
- **bgp-evpn>ip-route-advertisement**
- **arp-populate**
- **authentication-policy**

Dynamic routing protocols such as IS-IS, RIP, and OSPF are also not supported.

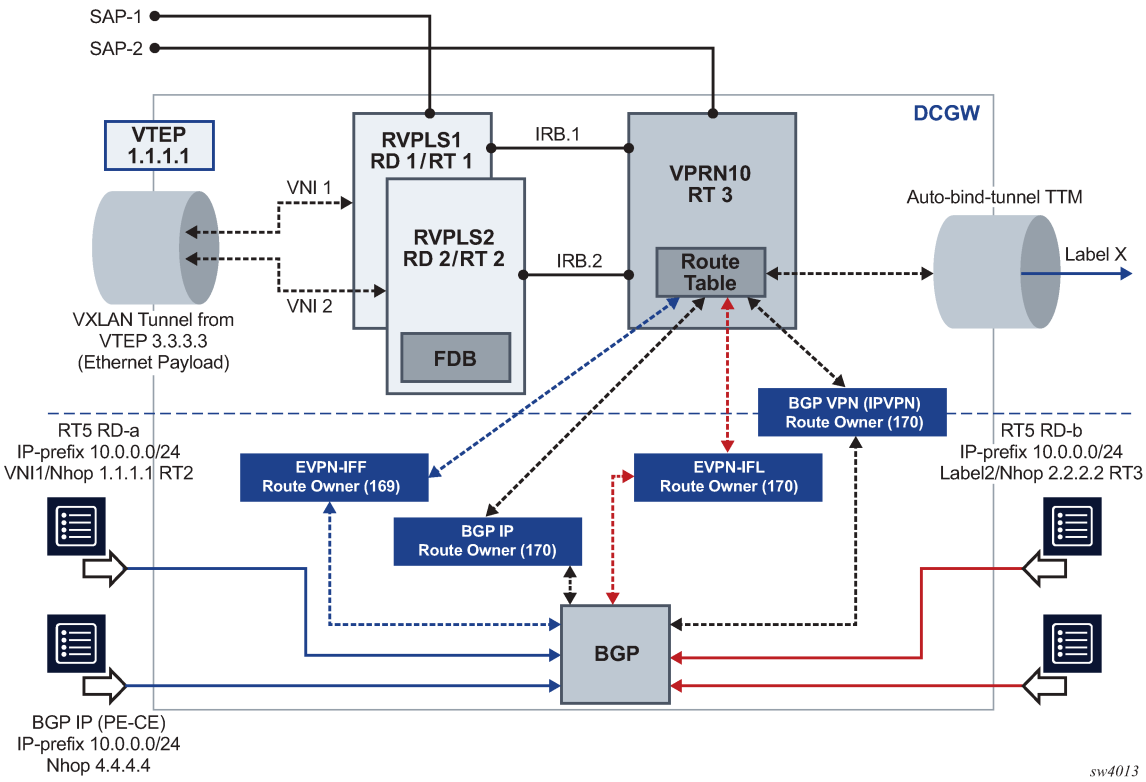
6.5.23 Interaction of EVPN with BGP owners in the same VPRN service

SR OS allows multiple BGP owners in the same VPRN service to receive or advertise IP prefixes contained in the VPRN's route table. Specifically, the same VPRN route table can simultaneously install and process IPv4 or IPv6 prefixes for the following owners:

- EVPN-IFL (EVPN Interface-less IP prefix routes)
- EVPN-IFF (EVPN Interface-ful IP prefix routes)
- VPN-IP (also referred to as IPVPN routes)
- IP (also referred to as BGP PE-CE routes)

Figure 236: Different owners supported on the same VPRN shows the service architecture and the concept of different owners supported on the same VPRN.

Figure 236: Different owners supported on the same VPRN



In the example shown in Figure 236: Different owners supported on the same VPRN, VPRN 10 is configured with regular interfaces and R-VPLS interfaces and receives the same prefix 10.0.0.0/24 via the four owners.

EVPN-IFL routes are EVPN IP-Prefix (or type 5) routes that are imported and exported based on the VPRN **bgp-evpn**>**mpls** configuration, as described in Interface-less IP-VRF-to-IP-VRF model (IP encapsulation) for MPLS tunnels.

EVPN-IFF routes are EVPN IP-Prefix (or type 5) routes that are imported and exported based on the configuration of the R-VPLS services attached to the VPRN. EVPN-IFF routes are advertised and processed if the R-VPLS services are configured with the **configure>service>vpls>bgp-evpn>ip-route-advertisement** command. Although installed in the VPRN service, EVPN-IFF routes use the route distinguisher and route targets determined by the configuration in the R-VPLS, and are supported in R-VPLS services with VXLAN or MPLS encapsulations. See Interface-ful IP-VRF-to-IP-VRF with SBD IRB model for more information about EVPN-IFF routes.

In addition to EVPN-IFL and EVPN-IFF routes, BGP IP and VPN-IP families are supported on the same VPRN.

6.5.23.1 Interworking of EVPN-IFL and IPVPN in the same VPRN

This section describes the SR OS interworking details for BGP owners in the same VPRN. The behavior is compliant with *draft-ietf-bess-evpn-ipvpn-interworking*.

A VPRN service can be configured to support EVPN-IFL and IPVPN simultaneously. For example, the following MD CLI excerpt shows a VPRN service configured for EVPN-IFL (**vprn**>**bgp-evpn** context) and IPVPN (**vprn**>**bgp-ipvpn** context):

```
[ex:/configure service vprn "vprn-ipvpn-evpnifl-AL-80"]
A:admin@PE-2# info
    admin-state enable
    service-id 80
    customer "1"
    bgp-evpn {
        mpls 1 {
            admin-state enable
            route-distinguisher "192.0.2.2:80"
            vrf-target {
                community "target:64500:80"
            }
            auto-bind-tunnel {
                resolution any
            }
        }
    }
    bgp-ipvpn {
        mpls {
            admin-state enable
            route-distinguisher "192.0.2.2:80"
            vrf-target {
                community "target:64500:80"
            }
            auto-bind-tunnel {
                resolution any
            }
        }
    }
    interface "lo0" {
        loopback true
        ipv4 {
            primary {
                address 2.2.2.2
                prefix-length 32
            }
        }
    }
```

When EVPN-IFL and IPVPN are both enabled on the same VPRN, the following rules apply:

- IPVPN and EVPN-IFL routes are treated by BGP as separate routes; that is, the selection is done at route table level and not at the BGP level.
- At the route table level, IPVPN and EVPN-IFL routes may have the same route table preference (by default, this is 170 for both routes), route selection between IPVPN and EVPN-IFL routes is based on regular BGP path selection.
- ECMP across IPVPN and EVPN-IFL routes for the same prefix is not supported. When **vprn**>**ecmp** is configured to 2 or greater, installing multiple equal cost next hops for the same prefix in the VPRN route table is only supported within the same route owner, IPVPN or EVPN IFL.
- When EVPN-IFL and IPVPN are both enabled in the same VPRN, by default, EVPN-IFL routes are exported into IPVPN and the other way around (CLI configuration is not required).
- The **configure**>**service**>**vprn**>**allow-export-bgp-vpn** command is relevant within the same owner (either IPVPN or EVPN-IFL) and works as follows:

- The command re-exports a received EVPN-IFL route into an EVPN-IFL route to a different peer.
- The command also re-exports a received IPVPN route into an IPVPN route.
- If EVPN-IFL and IPVPN are both configured in the same VPRN, an EVPN-IFL route is automatically re-exported into an IPVPN route. Conversely, an IPVPN route is re-exported into an EVPN-IFL. This is true unless export policies prevent the automatic re-export function.

6.5.23.2 Route selection across EVPN-IFL and other owners in the VPRN service

This section describes the rules for route selection among EVPN-IFL, VPN-IP, and IP route table owners.

A PE may receive an IPv4 or IPv6 prefix in routes from different or same owners, and from the same or different BGP peer. For example, prefix 10.0.0.0/24 can be received as an EVPN-IFL route and also received as a VPN-IPv4 route. Or prefix 2001:db8:1::/64 can be received in two EVPN-IFL routes with different route distinguishers from different peers. In all these examples, the router selects the best route in a deterministic way.

For EVPN-IFF route selection rules, see Route selection for EVPN-IFF routes in the VPRN service. In SR OS, the VPRN route table route selection for all BGP routes, excluding EVPN-IFF, is performed using the following ordered, tie-breaking rules:

1. valid route wins over invalid route
2. lowest origin validation state (valid<not found<invalid) wins
3. lowest RTM (route table) preference wins
4. highest local preference wins
5. shortest D-PATH wins (skipped if **d-path-length-ignore** is configured)
6. lowest AIGP metric wins
7. shortest AS_PATH wins (skipped if the **as-path-ignore** command is configured for the route owner)
8. lowest origin wins
9. lowest MED wins
10. lowest owner type wins (BGP<BGP-LABEL<BGP-VPN)

    Note: BGP-VPN refers to VPN-IP and EVPN-IFL in this context.

11. EBGP wins
12. lowest route table or tunnel-table cost to the next hop (skipped if the **ignore-nh-metric** command is configured)
13. lowest next-hop type wins (resolution of next hop in TTM wins vs RTM) (skipped if the **ignore-nh-metric** command is configured)
14. lowest router ID wins (skipped if the **ignore-router-id** command is configured)
15. shortest cluster_list length wins
16. lowest IP address

    Note: The IP address refers to the peer that advertised the route.

17. EVPN-IFL wins over IPVPN routes
18. next-hop check (IPv4 next hop wins over IPv6 next hop, and then lowest next hop wins)

    Note: This is a tiebreaker if BGP receives the same prefix for VPN-IPv6 and IFL. An IPv6-Prefix received as VPN-IPv6 is mapped as IPv6 next hop, whereas the same IPv6 prefix received as IFL could have an IPv4 next hop.

19. RD check for RTM (lowest RD wins)

ECMP is not supported across EVPN-IFL and other owners, but it is supported within the EVPN-IFL owner for multiple EVPN-IFL routes received with the same IP prefix. When ECMP is configured with N number of paths in the VPRN, BGP orders the routes based on the previously described tie-break criteria breaking out after step 13 (lowest next-hop type). At that point, BGP creates an ECMP set with the best N routes.

Example:

In a scenario in which two EVPN-IFL routes are received on the same VPRN with same prefix, 10.0.0.0/24; different RDs 192.0.2.1:1 and 192.0.2.2; and different router ID, 192.0.2.1 and 192.0.2.2; the following tie-breaking criteria are considered.

- Assuming everything else is the same, BGP orders the routes based on the preceding criteria and prefers the route with the lowest router ID.
- If vprn>ecmp=2, the two routes are treated as equal in the route table and added to the same ECMP set.

6.5.23.3 Route selection for EVPN-IFF routes in the VPRN service

While the route selection in VPRN for other BGP owners described in Route selection across EVPN-IFL and other owners in the VPRN service follows similar criteria, the default selection for EVPN-IFF routes in the VPRN route table follow different rules:

- By default, EVPN-IFF routes have a VPRN route table preference of 169; therefore, EVPN-IFF routes are preferred over EVPN-IFL, VPN-IP, or IP owners that have a preference of 170.
- When two or more EVPN-IFF routes with the same IPv4 or IPv6 prefix and length, but with different route keys are received (for example, two routes with the same prefix and length but with different RDs), BGP hands the EVPN-IFF routes over to the EVPN application for selection. In this case, EVPN orders the routes based on their {R-VPLS Ifindex, RD, Ethernet Tag} and considers the top one for installing in the route table if ecmp is 1. If ecmp is N, the top N routes for the prefix are selected.

  Example:

  – Consider the following two IP-Prefix routes that are received on the same R-VPLS service:

    Route 1: (RD=192.0.0.1:30, Ethernet Tag=0, Prefix=10.0.0.0/24, next-hop 192.0.0.1)

    Route 2: (RD=192.0.0.2:30, Ethernet Tag=0, Prefix=10.0.0.0/24, next-hop 192.0.0.2)

  – Because their route key is different (their RDs do not match), EVPN orders them based on R-VPLS Ifindex first, then RD, and then Ethernet Tag. Because they are received on the same R-VPLS, the Ifindex is the same on both. The top route on the priority list is Route 1, based on its lower RD. If the VPRN's **ecmp** command has a value of 1, only Route 1 is installed in the VPRN's route table.
- If the previously described way of selecting EVPN-IFF routes in the VPRN does not satisfy the user requirements, the **configure service system bgp evpn ip-prefix-routes iff-bgp-path-selection** command enables a BGP-based path selection for EVPN-IFF routes, which is equivalent to the selection followed for EVPN-IFL or IPVPN routes with the following considerations:

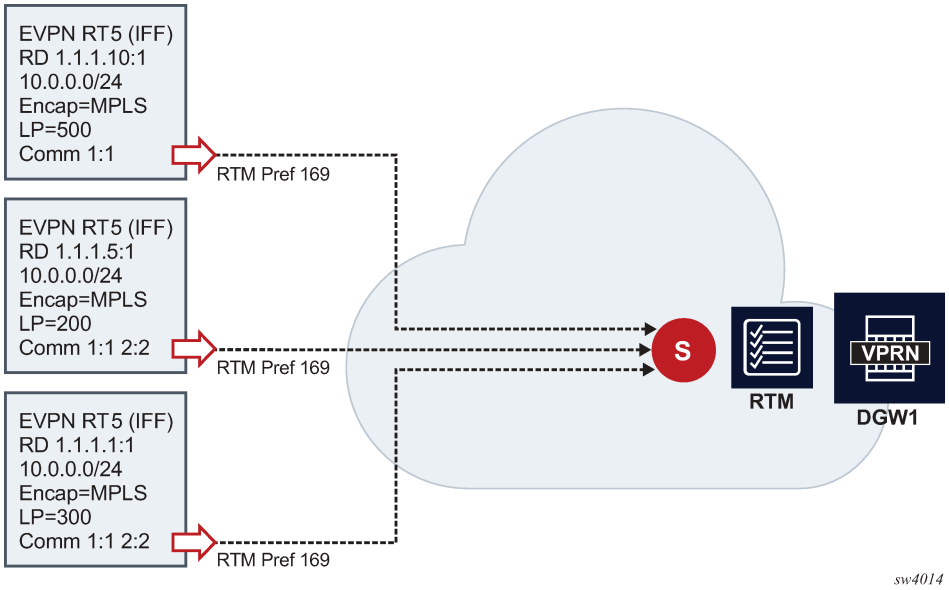
– All the tie breakers in Route selection across EVPN-IFL and other owners in the VPRN service are valid for EVPN-IFF, except for the lowest owner type tie breaker that does not affect EVNP-IFF routes.

Note: The **ignore-nh-metric** command does not exist for EVPN-IFF routes.

– While the tie breakers in Route selection across EVPN-IFL and other owners in the VPRN service are also valid for comparing an EVPN-IFL and an IPVPN route for the same prefix, an EVPN-IFF and IPVPN route are never compared based on those tie breakers. They are only used to compare multiple EVPN-IFF routes for the same prefix, and only when the **iff-bgp-path-selection** command is configured.

- If the **iff-bgp-path-selection** command is configured, the EVPN-IFF path selection for the N routes that form the ECMP set follow the same rules as in Route selection across EVPN-IFL and other owners in the VPRN service for EVPN-IFL or IPVPN routes.

Figure 237: EVPN-IFF path selection for N routes with iff-bgp-path-selection configured



Upon receiving the same IP prefix 10.0.0.0/24 for the same VPRN in EVPN-IFF routes with different RDs, as shown in Figure 237: EVPN-IFF path selection for N routes with iff-bgp-path-selection configured, the following selection criteria is used:

- If the **iff-bgp-path-selection** command is configured, the selection is based on BGP path selection, and the selected route is the top route, based on the highest Local Preference (LP)(500>300>200).
- However, if the **iff-bgp-path-selection** command is not configured, the bottom route is selected assuming the three routes are received on the same R-VPLS, and based on the lower RD (1.1.1.1:1<1.1.1.5:1<1.1.1.10:1).

Although, by default, EVPN-IFF routes have an RTM preference of 169 and they are preferred over the RTM preference of 170 used for the other BGP route owners, a selection across EVPN-IFF and the other owners may result if the RTM preference is changed and made equal (via import policy or **config**>**router**>**bgp**>**preference**). Note that the route table preference for EVPN-IFF routes can be

changed from the default value 169 if the **iff-attribute-uniform-propagation** command is enabled and an import policy or the **config**>**router**>**bgp**>**preference** command is configured to change it.

In case the RTM preference is changed and made equal to the same as for EVPN-IFF routes, if multiple routes with the same key (different RD) are received for EVPN-IFF and another owner in the same VPRN, the selection order is as follows:

1. BGP (IPv4 or IPv6)
2. BGP-IPVPN
3. EVPN-IFF
4. EVPN-IFL

Note: The previous selection order applies to EVPN-IFF routes when compared with others. When BGP-IPVPN, BGP IP and EVPN-IFL routes are compared, regular BGP path selection is used as described in Route selection across EVPN-IFL and other owners in the VPRN service.

6.5.23.4 BGP path attribute propagation

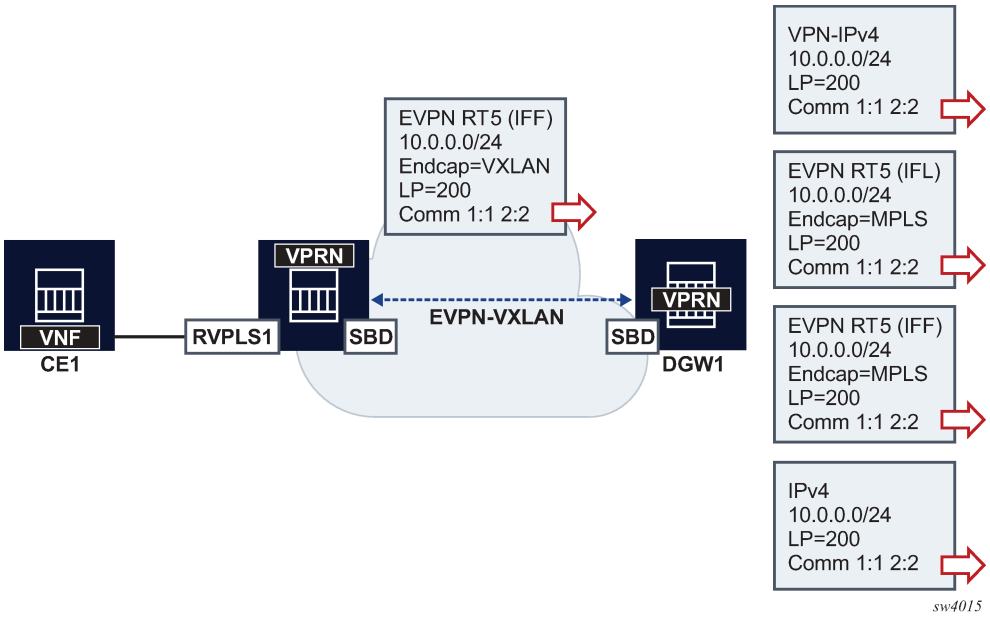
A VPRN can receive and install routes for a specific BGP for a specific BGP owner. The routes may be re-exported in the context of the same VPRN and to the same BGP owner or a different one. For example, an EVPN-IFL route can be received from peer N, installed in VPRN 1, and re-exported to peer M using family VPN-IPv4.

When re-exporting BGP routes, the original BGP path attributes are preserved without any configuration in the following cases:

- EVPN-IFL route re-exported into an IPVPN route, and the other way around
- EVPN-IFL route re-exported into a BGP IP route (PE-CE), and the other way around
- IPVPN route re-exported into a BGP IP route (PE-CE), and the other way around
- EVPN-IFL, IPVPN or BGP IP routes re-exported into a route of the same owner. For example, EVPN-IFL to EVPN-IFL, when the **allow-export-bgp-vpn** command is configured.

BGP path attributes to or from EVPN-IFF are not preserved by default. If BGP Path Attribute propagation is required, the **configure service system bgp-evpn ip-prefix-routes iff-attribute-uniform-propagation** command must be configured. Figure 238: BGP path attribute propagation when iff-attribute-uniform-propagation is configured shows an example of BGP Path Attribute propagation from EVPN-IFF to the other BGP owners in the VPRN when the **iff-attribute-uniform-propagation** command is configured.

Figure 238: BGP path attribute propagation when iff-attribute-uniform-propagation is configured



In the example in Figure 238: BGP path attribute propagation when iff-attribute-uniform-propagation is configured, DGW1 propagates the received LP and communities on an EVPN-IFF route, when advertising the same prefix into any type of BGP owner route, including VPN-IPv4/6, EVPN-IFL, EVPN-IFF, IPv4, or IPv6. If the **iff-attribute-uniform-propagation** command is not configured on DCW1, no BGP path attributes are propagated, but are re-originated instead. The propagation in the opposite direction follows the same rules; configuration of the **iff-attribute-uniform-propagation** command is required.

When propagating BGP path attributes, the following criteria are considered:

- The propagation is compliant with the uniform propagation described in *draft-ietf-bess-evpn-ipvpn-interworking*.
- The following extended communities are filtered or excluded when propagating attributes:
  - all extended communities of type 0x06 (EVPN type). In particular, all those that are supported by routes type 5:
    - MAC Mobility extended community (sub-type 0x00)
    - EVPN Router's MAC extended community (sub-type 0x03)
  - BGP encapsulation extended community
  - all Route Target extended communities
- The BGP Path Attribute propagation within the same owner is supported in the following cases:
  - EVPN-IFF to EVPN-IFF (route received on R-VPLS and advertised in a different R-VPLS context), assuming the **iff-attribute-uniform-propagation** command is configured
  - EVPN-IFL to EVPN-IFL (route received on a VPRN and re-advertised based on the configuration of **vprn**>**allow-export-bgp-vpn**)
  - VPN-IPv4/6 to VPN-IPv4/6 (route received on a VPRN and re-advertised based on the configuration of **vprn**>**allow-export-bgp-vpn**)
- The propagation is supported for iBGP and eBGP as follows:

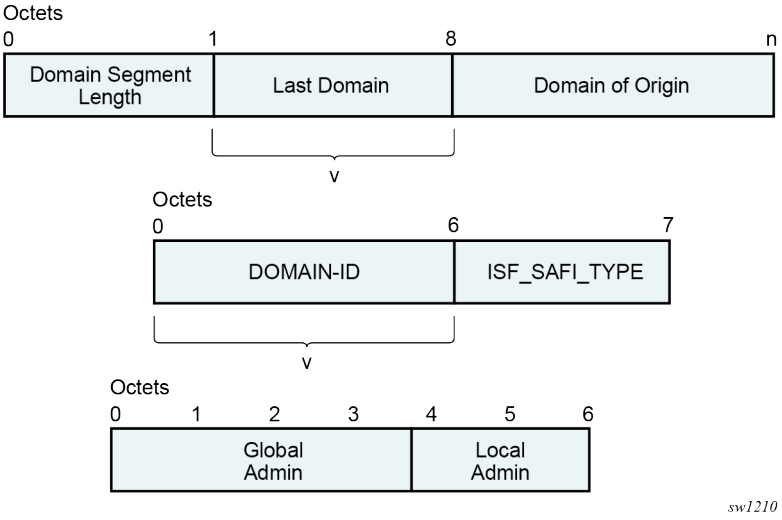
  - iBGP-only attributes can only be propagated to iBGP peers
  - non-transitive attributes are propagated based on existing rules
  - when peering an eBGP neighbor, the AS_PATH is prepended by the VPRN ASN
- If ECMP is enabled in the VPRN and multiple routes of the same BGP owner with different Route Distinguishers are installed in the route table, only the BGP path attributes of the best route are subject for propagation.

6.5.23.5 BGP D-PATH attribute for Layer 3 loop protection

SR OS has a full implementation of the D-PATH attribute as described in *draft-ietf-bess-evpn-ipvpn-interworking*.

D-PATH is composed of a sequence of domain segments (similar to AS_PATH). Each domain segment is graphically represented as shown in the following figure.

Figure 239: D-PATH attribute

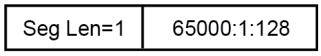


Where:

- Each domain segment comprises of <domain_segment_length, domain_segment_value>, where the domain segment value is a sequence of one or more domains.
- Each domain is represented by <DOMAIN-ID:ISF_SAFI_TYPE>, where the newly added domain is added by a gateway, is always prepended at the left of the existing last domain.
- The supported ISF_SAFI_TYPE values are:
  - 0 = Local ISF route
  - 1 = safi 1 (typically identifies PE-CE BGP domains)
  - 70 = evpn
  - 128 = safi 128 (IPVPN domains)
- Labeled unicast IP routes do not support D-PATH.

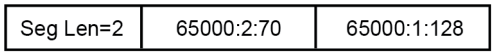
- The D-PATH attribute is only modified by a gateway and not by an ABR/ASBR or RR. A gateway is defined as a PE where a VPRN is instantiated, and that VPRN advertises or receives routes from multiple BGP owners (for example, EVPN-IFL and BGP-IPVPN) or multiple instances of the same owner (for example, VPRN with two BGP-IPVPN instances)

Suppose a router receives prefix P in an EVPN-IFL instance with the following D-PATH from neighbor N.



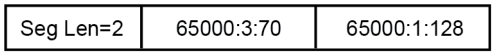
sw1312

If the router imports the route in VPRN-1, BGP-EVPN SRv6 instance with domain 65000:2, the router readvertises the route to its BGP-IPVPN MPLS instance as follows.



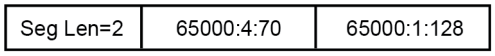
sw1313

If the router imports the route in VPRN-1, BGP-EVPN SRv6 instance with domain 65000:3, the router readvertises the route to its BGP-EVPN MPLS instance as follows.



sw1314

If the router imports the route in VPRN-1, BGP-EVPN MPLS instance with domain 65000:4, the router readvertises the route to its PE-CE BGP neighbor as follows.



sw1315

When a BGP route of families that support D-PATH is received and must be imported in a VPRN, the following rules apply:

- All domain IDs included in the D-PATH are compared with the local domain IDs configured in the VPRN. The local domain IDs for the VPRN include a list of (up to four) domain IDs configured at the **vprn** or **vprn bgp instance** level, including the domain IDs in local attached R-VPLS instances.
- If one or more D-PATH domain IDs match any local domain IDs for the VPRN, the route is not installed in the VPRN's route table.
- In the case where the IP-VPN or EVPN route matches the import route target in multiple VRFs, the D-PATH loop detection works per VPRN. For example, for each VPRN, BGP checks if the received domain IDs match any locally configured (maximum 4) domain IDs for that VPRN. A route may have a looped domain for one VPRN and not the other. In this case, BGP installs a route only in the VPRN route table that does not have a loop; the route is not installed in the VPRN that has the loop.
- A route that is not installed in any VPRN RTM (due to the domain ID matching any of the local domain IDs in the importing VPRNs) is still kept in the RIB-IN. The route is displayed in the **show router bgp routes** command with a **DPath Loop VRFs** field, indicating the VPRN in which the route is not installed due to a loop.
- Route target-based leaking between VPRNs and D-PATH loop detection is described in the following example.

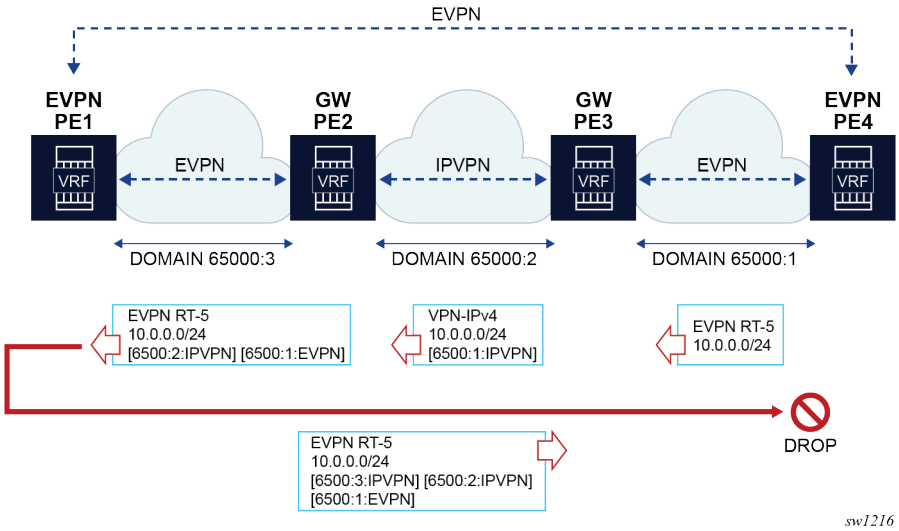
  Consider an EVPN-IFL route to prefix P imported in VPRN 20 (configured with domain 65000:20) is leaked into VPRN 30.

  When the route to prefix P is readvertised in the context of VPRN 30, which is enabled for BGP-IPVPN MPLS and BGP-EVPN MPLS, the readvertised BGP-IPVPN and BGP-EVPN routes have a D-PATH

with a prepended domain 65000:20:0. That is, leaked routes are readvertised with the domain ID of the VPRN of origin and an ISF_SAFI_TYPE = 0, as described in *draft-ietf-bess-evpn-ipvpn-interworking*.

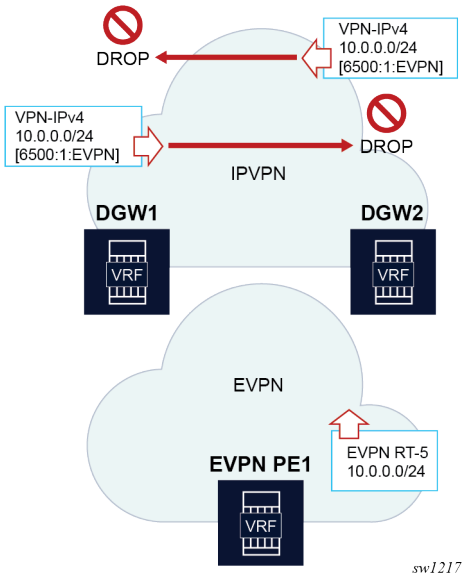
In the D-PATH example shown in the following figure, the different gateway PEs along the domains modify the D-PATH attribute by adding the source domain and family. If PE4 receives a route for the prefix with the domain of PE4 included in the D-PATH, PE4 does not install the route to avoid control plane loops.

Figure 240: D-PATH attribute example



In the D-PATH example shown in the following figure, DGW1 and DGW2 rely on the D-PATH attribute to automatically discard the prefixes received from the peer gateway in IPVPN and avoid loops by reinjecting the route back into the EVPN domain.

Figure 241: D-PATH attribute example two



Note: While site-of-origin extended communities and policies can be used in Figure 241: D-PATH attribute example two, the D-PATH method works across multiple domains and does not require policies.

6.5.23.5.1 BGP D-PATH configuration

The D-PATH attribute is modified on transmission or processed on reception based on the local VPRN or R-VPLS configuration. The domain ID is configured per-BGP instance and the ISF_SAFI_TYPE automatically derived from the instance type that imported the original route.

The **domain-id** is configured at **service bgp instance** level as a six-byte value that includes a global **admin** value and a local **admin** value, for example, 65000:1. Domain ID configuration is supported on:

- VPRN BGP-EVPN MPLS and SRv6 instances (EVPN-IFL)
- VPRN BGP-IPVPN MPLS and SRv6 instance
- R-VPLS BGP-EVPN MPLS and VXLAN instances (EVPN-IFF only – the R-VPLS is configured with the **evpn-tunnel** command)
- VPRN BGP neighbors (PE-CE)
- VPRN level (for local routes)

The following is an example CLI configuration:

```
// domain-id configuration

*[ex:configure service vprn "blue" bgp-evpn mpls 1]
*[ex:configure service vprn "blue" bgp-evpn segment-routing-v6 1]
*[ex:configure service vprn "blue" bgp-ipvpn mpls 1]
*[ex:configure service vprn "blue" bgp-ipvpn segment-routing-v6 1]
*[ex:configure service vprn "blue" bgp]
*[ex:configure service vpls "blue" bgp-evpn routes ip-prefix]
+-- domain-id <global-field:local-field>


*[ex:configure service vprn "blue"]
A:admin@PE-2#
+-- local-routes-domain-id <global-field:local-field>
// used as the domain-id for non-bgp routes in the VPRN.


// Example ‘a’

*[ex:configure service vprn "blue" bgp-ipvpn mpls 1]
    domain-id 65000:1
```

In the preceding "example 'a'", if a VPN-IPv4 route is received from a neighbor, imported in VPRN "blue" and exported to another neighbor as EVPN, the router prepends a D-PATH segment <65000:1:IPVPN> to the advertised EVPN RT5.

```

// Example ‘b’

*[ex:configure service vprn "blue"]
    local-routes-domain-id 65000:10
```

In the preceding "example 'b'", the **local-routes-domain-id** is configured at the **vprn** level. When configured, local routes (direct, static, IGP routes) are advertised with a D-PATH that contains the **vprn**>**local-routes-domain-id**.

The following additional considerations apply:

- If **vprn**>**local-routes-domain-id** is not configured, the local routes are advertised into the BGP instances with no D-PATH.
- If a VPRN BGP instance is not configured with a domain ID, the following handling applies:
  - Routes imported in the VPRN BGP instance are readvertised in a different instance without modifying the D-PATH.
  - Routes exported in the VPRN BGP instance are advertised with the D-PATH modified to include the domain ID of the instance that imported the route in the first place.
- Up to a maximum of four domain IDs per VPRN are supported. This includes domain IDs configured in the associated R-VPLS services.
- Modifying the domain IDs list initiates a route refresh for all address families associated with the VPRN.

6.5.23.5.2 BGP D-PATH and BGP best path selection

D-PATH is also considered for the BGP best path selection, as described in *draft-ietf-bess-evpn-ipvpn-interworking*.

As D-PATH is introduced in networks, not all the PEs may support D-PATH for BGP path selection. To guarantee compatibility in networks with PEs that do not support D-PATH, the following command determines if the D-PATH should be considered for BGP best-path selection.

```
ex:/configure]
A:admin@PE-3#
router “Base” bgp best-path-selection d-path-length-ignore <boolean> // default: false
service vprn <string> bgp best-path-selection d-path-length-ignore <boolean> // default: false
service vprn <string> d-path-length-ignore <boolean> // default: false

configure service system bgp evpn ip-prefix-routes d-path-length-ignore <boolean> // default:
 false
```

The following conditions apply to the **d-path-length-ignore** command usage:

- When **d-path-length-ignore** is configured at the base router level (or **vprn**>**bgp** level for PE-CE routes), BGP ignores the D-PATH domain segment length for best path selection purposes. This ignores **d-path-length** when comparing two VPN routes or two IFL routes within the same RD. These VPN or IFL routes are processed in main BGP instance.
- When **d-path-length-ignore** is configured at the VPRN router level, the VPRN RTM ignores the D-PATH domain segment length for best path selection purposes (for routes in VPRN).
- When **d-path-length-ignore** is configured at the **service system bgp evpn ip-prefix-routes** context, EVPN ignores the D-PATH length when **iff-bgp-path-selection** is enabled.
- When **d-path-length-ignore** is not configured, the D-PATH length is considered in the BGP best path selection process (at the BGP, the RTM, and IFF levels, respectively).

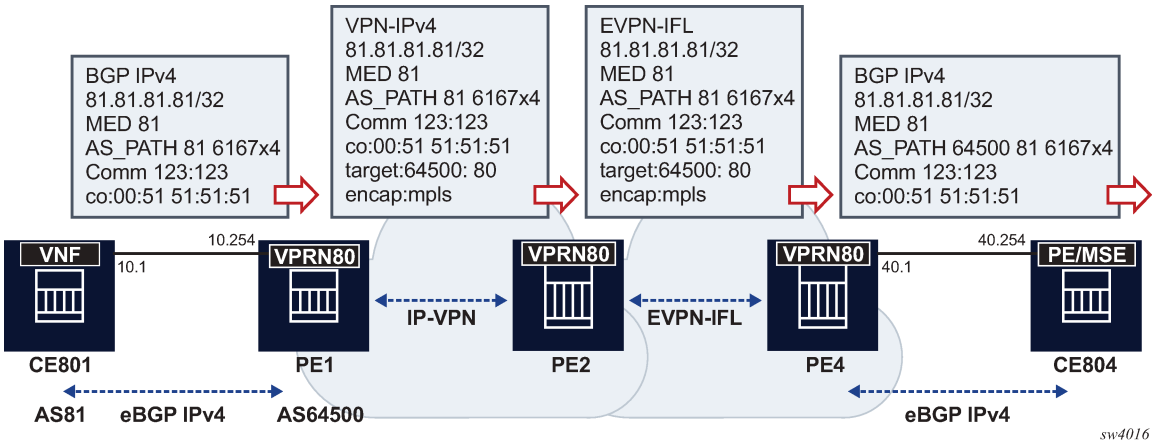
6.5.23.6 Configuration examples

This section describes configuration examples for stitching IPVPN and EVPN-IFL domains and the propagation of BGP path attributes for EVPN-IFF.

6.5.23.6.1 Example 1 - stitching IPVPN and EVPN-IFL domains

In this configuration example, IPVPN and EVPN-IFL are simultaneously configured in VPRN 80 of PE2. This allows the stitching of IPVPN and EVPN-IFL domains, as shown in Figure 242: Stitching IPVPN and EVPN-IFL domains.

Figure 242: Stitching IPVPN and EVPN-IFL domains



The following is an example configuration of PE1, PE2, and PE4 for VPRN 80.

Note: In this scenario, the BGP path attributes added by CE801 are propagated all the way up to CE804, across the VPRN-IPv4 and EVPN-IFL families.

```
// PE1's VPRN 80
A:PE-1# configure service vprn 80
A:PE-1>config>service>vprn# info
----------------------------------------------
            router-id 192.0.2.1
            autonomous-system 64500
            interface "lo0" create
                address 1.1.1.1/32
                loopback
            exit
            interface "local" create
                address 10.0.0.254/24
                sap 1/1/c1/1:80 create
                exit
            exit
            bgp-ipvpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    route-distinguisher 192.0.2.1:80
                    vrf-target target:64500:80
                    no shutdown
```

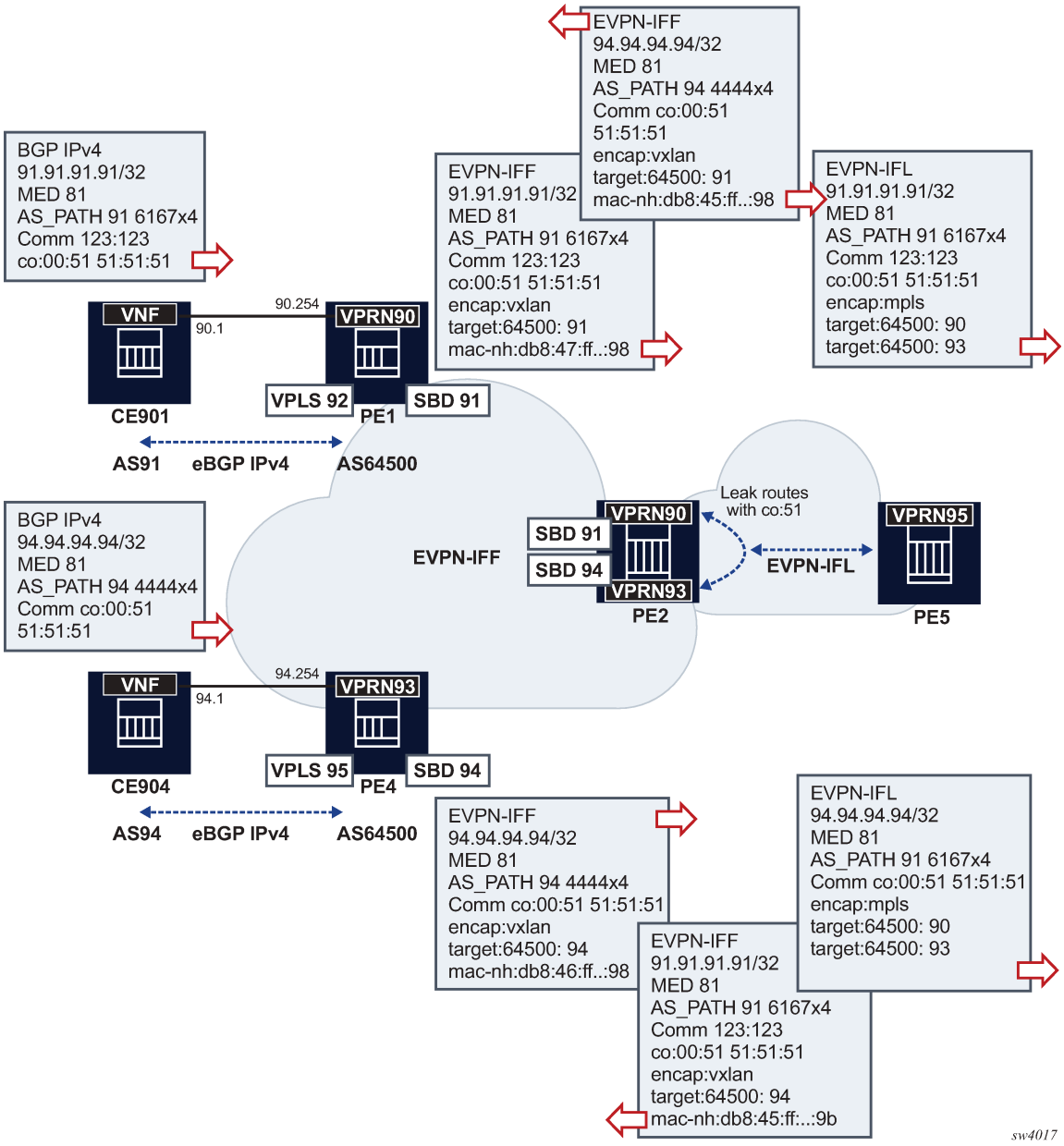
```
                exit
            exit
            bgp
                min-route-advertisement 1
                group "pe-ce"
                    family ipv4
                    type external
                    export "export-al-to-vnf"
                    neighbor 10.0.0.1
                        local-as 64500
                        peer-as 81
                    exit
                exit
                no shutdown
            exit
            no shutdown
// PE2's VPRN 80
A:PE-2# configure service vprn 80
A:PE-2>config>service>vprn# info
----------------------------------------------
            interface "lo0" create
                address 2.2.2.2/32
                loopback
            exit
            bgp-ipvpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    route-distinguisher 192.0.2.2:80
                    vrf-target target:64500:80
                    no shutdown
                exit
            exit
            bgp-evpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    route-distinguisher 192.0.2.2:80
                    vrf-target target:64500:80
                    no shutdown
                exit
            exit
            no shutdown
----------------------------------------------
// PE4's VPRN 80
A:PE-4# configure service vprn 80
A:PE-4>config>service>vprn# info
----------------------------------------------
            router-id 192.0.2.4
            autonomous-system 64500
            interface "lo0" create
                address 4.4.4.4/32
                loopback
            exit
            interface "local" create
                address 40.0.0.254/24
                sap 1/1/c1/1:80 create
                exit
            exit
            bgp-evpn
                mpls
                    auto-bind-tunnel
```

```
                    resolution any
                exit
                route-distinguisher 192.0.2.4:80
                vrf-target target:64500:80
                no shutdown
            exit
        exit
        bgp
            min-route-advertisement 1
            group "pe-ce"
                family ipv4
                type external
                export "export-bl-to-pe"
                neighbor 40.0.0.1
                    local-as 64500
                    peer-as 84
                exit
            exit
            no shutdown
        exit
        no shutdown
```

6.5.23.6.2 Example 2 - propagation of BGP path attributes for EVPN-IFF

In this configuration example, the DCGW PE2 re-exports EVPN-IFF routes into EVPN-IFF (leaked) routes and EVPN-IFL routes. The BGP path attributes are propagated as shown in Figure 243: Propagation of BGP path attributes for EVPN-IFF. As described in BGP path attribute propagation, EVPN extended communities, BGP encapsulation extended community and route targets are not propagated but instead, re-originated.

Figure 243: Propagation of BGP path attributes for EVPN-IFF



The following is an example configuration for PE4 and PE2 (PE1 has equivalent configuration as PE4).

```
// PE4 services for EVPN-IFF
A:PE-4>config>service>vprn# /configure service vprn 93
A:PE-4>config>service>vprn# info
----------------------------------------------
           router-id 4.4.4.4
           autonomous-system 64500
           interface "evi-95" create
               address 94.0.0.254/24
               vrrp 1 owner passive
                   backup 94.0.0.254
```

```
                exit
                vpls "evi-95"
                exit
            exit
            interface "evi-94" create
                vpls "evi-94"
                    evpn-tunnel
                exit
            exit
            bgp
                min-route-advertisement 1
                group "pe-ce"
                    family ipv4
                    type external
                    export "export-al-to-vnf"
                    neighbor 94.0.0.1
                        local-as 64500
                        peer-as 94
                    exit
                exit
                no shutdown
            exit
            no shutdown
-----------------------------------------------
A:PE-4>config>service>vprn# /configure service vpls 95
A:PE-4>config>service>vpls# info
-----------------------------------------------
            allow-ip-int-bind
            exit
            stp
                shutdown
            exit
            sap 1/1/c1/1:90 create
                no shutdown
            exit
            no shutdown
-----------------------------------------------
A:PE-4>config>service>vpls# /configure service vpls 94
A:PE-4>config>service>vpls# info
-----------------------------------------------
            allow-ip-int-bind
            exit
            vxlan instance 1 vni 94 create
            exit
            bgp
            exit
            bgp-evpn
                no mac-advertisement
                ip-route-advertisement
                evi 94
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
-----------------------------------------------
// PE2 config
A:PE-2# configure service vprn 90
A:PE-2>config>service>vprn# info
-----------------------------------------------
            interface "evi-91" create
```

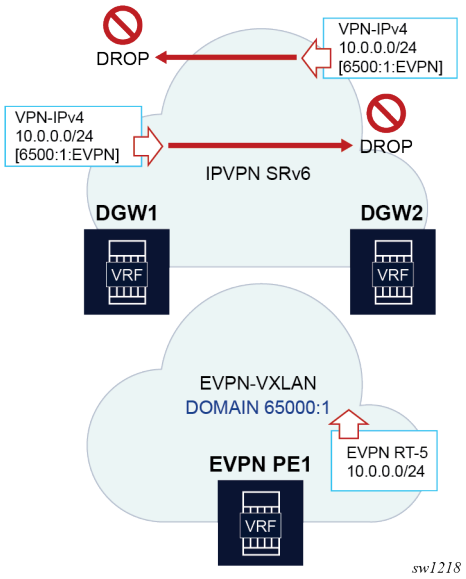
```
                vpls "evi-91"
                    evpn-tunnel
                exit
            exit
            bgp-evpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    route-distinguisher 192.0.2.2:90
                    vrf-export "leak-color-51-into-93"
                    vrf-target import target:64500:90
                    no shutdown
                exit
            exit
            no shutdown
-----------------------------------------------
A:PE-2>config>service>vprn# /configure service vpls 91
A:PE-2>config>service>vpls# info
-----------------------------------------------
            allow-ip-int-bind
            exit
            vxlan instance 1 vni 91 create
            exit
            bgp
            exit
            bgp-evpn
                no mac-advertisement
                ip-route-advertisement
                evi 91
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
-----------------------------------------------
A:PE-2>config>service>vpls# /configure service vprn 93
A:PE-2>config>service>vprn# info
-----------------------------------------------
            interface "evi-94" create
                vpls "evi-94"
                    evpn-tunnel
                exit
            exit
            bgp-evpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    route-distinguisher 192.0.2.2:93
                    vrf-export "leak-color-51-into-90"
                    vrf-target import target:64500:93
                    no shutdown
                exit
            exit
            no shutdown
-----------------------------------------------
A:PE-2>config>service>vprn# /configure service vpls 94
A:PE-2>config>service>vpls# info
-----------------------------------------------
            allow-ip-int-bind
```

```
            exit
            vxlan instance 1 vni 94 create
            exit
            bgp
            exit
            bgp-evpn
                no mac-advertisement
                ip-route-advertisement
                evi 94
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
-----------------------------------------------
A:PE-2>config>service>vpls# /show router policy "leak-color-51-into-90"
    entry 10
        from
            community "color-51"
        exit
        action accept
            community add "RT64500:90" "RT64500:93"
        exit
    exit
    default-action accept
        community add "RT64500:93"
    exit
A:PE-2>config>service>vpls# /show router policy "leak-color-51-into-93"
    entry 10
        from
            community "color-51"
        exit
        action accept
            community add "RT64500:90" "RT64500:93"
        exit
    exit
    default-action accept
        community add "RT64500:90"
    exit
```

6.5.23.6.3 Example 3 - D-PATH configuration

The example in the following figure shows a typical Layer 3 EVPN DC gateway scenario where EVPN-IFF routes are translated into IPVPN routes, and vice versa. Because redundant gateways are used, this scenario is subject to Layer 3 routing loops, and the D-PATH attribute helps preventing these loops in an automatic way, without the need for extra routing policies to tag or drop routes.

Figure 244: Use of D-PATH for Layer 3 DC gateway redundancy



The following is the configuration of the VPRN or R-VPLS services in DGW1 and DGW2 in the preceding figure.

```
A:DGW1# configure service vprn 20
A:DGW1>config>service>vprn# info
----------------------------------------------
            interface "sbd-1" create
                vpls “sbd-1”
                    evpn-tunnel
                exit
            exit
            segment-routing-v6 1 create
                locator "LOC-1"
                    function
                        end-dt46
                    exit
                exit
            exit
            bgp-ipvpn
                segment-routing-v6
                    route-distinguisher 192.0.2.1:20
                    srv6-instance 1 default-locator "LOC-1"
                    source-address 2001:db8::1
                    vrf-target target:64500:20
                    domain-id 65000:2
                    no shutdown
                exit
            exit
            no shutdown
*A:DGW1# configure service vpls "sbd-1"
*A:DGW1>config>service>vpls# info
----------------------------------------------
            allow-ip-int-bind
            exit
            vxlan instance 1 vni 1 create
            exit
            bgp
```

```
            exit
            bgp-evpn
                evi 1
                ip-route-advertisement domain-id 65000:1
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit

A:DGW2# configure service vprn 20
A:DGW2>config>service>vprn# info
----------------------------------------------
            interface "sbd-1" create
                vpls “sbd-1”
                    evpn-tunnel
                exit
            exit
            segment-routing-v6 1 create
                locator "LOC-1"
                    function
                        end-dt46
                    exit
                exit
            exit
            bgp-ipvpn
                segment-routing-v6
                    route-distinguisher 192.0.2.2:20
                    srv6-instance 1 default-locator "LOC-1"
                    source-address 2001:db8::2
                    vrf-target target:64500:20
                    domain-id 65000:2
                    no shutdown
                exit
            exit
            no shutdown
*A:DGW2# configure service vpls "sbd-1"
*A:DGW2>config>service>vpls# info
----------------------------------------------
            allow-ip-int-bind
            exit
            vxlan instance 1 vni 1 create
            exit
            bgp
            exit
            bgp-evpn
                evi 1
                ip-route-advertisement domain-id 65000:1
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
```

The following considerations apply to the example configuration shown in Figure 244: Use of D-PATH for Layer 3 DC gateway redundancy.

- Imported VPN-IP SRv6 routes are readvertised as EVPN-IFF VXLAN routes with a prepended D-PATH domain 65000:2:128.

- Imported EVPN-IFF VXLAN routes are readvertised as VPN-IP SRv6 routes with a prepended D-PATH domain 65000:1:70.

If PE1 sends an EVPN-IFF route 10.0.0.0/24 that is imported by both DGW1 and DGW2, then, when DGW1 and DGW2 receive each other's routes, they identify the D-PATH attribute and compare the list of domains with the locally configured domains in the VPRN. Since the domain matches one of the local domains, the route is not installed in the VPRN route table and it is flagged as a looped route (the **show router bgp routes detail** or **hunt** commands show **DPath Loop VRFs: 20**). In this way loops are prevented.

6.5.24 Routing policies for BGP EVPN routes

Routing policies match on specific fields when EVPN routes are imported or exported. These matching fields (excluding route table evpn ip-prefix routes, unless explicitly mentioned), are:

- communities, extended-communities, and large-communities
- well-known communities (**no-export** | **no-export-subconfed** | **no-advertise**)
- family EVPN
- protocol BGP-VPN (this term also matches VPN-IPv4 and VPN-IPv6 routes)
- prefix lists for routes type 2 when they contain an IP address, and for type 5
- route tags that can be passed by EVPN to BGP from:
  - **service>epipe/vpls>bgp-evpn>mpls/vxlan>default-route-tag** (this route-tag can be matched on export only)
  - **service>vpls>proxy-arp/nd>evpn-route-tag** (this route tag can be matched on export only)
  - route table route-tags when exporting EVPN IP-prefix routes
- EVPN type
- BGP attributes that are applicable to EVPN routes (such as AS-path, local-preference, next-hop)

Additionally, the route tags can be used on export policies to match EVPN routes that belong to a service and BGP instance, routes that are created by the proxy-arp or proxy-nd application, or IP-Prefix routes that are added to the route table with a route tag.

EVPN can pass only one route tag to BGP to achieve matching on export policies. In case of a conflict, the **default-route-tag** has the least priority of the three potential tags added by EVPN.

For instance, if VPLS 10 is configured with **proxy-arp>evpn-route-tag 20** and **bgp-evpn>mpls>default-route-tag 10**, all MAC/IP routes, which are generated by the proxy-arp application, uses route tag 20. Export policies can then use "from tag 20" to match all those routes. In this case, inclusive Multicast routes are matched by using "from tag 10".

6.5.24.1 Routing policies for BGP EVPN IP prefixes

BGP routing policies are supported for IP prefixes imported or exported through BGP-EVPN in R-VPLS services (EVPN-IFF routes) or VPRN services (EVPN-IFL routes).

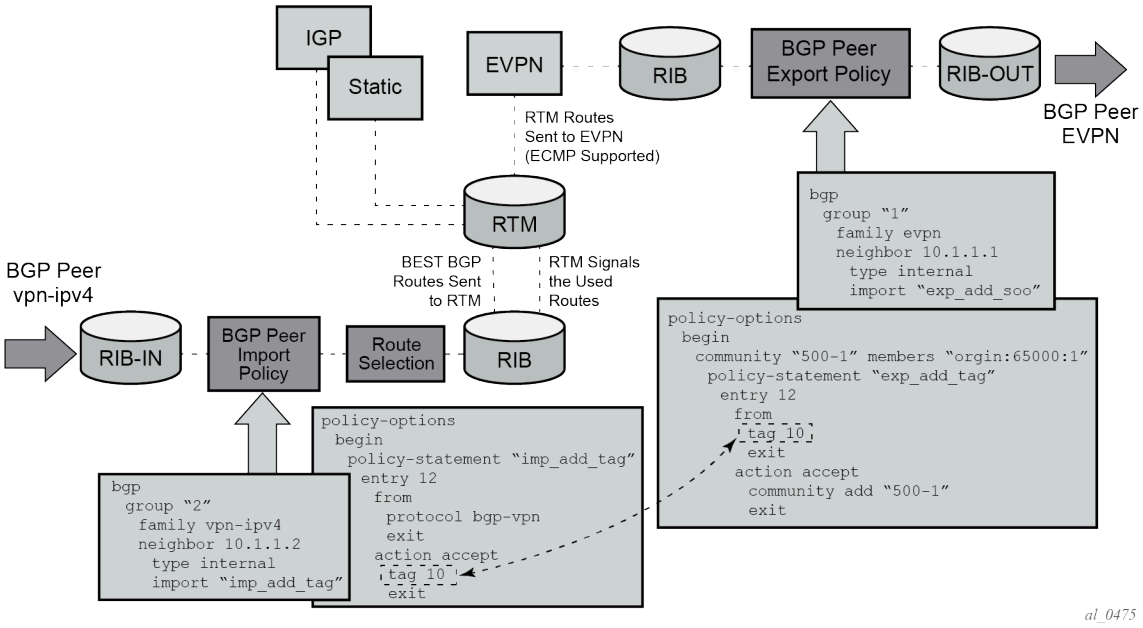
When applying routing policies to control the distribution of prefixes between EVPN-IFF and IP-VPN (or EVPN-IFL), the user must consider that these owners are completely separate as far as BGP is concerned and when prefixes are imported in the VPRN routing table, the BGP attributes are lost to the other owner, unless the **iff-attribute-uniform-propagation** command is configured on the router.

If the **iff-attribute-uniform-propagation** command is disabled, the use of route tags allows the controlled distribution of prefixes across the two families.

Figure 245: IP-VPN import and EVPN export BGP workflow shows an example of how VPN-IPv4 routes are imported into the RTM (Routing Table Manager) and then passed to EVPN for its own process.

Note: VPN-IPv4 routes can be tagged at ingress and that tag is preserved throughout the RTM and EVPN processing so that the tag can be matched at the egress BGP routing policy.

Figure 245: IP-VPN import and EVPN export BGP workflow

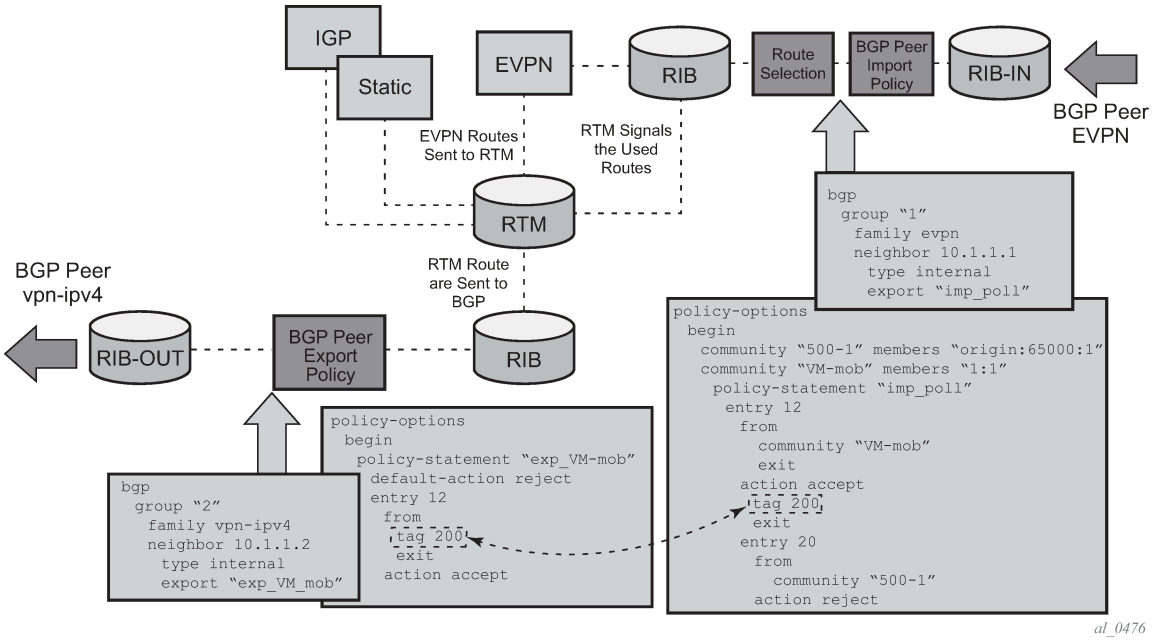


Policy tags can be used to match EVPN IP prefixes that were learned not only from BGP VPN-IPv4 but also from other routing protocols. The tag range supported for each protocol is different, as follows:

```
<tag>  : accepts in decimal or hex
        [0x1..0xFFFFFFFF]H (for OSPF and IS-IS)
        [0x1..0xFFFF]H (for RIP)
        [0x1..0xFF]H (for BGP)
```

Figure 246: EVPN import and I-VPN export BGP workflow shows an example of the reverse workflow where routes are imported from EVPN and exported from RTM to BGP VPN-IPv4.

Figure 246: EVPN import and I-VPN export BGP workflow



The preceding described behavior and the use of tags is also valid for vsi-import and vsi-export policies in the R-VPLS.

The following is a summary of the policy behavior for EVPN-IFF IP-prefixes when **iff-attribute-uniform-propagation** is disabled.

- For EVPN-IFF routes received and imported in RTM, policy entries (peer or vsi-import) match on communities or any of the following fields, and can add tags (as action):
  - communities, extended-communities or large communities
  - well-known communities
  - family EVPN
  - protocol bgp-vpn
  - prefix-lists
  - EVPN route type
  - BGP attributes (as-path, local-preference, next-hop)
- For exporting RTM to EVPN-IFF prefix routes, policy entries only match on tags, and based on this matching, add communities, accept, or reject. This applies to the peer level or on the VSI export level. Policy entries can also add tags for static routes, RIP, OSPF, IS-IS, BGP, and ARP-ND routes, which can then be matched on the BGP peer export policy, or on the VSI export policy for EVPN-IFF routes.

The following applies if the **iff-attribute-uniform-propagation** command is enabled.

For exporting RTM to EVPN-IFF prefix routes, in addition to matching on tags, matching path attributes on EVPN-IFF routes is supported in the following:

- vrf-export (when exporting the prefixes in VPN-IP or EVPN IFL or IP routes)
- vsi-export policies (when exporting the prefixes in EVPN-IFF routes)

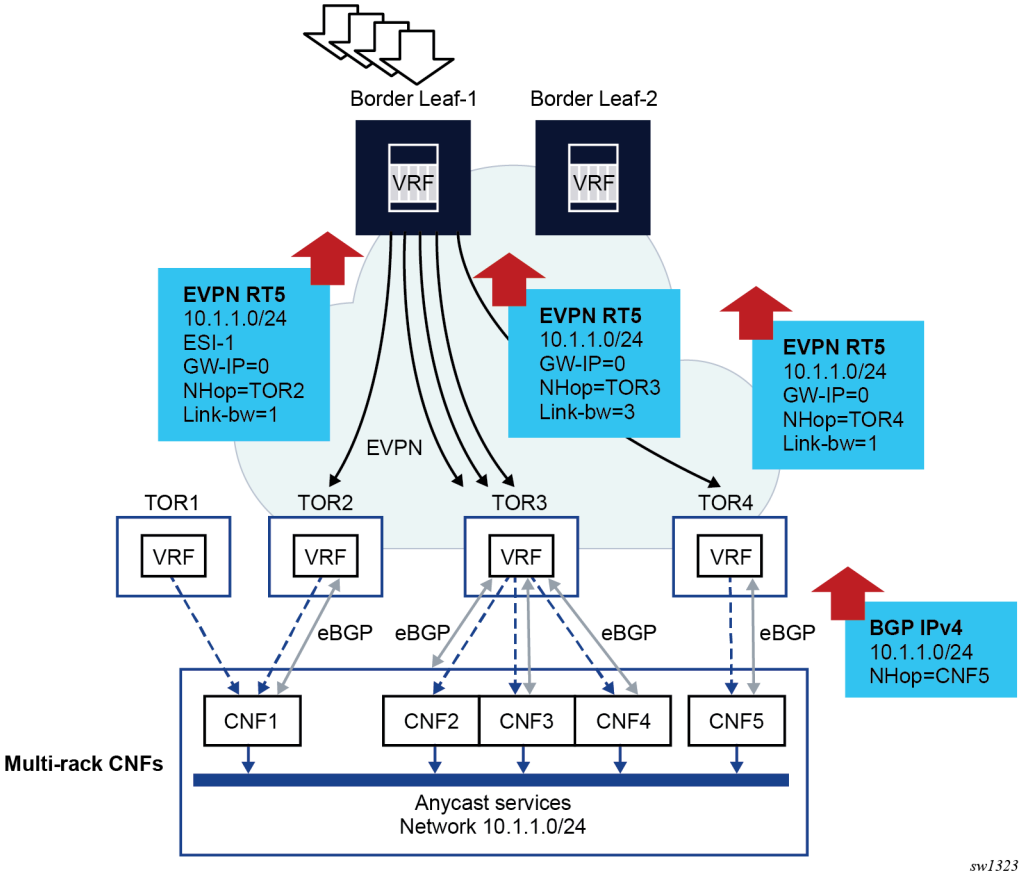
- for non-BGP route-owners (RIP, OSPF, IS-IS, static, ARP-ND), there are no changes and the only match criterion in vsi-export for EVPN-IFF routes is tags

6.5.25 EVPN Weighted ECMP for IP prefix routes

SR OS supports Weighted ECMP for EVPN IP Prefix routes (IPv4 and IPv6), in the EVPN Interface-less (EVPN-IFL) and EVPN Interface-ful (EVPN-IFF) models.

Based on *draft-ietf-bess-evpn-unequal-lb*, the EVPN Link Bandwidth extended community is used in the IP Prefix routes to indicate a weight that the receiver PE must consider when load balancing traffic to multiple EVPN, CE, or both next hops. The supported weight in the extended community is of type Generalized weight and encodes the count of CEs that advertised prefix N to a PE in a BGP PE-CE route. The following figure shows the use of EVPN Weighted ECMP.

Figure 247: Weighted ECMP for IP Prefix routes use case



In the preceding figure, some multi-rack Container Network Functions (CNFs) are connected to a few TORs in the EVPN network. Each CNF advertises the same anycast service network 10.1.1.0/24 using a single PE-CE BGP session. Without Weighted ECMP, the TOR2, TOR3 and TO4 would re-advertise the prefix in an EVPN IP-Prefix route and flows to 10.1.1.0/24 from the Border Leaf-1 would be equally distributed among TOR2, TOR3 and TOR4. However, the needed load balancing distribution is based on the count of CNFs that are attached to each TOR. That is, out of five flows to 10.1.1.0/24, three should be

directed to TOR3 (because it has three CNFs attached), one to TOR4 and one to either TOR2 or TOR1 (since CNF1 is dual-homed to both).

Weighted ECMP achieves the needed unequal load balancing based on the CNF count on each TOR. In the Figure 247: Weighted ECMP for IP Prefix routes use case example, if Weighted ECMP is enabled, the TORs add a weight encoded in the EVPN IP Prefix route, where the weight matches the count of CNFs that each TOR has locally . The Border Leaf creates an ECMP set for prefix 10.1.1.0/24 were the weights are considered when distributing the load to the prefix.

The procedures associated with EVPN Weighted ECMP for IP Prefix routes can be divided into advertising and receiving procedures:

- The advertising procedures are enabled by using the **config>service>vprn>bgp-evpn>mpls>evpn-link-bandwidth advertise** (EVPN-IFL) or the **config>service>vpls>bgp-evpn>ip-route-link-bandwidth advertise** (EVPN-IFF) commands. The **advertise** command triggers the advertisement of the EVPN Link Bandwidth extended community with a weight that matches the CE count advertised by the route. The dynamic weight can, optionally, be overridden by a configuring the **advertise weight** *weight* value.
- The receiving procedures are enabled by using the **config>service>vprn>bgp-evpn>mpls>evpn-link-bandwidth weighted-ecmp** (EVPN-IFL) or **config>service>vpls>bgp-evpn>ip-route-link-bandwidth weighted-ecmp** (EVPN-IFF) commands. When **weighted-ecmp** is enabled, the receiving PE installs IP Prefix routes in the VPRN route-table associated with a normalized weight that is derived from the signaled weight.
  - For EVPN-IFL, for weighted ECMP across EVPN next hops and CE next hops, the **config>service>vprn>bgp>group>evpn-link-bandwidth>add-to-received-bgp** *number* and the **config>service>vprn>bgp eibgp-loadbalance** commands must be configured.
  - For EVPN-IFF, Weighted ECMP can only be applied to EVPN next hops and not to **eibgp-loadbalance**.

Example: EVPN-IFL service configuration

Suppose PE2, PE4, and PE5 are attached to the same EVPN-IFL service on vprn 2000. PE4 is connected to two CEs (CE-41 and CE-42) and PE5 to one CE (CE-51). The three CEs advertise the same prefix 192.168.1.0/24 using PE-CE BGP and the goal is for PE2 to distribute to PE4 twice as many flows (to 192.168.1.0/24) as for PE5.

The configuration of PE4 and PE5 follows:

```
*A:PE-4# configure service vprn 2000
*A:PE-4>config>service>vprn# info
----------------------------------------------
            ecmp 10
            autonomous-system 64500
            interface "to-CE41" create
                address 10.41.0.1/24
                sap pxc-3.a:401 create
                exit
            exit
            interface "to-CE42" create
                address 10.42.0.1/24
                sap pxc-3.a:402 create
                exit
            exit
            bgp-evpn
                mpls
                    auto-bind-tunnel
                        resolution any
```

```
                    exit
                    evi 2000
                    evpn-link-bandwidth
                        advertise
                        weighted-ecmp
                    exit
                    route-distinguisher 192.0.2.4:2000
                    vrf-target target:64500:2000
                    no shutdown
                exit
            exit
            bgp
                multi-path
                    ipv4 10
                exit
                eibgp-loadbalance
                router-id 4.4.4.4
                rapid-withdrawal
                group "pe-ce"
                    family ipv4 ipv6
                    neighbor 10.41.0.2
                        peer-as 64541
                        evpn-link-bandwidth
                            add-to-received-bgp 1
                        exit
                    exit
                    neighbor 10.42.0.2
                        peer-as 64542
                        evpn-link-bandwidth
                            add-to-received-bgp 1
                        exit
                    exit
                exit
                no shutdown
            exit
            no shutdown


A:PE-5# configure service vprn 2000
A:PE-5>config>service>vprn# info
----------------------------------------------
            autonomous-system 64500
            interface "to-CE51" create
                address 10.51.0.1/24
                sap pxc-3.a:501 create
                exit
            exit
            bgp-evpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    evi 2000
                    evpn-link-bandwidth
                        advertise
                        weighted-ecmp
                    exit
                    route-distinguisher 192.0.2.5:2000
                    vrf-target target:64500:2000
                    no shutdown
                exit
            exit
            bgp
                multi-path
```

```
                    ipv4 10
                exit
                eibgp-loadbalance
                router-id 5.5.5.5
                rapid-withdrawal
                group "pe-ce"
                    family ipv4 ipv6
                    neighbor 10.51.0.2
                        peer-as 64551
                        evpn-link-bandwidth
                            add-to-received-bgp 1
                        exit
                    exit
                exit
                no shutdown
            exit
            no shutdown
```

The configuration on PE2 follows:

```
*A:PE-2# configure service vprn 2000
*A:PE-2>config>service>vprn# info
----------------------------------------------
            ecmp 10
            interface "to-PE" create
                address 20.10.0.1/24
                sap pxc-3.a:2000 create
                exit
            exit
            bgp-evpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    evi 2000
                    evpn-link-bandwidth
                        advertise
                        weighted-ecmp
                    exit
                    route-distinguisher 192.0.2.2:2000
                    vrf-target target:64500:2000
                    no shutdown
                exit
            exit
            no shutdown
```

Example: PE4 and PE5 IP Prefix route advertisement

As a result of the preceding configuration, PE4 (next-hop 2001:db8::4) and PE5 (next-hop 2001:db8::5) advertise the IP Prefix route from the CEs with weights 2 and 1 respectively:

```
*A:PE-2# show router bgp routes evpn ip-prefix prefix 192.168.1.0/24 community
 target:64500:2000 hunt
===============================================================================
 BGP Router ID:192.0.2.2        AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked, x - stale, > - best, b - backup, p - purge
 Origin codes  : i - IGP, e - EGP, ? - incomplete

===============================================================================
```

```
BGP EVPN IP-Prefix Routes
===============================================================================
-------------------------------------------------------------------------------
RIB In Entries
-------------------------------------------------------------------------------
Network        : n/a
Nexthop        : 2001:db8::4
Path Id        : None
From           : 2001:db8::4
Res. Nexthop   : fe80::b446:ffff:fe00:142
Local Pref.    : 100                    Interface Name : int-PE-2-PE-4
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : None
AIGP Metric    : None                   IGP Cost       : 10
Connector      : None
Community      : target:64500:2000 evpn-bandwidth:1:2
                 bgp-tunnel-encap:MPLS
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.0.2.4
Flags          : Used Valid Best IGP
Route Source   : Internal
AS-Path        : 64541
EVPN type      : IP-PREFIX
ESI            : ESI-0
Tag            : 0
Gateway Address: 00:00:00:00:00:00
Prefix         : 192.168.1.0/24
Route Dist.    : 192.0.2.4:2000
MPLS Label     : LABEL 524283
Route Tag      : 0
Neighbor-AS    : 64541
Orig Validation: N/A
Source Class   : 0                      Dest Class     : 0
Add Paths Send : Default
Last Modified  : 01h19m43s

Network        : n/a
Nexthop        : 2001:db8::5
Path Id        : None
From           : 2001:db8::5
Res. Nexthop   : fe80::b449:1ff:fe01:1f
Local Pref.    : 100                    Interface Name : int-PE-2-PE-5
Aggregator AS  : None                   Aggregator     : None
Atomic Aggr.   : Not Atomic             MED            : None
AIGP Metric    : None                   IGP Cost       : 10
Connector      : None
Community      : target:64500:2000 evpn-bandwidth:1:1
                 bgp-tunnel-encap:MPLS
Cluster        : No Cluster Members
Originator Id  : None                   Peer Router Id : 192.0.2.5
Flags          : Used Valid Best IGP
Route Source   : Internal
AS-Path        : 64551
EVPN type      : IP-PREFIX
ESI            : ESI-0
Tag            : 0
Gateway Address: 00:00:00:00:00:00
Prefix         : 192.168.1.0/24
Route Dist.    : 192.0.2.5:2000
MPLS Label     : LABEL 524285
Route Tag      : 0
Neighbor-AS    : 64551
Orig Validation: N/A
Source Class   : 0                      Dest Class     : 0
```

```
Add Paths Send : Default
Last Modified  : 00h08m45s

-------------------------------------------------------------------------------
RIB Out Entries
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
Routes : 2
===============================================================================
```

Example: PE2 prefix installation

The **show router** *id* **route-table extensive** command performed on PE2, shows that PE2 installs the prefix with weights 2 and 1 respectively for PE4 and PE5:

```
*A:PE-2# show router 2000 route-table 192.168.1.0/24 extensive

===============================================================================
Route Table (Service: 2000)
===============================================================================
Dest Prefix             : 192.168.1.0/24
  Protocol              : EVPN-IFL
  Age                   : 01h22m47s
  Preference            : 170
  Indirect Next-Hop     : 2001:db8::4
    Label               : 524283
    QoS                 : Priority=n/c, FC=n/c
    Source-Class        : 0
    Dest-Class          : 0
    ECMP-Weight         : 2
    Resolving Next-Hop  : 2001:db8::4 (LDP tunnel)
      Metric            : 10
      ECMP-Weight       : N/A
  Indirect Next-Hop     : 2001:db8::5
    Label               : 524285
    QoS                 : Priority=n/c, FC=n/c
    Source-Class        : 0
    Dest-Class          : 0
    ECMP-Weight         : 1
    Resolving Next-Hop  : 2001:db8::5 (LDP tunnel)
      Metric            : 10
      ECMP-Weight       : N/A
-------------------------------------------------------------------------------
No. of Destinations: 1
===============================================================================

*A:PE-2# show router 2000 fib 1 192.168.1.0/24 extensive


===============================================================================
FIB Display (Service: 2000)
===============================================================================
Dest Prefix             : 192.168.1.0/24
  Protocol              : EVPN-IFL
  Installed             : Y
  Indirect Next-Hop     : 2001:db8::4
    Label               : 524283
    QoS                 : Priority=n/c, FC=n/c
    Source-Class        : 0
    Dest-Class          : 0
    ECMP-Weight         : 2
    Resolving Next-Hop  : 2001:db8::4 (LDP tunnel)
      ECMP-Weight       : 1
```

```
  Indirect Next-Hop     : 2001:db8::5
    Label               : 524285
    QoS                 : Priority=n/c, FC=n/c
    Source-Class        : 0
    Dest-Class          : 0
    ECMP-Weight         : 1
    Resolving Next-Hop  : 2001:db8::5 (LDP tunnel)
      ECMP-Weight       : 1
===============================================================================
Total Entries : 1
===============================================================================
```

Example: EVPN-IFL handling

In case of EVPN-IFL, Weighted ECMP is also supported for EIBGP load balancing among EVPN and CE next hops. For example, PE4 installs the same prefix with an EVPN-IFL next hop and two CE next hops, and each one with its normalized weight:

```
*A:PE-4# /show router 2000 route-table 192.168.1.0/24 extensive

===============================================================================
Route Table (Service: 2000)
===============================================================================
Dest Prefix             : 192.168.1.0/24
  Protocol              : BGP
  Age                   : 00h02m27s
  Preference            : 170
  Indirect Next-Hop     : 10.41.0.2
    QoS                 : Priority=n/c, FC=n/c
    Source-Class        : 0
    Dest-Class          : 0
    ECMP-Weight         : 1
    Resolving Next-Hop  : 10.41.0.2
      Interface         : to-CE41
      Metric            : 0
      ECMP-Weight       : N/A
  Indirect Next-Hop     : 10.42.0.2
    QoS                 : Priority=n/c, FC=n/c
    Source-Class        : 0
    Dest-Class          : 0
    ECMP-Weight         : 1
    Resolving Next-Hop  : 10.42.0.2
      Interface         : to-CE42
      Metric            : 0
      ECMP-Weight       : N/A
  Indirect Next-Hop     : 2001:db8::5
    Label               : 524285
    QoS                 : Priority=n/c, FC=n/c
    Source-Class        : 0
    Dest-Class          : 0
    ECMP-Weight         : 1
    Resolving Next-Hop  : 2001:db8::5 (LDP tunnel)
      Metric            : 10
      ECMP-Weight       : N/A
-------------------------------------------------------------------------------
No. of Destinations: 1
===============================================================================
```

6.5.26 EVPN IP aliasing for IP prefix routes

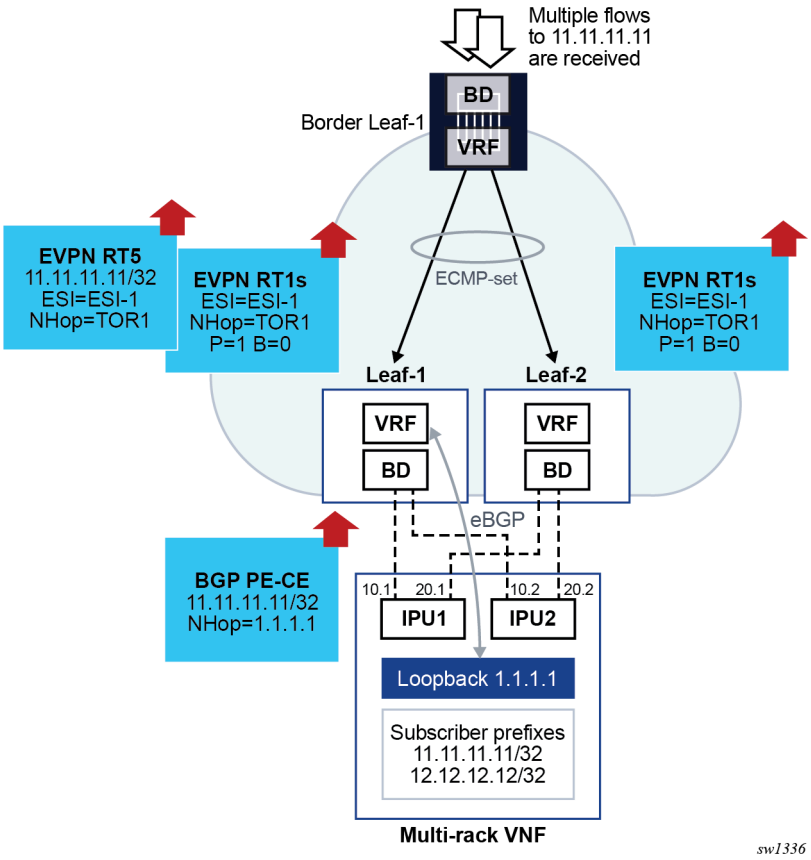
SR OS supports IP aliasing for EVPN IP prefix routes in the EVPN IFL (Interface-less) or EVPN IFF (Interface-ful) models and as described in *draft-sajassi-bess-evpn-ip-aliasing*.

IP aliasing allows PEs to load-balance flows to multiple PEs attached to the same prefix, even if not all of them advertise reachability to the prefix in IP prefix routes. IP aliasing works based on the following principles:

- It requires the configuration of a virtual Ethernet Segment (ES), for example, ES-1, that is associated with a **vprn-next-hop** and an **evi** configured in the **vprn** context. All PEs with reachability to the **vprn-next-hop**, via the non-EVPN route, advertise their attachment to the ES using EVPN Auto-Discovery per ES and per EVI routes in the VPRN service context.
- Any PE that receives a BGP PE-CE route for a prefix P via next-hop N, where N matches the active **vprn-next-hop**, advertises an IP prefix route for P with the ESI of the ES; for example, ESI-1.
- On reception, PEs importing IP prefix routes with ESI-1 install the prefix P in the route table using the next hops of the AD per-EVI routes for ESI-1, instead of the next hop of the IP prefix route.

Figure 248: EVPN IP aliasing in an EVPN-IFL model is an example of the use of IP aliasing in an EVPN-IFL model.

Figure 248: EVPN IP aliasing in an EVPN-IFL model



In the Figure 248: EVPN IP aliasing in an EVPN-IFL model example shown in the preceding figure, a multi-rack Virtual Network Function (VNF) is attached to Leaf-1 and Leaf-2. Although the VNF supports a single

PE-CE eBGP session to Leaf-1, the preferred behavior is for the Border-Leaf-1 to load balance the traffic toward the VNF using both Leaf-1 and Leaf-2 as next hops. EVPN IP aliasing achieves that preferred behavior based on the following configuration.

An ES L3-ES-1 is configured in Leaf-1 and Leaf-2. The ES is configured for **all-active** mode and is associated with the **vprn-next-hop** of the VNF. The association with the **evi** of the VPRN where the next hop is installed is also required.

Example: Leaf-1 and Leaf-2 ES configuration (MD-CLI)

```
[ex:/configure service system bgp evpn]
A:admin@node2# info
    ethernet-segment "L3-ES-1" {
        admin-state enable
        type virtual
        esi 0x01010101010000000000
        multi-homing-mode all-active
        association {
            vprn-next-hop 1.1.1.1 {
                virtual-ranges {
                    evi 2500 { }
                }
            }
        }
    }
```

Example: Leaf-1 and Leaf-2 ES configuration (classic CLI)

```
config>service>system>bgp-evpn# info
----------------------------------------------
ethernet-segment "L3-ES-1" virtual create
                   esi 01:01:01:01:01:00:00:00:00:00
                   service-carving
                       mode auto
                   exit
                   multi-homing all-active
                   vprn-next-hop 1.1.1.1
                   evi
                       evi-range 2500
                   exit
                   no shutdown
               exit
```

The VPRN service configuration in Leaf-1 and Leaf-2 requires the configuration of the **evi** so that the ES is active on the service.

Example: Leaf-1 VPRN configuration (MD-CLI)

```
[ex:/configure service vprn "2500"]
A:admin@node2# info
    admin-state enable
    customer "1"
    autonomous-system 64502
    bgp-evpn {
        mpls 1 {
            admin-state enable
            route-distinguisher "192.168.0.1:2500"
            evi 2500
            vrf-target {
                community "target:64500:2500"
            }
```

```
            auto-bind-tunnel {
                resolution any
            }
        }
    }
    bgp {
        min-route-advertisement 1
        router-id 2.2.2.2
        rapid-withdrawal true
        ebgp-default-reject-policy {
            import false
            export false
        }
        next-hop-resolution {
            use-bgp-routes true
        }
        group "pe-ce" {
            multihop 10
            family {
                ipv4 true
                ipv6 true
            }
        }
        neighbor "1.1.1.1" {
            group "pe-ce"
            peer-as 64501
        }
    }
    interface "irb1" {
        ipv4 {
            primary {
                address 10.10.10.254
                prefix-length 24
            }
            vrrp 1 {
                backup [10.10.10.254]
                owner true
                passive true
            }
        }
        vpls "BD2501" {
            evpn {
                arp {
                    learn-dynamic false
                    advertise dynamic {
                    }
                }
            }
        }
    }
    interface "lo1" {
        loopback true
        ipv4 {
            primary {
                address 2.2.2.2
                prefix-length 32
            }
        }
    }
    static-routes {
        route 1.1.1.1/32 route-type unicast {
            next-hop "10.10.10.1" {
                admin-state enable
            }
```

```
        }
    }
```

Example: Leaf-1 VPRN configuration (classic CLI)

```
config>service>vprn 2500 # info
----------------------------------------------
      autonomous-system 64502
            interface "irb1" create
                address 10.10.10.254/24
                vrrp 1 owner passive
                    backup 10.10.10.254
                exit
                vpls "BD2501"
                    evpn
                        arp
                            no learn-dynamic
                            advertise dynamic
                        exit
                    exit
                exit
            exit
            interface "lo1" create
                address 2.2.2.2/32
                loopback
            exit
            static-route-entry 1.1.1.1/32
                next-hop 10.10.10.1
                    no shutdown
                exit
            exit
            bgp-evpn
                mpls
                    auto-bind-tunnel
                        resolution any
                    exit
                    evi 2500
                    route-distinguisher 192.168.0.1:2500
                    vrf-target target:64500:2500
                    no shutdown
                exit
            exit
            bgp
                min-route-advertisement 1
                router-id 2.2.2.2
                rapid-withdrawal
                next-hop-resolution
                    use-bgp-routes
                exit
                group "pe-ce"
                    family ipv4 ipv6
                    multihop 10
                    neighbor 1.1.1.1
                        peer-as 64501
                    exit
                exit
                no shutdown
            exit
            no shutdown
```

Example: Leaf-2 VPRN configuration(MD-CLI)

```
[ex:/configure service vprn "2500"]
A:admin@node2# info
    admin-state enable
    customer "1"
    bgp-evpn {
        mpls 1 {
            admin-state enable
            route-distinguisher "192.168.0.2:2500"
            evi 2500
            vrf-target {
                community "target:64500:2500"
            }
            auto-bind-tunnel {
                resolution any
            }
        }
    }
    interface "irb1" {
        ipv4 {
            primary {
                address 10.10.10.254
                prefix-length 24
            }
            vrrp 1 {
                backup [10.10.10.254]
                owner true
                passive true
            }
        }
        vpls "BD2501" {
            evpn {
                arp {
                    learn-dynamic false
                    advertise dynamic {
                    }
                }
            }
        }
    }
    static-routes {
        route 1.1.1.1/32 route-type unicast {
            next-hop "10.10.10.1" {
                admin-state enable
            }
        }
    }
```

Example: Leaf-2 VPRN configuration (classic CLI)

```
config>service>vprn 2500 # info
----------------------------------------------
            interface "irb1" create
                address 10.10.10.254/24
                vrrp 1 owner passive
                    backup 10.10.10.254
                exit
                vpls "BD2501"
                    evpn
                        arp
                            no learn-dynamic
                            advertise dynamic
```

```
                    exit
                exit
            exit
        exit
        static-route-entry 1.1.1.1/32
            next-hop 10.10.10.1
                no shutdown
            exit
        exit

        bgp-evpn
            mpls
                auto-bind-tunnel
                    resolution any
                exit
                evi 2500
                route-distinguisher 192.168.0.2:2500
                vrf-target target:64500:2500
                no shutdown
            exit
        exit
        no shutdown
```

The Border-Leaf-1 configuration also needs the addition of the **evi** in the VPRN. This allows the creation of ECMP-sets where the next hops of the received IP prefixes are linked to the AD per-EVI routes next hops.

Example: Border-Leaf-1 VPRN configuration (MD-CLI)

```
[ex:/configure service vprn "2500"]
A:admin@node2# info
    admin-state enable
    customer "1"
    ecmp 4
    bgp-evpn {
        mpls 1 {
            admin-state enable
            route-distinguisher "192.168.0.3:2500"
            evi 2500
            vrf-target {
                community "target:64500:2500"
            }
            auto-bind-tunnel {
                resolution any
            }
        }
    }
```

Example: Border-Leaf-1 VPRN configuration (classic CLI)

```
config>service>vprn 2500 # info
----------------------------------------------
        ecmp 4
        bgp-evpn
            mpls
                auto-bind-tunnel
                    resolution any
                exit
                evi 2500
                route-distinguisher 192.168.0.3:2500
                vrf-target target:64500:2500
                no shutdown
```

```
            exit
        exit
        no shutdown
```

Based on the preceding configuration and the reachability of next-hop 1.1.1.1 via non-EVPN route, the two leaf nodes advertise their attachment to the ES via AD per-ES or EVI routes. Use the following command to display the advertisement status for ESI routes.

```
show router bgp routes evpn auto-disc esi 01:01:01:01:01:00:00:00:00:00
```

Output example: Advertisement of Auto-Discovery per-ES routes

```
===============================================================================
 BGP Router ID:192.0.2.3        AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked, x - stale, > - best, b - backup, p - purge
 Origin codes  : i - IGP, e - EGP, ? - incomplete

===============================================================================
BGP EVPN Auto-Disc Routes
===============================================================================
Flag  Route Dist.         ESI                           NextHop
      Tag                                               Label
-------------------------------------------------------------------------------
u*>i  192.168.0.1:2500    01:01:01:01:01:00:00:00:00:00 192.168.0.1
      0                                                 LABEL 524282

u*>i  192.168.0.1:2500    01:01:01:01:01:00:00:00:00:00 192.168.0.1
      MAX-ET                                            LABEL 0

u*>i  192.168.0.2:2500    01:01:01:01:01:00:00:00:00:00 192.168.0.2
      0                                                 LABEL 524283

u*>i  192.168.0.2:2500    01:01:01:01:01:00:00:00:00:00 192.168.0.2
      MAX-ET                                            LABEL 0

-------------------------------------------------------------------------------
Routes : 4
===============================================================================
```

At the same time, upon reception of the BGP PE-CE route from the VNF with prefix 11.11.11.11/32 (with next-hop 1.1.1.1, matching the **vprn-next-hop**), Leaf-1 readvertises the route in an IP prefix route with the ESI of the IP aliasing ES. Use the following command.

```
show router bgp routes evpn ip-prefix prefix 11.11.11.11/32
```

Output example: Leaf-1 readvertises the route in an IP Prefix route with the ESI of the IP Aliasing ES

```
===============================================================================
 BGP Router ID:192.0.2.3        AS:64500       Local AS:64500
===============================================================================
 Legend -
 Status codes  : u - used, s - suppressed, h - history, d - decayed, * - valid
                 l - leaked, x - stale, > - best, b - backup, p - purge
 Origin codes  : i - IGP, e - EGP, ? - incomplete
```

```
===============================================================================
BGP EVPN IP-Prefix Routes
===============================================================================
Flag  Route Dist.         Prefix
      Tag                 Gw Address
                          NextHop
                          Label
                          ESI
-------------------------------------------------------------------------------
u*>i  192.168.0.3:2500    11.11.11.11/32
      0                   00:00:00:00:00:00
                          192.168.0.1
                          LABEL 524279
                          01:01:01:01:01:00:00:00:00:00

-------------------------------------------------------------------------------
Routes : 1
===============================================================================
```

The IP prefix routes with non-zero ESI are also processed and recursively resolved on the PEs that are part of the ES. In the Figure 248: EVPN IP aliasing in an EVPN-IFL model example, Leaf-2 installs the prefix with the next hop associated with the ES instead of the next hop of the IP Prefix route; that is, the resolved next-hop is 1.1.1.1 instead of the IP prefix route next-hop 192.168.1.1. Use the following command to display the prefix with next hop association.

```
show router 2500 route-table 11.11.11.11/32 extensive
```

Output example: Prefix with next hop associated with ES

```
===============================================================================
Route Table (Service: 2500)
===============================================================================
Dest Prefix             : 11.11.11.11/32
  Protocol              : EVPN-IFL
  Age                   : 22h41m44s
  Preference            : 170
  Indirect Next-Hop     : 1.1.1.1
    QoS                 : Priority=n/c, FC=n/c
    Source-Class        : 0
    Dest-Class          : 0
    ECMP-Weight         : N/A
    Resolving Next-Hop  : 1.1.1.1
      Interface         : irb1
      Metric            : 0
      ECMP-Weight       : N/A
-------------------------------------------------------------------------------
No. of Destinations: 1
===============================================================================
```

Although the preceding example is based on an EVPN IFL model, **vprn-next-hop** ES can also be associated with VPRNs that use the EVPN IFF model to exchange IP Prefix routes. In an EVPN IFF model, the **vprn-next-hop** ES is associated with the VPRN if its R-VPLS connected to the EVPN tunnel contains the **evi** configured in the ES.

The following considerations about **vprn-next-hop** ES apply:

- The ES is operationally up as long as it is administratively enabled. The operational state does not reflect the presence of the VPRN next hop in the VPRN's route table.

- The AD per-ES or EVI routes for the ES are advertised as long as the VPRN next hop is installed in the route table (as a non-EVPN route). If the **vprn-next-hop** is installed in the VPRN's route table as an EVPN IP prefix route, the AD per-ES or EVI routes are not advertised.
- A node can generate an IP prefix route with the ESI of a **vprn-next-hop** as long as the node has the **vprn-next-hop** installed in its VPRN's route table, even as an EVPN IP prefix route.
- The AD per-ES or EVI routes are advertised with the RD and route target of the VPRN instance associated with the **evi** configured for the **vprn-next-hop**.
- ES routes are also advertised for the ES and are responsible for the DF Election in the ES in the case of **single-active** mode.
- All the non-DF PEs in the ES advertise their AD per-EVI route with bit P=0 and bit B=1, whereas the DF PE advertises its AD per-EVI with P=1 and B=0. When creating the ECMP-set for a prefix associated with an ESI, the remote PEs exclude those PEs for which their AD per-EVI routes indicate P=0.

6.6 Configuring an EVPN service with CLI

This section provides information to configure VPLS using the command line interface.

6.6.1 EVPN-VXLAN configuration examples

6.6.1.1 Layer 2 PE example

This section shows a configuration example for three PEs in a Data Center, all the following assumptions are considered:

- PE-1 is a Data Center Network Virtualization Edge device (NVE) where service VPLS 2000 is configured.
- PE-2 and PE-3 are redundant Data Center Gateways providing Layer 2 connectivity to the WAN for service VPLS 2000.

DC PE-1 configuration for service VPLS 2000

DC PE-2 and PE-3 configuration with SAPs at the WAN side (advertisement of all macs and unknown-mac-route):

```
vpls 2000 name "2000" customer 1 create
                vxlan instance 1 vni 2000 create
                exit
                bgp
                    route-distinguisher 65001:2000
                    route-target export target:65000:2000 import target:65000:2000
                exit
                bgp-evpn
                    unknown-mac-route
                    vxlan bgp 1 vxlan-instance 1
                        no shutdown
                    exit
                exit
                site "site-1" create
                    site-id 1
```

```
                    sap 1/1/1:1
                    no shutdown
                exit
                sap 1/1/1:1 create
                    no shutdown
                exit
                no shutdown
            exit
```

DC PE-2 and PE-3 configuration with BGP-AD spoke-SDPs at the WAN side (mac-advertisement disable, only unknown-mac-route advertised):

```
service vpls 2000 name "vpls2000" customer 1 create
    vxlan instance 1 vni 2000 create
    bgp
        pw-template-binding 1 split-horizon-group “to-WAN” import-
rt target:65000:2500
        vsi-export “export-policy-1” #policy exporting the WAN and DC RTs
        vsi-import “import-policy-1” #policy importing the WAN and DC RTs
        route-distinguisher 65001:2000
    bgp-ad
        no shutdown
        vpls-id 65000:2000
    bgp-evpn
        mac-advertisement disable
        unknown-mac-route
        vxlan bgp 1 vxlan-instance 1
            no shutdown
    site site-1 create
        split-horizon-group “to-WAN”
        no shutdown
        site-id 1
```

6.6.1.2 EVPN for VXLAN in R-VPLS services example

This section shows a configuration example for three 7750 SR, 7450 ESS, or 7950 XRS PEs in a Data Center, based on the following assumptions:

PE-1 is a Data Center Network Virtualization Edge device (NVE) where the following services are configured:

- R-VPLS 2001 and R-VPLS 2002 are subnets where Tenant Systems are connected
- VPRN 500 is a VPRN instance providing inter-subnet forwarding between the local subnets and from local subnets to the WAN subnets
- R-VPLS 501 is an IRB backhaul R-VPLS service that provides EVPN-VXLAN connectivity to the VPRNs in PE-2 and PE-3

```
*A:PE-1>config>service# info
        vprn 500 name "vprn500" customer 1 create
            ecmp 4
            route-distinguisher 65071:500
            vrf-target target:65000:500
            interface "evi-501" create
              address 10.30.30.1/24
                vpls "evpn-vxlan-501"
                exit
            exit
            interface "subnet-2001" create
```

```
                address 10.10.10.1/24
                vpls "r-vpls 2001"
                exit
            exit
            interface "subnet-2002" create
                address 10.20.20.1/24
                vpls "r-vpls 2002"
                exit
            exit
            no shutdown
        exit
        vpls 501 name “evpn-vxlan-501” customer 1 create
            allow-ip-int-bind
            vxlan instance 1 vni 501 create
            exit
            bgp
                route-distinguisher 65071:501
                route-target export target:65000:501 import target:65000:501
          exit
            bgp-evpn
                ip-route-advertisement incl-host
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
        exit
        vpls 2001 name “r-vpls 2001” customer 1 create
            allow-ip-int-bind
            sap 1/1/1:21 create
            exit
            sap 1/1/1:501 create
            exit
            no shutdown
        exit
        vpls 2002 name “r-vpls 2002” customer 1 create
            allow-ip-int-bind
            sap 1/1/1:22 create
            exit
            sap 1/1/1:502 create
            exit
            no shutdown
        exit
```

PE-2 and PE-3 are redundant Data Center Gateways providing Layer 3 connectivity to the WAN for subnets "subnet-2001” and "subnet-2002”. The following configuration excerpt shows an example for PE-2. PE-3 would have an equivalent configuration.

```
*A:PE-2>config>service# info
        vprn 500  name "vprn500" customer 1 create
            ecmp 4
            route-distinguisher 65072:500
            auto-bind-tunnel
              resolution-filter
                gre
                ldp
                rsvp
              exit
              resolution filter
            exit
```

```
            vrf-target target:65000:500
            interface "evi-501" create
              address 10.30.30.2/24
                vpls "evpn-vxlan-501"
                exit
            exit
            no shutdown
        exit
        vpls 501 name “evpn-vxlan-501” customer 1 create
            allow-ip-int-bind
            vxlan instance 1 vni 501 create
            exit
            bgp
                route-distinguisher 65072:501
                route-target export target:65000:501 import target:65000:501
            exit
            bgp-evpn
                ip-route-advertisement incl-host
                vxlan bgp 1 vxlan-instance 1
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
        exit
```

6.6.1.3 EVPN for VXLAN in EVPN tunnel R-VPLS services example

The example in EVPN for VXLAN in R-VPLS services example can be optimized by using EVPN tunnel R-VPLS services instead of regular IRB backhaul R-VPLS services. If EVPN tunnels are used, the corresponding R-VPLS services cannot contain SAPs or SDP-bindings and the VPRN interfaces do not need IP addresses.

The following excerpt shows the configuration in PE-1 for the VPRN 500. The R-VPLS 501, 2001 and 2002 can keep the same configuration as shown in the previous section.

```
*A:PE-1>config>service# info
        vprn 500 name "vprn500" customer 1 create
            ecmp 4
            route-distinguisher 65071:500
            vrf-target target:65000:500
            interface "evi-501" create
                vpls "evpn-vxlan-501"
                    evpn-tunnel# no need to configure an IP address
                exit
            exit
            interface "subnet-2001" create
                address 10.10.10.1/24
                vpls "r-vpls 2001"
                exit
            exit
            interface "subnet-2002" create
                address 20.20.20.1/24
                vpls "r-vpls 2002"
                exit
            exit
            no shutdown
        exit
```

The VPRN 500 configuration in PE-2 and PE-3 would be changed in the same way by adding the evpn-tunnel and removing the IP address of the EVPN-tunnel R-VPLS interface. No other changes are required.

```
*A:PE-2>config>service# info
        vprn 500 name "vprn500" customer 1 create
            ecmp 4
            route-distinguisher 65072:500
            auto-bind-tunnel
              resolution-filter
                gre
                ldp
                rsvp
              exit
              resolution filter
            exit
            vrf-target target:65000:500
            interface "evi-501" create
                vpls "evpn-vxlan-501"
                    evpn-tunnel# no need to configure an IP address
                exit
            exit
            no shutdown
        exit
```

6.6.1.4 EVPN for VXLAN in R-VPLS services with IPv6 interfaces and prefixes example

In the following configuration example, PE1 is connected to CE1 in VPRN 30 through a dual-stack IP interface. VPRN 30 is connected to an EVPN-tunnel R-VPLS interface enabled for IPv6.

In the following excerpt configuration the PE1 advertises, in BGP EVPN, the 172.16.0.0/24 and 2001:db8:1000::1 prefixes in two separate NLRIs. The NLRI for the IPv4 prefix uses gateway IP = 0 and a non-zero gateway MAC, whereas the NLRI for the IPv6 prefix is sent with gateway IP = Link-Local Address for interface "int-evi-301" and no gateway MAC.

```
*A:PE1>config>service# info
        vprn 30 name "vprn30" customer 1 create
            route-distinguisher 192.0.2.1:30
            vrf-target target:64500:30
            interface "int-PE-1-CE-1" create
                enable-ingress-stats
                address 172.16.0.254/24
                ipv6
                    address 2001:db8:1000::1/64
                exit
                sap 1/1/1:30 create
                exit
            exit
            interface "int-evi-301" create
                ipv6
                exit
                vpls "evi-301"
                    evpn-tunnel
                exit
            exit
            no shutdown
----------------------------------------------
```

6.6.2 EVPN-MPLS configuration examples

6.6.2.1 EVPN all-active multihoming example

This section shows a configuration example for three 7750 SR, 7450 ESS, or 7950 XRS PEs, all the following assumptions are considered:

- PE-1 and PE-2 are multihomed to CE-12 that uses a LAG to get connected to the network. CE-12 is connected to LAG SAPs configured in an all-active multihoming Ethernet segment.
- PE-3 is a remote PE that performs aliasing for traffic destined for the CE-12.

The following configuration excerpt applies to a VPLS-1 on PE-1 and PE-2, as well as the corresponding Ethernet-segment and LAG commands.

```
A:PE1# configure lag 1
A:PE1>config>lag# info
----------------------------------------------
        mode access
        encap-type dot1q
        port 1/1/2
        lacp active administrative-key 1 system-id 00:00:00:00:69:72
        no shutdown
----------------------------------------------
A:PE1>config>lag# /configure service system bgp-evpn
A:PE1>config>service>system>bgp-evpn# info
----------------------------------------------
                route-distinguisher 192.0.2.69:0
                ethernet-segment "ESI-71" create
                    esi 0x01000000007100000001
                    es-activation-timer 10
                    service-carving
                        mode auto
                    exit
                    multi-homing all-active
                    lag 1
                    no shutdown
                exit
----------------------------------------------
A:PE1>config>service>system>bgp-evpn# /configure service vpls 1
A:PE1>config>service>vpls# info
----------------------------------------------
            bgp
            exit
            bgp-evpn
                cfm-mac-advertisement
                evi 1
                vxlan
                    shutdown
                exit
                mpls bgp 1
                    ingress-replication-bum-label
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            stp
                shutdown
```

```
            exit
            sap lag-1:1 create

            exit
            no shutdown
-----------------------------------------------

A:PE2# configure lag 1
A:PE2>config>lag# info
-----------------------------------------------
        mode access
        encap-type dot1q
        port 1/1/3
        lacp active administrative-key 1 system-id 00:00:00:00:69:72
        no shutdown
-----------------------------------------------
A:PE2>config>lag# /configure service system bgp-evpn
A:PE2>config>service>system>bgp-evpn# info
-----------------------------------------------
                route-distinguisher 192.0.2.72:0
                ethernet-segment "ESI-71" create
                    esi 0x01000000007100000001
                    es-activation-timer 10
                    service-carving
                        mode auto
                    exit
                    multi-homing all-active
                    lag 1
                    no shutdown
                exit
-----------------------------------------------
A:PE2>config>service>system>bgp-evpn# /configure service vpls 1
A:PE2>config>service>vpls# info
-----------------------------------------------
            bgp
            exit
            bgp-evpn
                cfm-mac-advertisement
                evi 1
                vxlan
                    shutdown
                exit
                mpls bgp 1
                    ingress-replication-bum-label
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            sap lag-1:1 create
            exit
            no shutdown
-----------------------------------------------
```

The configuration on the remote PE (PE-3), which supports aliasing to PE-1 and PE-2 is shown below. PE-3 does not have any Ethernet-segment configured. It only requires the VPLS-1 configuration and ecmp>1 to perform aliasing.

```
*A:PE3>config>service>vpls# info
```

```
-----------------------------------------------
            bgp
            exit
            bgp-evpn
                cfm-mac-advertisement
                evi 1
                mpls bgp 1
                    ingress-replication-bum-label
                    ecmp 4
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            sap 1/1/1:1 create
            exit
            spoke-sdp 4:13 create
                no shutdown
            exit
            no shutdown
-----------------------------------------------
```

6.6.2.2 EVPN single-active multihoming example

If we wanted to use **single-active** multihoming on PE-1 and PE-2 instead of **all-active** multihoming, we would only need to modify the following:

- change the LAG configuration to **single-active**

  The CE-12 is now configured with two different LAGs, therefore, the key/system-id/system-priority must be different on PE-1 and PE-2

- change the Ethernet-segment configuration to **single-active**

No changes are needed at service level on any of the three PEs.

The differences between single-active versus all-active multihoming are highlighted in **bold** in the following example excerpts:

```
A:PE1# configure lag 1
A:PE1>config>lag# info
-----------------------------------------------
        mode access
        encap-type dot1q
        port 1/1/2
        lacp active administrative-key 1 system-id 00:00:00:00:69:69
        no shutdown
-----------------------------------------------
A:PE1>config>lag# /configure service system bgp-evpn
A:PE1>config>service>system>bgp-evpn# info
-----------------------------------------------
                route-distinguisher 192.0.2.69:0
                ethernet-segment "ESI-71" create
                    esi 0x01000000007100000001
                    es-activation-timer 10
                    service-carving
                        mode auto
                    exit
```

```
                    multi-homing single-active
                    lag 1
                    no shutdown
                exit
-----------------------------------------------

A:PE2# configure lag 1
A:PE2>config>lag# info
-----------------------------------------------
        mode access
        encap-type dot1q
        port 1/1/3
        lacp active administrative-key 1 system-id 00:00:00:00:72:72
        no shutdown
-----------------------------------------------
A:PE2>config>lag# /configure service system bgp-evpn
A:PE2>config>service>system>bgp-evpn# info
-----------------------------------------------
                route-distinguisher 192.0.2.72:0
                ethernet-segment "ESI-71" create
                    esi 0x01000000007100000001
                    es-activation-timer 10
                    service-carving
                        mode auto
                    exit
                    multi-homing single-active
                    lag 1
                    no shutdown
                exit
-----------------------------------------------
```

6.6.3 PBB-EVPN configuration examples

6.6.3.1 PBB-EVPN all-active multihoming example

As in the EVPN all-active multihoming example , this section also shows a configuration example for three 7750 SR, 7450 ESS, or 7950 XRS PEs, however, PBB-EVPN is used in this excerpt, as follows:

- PE-1 and PE-2 are multihomed to CE-12 that uses a LAG to get connected to I-VPLS 20001. CE-12 is connected to LAG SAPs configured in an **all-active** multihoming Ethernet-segment.
- PE-3 is a remote PE that performs aliasing for traffic destined for the CE-12.
- The three PEs are connected through B-VPLS 20000, a Backbone VPLS where EVPN is enabled.

The following excerpt shows the example configuration for I-VPLS 20001 and B-VPLS 20000 on PE-1 and PE-2, as well as the corresponding Ethernet-segment and LAG commands:

```
*A:PE1# configure lag 1
*A:PE1>config>lag# info
-----------------------------------------------
        mode access
        encap-type dot1q
        port 1/1/2
        lacp active administrative-key 1 system-id 00:00:00:00:69:72
        no shutdown
-----------------------------------------------
*A:PE1>config>lag# /configure service system bgp-evpn
```

```
*A:PE1>config>service>system>bgp-evpn# info
----------------------------------------------
                route-distinguisher 192.0.2.69:0
                ethernet-segment "ESI-71" create
                    esi 01:00:00:00:00:71:00:00:00:01
                    source-bmac-lsb 71-71 es-bmac-table-size 8
                    es-activation-timer 5
                    service-carving
                        mode auto
                    exit
                    multi-homing all-active
                    lag 1
                    no shutdown
                exit
----------------------------------------------
*A:PE1>config>service>system>bgp-evpn# /configure service vpls 20001
*A:PE1>config>service>vpls# info
----------------------------------------------
            pbb
                backbone-vpls 20000
                exit
            exit
            stp
                shutdown
            exit
            sap lag-1:71 create
            exit
            no shutdown
----------------------------------------------
*A:PE1>config>service>vpls# /configure service vpls 20000
*A:PE1>config>service>vpls# info
----------------------------------------------
            service-mtu 2000
            pbb
                source-bmac 00:00:00:00:00:69
                use-es-bmac
            exit
            bgp-evpn
                evi 20000
                mpls bgp 1
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
----------------------------------------------

*A:PE2# configure lag 1
*A:PE2>config>lag# info
----------------------------------------------
        mode access
        encap-type dot1q
        port 1/1/3
        lacp active administrative-key 1 system-id 00:00:00:00:69:72
        no shutdown
----------------------------------------------
*A:PE2>config>lag# /configure service system bgp-evpn
*A:PE2>config>service>system>bgp-evpn# info
----------------------------------------------
```

```
                route-distinguisher 192.0.2.72:0
                ethernet-segment "ESI-71" create
                    esi 01:00:00:00:00:71:00:00:00:01
                    source-bmac-lsb 71-71 es-bmac-table-size 8
                    es-activation-timer 5
                    service-carving
                        mode auto
                    exit
                    multi-homing all-active
                    lag 1
                    no shutdown
                exit
-----------------------------------------------
*A:PE2>config>service>system>bgp-evpn# /configure service vpls 20001
*A:PE2>config>service>vpls# info
-----------------------------------------------
            pbb
                backbone-vpls 20000
                exit
            exit
            stp
                shutdown
            exit
            sap lag-1:71 create
            exit
            no shutdown
-----------------------------------------------
*A:PE2>config>service>vpls# /configure service vpls 20000
*A:PE2>config>service>vpls# info
-----------------------------------------------
            service-mtu 2000
            pbb
                source-bmac 00:00:00:00:00:72
                use-es-bmac
            exit
            bgp-evpn
                evi 20000
                mpls bgp 1
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
-----------------------------------------------
*A:PE2>config>service>vpls#
```

The combination of the pbb **source-bmac** and the Ethernet-segment **source-bmac-lsb** create the same BMAC for all the packets sourced from both PE-1 and PE-2 for Ethernet-segment "ESI-71".

6.6.3.2 PBB-EVPN single-active multihoming example

In the following configuration example, PE-70 and PE-73 are part of the same single-active multihoming, Ethernet-segment ESI-7413. In this case, the CE is connected to PE-70 and PE-73 through spoke-SDPs 4:74 and 34:74, respectively.

In this example PE-70 and PE-73 use a different source-bmac for packets coming from ESI-7413 and it is not an **es-bmac** as shown in the PBB-EVPN all-active multihoming example .

```
*A:PE70# configure service system bgp-evpn
*A:PE70>config>service>system>bgp-evpn# info
----------------------------------------------
                route-distinguisher 192.0.2.70:0
                ethernet-segment "ESI-7413" create
                    esi 01:74:13:00:74:13:00:00:74:13
                    es-activation-timer 0
                    service-carving
                        mode auto
                    exit
                    multi-homing single-active
                    sdp 4
                    no shutdown
                exit
----------------------------------------------
*A:PE70>config>service>system>bgp-evpn# /configure service vpls 20001
*A:PE70>config>service>vpls# info
----------------------------------------------
            pbb
                backbone-vpls 20000
                exit
            exit
            stp
                shutdown
            exit
            spoke-sdp 4:74 create
                no shutdown
            exit
            no shutdown
----------------------------------------------
*A:PE70>config>service>vpls# /configure service vpls 20000
*A:PE70>config>service>vpls# info
----------------------------------------------
            service-mtu 2000
            pbb
                source-bmac 00:00:00:00:00:70
            exit
            bgp-evpn
                evi 20000
                mpls bgp 1
                    ecmp 2
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
----------------------------------------------
*A:PE70>config>service>vpls#


A:PE73>config>service>system>bgp-evpn# info
----------------------------------------------
                route-distinguisher 192.0.2.73:0
                ethernet-segment "ESI-7413" create
                    esi 01:74:13:00:74:13:00:00:74:13
```

```
                    es-activation-timer 0
                    service-carving
                        mode auto
                    exit
                    multi-homing single-active
                    sdp 34
                    no shutdown
                exit
-----------------------------------------------
A:PE73>config>service>system>bgp-evpn# /configure service vpls 20001
A:PE73>config>service>vpls# info
-----------------------------------------------
            pbb
                backbone-vpls 20000
                exit
            exit
            stp
                shutdown
            exit
            spoke-sdp 34:74 create
                no shutdown
            exit
            no shutdown
-----------------------------------------------
A:PE73>config>service>vpls# /configure service vpls 20000
A:PE73>config>service>vpls# info
-----------------------------------------------
            service-mtu 2000
            pbb
                source-bmac 00:00:00:00:00:73
            exit
            bgp-evpn
                evi 20000
                mpls bgp 1
                    auto-bind-tunnel
                        resolution any
                    exit
                    no shutdown
                exit
            exit
            stp
                shutdown
            exit
            no shutdown
-----------------------------------------------
A:PE73>config>service>vpls#
```

7 Pseudowire ports

This chapter provides information about pseudowire ports (PW ports), process overview, and implementation notes.

7.1 Overview

A PW port is primarily used to provide PW termination with the following characteristics:

- Provide access (SAP) based capabilities to a PW which has traditionally been a network port based concept within SR OS. For example, PW payload can be extracted onto a PW-port-based SAPs with granular queuing capabilities (queuing per SAP). This is in contrast with traditional PW termination on network ports where queuing is instantiated per physical port on egress or per MDA on ingress.
- Lookup dot1q and qinq VLAN tags underneath the PW labels and map the traffic to different services.
- Terminate subscriber traffic carried within the PW on a BNG. In this case PW-port-based SAPs are instantiated under a group interface with Enhanced Subscriber Management (ESM). In this case, a PW-port-based SAP is treated as any other regular SAP created directly on a physical port with full ESM capabilities.

The PW-port coverage expands beyond the TLDP-signaled pseudowires and encompass termination of all tunnel types (MPLS, GRE, VXLAN, L2oGRE, SRv6, and so on) and signaling methods (TLDP, BGP-EVPN, BGP-VPSW, and so on) that can be configured under an Epipe service.

Mapping between PWs and PW ports is performed on one-to-one basis.

There are two modes in which PW port can operate:

- **a PW port bound to a specific physical port (I/O port)**

  A successful mapping between the PW and PW port requires that the PW terminates on the same physical port (I/O port) to which the PW port is bound. In this mode of operation, PW ports do not support re-routing of PWs between the I/O ports. For example, if a PW is rerouted to an alternate physical port because of a network failure, the PW port becomes non-operational.
- **a PW port independent of the physical port (I/O port) on which the PW is terminated**

  This capability relies on FPE functionality, therefore, the name FPE based PW port. The benefit of such PW port is that it can provide services in cases where traffic within PW is rerouted between I/O ports because of a network failure.

When the PW port is created, the mapping between the PW port and PW depends on the mode of operation and application.

PW port creation:

```
configure
   pw-port <id>
       encap-type {dot1q|qinq}
```

Similar to any other Ethernet-based port, the PW port supports two encapsulation types, dot1q and qinq. Ether-type on a PW port is not configurable and it is set to a fixed value of 0x8100 for dot1q and qinq encapsulation.

7.2 PW port bound to a physical port

In this mode of operation, the PW port is bound to a specific physical port through an SDP binding context:

```
configure
    service
        sdp 1 mpls create
            far-end 10.10.10.10
            ldp
            binding
                port 1/1/1
                pw-port 1 vc-id 11 create
                    egress
                    shaping inter-dest-id vport-1
```

In this example, pw-port 1 is bound to a physical port 1/1/1.This PW port is mapped to the PW with vc-id 11 under the sdp 1 which must be terminated on port 1/1/1.A PW port is shaped by a virtual port scheduler (Vport) construct named vport-1 configured under port 1/1/1. SAPs created under such PW ports can be terminated in ESM, Layer 3 IES/VPRN interface or in an Epipe.

7.3 FPE-based PW port

The FPE based PW-port is primarily used to extract a PW payload onto an access based PW-port SAP, independent of the network I/O ports. FPE uses Port Cross-Connect (PXC) ports and provides an anchoring point for PW-port, independent of I/O ports, the term anchored PW-port can be interchangeably used with the term FPE based PW-port. The following are examples of applications which rely on FPE based PW-port:

- ESM over PW where MPLS/GRE based PW can be rerouted between I/O ports on an SR OS node without affecting ESM service
- Granular QoS per PW because the PW payload is terminated on an access based PW-port SAP → ingress/egress queues are created per SAP (as opposed to per network port on egress and per MDA on network ingress)
- PW-SAP with MPLS resiliency, where the LSP used by the PW terminated on a PW Ports is protected using MPLS mechanisms such as FRR and could therefore use any port on the system
- PW-port using LDP-over-RSVP tunnels
- A PW Port using a BGP VPWS

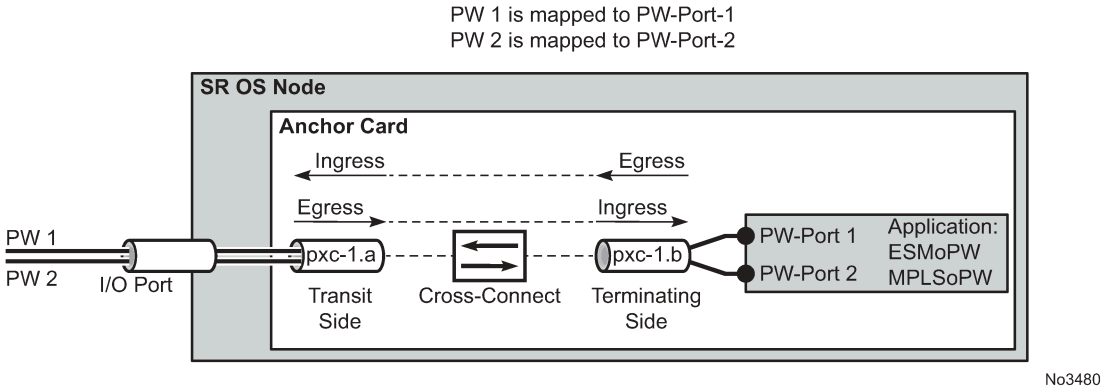
Although the primary role of FPE based PW-port is to terminate an external PW, in certain cases PW-port can be used to terminate traffic from regular SAP on I/O ports. This can be used to:

- Separate service termination point from the SAPs which are tied to I/O ports.
- Distribute load from a single I/O port to multiple line cards based on S-Tag (traffic from each S-tag can be mapped to a separate PW associated with different PXCs residing on different line cards).

7.3.1 Cross-connect between the external PW and the FPE-based PW-port

PW payload delivery from the I/O ports to the FPE based PW-port (and SAP) is facilitated via an internal cross-connect which is built on top of PXC sub-ports. Such cross-connect allows for mapping between PWs and PW-ports even in cases where PW payloads have overlapping VLANS. This concept is shown in Figure 249: Multiplexing PWs over PXC-based internal cross-connect .

Figure 249: Multiplexing PWs over PXC-based internal cross-connect



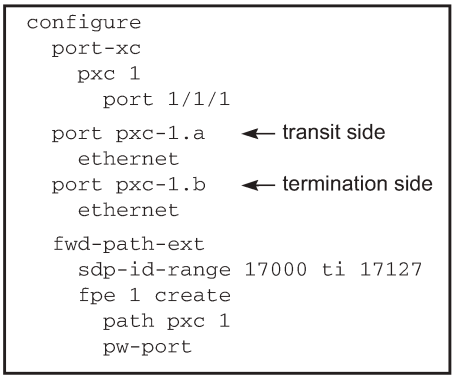
Parameters associated with the PXC sub-ports or PXC based LAGs (QoS, lag-profiles, and so on) are accessible/configurable through CLI. For example, the operator may apply an egress port-scheduler on sub-port pxc-1.b in Figure 249: Multiplexing PWs over PXC-based internal cross-connect to manage the sum of the bandwidth from associated PW-ports (PW-ports 1 and 2). To avoid confusion during configuration of PXC sub-ports /LAGs, a clear definition of reference points on the cross-connect created through FPE is required:

- Terminating side of the cross-connect is closer to PW-ports (.b side)
- Transit side of the cross-connect is closer to I/O ports (.a side)

Because the creation of the cross-connect on FPE based PW-ports is highly automated through and FPE configurations, the SR OS system:

- Assign PXC sub-ports .a to the transit side, and PXC sub-ports .b to the terminating side in case that a single PXC is used; see Figure 250: Assign PXC sub ports .

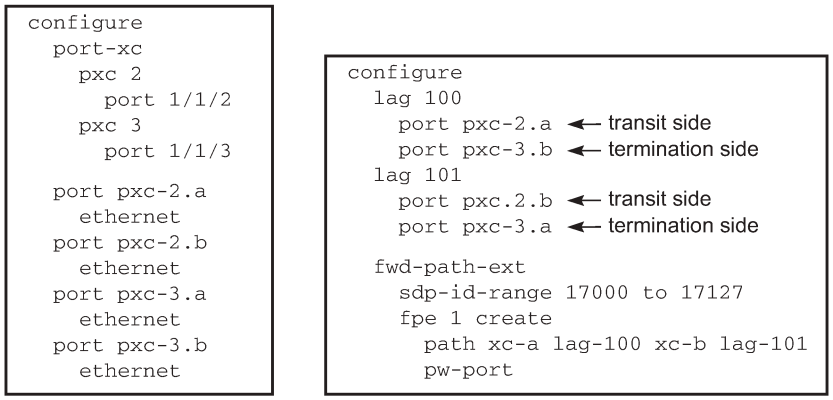
Figure 250: Assign PXC sub ports



No3481

- Assign the xc-a LAG to the transit side, and the xc-b LAG to the terminating side if that a PXC based LAG is used; see Figure 251: Assign the LAG .

  Figure 251: Assign the LAG



No3494

xc-a and xc-b can be associated with any PXC based LAG ID. For example, the following path configuration is allowed: xc-a with lag-id 100 (which includes pxc sub-ports pxc-2.a and pxc-3.b) and xc-b with lag-id 101 (which includes pxc sub-ports pxc-3.a and pxc-2.b). Regardless of the pxc sub-ports that are assigned to respective LAGs, the xc-a side of the path is used as the transit side of the cross-connect, while the xc-b side of the path is used as the termination side of the cross-connect.

7.3.2 PXC-based PW-port — building the cross-connect

From a logical perspective, the internal cross-connect that maps the external PW to a PW-port is implemented as a switched Epipe service (**vc-switching**). This switched Epipe service switches an external PW to the internal PW that is terminated on a FPE based PW-port. In this fashion, the PW-port becomes independent of the I/O ports. Assuming that PXC and PW-port are already configured in the system, the following are the three main configuration steps required to terminate the payload carried over external PW on the PW-port SAP:

1. Auto-setup of the internal transport tunnel over which the cross-connect is built
2. Auto-setup of the internal PW, switching the external PW to the internal PW and terminating the PW on the FPE based PW-port
3. Terminating the service on the PW-SAP

The status of the internally built constructs can be examined via various show commands (for example, **show service id** <**epipe-id**> **1 sdp**). The internal SDP ID is allocated from the user space. To avoid conflict between the user provisioned SDP IDs and the system provisioned SDP IDs, a range of SDP ids that is used for internal consumption must be reserved in advance. This is accomplished via the **sdp-id-range** commands under the **config>fwd-path-ext** hierarchy.

Configuration steps necessary to build PW-port based cross-connect over PXC are shown in the following diagrams (a single PXC is used in this example).

7.3.2.1 Building the internal transport tunnel

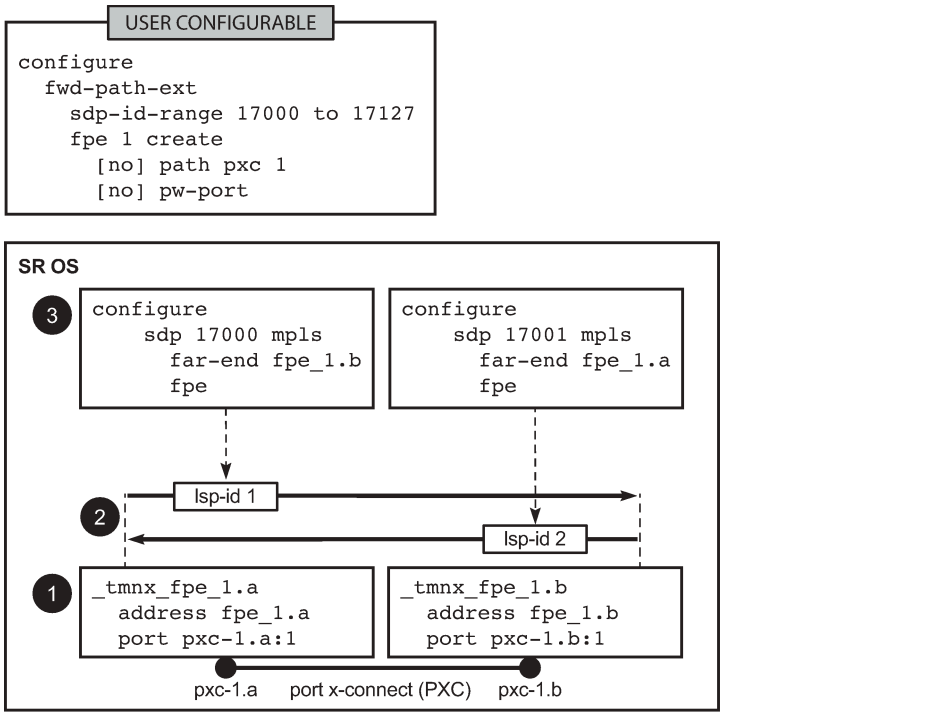
The **fpe** command instructs the SR OS system to build an LSP tunnel over the PXC. This tunnel is used to multiplex PW traffic to respective PW-ports. Each external PW is switched to an internal PW (on top of this tunnel) and its payload is off-loaded to a respective PW-port.

After the **fpe** is configured (see the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*, "*Forwarding Path Extensions*"), the SR OS system automatically configures steps 1, 2 and 3 in Figure 252: Building the internal LSP over PXCs. The objects created in steps 1, 2 and 3 can be seen via show commands. However, they are not visible to the operator in the configure branch of CLI.

The significance of the **pw-port** command under the FPE is to inform the system about the kind of cross-connect that needs to be built over PXC – in this case this cross-connect is PW-port specific. Applications other than PW-port may require different functionality over PXCs and this is reflected by a different command under the FPE CLI hierarchy (for example, **vxlan-termination** command instead of **pw-port**).

Note that the IP addresses setup on internal interfaces on PXC sub-ports are Martian IP addresses and they are shown in CLI as fpe_<id>.a and fpe_<id>.b.

Figure 252: Building the internal LSP over PXCs



No3483

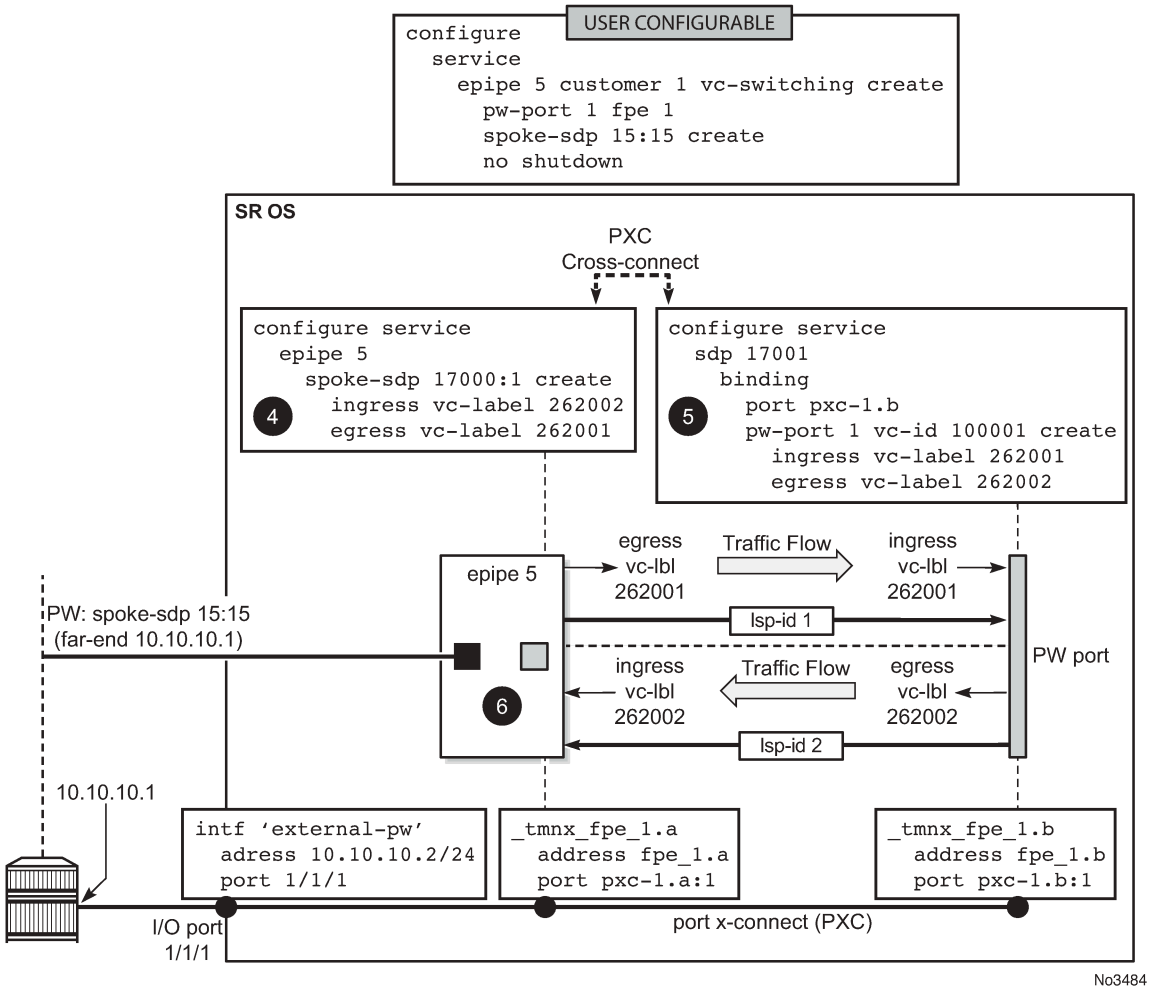
7.3.2.2 Mapping the external PW to the PW-port

Mapping between the external PW and the FPE based PW-port is performed via an Epipe of type vc-switching. The user configurable Epipe (id 5 in this example) aids in setting up steps 4, 5 and 6 in Figure 253: Mapping between the external PW and the PXC based PW-port:

1. An internal PW is automatically added to the user configured Epipe 5.
2. A bind is created between the internal PW and the PW-port attached to PXC.
3. External PW is switched to the internal PW.

At this stage, the external PW is mapped to the **pw-port 1**, as shown in Figure 253: Mapping between the external PW and the PXC based PW-port. The **spoke-sdp 17000:1** and the binding under SDP 17001 (**spoke-sdp 17001:100001**) created in steps 4 and 5 (Figure 253: Mapping between the external PW and the PXC based PW-port) can be seen via show commands. However, they are not visible to the operator in the configure branch of CLI.

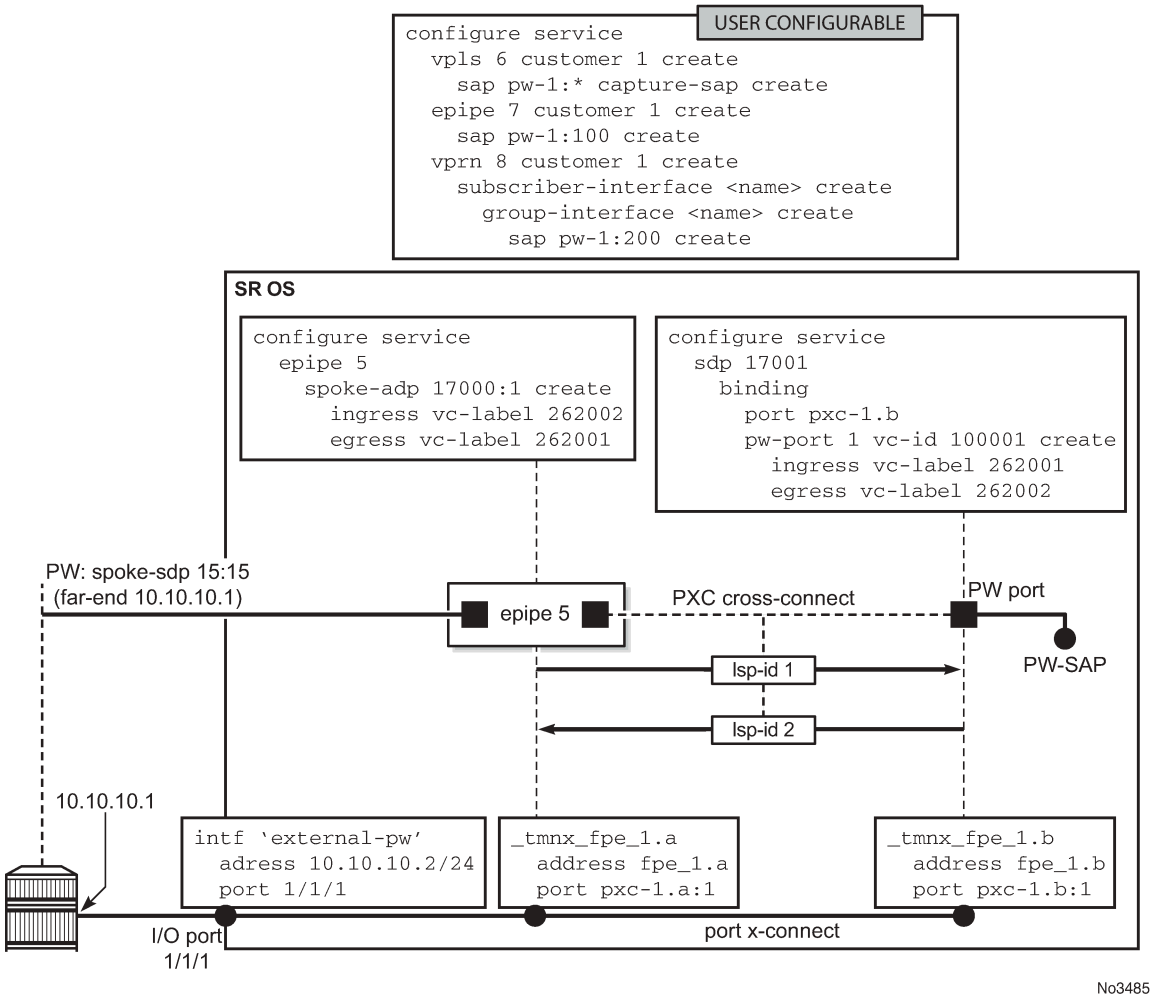
Figure 253: Mapping between the external PW and the PXC based PW-port



7.3.2.3 Terminating the service on PW-SAP

In the final step, PW-port SAP is applied to a service (Figure 254: Service termination on PW-SAP).

Figure 254: Service termination on PW-SAP



7.3.3 FPE-based PW port operational state

The ability of the stitching service to forward traffic drives the FPE-based PW port operational state. This includes the operational status of the stitching service and, if the external PW is TLDP signaled, the PW status bits. The stitching service operational status transitions to a non-operational state if the EVPN network destination, a BGP-VPWS spoke-SDP, or a configured spoke-SDP is operationally down.

With a TLDP-signaled PW, the operational state of the PW port depends on the PW status bits received from the peer, even if the stitching service is operationally up:

- The PW port transitions into a non-operational state if the PW Preferential Forwarding bit (pwFwdingStandby) is received from the peer (the PW is in standby mode).
- The PW port transitions into a non-operational state if the Local Attachment Circuit (LAC) or Local Packet Switched Network (PSN) faults are received from the peer (lacIngressFault, lacEgressFault, psnIngressFault, and psnEgressFault). The operator must explicitly enable this behavior through CLI. By default, the aforementioned fault bits received from the peer do not affect the state of the PW port.

The operational flag for a non-operational PW-port is set to **stitchingSvcTxDown**.

Transitioning of the PW port into the down state because of a PXC failure (for example physical port fails) brings the stitching service down with the following result:

- In case of a TLDP-signaled PW, the **psnIngressFault** and **psnEgressFault** PW status bits are propagated to the remote end, indicating that the local stitching service is down.
- In case of an EVPN, the EVPN route is withdrawn, indicating that the local stitching service is down.
- In case of a BGP-VPWS, the BGP-VPWS 'D' bit of the Layer 2 Information Extended Community flag field is set, indicating that the local stitching service is down.

By default, if a PW port is mated to a T-LDP-signaled PW (for example, a spoke-SDP) across an FPE in an Epipe service, the PW port operational state only reacts to the following PW status bit being set on the received T-LDP status message:

```
pwFwdingStandby (5) -- Pseudo Wire in Standby mode
```

When the **down-on-peer-tldp-pw-status-faults** command is configured, the PW port only goes locally operationally down if any of the following PW status bits are received on the mate spoke-SDP:

```
pwNotForwarding (0), -- Pseudo Wire Not Forwarding
lacIngressFault (1), -- Local Attachment Circuit Rx Fault
lacEgresssFault (2), -- Local Attachment Circuit Tx Fault
psnIngressFault (3), -- Local PSN-facing PW Rx Fault
psnEgressFault (4), -- Local PSN-facing PW Tx Fault
```

If the configuration is removed, the system no longer takes the mate PW status fault bits into account in the operational state of the PW port.

Note: The stitching service in this context is an Epipe service in vc-switching mode for BGP-VPWS or T-LDP signaled PW, as follows:

- **Classic CLI commands**

```
configure
      service epipe epipe-id customer cust-id vc-switching [create]
            pw-port pw-port-id fpe fpe-id
            spoke-sdp sdp-id:vc-id [create]
                  or
            bgp-vpws
```

- **MD-CLI commands**

```
configure
      service epipe string
            customer reference vc-switching
            spoke-sdp sdp-id:vc-id
                  or
            bgp-vpws
      pw-port number
            epipe epipe-id
                fpe-id reference
```

Note: The stitching service in this context is an Epipe service in a non-VC-switching mode for EVPN, as follows:

- **Classic CLI commands**

```
configure
      service epipe epipe-id customer cust-id create
            pw-port pw-port-id fpe fpe-id
            bgp-evpn
```

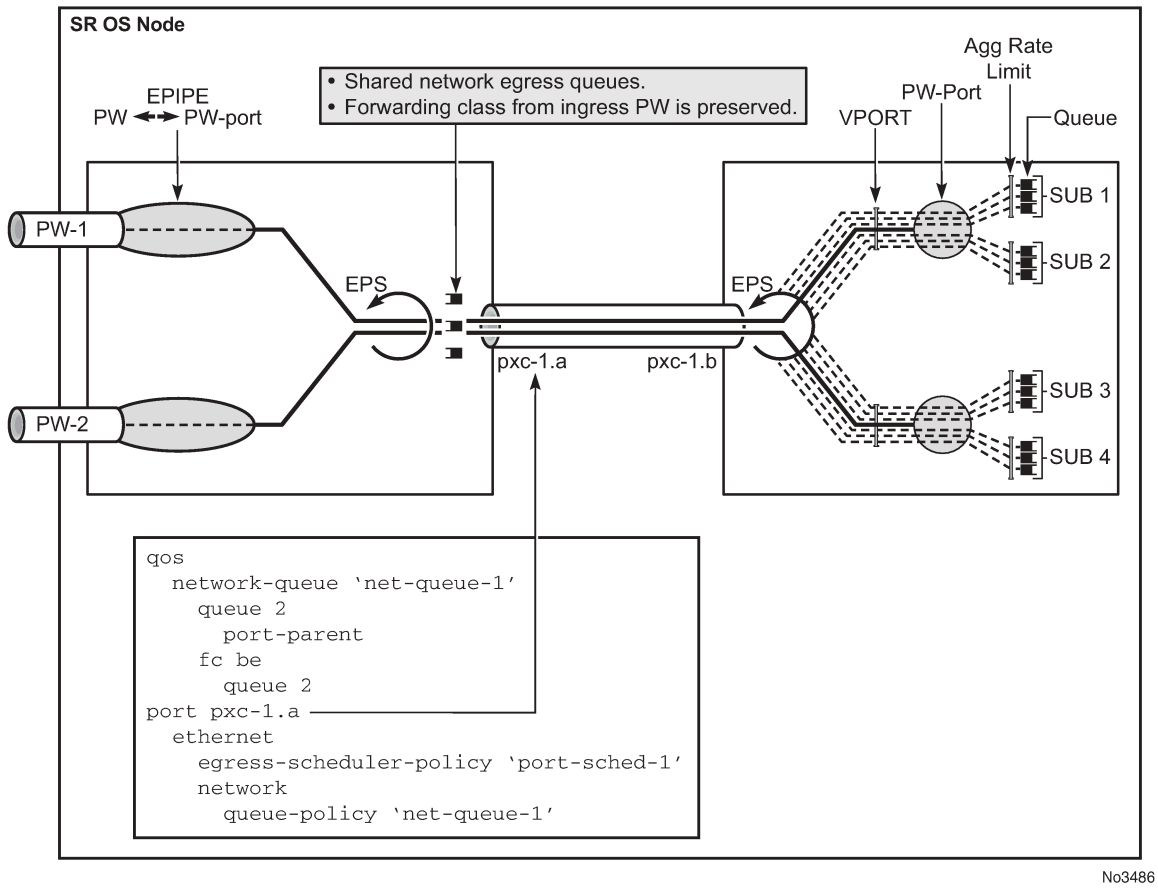
- **MD-CLI commands**

```
configure
      service epipe string
            customer reference
            bgp-evpn
      pw-port number
            epipe epipe-id
                fpe-id reference
```

7.3.4 QoS

QoS fundamentals for the case where multiple PWs are multiplexed over a single cross-connect are shown in Figure 255: QoS on FPE-based PW-port.

Figure 255: QoS on FPE-based PW-port



Egress QoS may be applied on both sides of the cross-connect (PXC sub-ports .a and .b) to control congestion on the cross-connect itself. This can be accomplished via an Egress Port Scheduler (EPS) applied to each sub-port.

EPS applied to pxc-1.a (transit side) manages congestion on the cross-connect for traffic coming from the external PWs. A single set of queues is shared by all PWs utilizing this cross-connect in this direction.

EPS applied to the pxc-1.b (terminating side) is used to manage congestion on the cross-connect for traffic going toward the PWs (leaving the SR OS node). A set of queues is dedicated to each PW-port SAP.

QoS on PXC sub-ports is described in the *7450 ESS, 7750 SR, 7950 XRS, and VSR Interface Configuration Guide*, "*PXC*".

7.3.4.1 Preservation of forwarding class across PXC

The internal cross-connect used by FPE based PW-port is relying on an MPLS tunnel built over internal network interfaces configured on PXCs. Those internal network interfaces are using a default network policy 1 for egress traffic classification, remarking and marking purposes. Because the PXC cross-connect

is MPLS based, the EXP bits in newly added MPLS header is marked according to the default network policy (for brevity reasons, only the relevant parts of the network policy are shown here).

```
*A:node-1>config>qos>network# info detail
----------------------------------------------
            description "Default network QoS policy."
            scope template
            egress
                fc af
                    lsp-exp-in-profile 3
                    lsp-exp-out-profile 2
                exit
                fc be
                    lsp-exp-in-profile 0
                    lsp-exp-out-profile 0
                exit
                fc ef
                    lsp-exp-in-profile 5
                    lsp-exp-out-profile 5
                exit
                fc h1
                    lsp-exp-in-profile 6
                    lsp-exp-out-profile 6
                exit
                fc h2
                    lsp-exp-in-profile 4
                    lsp-exp-out-profile 4
                exit
                fc l1
                    lsp-exp-in-profile 3
                    lsp-exp-out-profile 2
                exit
                fc l2
                    lsp-exp-in-profile 1
                    lsp-exp-out-profile 1
                exit
                fc nc
                    lsp-exp-in-profile 7
                    lsp-exp-out-profile 7
                exit
            exit
----------------------------------------------
```

As seen in this excerpt from the default network egress policy, the forwarding classes AF and L1 marks the EXP bits with the same values. This renders the forwarding classes AF and L1 set on one side of PXC, indistinguishable from each other on the other side of the PXC.

This effectively reduces the number of forwarding classes from 8 to 7 in deployment scenarios where the QoS treatment of traffic depends on preservation of forwarding classes across PXC. That is, in such scenarios, one of the forwarding classes AF or L1 should not be used.

7.3.5 Statistics on the FPE based PW-port

An FPE-based PW-port is associated with an internal spoke-SDP as described in PXC-based PW-port — building the cross-connect and FPE-based PW port operational state. Statistics for the number of forwarded/dropped packets/octets per direction on a PW-port are therefore maintained per this internal spoke-SDP. Octets field counts octets in customer frame (including customer's Ethernet header with VLAN

tags). The following command is used to display PW-port statistics along with the status of the internal spoke-SDP associated with the PW-Port:

```
*A:Dut-B# show pw-port 3 statistics
===============================================================================
Service Destination Point (Sdp Id 17000 Pw-Port 3)
===============================================================================
SDP Binding port     : pxc-1.b
VC-Id                : 100003               Admin Status       : up
Encap                : dot1q                Oper Status        : up
VC Type              : ether
Admin Ingress label  : 262135               Admin Egress label : 262136
Oper Flags           : (Not Specified)
Monitor Oper-Group   : (Not Specified)
Statistics            :
I. Fwd. Pkts.        : 12000                I. Dro. Pkts.      : 0
I. Fwd. Octs.        : 720000               I. Dro. Octs.      : 0
E. Fwd. Pkts.        : 12000                E. Fwd. Octets     : 720000
===============================================================================
```

7.3.6 Intra-chassis redundancy models for PXC-based PW port

Intra-chassis redundancy models rely on PXC-based LAG. PXC-based LAG can contain multiple PXCs on the same line card (port redundancy) or PXCs across different line cards (port- and card-level redundancy).

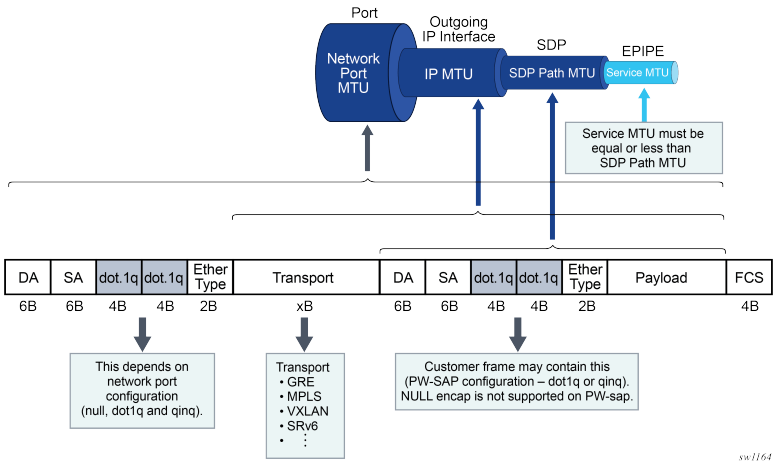
FPE-based PW ports also provide network level-redundancy where MPLS/IP can be rerouted to different I/O ports (because of network failure) without interruption of service.

7.4 PW ports and MTU

PW port based traffic is subject to a number of MTU checks, some of which depend on the tunnel type and signaling method. Downstream traffic (toward the remote end of the tunnel) is forced through several MTU checks in the data plane, and an MTU size violation can cause fragmentation or a packet drop. Other MTU checks are performed only in the control plane.

SDP-based tunnels (TLDP-based MPLS and GRE tunnels, and separately, L2oGRE tunnels) represent the most extensive example of configurable MTUs as shown in Figure 256: PW MTU.

Figure 256: PW MTU



In TLDP tunnels, the service MTU is negotiated through signaling in the control plane where values on both sides of the tunnel must match, otherwise, the tunnel fails to transition into an operational state.

A generic SDP-based tunnel (such as a L2oGRE or TLDP GRE/MPLS tunnel) under an Epipe service has the following configurable MTUs:

- Port MTU represents the maximum frame size on the outgoing physical port. This is configured under the physical port and is enforced in the data plane.
- IP MTU is the maximum IP packet size on the outgoing IP interface. This is configurable under an IP interface and is enforced in the data plane.
- SDP path MTU represents the maximum size of the frame that is encapsulated within the tunnel (excluding the transport header). Its value is determined based on the smallest MTU size on the path between the endpoints of the tunnel. In SR OS, this SDP path MTU is calculated automatically by subtracting the size of the transport header from the configured IP MTU of the outgoing interface. This is configurable under the SDP and is enforced in the data plane.
- Service MTU indicates the maximum frame size of the customer payload that can be transmitted over the service. Its value is determined by the MTU size within the customer's network. The service MTU is significant because it can customize the size of the customer's payload independently of the MTUs in the transport network. The service MTU is negotiated using signaling and must match on both sides of the tunnel. This MTU is not enforced in the data plane.
  The service MTU is configured under the Epipe service, which is also the basic configuration construct used for FPE-based PW ports. Hence, the service MTU configured under the Epipe service also applies to FPE-based PW ports.

  However, fixed PW ports (bound to a fixed port) are not configured under an Epipe service and therefore the service MTU configuration under the Epipe is inaccessible. Fixed PW ports are stitched to the tunnel within the SDP context with the **configure service sdp binding** command. This requires a separate configuration for the service MTU. Considering that fixed PW ports support only TLDP-signaled MPLS and GRE pseudowires in which the service MTU is advertised from each endpoint of the pseudowire, the **adv-service-mtu** command signals the service MTU for fixed PW ports.

Classic CLI

```
>config>service# info
----------------------------------------------
        sdp 1 mpls create
```

```
            far-end 10.20.1.3
            ldp
            binding
                port lag-1
                pw-port 1 vc-id 10 create
                    [no] adv-service-mtu <1..9782>
                    no shutdown
                exit
```

MD-CLI

```
[gl:configure pw-port 1]
    sdp 1 {
        admin-state enable
        vc-id 10
        adv-service-mtu 1514
    }
```

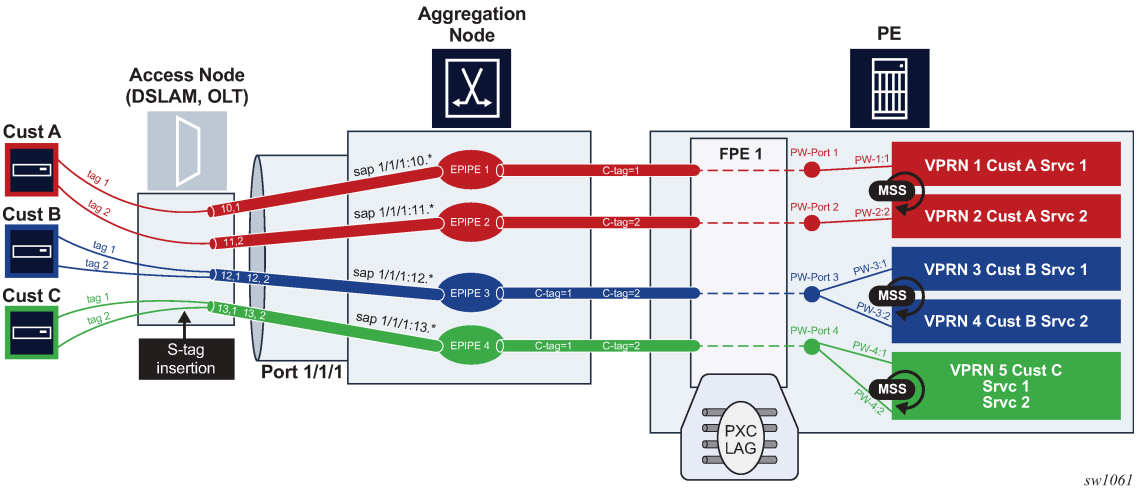
7.5 MSS and PW ports

A scheduler with a Multi-Service-Site (MSS) can be used to control aggregate bandwidth over multiple PW SAPs. This MSS can be associated with PW SAPs created under different PW ports that are bound to the same physical port, LAG, or an FPE object. This functionality is supported for the following SR OS PW port types:

- Fixed PW-port that is bound to a physical port or LAG
- FPE based PW-port that is using the PXC concept

An example of MSS applied to PW SAPs is shown in Figure 257: MSS on PW SAPs where three MSSes are applied to PW SAPs across multiple PW ports and VPRNs. Although the PW SAPs can span multiple VPRNs and PW ports, the underlying physical port, LAG or an FPE object must be shared for a set of PW SAPs under the same MSS instance.

Figure 257: MSS on PW SAPs



7.5.1 Configuration examples

The following are examples for fixed PW ports.

- PW port declaration

```
configure pw-port 1
    encap-type
    qinq-type
    :
configure pw-port 2
    encap-type
    qinq-type
```

- Binding a PW port to a spoke SDP

```
configure service sdp 2
    far-end <ip-address>
    binding
            port lag-1
                    pw-port 1 vc-id 10
                    pw-port 2 vc-id 11
```

- Create an MSS

```
configure service customer 1
    multi-service-site “mss-PW port”
            assignment port lag-1
```

- Apply MSS to all PW SAPs (associated with the same customer ID), and possibly belonging to different PW ports for aggregate bandwidth control within a Layer 3 (Intf) or a Layer 2 (Epipe) service. In this example, PW SAPs pw-1:1.1 and pw-1:2.2 belong to two different PW ports (1 and 2) on the same physical port/LAG.

```
configure service IES/VPRN/EPIPE
    sap pw-1:1.1 create
            multi-service-site mss-pw port
                    egress
            qos <id>

service ies/vprn/epipe
    sap pw-2:2.2 create
            multi-service-site mss-pw port
                    egress
                            qos <id>
```

The following are configuration examples for FPE-based PW ports:

- PW port declarations

```
configure pw-port 1
    encap-type
    qinq-type
            :
configure pw-port 2
```

```
        encap-type
        qinq-type
                :
```

- Configure PXC (with specific ports)

```
   configure port-xc
       pxc 1
               port 1/1/1
       pxc 2
               port 2/2/2
```

In classic CLI, sub-ports are automatically created.

```
  configure
      port pxc-1.a
      port pxc-1.b
      port pxc-2.a
      port pxc-2.b
```

In MDI-CLI, sub-ports must be manually created.

- PXC LAG configuration

```
   configure lag 100
       port pxc-1.a
       port pxc-2.a
   configure lag 200
       port pxc-1.a
       port pxc-2.b
```

- FPE configuration

```
  configure fwd-path-ext
              fpe 1
                      path xc-a lag-100 xc-b lag 200
                      pw-port
```

- PW port binding

```
   configure service
       epipe 5 customer 1 vc-switching
               pw-port 1 fpe 1
               spoke-sdp 1:2

   epipe 6 customer 1 vc-switching
       pw-port 2 fpe 1
       spoke-sdp 1:3
```

- Create an MSS

```
   configure service customer 1 create
       multi-service-site mss-fpe-pw-port
               assignment fpe 1
```

- Apply MSS to all PW SAPs that are associated with the same customer ID and possibly belonging to different PW ports for aggregate bandwidth control within a Layer 3 (Intf) or a Layer 2 (Epipe) services. In this example, PW SAPs pw-1:1.1 and pw-1:2.2 belong to two different PW ports (1 and 2) on the same physical port/LAG.

```
configure service ies/vprn/epipe
...
        sap pw-1:1.1 create
                multi-service-site mss-pw-port
                egress
                        qos <id>
```

- ```
  configure service ies/vprn/epipe
  ...
          sap pw-2:2.2 create
                  multi-service-site mss-pw-port
                  egress
                          qos <id>
  ```

7.5.2 Concurrent scheduling QoS mechanisms on a PW port

An assortment of PW SAPs assigned to ESM and business services (non-ESM) can coexist under the same PW port. The scheduling hierarchy for each set of SAPs is configured differently and supported concurrently. For example, ESM subscribers can use their own scheduling hierarchy, while the business service's PW SAPs on the same PW port can continue to use their own scheduling (via MSS, or however the business SAP scheduling hierarchy is set up).

7.5.3 Show command examples

The following are examples of the MSS and PW port command output.

```
show service customer 1 site "mss"
===============================================================================
Customer  1
===============================================================================
Customer-ID        : 1
Customer Name      : 1
Contact            : (Not Specified)
Description        : Default customer
Phone              : (Not Specified)
Creation Origin    : manual

-------------------------------------------------------------------------------
Multi Service Site
-------------------------------------------------------------------------------
Site               : mss
Description        : (Not Specified)
Assignment         : FPE 1
I. Sched Pol       : (Not Specified)
E. Sched Pol       : (Not Specified)
Egr Agg Rate Limit : 50000
Q Frame-Based Acct : Disabled
Limit Unused BW    : Disabled
E. Plcr Ctrl Polcy : (Not Specified)
I. Plcr Ctrl Polcy : (Not Specified)
```
```

```
-------------------------------------------------------------------------------
Service Association
-------------------------------------------------------------------------------
Service-Id         : 1 (Epipe)
 - SAP : pw-1:100
Service-Id         : 2 (VPRN)
 - SAP : pw-2:100
show qos agg-rate customer 1 site "mss"
===============================================================================
Aggregate Rate Information - Customer 1 MSS mss
===============================================================================
Root (Egr)
| slot(3)
|    AdminRate              : 50000
|    OperRate               : 50000
|    Limit Unused Bandwidth : disabled
|    OnTheWireRates         : false
|    LastMileOnTheWireRates : false
|
|--(Q) : 2->pw-2:100->8  (Port pxc-1.b)
|
|--(Q) : 2->pw-2:100->7  (Port pxc-1.b)
|
|--(Q) : 2->pw-2:100->6  (Port pxc-1.b)
|
|--(Q) : 2->pw-2:100->5  (Port pxc-1.b)
|
|--(Q) : 1->pw-1:100->8  (Port pxc-1.b)
|
|--(Q) : 1->pw-1:100->7  (Port pxc-1.b)
|
|--(Q) : 1->pw-1:100->6  (Port pxc-1.b)
|
|--(Q) : 1->pw-1:100->5  (Port pxc-1.b)
show qos scheduler-hierarchy customer 1 site "mss"
===============================================================================
Scheduler Hierarchy - Customer 1 MSS mss
===============================================================================
Root (Ing)
|
No Active Members Found on slot 3
Root (Egr)
| slot(3)
|--(Q) : 2->pw-2:100->8  (Port pxc-1.b)
|
|--(Q) : 2->pw-2:100->7  (Port pxc-1.b)
|
|--(Q) : 2->pw-2:100->6  (Port pxc-1.b)
|
|--(Q) : 2->pw-2:100->5  (Port pxc-1.b)
|
|--(Q) : 1->pw-1:100->8  (Port pxc-1.b)
|
|--(Q) : 1->pw-1:100->7  (Port pxc-1.b)
|
|--(Q) : 1->pw-1:100->6  (Port pxc-1.b)
|
|--(Q) : 1->pw-1:100->5  (Port pxc-1.b)
|
show>qos>policer-hierarchy# customer 1 site mss
===============================================================================
Policer Hierarchy - Customer 1 MSS mss
===============================================================================
root (Ing)
```

```
|
No Active Members Found on slot 1
root (Egr)
|
| slot(1/1)
|    Profile-preferred:Disabled
|
|--(A) : policer1 (Customer: 1, MSS: mss)
|   |
|   |--(P) : Policer 2->pxc-1.b:100->1
|   |   |
|   |   |    [Level 1 Weight 1]
|   |   |    Assigned PIR:5000       Offered:5000
|   |   |    Consumed:5000
|   |   |
|   |   |    Assigned FIR:5000
|   |
|   |--(P) : Policer 1->pxc-1.b:100->1
|   |   |
|   |   |    [Level 1 Weight 1]
|   |   |    Assigned PIR:5000       Offered:10000
|   |   |    Consumed:5000
|   |   |
|   |   |    Assigned FIR:5000
|
|--(A) : policer2 (Customer: 1, MSS: mss)
|   |
|   |--(P) : Policer 2->pxc-1.b:100->2
|   |   |
|   |   |    [Level 1 Weight 1]
|   |   |    Assigned PIR:5000       Offered:15000
|   |   |    Consumed:5000
|   |   |
|   |   |    Assigned FIR:5000
|   |
|   |--(P) : Policer 1->pxc-1.b:100->2
|   |   |
|   |   |    [Level 1 Weight 1]
|   |   |    Assigned PIR:5000       Offered:20000
|   |   |    Consumed:5000
|   |   |
|   |   |    Assigned FIR:5000
|
|--(A) : policer34 (Customer: 1, MSS: mss)
|   |
|   |--(A) : policer3 (Customer: 1, MSS: mss)
|   |   |
|   |   |--(P) : Policer 2->pxc-1.b:100->3
|   |   |   |
|   |   |   |    [Level 1 Weight 1]
|   |   |   |    Assigned PIR:10000      Offered:25000
|   |   |   |    Consumed:10000
|   |   |   |
|   |   |   |    Assigned FIR:10000
|   |   |
|   |   |--(P) : Policer 1->pxc-1.b:100->3
|   |   |   |
|   |   |   |    [Level 1 Weight 1]
|   |   |   |    Assigned PIR:10000      Offered:30000
|   |   |   |    Consumed:10000
|   |   |   |
|   |   |   |    Assigned FIR:10000
|   |
|   |--(A) : policer4 (Customer: 1, MSS: mss)
```

```
|   |   |
|   |   |--(P) : Policer 2->pxc-1.b:100->4
|   |   |   |
|   |   |   |    [Level 1 Weight 1]
|   |   |   |    Assigned PIR:0          Offered:35000
|   |   |   |    Consumed:0
|   |   |   |
|   |   |   |    Assigned FIR:0
|   |   |
|   |   |--(P) : Policer 1->pxc-1.b:100->4
|   |   |   |
|   |   |   |    [Level 1 Weight 1]
|   |   |   |    Assigned PIR:0          Offered:40000
|   |   |   |    Consumed:0
|   |   |   |
|   |   |   |    Assigned FIR:0
```

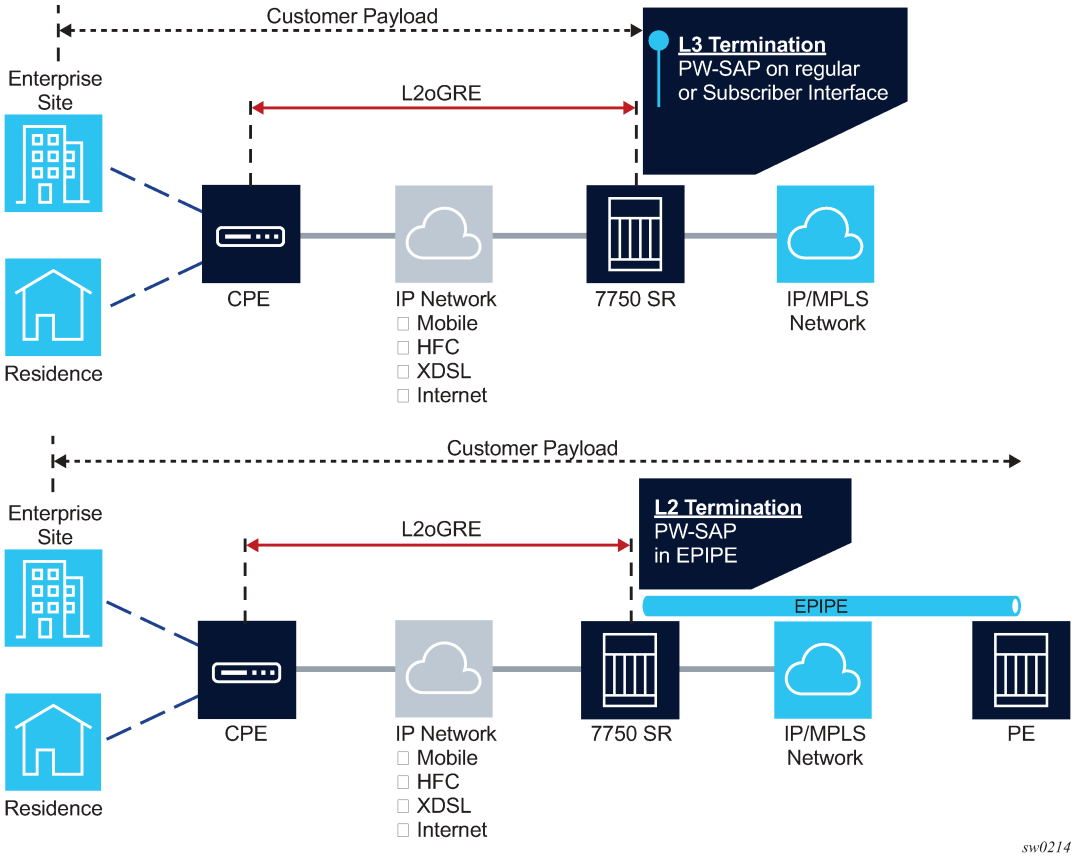
7.6 L2oGRE termination on FPE-based PW port

L2oGRE termination on an FPE-based PW port allows Layer 2 customer traffic to be transported over an IP network to an SR OS node deeper in the network. In the SR OS node, the customer's payload delivered within L2oGRE tunnel is extracted onto a PW SAP (configured under an interface, group interface, or Epipe) and handed off to an Layer 2 or Layer 3 service. This allows the operator to quickly and simply expand their service offering to their customers over an existing IP network. New service offerings become independent of the existing IP network to which CEs are attached.

For secure operation, the transit IPv4/IPv6 network should be trusted or alternatively, GRE traffic can be secured by IPsec (L2oGREoIPSec).

Figure 258: L2oGRE network examples shows a typical example where a customer payload is tunneled in GRE through an IPv4 or IPv6 network for an Layer 2 or Layer 3 handoff in an SR OS node that is placed deeper in the network.

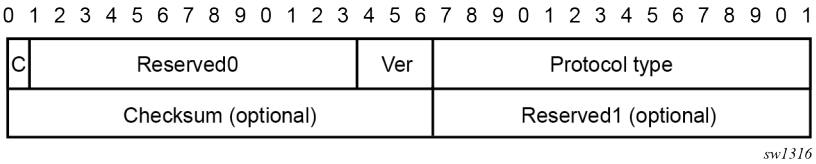
Figure 258: L2oGRE network examples



7.6.1 L2oGRE packet format

The following figure shows the L2oGRE packet format.

Figure 259: L2oGRE packet format



The supported GRE header in this context is defined in RFC 2784, *Generic Routing Encapsulation (GRE)*. The protocol type is set to 0x6558 (bridged Ethernet), and the Checksum and Reserved1 fields are normally omitted. The SR OS can accept headers with those two fields present, but the system omits them when encapsulating packets on transmission. Therefore, the transmitted GRE header length in the SR OS is 4 bytes.

Key and sequence number extensions to GRE as defined in RFC 2890, *Key and Sequence Number Extensions to GRE*, are not supported and received packets containing key or sequence numbers are dropped in the SR.

7.6.2 GRE delivery protocol

The GRE delivery protocol (transport protocol) can be IPv4 or IPv6.

If the GRE delivery protocol is IPv4, then the protocol field in IPv4 header is set to value 47 (GRE). The same value (47 – GRE) is used on the Next Header field if the delivery protocol is IPv6. IPv6 extension headers in the L2oGRE transport IPv6 header are not supported, and packets containing these IPv6 extension headers are dropped.

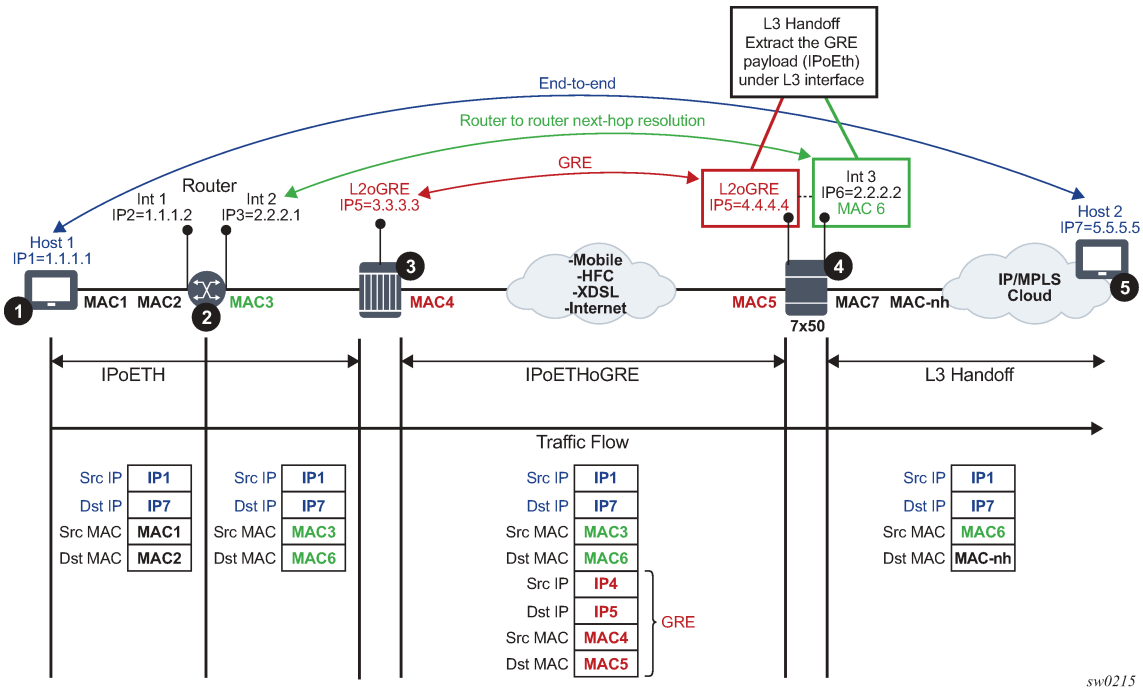
7.6.3 Tracking payloads and service termination points

A customer payload within L2oGRE can be extracted onto a PW SAP inside an SR OS node. This PW SAP can be configured under an interface, subscriber interface, or an Epipe. When on a PW SAP, customer traffic can be passed further into the network to its destination using Layer 2 or Layer 3 services. End-to-end Layer 2 and Layer 3 scenarios are described in the following sections.

7.6.3.1 Plain L3 termination

Figure 260: L2oGRE MTUs shows an example of plain Layer 3 termination with MTUs.

Figure 260: L2oGRE MTUs



In this example:

- Communication occurs between points 1 and 5.
- There may be a router present at point 2. A router at point 2 would see the SR OS node as Layer 3 next-hop.

- The device in point 3 encapsulates Layer 2 Ethernet frames into GRE and sends them to the SR OS node.
- The SR OS node at point 4 de-encapsulates the packet and performs an Layer 3 lookup on the inner packet to deliver it to the destination.

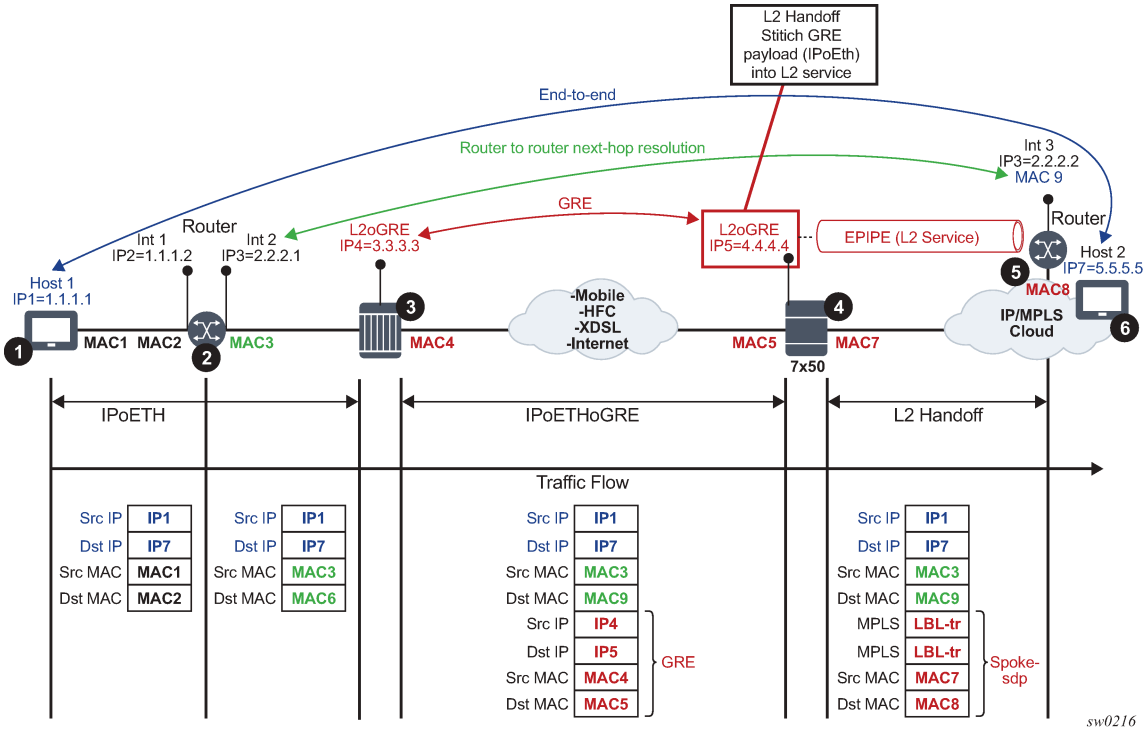
The following is an example where PW SAP is configured under a Layer 3 interface with a PW carrying IP over Ethernet:

```
configure
service vprn 1 customer 1 create
interface example-if
address 192.168.1.1/24
sap pw-1:5.5 create
ingress
filter ip 1000
egress
filter ip 2000
```

7.6.3.2 Layer 2 termination

Figure 261: Layer 2 termination and hand-off shows an example of Layer 2 termination and hand-off.

Figure 261: Layer 2 termination and hand-off



In this example:

- Communication occurs between points 1 and 6.
- There may be a router present at point 2. A router at point 2 would see a Layer 3 device at point 5 as a Layer 3 next-hop. Everything in between is Layer 2.

- The device at point 3 encapsulates Layer 2 Ethernet frames into GRE and sends them to the SR OS node (7x50).
- The SR OS node at point 4 de-encapsulates the packet and sends it into the Layer 2 service that leads to the node at point 5.

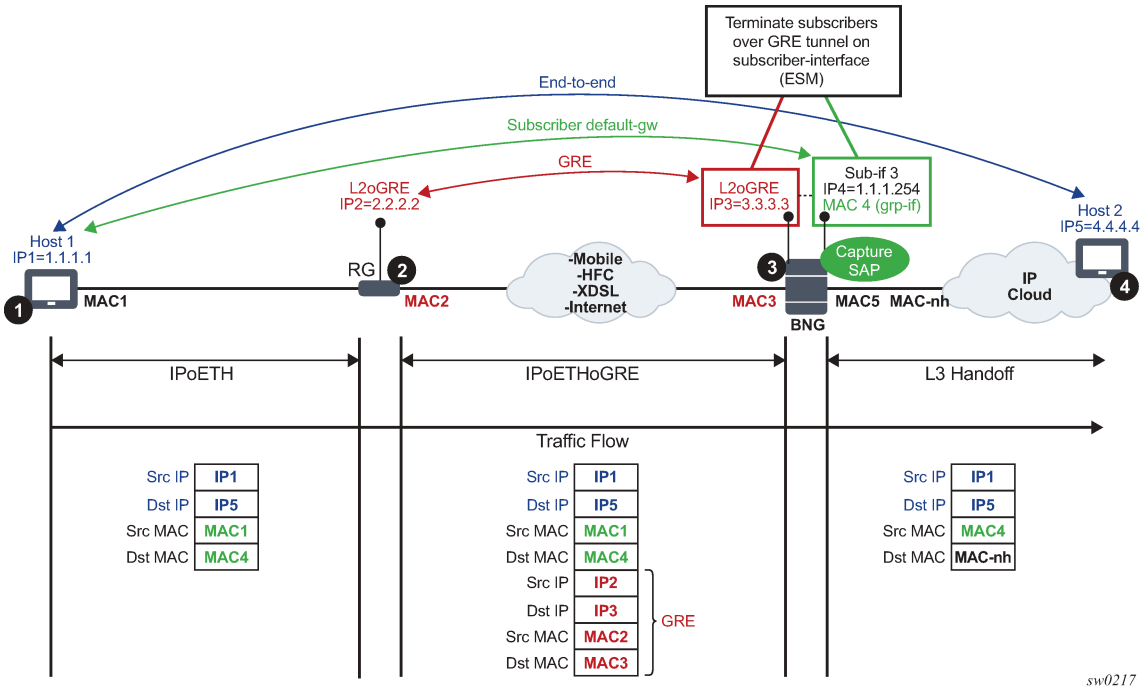
The following shows an example of PW SAP configured under an Epipe:

```
configure
service epipe 4 customer 1 create
sap pw-1:2 create
spoke-sdp 1:1
```

7.6.3.3 ESM termination

The primary case for ESM termination is business services. Figure 262: ESM termination shows an example of ESM termination.

Figure 262: ESM termination



In this example:

- Communication occurs between points 1 and 4.
- RG (Residential Gateway) at point 2 encapsulates L2 customer frames into GRE and sends them the SR OS node (7x50 BNG).
- The BNG node at point 3 terminates the subscriber traffic, performs an Layer 3 lookup, and sends it to the destination.

The following shows an example where a PW SAP is configured under a subscriber interface:

```
configure
service vprn 3 customer 1 create
interface subscriber-interface <sub-if-name>
address 192.168.1.1/24
group-interface  <grp-if-name>
sap pw-1:10.10 create
```

The following shows an example with the capture of a PW SAP configured:

```
configure
service vpls 2 customer 1 create
trigger-packet dhcp pppoe
sap pw-1:*.* capture-sap create
```

7.6.4 Configuration steps

L2oGRE tunnels are emulated as an SDP of type **gre-eth-bridged** (shown as GRE-B in the output of relevant show commands). This SDP defines two end-points on the tunnel:

- **far-end IP address**

  This defines the IP address of the remote device that terminates the tunnel.

- **local IP address where the tunnel is terminated within an SR OS**

  This is a special IP address within an SR OS node that is not associated with any interface. It is only used for L2oGRE tunnel termination.

Binding an L2oGRE tunnel to an FPE-based PW port within the SR OS is performed through an Epipe service. When the connection is established, the tunnel payload can be extracted to a PW SAP that can be used similarly to a regular SAP under Layer 3 interfaces, subscriber interfaces, or an Epipe.

Table 26: L2oGRE tunnel example configuration describes the L2oGRE example configuration steps.

Table 26: L2oGRE tunnel example configuration

Step	Example CLI	Comments
PXC-based PW Port related configuration		

PW Port creation	`pw-port 1`
	`encap-type dot1q`
	`dot1q-etype 0x8100`
		L2oGRE tunnel is terminated on this PW port.

Port-XC creation	`port-xc`
	`pxc 1 create`
	`port 1/1/1`
		This command triggers automatic creation of PXC sub-ports:
		`configure`
		`port pxc-1.a`
		`port pxc-1.b`

Step	Example CLI	Comments
		This is where the L2oGRE terminating PW port is anchored.

Creation of FPE that is used for PW port anchoring	`fwd-path-ext`
	`sdp-id-range from 17400 to 17500`
	`fpe 1 create`
	`path pxc 1`
	`pw-port`
		The application under this FPE is the PW port termination. The use of PW port in this case is versatile and can be used to terminate an L2oGRE or MPLS/GRE-based PW. In this example, it is used to terminate an L2oGRE tunnel.

L2oGRE tunnel definition		

Configuration of GRE-bridged tunnel termination IPv4/IPv6 addresses.	`service>system>gre-eth-bridged`
	`tunnel-termination 10.1.1.2 fpe 1`
	`service>system>gre-eth-bridged`
	`tunnel-termination 2001:db8 ::1 fpe 1`
		This is a special IPv4/IPv6 address that is not configured under any Layer 3 interface and it must not overlap with any IPv4/IPv6 address configured under an Layer 3 interface in Base router. Multiple termination IPv4/IPv6 addresses are supported.

Configuration of L2oGRE SDP	`service> sdp 2 gre-eth-bridged create`
	`far-end 10.1.1.2 local-end 10.1.1.2`
	or
	`service>`
	`sdp 2 gre-eth-bridged create`
	`far-end 2001:db8::2 local-end 2001:db8::1`
		This represents the L2oGRE tunnel within SR OS as defined by the tunnel end-point IPv4/IPv6 addresses.

Stitching L2oGRE tunnel to an anchored PW port		
Association between the PW port and a PXC port via FPE.	`service>epipe 1` `pw-port 1 fpe 1`	This command anchors the PW port 1 to a PXC port referenced in FPE 1.

Binding between L2oGRE tunnel and the PW port	`service>epipe 1`
	`pw-port 1 fpe 1`
	`spoke-sdp 2:1`
		L2oGRE is terminated on a PW port and the Layer 2 payload within the tunnel is extracted on the PW SAP

PW SAP service association		

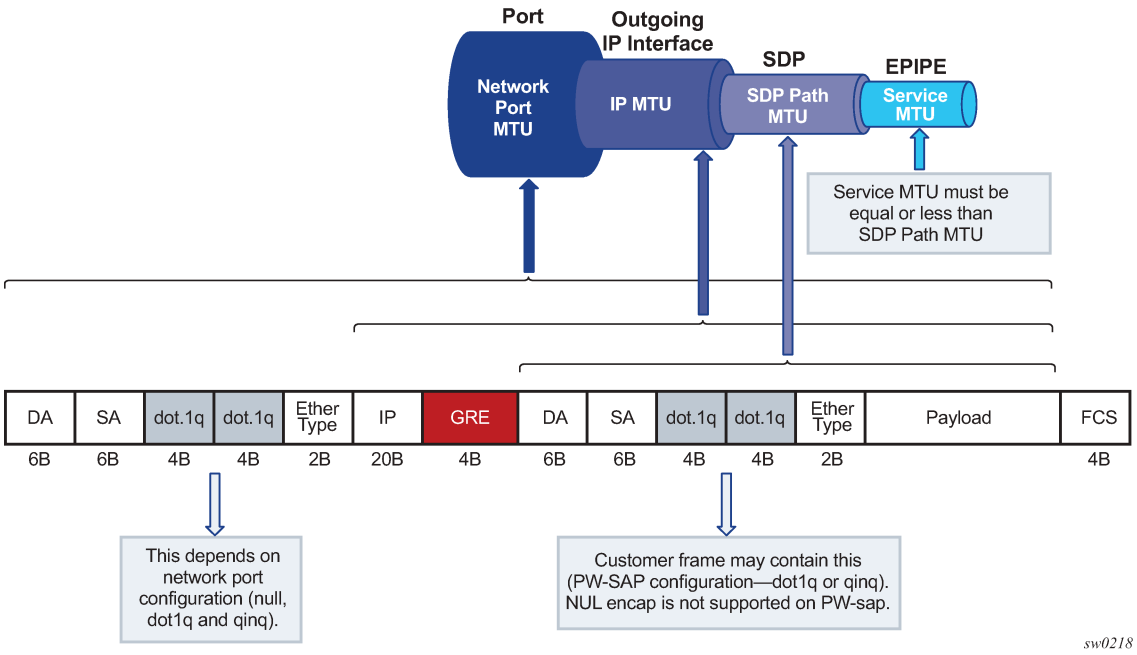
Step	Example CLI	Comments

Creation of services that use PW SAP	`service>epipe 100`
	`sap pw-1:100 create`
	`service>vprn 101>if`
	`sap pw-1:101 create`	

7.6.5 Fragmentation and MTU configuration

IP fragmentation is only supported for L2oGRE with IPv4 transport. Traffic is subjected to several MTU checks in the downstream direction (toward the remote end of the L2oGRE tunnel) within the SR OS node, as shown in Figure 263: L2oGRE MTUs.

Figure 263: L2oGRE MTUs



In the example:

- Port MTU represents the maximum frame size on the outgoing physical port.
- IP MTU is the maximum IP packet size on the outgoing IP interface.
- SDP Path MTU represents the maximum size of a frame that is encapsulated with the GRE tunnel. Its value is determined by the smallest MTU size on the path between the two GRE tunnel terminating end-points. The SDP Path MTU is calculated automatically by subtracting transport IP and GRE header bytes from the configured IP MTU of the outgoing interface.
- Service MTU indicates the maximum frame size that the customer can accept over the service (PW SAP). Its value is determined by the MTU size within the customer's network. The service MTU is

configured within the VC-switching Epipe that stitches the L2oGRE spoke SDP to a PW port. The default value is set to 1514 bytes.

MTU values:

Figure 262: ESM termination shows an example of IPv4 as the GRE delivery protocol.

- Port MTU = 1600 bytes (this is operator's configured value)
- IP MTU (of the outgoing interface) = 1600 bytes- 22 bytes= 1578 bytes (this is operator's configured value)
- SDP Path MTU = automatically calculated and set to 1578 bytes - 24 bytes = 1554 bytes.
- Service MTU = This value must be configured to a value no higher than 1554 bytes (SDP Path MTU).

Frames within an SR OS cannot be fragmented on a service or SDP level. However, L2oGRE traffic can be fragmented at the port level and for IPv4 traffic at any downstream point, if the DF bit in the IP header is cleared. The DF bit setting is controlled by the **config**>**service**>**sdp**>**allow-fragmentation** and **config**>**service**>**pw-template**>**allow-fragmentation** commands.

L2oGRE-v6 frames are subjected to the same MTU checks as IPv4 frames. However, IPv6 frames are not fragmented if their size exceeds MTU, and instead, are dropped.

7.6.6 Reassembly

L2oGRE reassembly for IPv4 transport is supported through a generic reassembly function that requires an MS-ISA. As fragmented traffic enters an SR OS node, it is redirected to an MS-ISA via filters. When the traffic is reassembled in the MS-ISA, it is re-inserted into the forwarding complex where normal processing continues (as if the non-fragmented traffic originally entered the node).

Table 27: Configuring reassembly for GRE describes the configuration steps to support reassembly for GRE.

Table 27: Configuring reassembly for GRE

Step	Example CLI	Comments

Creation of a NAT-group that contains MS-ISAs	`configure isa nat-group 1`
	`mda 1/1`
	`mda 2/1`
		The reassembly function is performed in a NAT group that contains one or more MS-ISAs.

Referencing a reassembly group that is used for traffic in the Base routing context	`configure router` `reassembly-group 1`	Identification of the reassembly group that is used for traffic in the Base routing context. Upon reassembly, traffic is re-inserted in the same (Base) routing context. Reassembly group ID corresponds to the NAT group ID (in this case 1). There can be multiple NAT groups (reassembly groups) configured in the system and this command identifies the reassembly group that is used in the Base routing context.
Identifying and directing fragmented	`configure filter ip-filter <id>`	Fragmented GRE traffic is identified via a filter and redirected to the reassembly function. This filter

Step	Example CLI	Comments

traffic to the reassembly function.	`default-action forward`
	`entry <id> create`
	`match protocol gre`
	`fragment true`
	`exit`
	`action reassemble`
	`exit`
		must be applied to all ingress interfaces on which GRE traffic is expected to arrive.

8 VSR pseudowire ports

This chapter provides information about Virtualized Service Router (VSR) pseudowire ports (PW ports), process overview, and implementation notes.

8.1 Pseudowire ports

This chapter provides information about pseudowire ports (PW ports), process overview, and implementation notes.

8.1.1 PW port list

Each port eligible to transmit traffic on a Flex PW port, must be added to a pw-port-list.

```
config>service>system> PW port list# port ?
 - no port <port-id> [<port-id>... (16 max)]
- port <port-id> [<port-id>... (16 max)]
```

Only hybrid ports (**configure port port-id ethernet mode hybrid**) can be members of a PW port list.

A port used by Flex PW port can be shared with any other Layer 2 or Layer 3 service. For example, a Layer 3 interface using a regular SAP can be associated with a VPRN service, while the underlying port is also used by a Flex PW port. Another regular SAP from the same port can be associated with a VPLS or Epipe service at the same time.

Follow these rules when populating a PW port list:

- A port must be in hybrid mode before it is added to a PW port list.
- Before a port is removed from or added to a PW port list, all PW ports must be dissociated from the corresponding Epipe services (the PW ports must be unconfigured). This implies that all PW SAPs be deleted.
- Network interfaces (configured in Base routing context) can be configured only on ports that are in the PW port list.
- A port mode (access, network, or hybrid) cannot be changed while the port is in the PW port list.

From this, an operator can consider adding all ports that are in hybrid mode to a PW port list from the beginning of the system configuration. This ensures that those ports can be used by Flex PW port at any later time, independently of their current use.

8.1.2 Failover times

Traffic loss during port switchover depends on the following factors:

- Routing convergence; this depends on the number of routes in the network and the deployed routing protocol.

- The time it takes to associate PW SAPs with a new port. This action is performed within the VSR and the timing depends on the number of PW SAPs that are being moved from the old port to the new port. Note that PW SAPs are not recreated, instead the existing PW-SAPs are re-mapped to a new port.

The egress queues on the new port must be recreated. However, this does not incur additional downtime because a spare egress queue is always present on a port (referred to as a failover queue) and is used while per PW SAP egress queues are being created.

Depending on the scale and network load, downtime during a switchover can range from the sub-second range to a several seconds.

8.1.3 QoS

Egress queues are attached to the port that is used by a Flex PW port to forward traffic (a Flex PW port is bound to one of the ports in the PW port list). In similar fashion, if an egress port scheduler is used, it is attached to the same port. However, the egress port schedulers must be associated by configuration with every port in the PW port list while egress queues are instantiated only on a single port. During a port switchover, egress queues are recreated on the new port and while this is occurring, the failover queue is used to forward traffic. Each port has a single egress failover queue that is used to forward traffic while SAP or subscriber queues are being recreated during transitioning events.

On the other hand, egress port scheduler must be configured by the operator in advance on each port in the PW port list so that it can be ready to treat traffic immediately after its children queues are recreated on this port.

Policers are used on ingress and they do not need to be recreated during port switchover. Instead, they are re-mapped to a new port.

An example QoS configuration is provided below:

1. Egress port scheduler definition

```
port-scheduler-policy "flex" create
    max-rate 1000000
    group "test" create
    exit
    level 1 rate 100000
exit
```

2. Association between the egress port scheduler and ports

```
configure  port 1/1/1
    ethernet
       mode hybrid
       encap-type qinq
       egress-scheduler-policy "flex"
    exit
no shutdown
configure  port 1/1/2
    ethernet
       mode hybrid
       encap-type qinq
       egress-scheduler-policy "flex"
    exit
no shutdown
```

3. Association between subscriber queues or policers and the egress port scheduler

```
configure  qos sap-egress 2
    queue 1 create
       port-parent level 1
       rate 10000
    exit
    queue 2 create
       port-parent level 1
       rate 10000
    exit
    queue 3 create
       port-parent level 2
       rate 1000
    exit
```

4. Applying queue policy to an object:
   - Subscriber management

```
configure  subscriber-mgmt sla-profile "sla-profile-1"
    egress
       qos 2
    exit
```

   - PW SAP in a Layer 2 service

```
configure  service epipe 10
    sap pw-1:1.2
       egress
       qos 2
    exit
```

   - PW SAP in a Layer 3 service

```
configure  service vprn 11
    interface 'flex-int'
       address 1.1.1.1/24
       sap pw-1:1.3
          egress
             qos 2
          exit
    exit
exit
```

8.1.4 PW port termination for various tunnel types

The MPLS-based spoke SDP and L2oGRE-based spoke SDP tunnel types are supported on a Flex PW port.

8.1.4.1 MPLS-based spoke SDP

An MPLS-based spoke SDP can be rerouted between the ports defined in the PW port list and still be mapped to the same PW port based on the service label. Ethernet payload within the spoke SDP can be extracted onto a PW SAP with minimal traffic loss during port switchover.

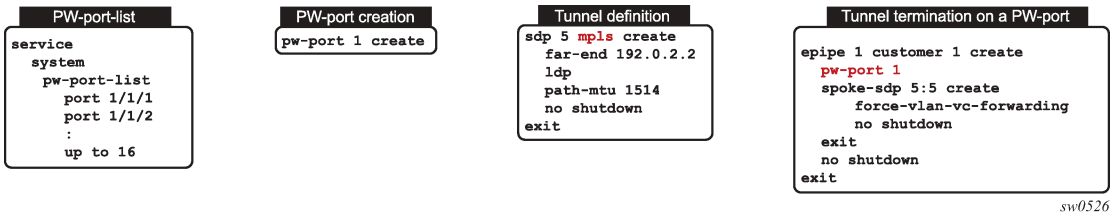
8.1.4.1.1 Provisioning

The termination of a MPLS-based spoke SDP on a Flex PW port follows the common provisioning framework:

1. Create a PW port list.
2. Add ports that are in hybrid mode to the PW port list.
3. Create a PW port.
4. Configure a tunnel.
5. Terminate a tunnel on a PW port via an Epipe service. A PW port must be configured within the Epipe before a spoke SDP is added to the same Epipe.

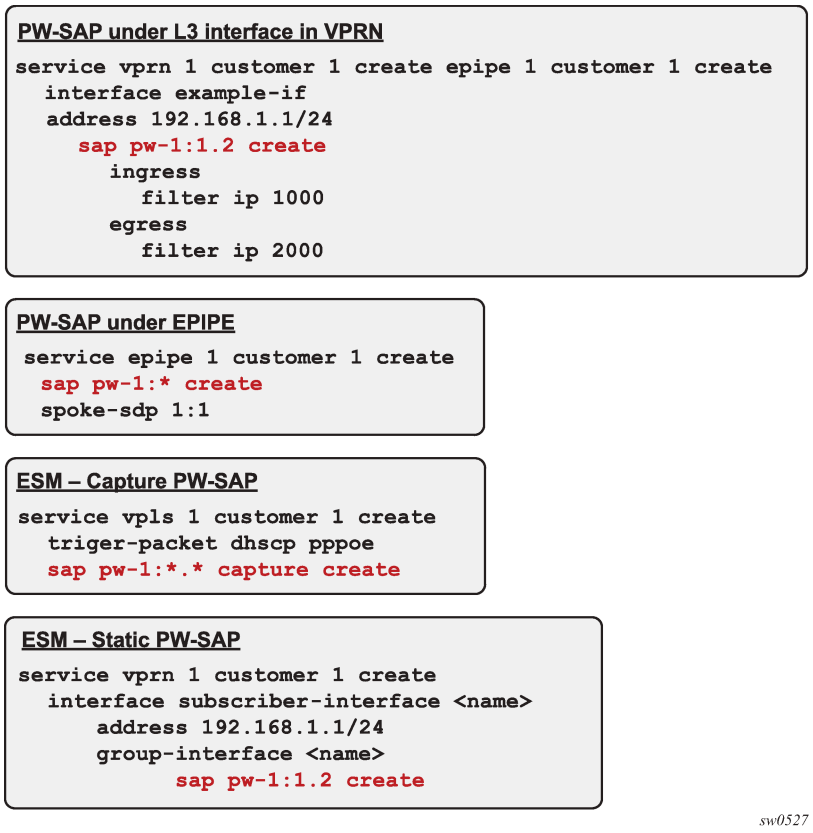
   The steps for MPLS-based spoke SDP termination on a Flex PW port are displayed in Figure 264: Provisioning MPLS-based spoke SDP termination on a flex PW port.

   Figure 264: Provisioning MPLS-based spoke SDP termination on a flex PW port



6. When a Flex PW port is associated with a tunnel, a payload from the tunnel can be extracted using service delimiting tags (Ethernet VLANs to S-tags, C-tags in the inner Ethernet header) on a PW SAP on a Layer 2 or Layer 3 service. See Figure 265: PW SAP configuration example

Figure 265: PW SAP configuration example



8.1.4.1.2 Flex PW-port operational state for MPLS based spoke SDP

The operational state of the Flex PW port is driven by the ability of the Epipe service (that ties the PW port to the spoke SDP) to forward traffic. The following events renders the PW port non-operational and triggers propagation of the PW status bits toward the remote end:

The Epipe service is shut down. This raises the following flags on the local end:

- lacIngressFault
- lacEgressFault
- psnEgressFault

The corresponding PW status bits that are propagated to the remote end raises the counterpart flags on the remote end.

The PW port within the Epipe service is shutdown. This raises the following flags on the local end:

- lacIngressFault
- lacEgressFault

The corresponding PW status bits that are propagated to the remote end raises the counterpart flags on the remote end.

MTU mismatch. This raises the following flags on the local end:

- lacIngressFault
- lacEgressFault
- pwNotForwarding

  The corresponding PW status bits that are propagated to the remote end raises the counterpart flags on the remote end.

In addition, PW port transitions into a non-operation state without propagating any PW status bits if the remote end cannot be reached.

The operation state of the Flex PW port state can be observed through the state of the underlying tunnel and the corresponding service via the following show command:

```
show pw-port 10 detail
===============================================================================
PW Port Information
===============================================================================
PW Port             : 10
Encap               : dot1q
IfIndex             : 1526726666
Description         : PW Port
Dot1Q Ethertype     : 0x8100
Service Id          : 10239
Admin Status        : up
Oper Status         : up
===============================================================================
```

8.1.4.1.3 Statistics

Statistics for the number of forwarded or dropped packets per octets per direction on a Flex PW port associated with a MPLS based spoke SDP are maintained per the spoke SDP. Octets field counts octets in customer frame (including customer's Ethernet header with VLAN tags).

The following command is used to display Flex PW port statistics along with the status of the spoke SDP associated with the PW port:

```
config>service>epipe# show pw-port 10 statistics
===============================================================================
Pw-Port 10
===============================================================================

Statistics            :
I. Fwd. Pkts.        : 110                  I. Dro. Pkts.      : 0
I. Fwd. Octs.        : 23060                I. Dro. Octs.      : 0
E. Fwd. Pkts.        : 76                   E. Fwd. Octets     : 16660

Grp Enc Stats         :
I. Dro. Inv. Spi.    : 0                    I. Dro. OthEncPkts.: 0
E. Dro. Enc. Pkts.   : 0
===============================================================================
```

8.1.4.2 L2oGRE-based spoke SDP

L2oGRE is supported for IPv4 and IPv6 transport with a termination IP address that must reside in the base router. Multiple L2oGRE tunnels can share the same termination IP address.

Each L2oGRE tunnel is represented by a unique pair of tunnel-end IP addresses. As the local endpoint address in VSR is usually shared between the tunnels, the tunnel far-end IP address becomes a differentiating field.

In VSR, an L2oGRE tunnel is represented by an SDP, which is then mapped as a spoke-SDP to a Flex PW port. Although it is mandatory to configure a VC-ID, in spoke-SDP the VC-ID loses its meaning because of the nature of L2oGRE tunnel: no sub-tunnels based on an MLPS label can be multiplexed within the two L2oGRE endpoints.

8.1.4.2.1 Provisioning

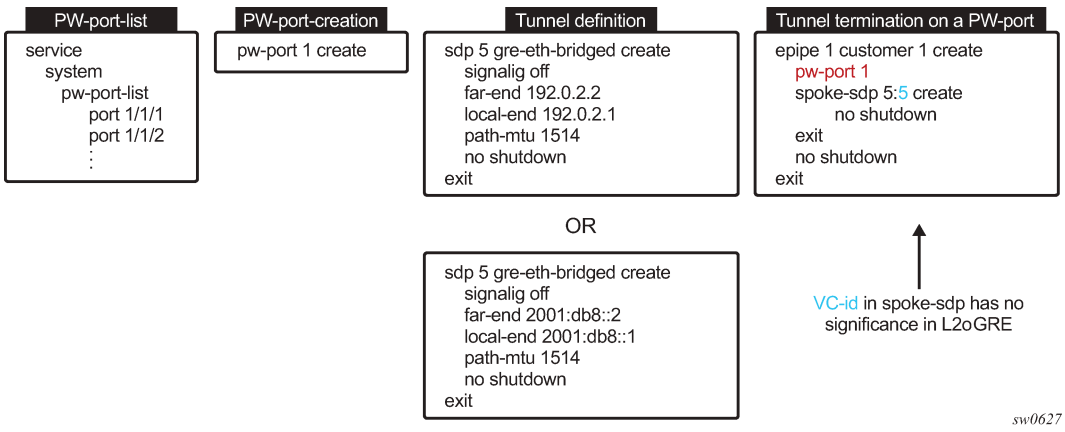
Perform the following common provisioning steps to terminate an L2oGRE tunnel on a Flex PW port:

1. Create a PW port list.
2. Add ports that are in hybrid mode to the PW port list.
3. Create a PW port.
4. Configure an L2oGRE tunnel using spoke-SDP.
5. Terminate a tunnel on a PW port using an Epipe service.

A PW port must be configured within the Epipe before a spoke-SDP is added to the same Epipe.

The steps for L2oGRE termination on a Flex PW port are displayed in Figure 266: Provisioning L2oGRE spoke-SDP termination on a flex PW port.

Figure 266: Provisioning L2oGRE spoke-SDP termination on a flex PW port



After a Flex PW port is associated with a tunnel, a payload from the tunnel can be extracted using service delimiting tags (such as S-tags or C-tags in the inner Ethernet header) on a PW SAP in a Layer 2 or Layer 3 service.

8.1.4.2.2 Flex PW-port operational state for L2oGRE-based spoke SDP

The operational state of the Flex PW port is determined by the ability of the stitching service (that is, the Epipe that ties the PW port to the tunnel using L2oGRE spoke-SDP) to forward traffic. This relationship can cause the stitching service's operational status to transition to a down state in the following cases:

- the SDP far-end is not reachable
- the route-table entry is missing
- SDP is down
- the Epipe service is administratively or operationally down

8.1.4.2.3 Reassembly

Reassembly of L2oGRE over IPv4 transport is supported through a generic reassembly function that requires a vISA. Filters redirect fragmented traffic, as it enters the VSR node, to a vISA. After the traffic is reassembled in the vISA, it is re-inserted into the vFP where normal processing continues, as if the non-fragmented traffic had originally entered the node.

Perform the steps in Table 28: Configuration steps for L2oGRE reassembly to configure reassembly for L2oGRE.

Table 28: Configuration steps for L2oGRE reassembly

Step	Example CLI	Comments
1. Create a NAT-group that contains MS-ISAs	`configure isa nat-group 1` `mda 1/1`	The reassembly function is performed in a NAT group that contains a vISA.
2. Reference a reassembly-group that is used for traffic in the base routing context	`configure router` `reassembly-group 1`	The reassembly-group that is used for traffic in the base routing context is identified. Upon reassembly, traffic is re-inserted in the same base routing context. The **reassembly-group id** corresponds to the **nat-group id** (in this case, the ID is 1).

3. Identify and direct fragmented traffic to the reassembly function	`configure filter ip-filter <id>`
	`default-action forward`
	`entry <id> create`
	`match protocol gre`
	`fragment true`
	`exit`
	`action reassemble`
	`exit`
		Fragmented GRE traffic is identified using a filter and is then redirected to the reassembly function. This filter must be applied to all ingress interfaces on which GRE traffic is expected to arrive.

9 Standards and protocol support

Note:
The information provided in this chapter is subject to change without notice and may not apply to all platforms.

Nokia assumes no responsibility for inaccuracies.

9.1 Access Node Control Protocol (ANCP)

draft-ietf-ancp-protocol-02, *Protocol for Access Node Control Mechanism in Broadband Networks*

RFC 5851, *Framework and Requirements for an Access Node Control Mechanism in Broadband Multi-Service Networks*

9.2 Bidirectional Forwarding Detection (BFD)

draft-ietf-idr-bgp-ls-sbfd-extensions-01, *BGP Link-State Extensions for Seamless BFD*

draft-ietf-lsr-ospf-bfd-strict-mode-10, *OSPF BFD Strict-Mode*

RFC 5880, *Bidirectional Forwarding Detection (BFD)*

RFC 5881, *Bidirectional Forwarding Detection (BFD) IPv4 and IPv6 (Single Hop)*

RFC 5882, *Generic Application of Bidirectional Forwarding Detection (BFD)*

RFC 5883, *Bidirectional Forwarding Detection (BFD) for Multihop Paths*

RFC 7130, *Bidirectional Forwarding Detection (BFD) on Link Aggregation Group (LAG) Interfaces*

RFC 7880, *Seamless Bidirectional Forwarding Detection (S-BFD)*

RFC 7881, *Seamless Bidirectional Forwarding Detection (S-BFD) for IPv4, IPv6, and MPLS*

RFC 7883, *Advertising Seamless Bidirectional Forwarding Detection (S-BFD) Discriminators in IS-IS*

RFC 7884, *OSPF Extensions to Advertise Seamless Bidirectional Forwarding Detection (S-BFD) Target Discriminators*

9.3 Border Gateway Protocol (BGP)

draft-gredler-idr-bgplu-epe-14, *Egress Peer Engineering using BGP-LU*

draft-hares-idr-update-attrib-low-bits-fix-01, *Update Attribute Flag Low Bits Clarification*

draft-ietf-idr-add-paths-guidelines-08, *Best Practices for Advertisement of Multiple Paths in IBGP*

draft-ietf-idr-best-external-03, *Advertisement of the best external route in BGP*

draft-ietf-idr-bgp-flowspec-oid-03, *Revised Validation Procedure for BGP Flow Specifications*

draft-ietf-idr-bgp-gr-notification-01, *Notification Message support for BGP Graceful Restart*

draft-ietf-idr-bgp-ls-app-specific-attr-16, *Application-Specific Attributes Advertisement with BGP Link-State*

draft-ietf-idr-bgp-ls-flex-algo-06, *Flexible Algorithm Definition Advertisement with BGP Link-State*

draft-ietf-idr-bgp-optimal-route-reflection-10, *BGP Optimal Route Reflection (BGP-ORR)*

draft-ietf-idr-error-handling-03, *Revised Error Handling for BGP UPDATE Messages*

draft-ietf-idr-flowspec-interfaceset-03, *Applying BGP flowspec rules on a specific interface set*

draft-ietf-idr-flowspec-path-redirect-05, *Flowspec Indirection-id Redirect* – localised ID

draft-ietf-idr-flowspec-redirect-ip-02, *BGP Flow-Spec Redirect to IP Action*

draft-ietf-idr-link-bandwidth-03, *BGP Link Bandwidth Extended Community*

draft-ietf-idr-long-lived-gr-00, *Support for Long-lived BGP Graceful Restart*

RFC 1772, *Application of the Border Gateway Protocol in the Internet*

RFC 1997, *BGP Communities Attribute*

RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*

RFC 2439, *BGP Route Flap Damping*

RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*

RFC 2858, *Multiprotocol Extensions for BGP-4*

RFC 2918, *Route Refresh Capability for BGP-4*

RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*

RFC 4360, *BGP Extended Communities Attribute*

RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*

RFC 4486, *Subcodes for BGP Cease Notification Message*

RFC 4659, *BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*

RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

RFC 4724, *Graceful Restart Mechanism for BGP* – helper mode

RFC 4760, *Multiprotocol Extensions for BGP-4*

RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*

RFC 5065, *Autonomous System Confederations for BGP*

RFC 5291, *Outbound Route Filtering Capability for BGP-4*

RFC 5396, *Textual Representation of Autonomous System (AS) Numbers* – asplain

RFC 5492, *Capabilities Advertisement with BGP-4*

RFC 5668, *4-Octet AS Specific BGP Extended Community*

RFC 6286, *Autonomous-System-Wide Unique BGP Identifier for BGP-4*

RFC 6793, *BGP Support for Four-Octet Autonomous System (AS) Number Space*

RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*

RFC 6811, *Prefix Origin Validation*

RFC 6996, *Autonomous System (AS) Reservation for Private Use*

RFC 7311, *The Accumulated IGP Metric Attribute for BGP*

RFC 7606, *Revised Error Handling for BGP UPDATE Messages*

RFC 7607, *Codification of AS 0 Processing*

RFC 7674, *Clarification of the Flowspec Redirect Extended Community*

RFC 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*

RFC 7854, *BGP Monitoring Protocol (BMP)*

RFC 7911, *Advertisement of Multiple Paths in BGP*

RFC 7999, *BLACKHOLE Community*

RFC 8092, *BGP Large Communities Attribute*

RFC 8097, *BGP Prefix Origin Validation State Extended Community*

RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*

RFC 8277, *Using BGP to Bind MPLS Labels to Address Prefixes*

RFC 8571, *BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions*

RFC 8950, *Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop*

RFC 8955, *Dissemination of Flow Specification Rules*

RFC 8956, *Dissemination of Flow Specification Rules for IPv6*

RFC 9086, *Border Gateway Protocol - Link State (BGP-LS) Extensions for Segment Routing BGP Egress Peer Engineering*

9.4 Broadband Network Gateway (BNG) Control and User Plane Separation (CUPS)

3GPP 23.007, *Restoration procedures*

3GPP 29.244, *Interface between the Control Plane and the User Plane nodes*

3GPP 29.281, *General Packet Radio System (GPRS) Tunnelling Protocol User Plane (GTPv1-U)*

BBF TR-459, *Control and User Plane Separation for a Disaggregated BNG*

BBF TR-459.2, *Multi-Service Disaggregated BNG with CUPS: Integrated Carrier Grade NAT function*

RFC 8300, *Network Service Header (NSH)*

9.5 Certificate management

RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*

RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*

RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*

RFC 6712, *Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)*

RFC 7030, *Enrollment over Secure Transport*

RFC 7468, *Textual Encodings of PKIX, PKCS, and CMS Structures*

9.6 Circuit emulation

RFC 4553, *Structure-Agnostic Time Division Multiplexing (TDM) over Packet (SAToP)*

RFC 5086, *Structure-Aware Time Division Multiplexed (TDM) Circuit Emulation Service over Packet Switched Network (CESoPSN)*

RFC 5287, *Control Protocol Extensions for the Setup of Time-Division Multiplexing (TDM) Pseudowires in MPLS Networks*

9.7 Ethernet

IEEE 802.1AB, *Station and Media Access Control Connectivity Discovery*

IEEE 802.1ad, *Provider Bridges*

IEEE 802.1ag, *Connectivity Fault Management*

IEEE 802.1ah, *Provider Backbone Bridges*

IEEE 802.1ak, *Multiple Registration Protocol*

IEEE 802.1aq, *Shortest Path Bridging*

IEEE 802.1ax, *Link Aggregation*

IEEE 802.1D, *MAC Bridges*

IEEE 802.1p, *Traffic Class Expediting*

IEEE 802.1Q, *Virtual LANs*

IEEE 802.1s, *Multiple Spanning Trees*

IEEE 802.1w, *Rapid Reconfiguration of Spanning Tree*

IEEE 802.1X, *Port Based Network Access Control*

IEEE 802.3ac, *VLAN Tag*

IEEE 802.3ad, *Link Aggregation*

IEEE 802.3ah, *Ethernet in the First Mile*

IEEE 802.3x, *Ethernet Flow Control*

ITU-T G.8031/Y.1342, *Ethernet Linear Protection Switching*

ITU-T G.8032/Y.1344, *Ethernet Ring Protection Switching*

ITU-T Y.1731, *OAM functions and mechanisms for Ethernet based networks*

9.8 Ethernet VPN (EVPN)

draft-ietf-bess-evpn-ipvpn-interworking-06, *EVPN Interworking with IPVPN*

draft-ietf-bess-evpn-irb-mcast-04, *EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding* – ingress replication

draft-ietf-bess-evpn-pref-df-06, *Preference-based EVPN DF Election*

draft-ietf-bess-evpn-unequal-lb-16, *Weighted Multi-Path Procedures for EVPN Multi-Homing* – section 9

draft-ietf-bess-evpn-virtual-eth-segment-06, *EVPN Virtual Ethernet Segment*

draft-ietf-bess-pbb-evpn-isid-cmacflush-00, *PBB-EVPN ISID-based CMAC-Flush*

draft-sajassi-bess-evpn-ip-aliasing-05, *EVPN Support for L3 Fast Convergence and Aliasing/Backup Path* – IP Prefix routes

RFC 7432, *BGP MPLS-Based Ethernet VPN*

RFC 7623, *Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)*

RFC 8214, *Virtual Private Wire Service Support in Ethernet VPN*

RFC 8317, *Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) an Provider Backbone Bridging EVPN (PBB-EVPN)*

RFC 8365, *A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)*

RFC 8560, *Seamless Integration of Ethernet VPN (EVPN) with Virtual Private LAN Service (VPLS) and Their Provider Backbone Bridge (PBB) Equivalents*

RFC 8584, *DF Election and AC-influenced DF Election*

RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*

RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)* – Asymmetric IRB Procedures and Mobility Procedure

RFC 9136, *IP Prefix Advertisement in Ethernet VPN (EVPN)*

RFC 9161, *Operational Aspects of Proxy ARP/ND in Ethernet Virtual Private Networks*

RFC 9251, *Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)*

9.9 gRPC Remote Procedure Calls (gRPC)

cert.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) Certificate Management Service*

file.proto version 0.1.0, *gRPC Network Operations Interface (gNOI) File Service*

gnmi.proto version 0.8.0, *gRPC Network Management Interface (gNMI) Service Specification*

PROTOCOL-HTTP2, *gRPC over HTTP2*

system.proto Version 1.0.0, *gRPC Network Operations Interface (gNOI) System Service*

9.10 Intermediate System to Intermediate System (IS-IS)

draft-ietf-isis-mi-02, *IS-IS Multi-Instance*

draft-kaplan-isis-ext-eth-02, *Extended Ethernet Frame Size Support*

ISO/IEC 10589:2002 Second Edition, *Intermediate system to Intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)*

RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*

RFC 2973, *IS-IS Mesh Groups*

RFC 3359, *Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System*

RFC 3719, *Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 3787, *Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS)*

RFC 4971, *Intermediate System to Intermediate System (IS-IS) Extensions for Advertising Router Information*

RFC 5120, *M-ISIS: Multi Topology (MT) Routing in IS-IS*

RFC 5130, *A Policy Control Mechanism in IS-IS Using Administrative Tags*

RFC 5301, *Dynamic Hostname Exchange Mechanism for IS-IS*

RFC 5302, *Domain-wide Prefix Distribution with Two-Level IS-IS*

RFC 5303, *Three-Way Handshake for IS-IS Point-to-Point Adjacencies*

RFC 5304, *IS-IS Cryptographic Authentication*

RFC 5305, *IS-IS Extensions for Traffic Engineering TE*

RFC 5306, *Restart Signaling for IS-IS* – helper mode

RFC 5308, *Routing IPv6 with IS-IS*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5310, *IS-IS Generic Cryptographic Authentication*

RFC 6119, *IPv6 Traffic Engineering in IS-IS*

RFC 6213, *IS-IS BFD-Enabled TLV*

RFC 6232, *Purge Originator Identification TLV for IS-IS*

RFC 6233, *IS-IS Registry Extension for Purges*

RFC 6329, *IS-IS Extensions Supporting IEEE 802.1aq Shortest Path Bridging*

RFC 7775, *IS-IS Route Preference for Extended IP and IPv6 Reachability*

RFC 7794, *IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability* – sections 2.1 and 2.3

RFC 7987, *IS-IS Minimum Remaining Lifetime*

RFC 8202, *IS-IS Multi-Instance* – single topology

RFC 8570, *IS-IS Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 8919, *IS-IS Application-Specific Link Attributes*

9.11 Internet Protocol (IP) Fast Reroute (FRR)

draft-ietf-rtgwg-lfa-manageability-08, *Operational management of Loop Free Alternates*

RFC 5286, *Basic Specification for IP Fast Reroute: Loop-Free Alternates*

RFC 7431, *Multicast-Only Fast Reroute*

RFC 7490, *Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)*

RFC 8518, *Selection of Loop-Free Alternates for Multi-Homed Prefixes*

9.12 Internet Protocol (IP) general

draft-grant-tacacs-02, *The TACACS+ Protocol*

RFC 768, *User Datagram Protocol*

RFC 793, *Transmission Control Protocol*

RFC 854, *Telnet Protocol Specifications*

RFC 1350, *The TFTP Protocol (revision 2)*

RFC 2347, *TFTP Option Extension*

RFC 2348, *TFTP Blocksize Option*

RFC 2349, *TFTP Timeout Interval and Transfer Size Options*

RFC 2428, *FTP Extensions for IPv6 and NATs*

RFC 2617, *HTTP Authentication: Basic and Digest Access Authentication*

RFC 2784, *Generic Routing Encapsulation (GRE)*

RFC 2818, *HTTP Over TLS*

RFC 2890, *Key and Sequence Number Extensions to GRE*

RFC 3164, *The BSD syslog Protocol*

RFC 4250, *The Secure Shell (SSH) Protocol Assigned Numbers*

RFC 4251, *The Secure Shell (SSH) Protocol Architecture*

RFC 4252, *The Secure Shell (SSH) Authentication Protocol* – publickey, password

RFC 4253, *The Secure Shell (SSH) Transport Layer Protocol*

RFC 4254, *The Secure Shell (SSH) Connection Protocol*

RFC 4511, *Lightweight Directory Access Protocol (LDAP): The Protocol*

RFC 4513, *Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms* – TLS

RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*

RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*

RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2* – TLS client, RSA public key

RFC 5425, *Transport Layer Security (TLS) Transport Mapping for Syslog* – RFC 3164 with TLS

RFC 5656, *Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer* – ECDSA

RFC 5925, *The TCP Authentication Option*

RFC 5926, *Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)*

RFC 6398, *IP Router Alert Considerations and Usage* – MLD

RFC 6528, *Defending against Sequence Number Attacks*

RFC 7011, *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*

RFC 7012, *Information Model for IP Flow Information Export*

RFC 7230, *Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing*

RFC 7231, *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*

RFC 7232, *Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests*

RFC 7301, *Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension*

RFC 7616, *HTTP Digest Access Authentication*

RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*

9.13 Internet Protocol (IP) multicast

cisco-ipmulticast/pim-autorp-spec01, *Auto-RP: Automatic discovery of Group-to-RP mappings for IP multicast* – version 1

draft-ietf-bier-pim-signaling-08, *PIM Signaling Through BIER Core*

draft-ietf-idmr-traceroute-ipm-07, *A "traceroute" facility for IP Multicast*

draft-ietf-l2vpn-vpls-pim-snooping-07, *Protocol Independent Multicast (PIM) over Virtual Private LAN Service (VPLS)*

RFC 1112, *Host Extensions for IP Multicasting*

RFC 2236, *Internet Group Management Protocol, Version 2*

RFC 2365, *Administratively Scoped IP Multicast*

RFC 2375, *IPv6 Multicast Address Assignments*

RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*

RFC 3306, *Unicast-Prefix-based IPv6 Multicast Addresses*

RFC 3376, *Internet Group Management Protocol, Version 3*

RFC 3446, *Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*

RFC 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*

RFC 3618, *Multicast Source Discovery Protocol (MSDP)*

RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*

RFC 3956, *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*

RFC 3973, *Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)* – auto-RP groups

RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*

RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*

RFC 4607, *Source-Specific Multicast for IP*

RFC 4608, *Source-Specific Protocol Independent Multicast in 232/8*

RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*

RFC 4611, *Multicast Source Discovery Protocol (MSDP) Deployment Scenarios*

RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*

RFC 5186, *Internet Group Management Protocol Version 3 (IGMPv3) / Multicast Listener Discovery Version 2 (MLDv2) and Multicast Routing Protocol Interaction*

RFC 5384, *The Protocol Independent Multicast (PIM) Join Attribute Format*

RFC 5496, *The Reverse Path Forwarding (RPF) Vector TLV*

RFC 6037, *Cisco Systems' Solution for Multicast in MPLS/BGP IP VPNs*

RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*

RFC 6513, *Multicast in MPLS/BGP IP VPNs*

RFC 6514, *BGP Encodings and Procedures for Multicast in MPLS/IP VPNs*

RFC 6515, *IPv4 and IPv6 Infrastructure Addresses in BGP Updates for Multicast VPNs*

RFC 6516, *IPv6 Multicast VPN (MVPN) Support Using PIM Control Plane and Selective Provider Multicast Service Interface (S-PMSI) Join Messages*

RFC 6625, *Wildcards in Multicast VPN Auto-Discover Routes*

RFC 6826, *Multipoint LDP In-Band Signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Path*

RFC 7246, *Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context*

RFC 7385, *IANA Registry for P-Multicast Service Interface (PMSI) Tunnel Type Code Points*

RFC 7716, *Global Table Multicast with BGP Multicast VPN (BGP-MVPN) Procedures*

RFC 7761, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*

RFC 8279, *Multicast Using Bit Index Explicit Replication (BIER)*

RFC 8296, *Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks* – MPLS encapsulation

RFC 8401, *Bit Index Explicit Replication (BIER) Support via IS-IS*

RFC 8444, *OSPFv2 Extensions for Bit Index Explicit Replication (BIER)*

RFC 8487, *Mtrace Version 2: Traceroute Facility for IP Multicast*

RFC 8534, *Explicit Tracking with Wildcard Routes in Multicast VPN* – (C-*,C-*) wildcard

RFC 8556, *Multicast VPN Using Bit Index Explicit Replication (BIER)*

9.14 Internet Protocol (IP) version 4

RFC 791, *Internet Protocol*

RFC 792, *Internet Control Message Protocol*

RFC 826, *An Ethernet Address Resolution Protocol*

RFC 951, *Bootstrap Protocol (BOOTP)* – relay

RFC 1034, *Domain Names - Concepts and Facilities*

RFC 1035, *Domain Names - Implementation and Specification*

RFC 1191, *Path MTU Discovery* – router specification

RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*

RFC 1534, *Interoperation between DHCP and BOOTP*

RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*

RFC 1812, *Requirements for IPv4 Routers*

RFC 1918, *Address Allocation for Private Internets*

RFC 2003, *IP Encapsulation within IP*

RFC 2131, *Dynamic Host Configuration Protocol*

RFC 2132, *DHCP Options and BOOTP Vendor Extensions*

RFC 2401, *Security Architecture for Internet Protocol*

RFC 3021, *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*

RFC 3046, *DHCP Relay Agent Information Option (Option 82)*

RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*

RFC 4884, *Extended ICMP to Support Multi-Part Messages* – ICMPv4 and ICMPv6 Time Exceeded

9.15 Internet Protocol (IP) version 6

RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*

RFC 2529, *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*

RFC 3122, *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*

RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3587, *IPv6 Global Unicast Address Format*

RFC 3596, *DNS Extensions to Support IP version 6*

RFC 3633, *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*

RFC 3646, *DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*

RFC 3971, *SEcure Neighbor Discovery (SEND)*

RFC 3972, *Cryptographically Generated Addresses (CGA)*

RFC 4007, *IPv6 Scoped Address Architecture*

RFC 4193, *Unique Local IPv6 Unicast Addresses*

RFC 4291, *Internet Protocol Version 6 (IPv6) Addressing Architecture*

RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*

RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*

RFC 4862, *IPv6 Stateless Address Autoconfiguration* – router functions

RFC 4890, *Recommendations for Filtering ICMPv6 Messages in Firewalls*

RFC 4941, *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*

RFC 5007, *DHCPv6 Leasequery*

RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6*

RFC 5722, *Handling of Overlapping IPv6 Fragments*

RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6* – IPv6

RFC 5952, *A Recommendation for IPv6 Address Text Representation*

RFC 6092, *Recommended Simple Security Capabilities in Customer Premises Equipment (CPE) for Providing Residential IPv6 Internet Service* – Internet Control and Management, Upper-Layer Transport Protocols, UDP Filters, IPsec and Internet Key Exchange (IKE), TCP Filters

RFC 6106, *IPv6 Router Advertisement Options for DNS Configuration*

RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*

RFC 6437, *IPv6 Flow Label Specification*

RFC 6603, *Prefix Exclude Option for DHCPv6-based Prefix Delegation*

RFC 8021, *Generation of IPv6 Atomic Fragments Considered Harmful*

RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*

RFC 8201, *Path MTU Discovery for IP version 6*

9.16 Internet Protocol Security (IPsec)

draft-ietf-ipsec-isakmp-mode-cfg-05, *The ISAKMP Configuration Method*

draft-ietf-ipsec-isakmp-xauth-06, *Extended Authentication within ISAKMP/Oakley (XAUTH)*

RFC 2401, *Security Architecture for the Internet Protocol*

RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*

RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH*

RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*

RFC 2406, *IP Encapsulating Security Payload (ESP)*

RFC 2407, *IPsec Domain of Interpretation for ISAKMP (IPsec DoI)*

RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*

RFC 2409, *The Internet Key Exchange (IKE)*

RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*

RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman group for Internet Key Exchange (IKE)*

RFC 3566, *The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec*

RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*

RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*

RFC 3947, *Negotiation of NAT-Traversal in the IKE*

RFC 3948, *UDP Encapsulation of IPsec ESP Packets*

RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec ESP*

RFC 4109, *Algorithms for Internet Key Exchange version 1 (IKEv1)*

RFC 4301, *Security Architecture for the Internet Protocol*

RFC 4303, *IP Encapsulating Security Payload*

RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*

RFC 4308, *Cryptographic Suites for IPsec*

RFC 4434, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*

RFC 4543, *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*

RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*

RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*

RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*

RFC 4945, *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2 and PKIX*

RFC 5019, *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*

RFC 5282, *Using Authenticated Encryption Algorithms with the Encrypted Payload of the IKEv2 Protocol*

RFC 5903, *ECP Groups for IKE and IKEv2*

RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*

RFC 5998, *An Extension for EAP-Only Authentication in IKEv2*

RFC 6379, *Suite B Cryptographic Suites for IPsec*

RFC 6380, *Suite B Profile for Internet Protocol Security (IPsec)*

RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*

RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*

RFC 7321, *Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)*

RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*

RFC 7427, *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*

9.17 Label Distribution Protocol (LDP)

draft-pdutta-mpls-ldp-adj-capability-00, *LDP Adjacency Capabilities*

draft-pdutta-mpls-ldp-v2-00, *LDP Version 2*

draft-pdutta-mpls-mldp-up-redundancy-00, *Upstream LSR Redundancy for Multi-point LDP Tunnels*

draft-pdutta-mpls-multi-ldp-instance-00, *Multiple LDP Instances*

draft-pdutta-mpls-tldp-hello-reduce-04, *Targeted LDP Hello Reduction*

RFC 3037, *LDP Applicability*

RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol* – helper mode

RFC 5036, *LDP Specification*

RFC 5283, *LDP Extension for Inter-Area Label Switched Paths (LSPs)*

RFC 5443, *LDP IGP Synchronization*

RFC 5561, *LDP Capabilities*

RFC 5919, *Signaling LDP Label Advertisement Completion*

RFC 6388, *Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*

RFC 6512, *Using Multipoint LDP When the Backbone Has No Route to the Root*

RFC 6826, *Multipoint LDP in-band signaling for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths*

RFC 7032, *LDP Downstream-on-Demand in Seamless MPLS*

RFC 7473, *Controlling State Advertisements of Non-negotiated LDP Applications*

RFC 7552, *Updates to LDP for IPv6*

9.18 Layer Two Tunneling Protocol (L2TP) Network Server (LNS)

draft-mammoliti-l2tp-accessline-avp-04, *Layer 2 Tunneling Protocol (L2TP) Access Line Information Attribute Value Pair (AVP) Extensions*

RFC 2661, *Layer Two Tunneling Protocol "L2TP"*

RFC 2809, *Implementation of L2TP Compulsory Tunneling via RADIUS*

RFC 3438, *Layer Two Tunneling Protocol (L2TP) Internet Assigned Numbers: Internet Assigned Numbers Authority (IANA) Considerations Update*

RFC 3931, *Layer Two Tunneling Protocol - Version 3 (L2TPv3)*

RFC 4719, *Transport of Ethernet Frames over Layer 2 Tunneling Protocol Version 3 (L2TPv3)*

RFC 4951, *Fail Over Extensions for Layer 2 Tunneling Protocol (L2TP) "failover"*

9.19 Multiprotocol Label Switching (MPLS)

draft-ietf-mpls-lsp-ping-ospfv3-codepoint-02, *OSPFv3 CodePoint for MPLS LSP Ping*

RFC 3031, *Multiprotocol Label Switching Architecture*

RFC 3032, *MPLS Label Stack Encoding*

RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services* – E-LSP

RFC 3443, *Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*

RFC 4023, *Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)*

RFC 4182, *Removing a Restriction on the use of MPLS Explicit NULL*

RFC 5332, *MPLS Multicast Encapsulations*

RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*

RFC 6374, *Packet Loss and Delay Measurement for MPLS Networks* – Delay Measurement, Channel Type 0x000C

RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*

RFC 6425, *Detecting Data Plane Failures in Point-to-Multipoint Multiprotocol Label Switching (MPLS) - Extensions to LSP Ping*

RFC 6790, *The Use of Entropy Labels in MPLS Forwarding*

RFC 7308, *Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)*

RFC 7510, *Encapsulating MPLS in UDP*

RFC 7746, *Label Switched Path (LSP) Self-Ping*

RFC 7876, *UDP Return Path for Packet Loss and Delay Measurement for MPLS Networks* – Delay Measurement

RFC 8029, *Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures*

9.20 Multiprotocol Label Switching - Transport Profile (MPLS-TP)

RFC 5586, *MPLS Generic Associated Channel*

RFC 5921, *A Framework for MPLS in Transport Networks*

RFC 5960, *MPLS Transport Profile Data Plane Architecture*

RFC 6370, *MPLS Transport Profile (MPLS-TP) Identifiers*

RFC 6378, *MPLS Transport Profile (MPLS-TP) Linear Protection*

RFC 6426, *MPLS On-Demand Connectivity and Route Tracing*

RFC 6427, *MPLS Fault Management Operations, Administration, and Maintenance (OAM)*

RFC 6428, *Proactive Connectivity Verification, Continuity Check and Remote Defect indication for MPLS Transport Profile*

RFC 6478, *Pseudowire Status for Static Pseudowires*

RFC 7213, *MPLS Transport Profile (MPLS-TP) Next-Hop Ethernet Addressing*

9.21 Network Address Translation (NAT)

draft-ietf-behave-address-format-10, *IPv6 Addressing of IPv4/IPv6 Translators*

draft-ietf-behave-v6v4-xlate-23, *IP/ICMP Translation Algorithm*

draft-miles-behave-l2nat-00, *Layer2-Aware NAT*

draft-nishitani-cgn-02, *Common Functions of Large Scale NAT (LSN)*

RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

RFC 5382, *NAT Behavioral Requirements for TCP*

RFC 5508, *NAT Behavioral Requirements for ICMP*

RFC 6146, *Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers*

RFC 6333, *Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion*

RFC 6334, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite*

RFC 6887, *Port Control Protocol (PCP)*

RFC 6888, *Common Requirements For Carrier-Grade NATs (CGNs)*

RFC 7753, *Port Control Protocol (PCP) Extension for Port-Set Allocation*

RFC 7915, *IP/ICMP Translation Algorithm*

9.22 Network Configuration Protocol (NETCONF)

RFC 5277, *NETCONF Event Notifications*

RFC 6020, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*

RFC 6022, *YANG Module for NETCONF Monitoring*

RFC 6241, *Network Configuration Protocol (NETCONF)*

RFC 6242, *Using the NETCONF Protocol over Secure Shell (SSH)*

RFC 6243, *With-defaults Capability for NETCONF*

RFC 8342, *Network Management Datastore Architecture (NMDA)* – Startup, Candidate, Running and Intended datastores

RFC 8525, *YANG Library*

RFC 8526, *NETCONF Extensions to Support the Network Management Datastore Architecture* – <get-data> operation

9.23 Open Shortest Path First (OSPF)

RFC 1765, *OSPF Database Overflow*

RFC 2328, *OSPF Version 2*

RFC 3101, *The OSPF Not-So-Stubby Area (NSSA) Option*

RFC 3509, *Alternative Implementations of OSPF Area Border Routers*

RFC 3623, *Graceful OSPF Restart Graceful OSPF Restart* – helper mode

RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*

RFC 4222, *Prioritized Treatment of Specific OSPF Version 2 Packets and Congestion Avoidance*

RFC 4552, *Authentication/Confidentiality for OSPFv3*

RFC 4576, *Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4577, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 5185, *OSPF Multi-Area Adjacency*

RFC 5187, *OSPFv3 Graceful Restart* – helper mode

RFC 5243, *OSPF Database Exchange Summary List Optimization*

RFC 5250, *The OSPF Opaque LSA Option*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5340, *OSPF for IPv6*

RFC 5642, *Dynamic Hostname Exchange Mechanism for OSPF*

RFC 5709, *OSPFv2 HMAC-SHA Cryptographic Authentication*

RFC 5838, *Support of Address Families in OSPFv3*

RFC 6549, *OSPFv2 Multi-Instance Extensions*

RFC 6987, *OSPF Stub Router Advertisement*

RFC 7471, *OSPF Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE

RFC 7684, *OSPFv2 Prefix/Link Attribute Advertisement*

RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*

RFC 8362, *OSPFv3 Link State Advertisement (LSA) Extensibility*

RFC 8920, *OSPF Application-Specific Link Attributes*

9.24 OpenFlow

TS-007 Version 1.3.1, *OpenFlow Switch Specification* – OpenFlow-hybrid switches

9.25 Path Computation Element Protocol (PCEP)

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks.* – MPLS binding SIDs

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-dhs-spring-pce-sr-p2mp-policy-00, *PCEP extensions for p2mp sr policy*

RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)*

RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*

RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*

RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*

RFC 8408, *Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages*

RFC 8664, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*

9.26 Point-to-Point Protocol (PPP)

RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP)*

RFC 1990, *The PPP Multilink Protocol (MP)*

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*

RFC 2516, *A Method for Transmitting PPP Over Ethernet (PPPoE)*

RFC 4638, *Accommodating a Maximum Transit Unit/Maximum Receive Unit (MTU/MRU) Greater Than 1492 in the Point-to-Point Protocol over Ethernet (PPPoE)*

RFC 5072, *IP Version 6 over PPP*

9.27 Policy management and credit control

3GPP TS 29.212 Release 11, *Policy and Charging Control (PCC); Reference points* – Gx support as it applies to wireline environment (BNG)

RFC 4006, *Diameter Credit-Control Application*

RFC 6733, *Diameter Base Protocol*

9.28 Pseudowire (PW)

draft-ietf-l2vpn-vpws-iw-oam-04, *OAM Procedures for VPWS Interworking*

MFA Forum 12.0.0, *Multiservice Interworking - Ethernet over MPLS*

MFA Forum 13.0.0, *Fault Management for Multiservice Interworking v1.0*

MFA Forum 16.0.0, *Multiservice Interworking - IP over MPLS*

RFC 3916, *Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*

RFC 3985, *Pseudo Wire Emulation Edge-to-Edge (PWE3)*

RFC 4385, *Pseudo Wire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*

RFC 4446, *IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)*

RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*

RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*

RFC 5085, *Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires*

RFC 5659, *An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge*

RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*

RFC 6073, *Segmented Pseudowire*

RFC 6310, *Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping*

RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*

RFC 6575, *Address Resolution Protocol (ARP) Mediation for IP Interworking of Layer 2 VPNs*

RFC 6718, *Pseudowire Redundancy*

RFC 6829, *Label Switched Path (LSP) Ping for Pseudowire Forwarding Equivalence Classes (FECs) Advertised over IPv6*

RFC 6870, *Pseudowire Preferential Forwarding Status bit*

RFC 7023, *MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking*

RFC 7267, *Dynamic Placement of Multi-Segment Pseudowires*

RFC 7392, *Explicit Path Routing for Dynamic Multi-Segment Pseudowires* – ER-TLV and ER-HOP IPv4 Prefix

RFC 8395, *Extensions to BGP-Signaled Pseudowires to Support Flow-Aware Transport Labels*

9.29 Quality of Service (QoS)

RFC 2430, *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*

RFC 2597, *Assured Forwarding PHB Group*

RFC 3140, *Per Hop Behavior Identification Codes*

RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*

9.30 Remote Authentication Dial In User Service (RADIUS)

draft-oscca-cfrg-sm3-02, *The SM3 Cryptographic Hash Function*

RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*

RFC 2866, *RADIUS Accounting*

RFC 2867, *RADIUS Accounting Modifications for Tunnel Protocol Support*

RFC 2868, *RADIUS Attributes for Tunnel Protocol Support*

RFC 2869, *RADIUS Extensions*

RFC 3162, *RADIUS and IPv6*

RFC 4818, *RADIUS Delegated-IPv6-Prefix Attribute*

RFC 5176, *Dynamic Authorization Extensions to RADIUS*

RFC 6613, *RADIUS over TCP* – with TLS

RFC 6614, *Transport Layer Security (TLS) Encryption for RADIUS*

RFC 6929, *Remote Authentication Dial-In User Service (RADIUS) Protocol Extensions*

RFC 6911, *RADIUS attributes for IPv6 Access Networks*

9.31 Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

draft-newton-mpls-te-dynamic-overbooking-00, *A Diffserv-TE Implementation Model to dynamically change booking factors during failure events*

RFC 2702, *Requirements for Traffic Engineering over MPLS*

RFC 2747, *RSVP Cryptographic Authentication*

RFC 2961, *RSVP Refresh Overhead Reduction Extensions*

RFC 3097, *RSVP Cryptographic Authentication -- Updated Message Type Value*

RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*

RFC 3477, *Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)*

RFC 3564, *Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering*

RFC 3906, *Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels*

RFC 4090, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*

RFC 4124, *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*

RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*

RFC 4875, *Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)*

RFC 4950, *ICMP Extensions for Multiprotocol Label Switching*

RFC 5712, *MPLS Traffic Engineering Soft Preemption*

RFC 5817, *Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks*

9.32 Routing Information Protocol (RIP)

RFC 1058, *Routing Information Protocol*

RFC 2080, *RIPng for IPv6*

RFC 2082, *RIP-2 MD5 Authentication*

RFC 2453, *RIP Version 2*

9.33 Segment Routing (SR)

draft-bashandy-rtgwg-segment-routing-uloop-06, *Loop avoidance using Segment Routing*

draft-filsfils-spring-net-pgm-extension-srv6-usid-13, *Network Programming extension: SRv6 uSID instruction*

draft-filsfils-spring-srv6-net-pgm-insertion-04, *SRv6 NET-PGM extension: Insertion*

draft-ietf-6man-spring-srv6-oam-10, *Operations, Administration, and Maintenance (OAM) in Segment Routing Networks with IPv6 Data plane (SRv6)*

draft-ietf-idr-bgp-ls-segment-routing-ext-16, *BGP Link-State extensions for Segment Routing*

draft-ietf-idr-bgpls-srv6-ext-13, *BGP Link State Extensions for SRv6*

draft-ietf-idr-segment-routing-te-policy-11, *Advertising Segment Routing Policies in BGP*

draft-ietf-isis-mpls-elc-10, *Signaling Entropy Label Capability and Entropy Readable Label Depth Using IS-IS* – advertising ELC

draft-ietf-lsr-flex-algo-16, *IGP Flexible Algorithm*

draft-ietf-lsr-isis-srv6-extensions-14, *IS-IS Extension to Support Segment Routing over IPv6 Dataplane*

draft-ietf-ospf-mpls-elc-12, *Signaling Entropy Label Capability and Entropy Readable Label-stack Depth Using OSPF* – advertising ELC

draft-ietf-rtgwg-segment-routing-ti-lfa-01, *Topology Independent Fast Reroute using Segment Routing*

draft-ietf-spring-conflict-resolution-05, *Segment Routing MPLS Conflict Resolution*

draft-ietf-spring-segment-routing-policy-08, *Segment Routing Policy Architecture*

draft-ietf-teas-sr-rsvp-coexistence-rec-02, *Recommendations for RSVP-TE and Segment Routing LSP co-existence*

draft-voyer-6man-extension-header-insertion-10, *Deployments With Insertion of IPv6 Segment Routing Headers*

draft-voyer-pim-sr-p2mp-policy-02, *Segment Routing Point-to-Multipoint Policy*

draft-voyer-spring-sr-p2mp-policy-03, *SR Replication Policy for P2MP Service Delivery*

RFC 8287, *Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes*

RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF* – node MSD

RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS* – node MSD

RFC 8660, *Segment Routing with the MPLS Data Plane*

RFC 8661, *Segment Routing MPLS Interworking with LDP*

RFC 8663, *MPLS Segment Routing over IP* – BGP SR with SR-MPLS-over-UDP/IP

RFC 8665, *OSPF Extensions for Segment Routing*

RFC 8666, *OSPFv3 Extensions for Segment Routing*

RFC 8667, *IS-IS Extensions for Segment Routing*

RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*

RFC 8754, *IPv6 Segment Routing Header (SRH)*

RFC 8814, *Signaling Maximum SID Depth (MSD) Using the Border Gateway Protocol - Link State*

RFC 8986, *Segment Routing over IPv6 (SRv6) Network Programming*

RFC 9252, *BGP Overlay Services Based on Segment Routing over IPv6 (SRv6)*

9.34 Simple Network Management Protocol (SNMP)

draft-blumenthal-aes-usm-04, *The AES Cipher Algorithm in the SNMP's User-based Security Model* – CFB128-AES-192 and CFB128-AES-256

draft-ietf-isis-wg-mib-06, *Management Information Base for Intermediate System to Intermediate System (IS-IS)*

draft-ietf-mboned-msdp-mib-01, *Multicast Source Discovery protocol MIB*

draft-ietf-mpls-ldp-mib-07, *Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)*

draft-ietf-mpls-lsr-mib-06, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base Using SMIv2*

draft-ietf-mpls-te-mib-04, *Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base*

draft-ietf-ospf-mib-update-08, *OSPF Version 2 Management Information Base*

draft-ietf-vrrp-unified-mib-06, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6* – IPv6

ESO-CONSORTIUM-MIB revision 200406230000Z, *esoConsortiumMIB*

IANA-ADDRESS-FAMILY-NUMBERS-MIB revision 200203140000Z, *ianaAddressFamilyNumbers*

IANAifType-MIB revision 200505270000Z, *ianaifType*

IANA-RTPROTO-MIB revision 200009260000Z, *ianaRtProtoMIB*

IEEE8021-CFM-MIB revision 200706100000Z, *ieee8021CfmMib*

IEEE8021-PAE-MIB revision 200101160000Z, *ieee8021paeMIB*

IEEE8023-LAG-MIB revision 200006270000Z, *lagMIB*

LLDP-MIB revision 200505060000Z, *lldpMIB*

RFC 1157, *A Simple Network Management Protocol (SNMP)*

RFC 1212, *Concise MIB Definitions*

RFC 1215, *A Convention for Defining Traps for use with the SNMP*

RFC 1724, *RIP Version 2 MIB Extension*

RFC 1901, *Introduction to Community-based SNMPv2*

RFC 2021, *Remote Network Monitoring Management Information Base Version 2 using SMIv2*

RFC 2206, *RSVP Management Information Base using SMIv2*

RFC 2213, *Integrated Services Management Information Base using SMIv2*

RFC 2494, *Definitions of Managed Objects for the DS0 and DS0 Bundle Interface Type*

RFC 2578, *Structure of Management Information Version 2 (SMIv2)*

RFC 2579, *Textual Conventions for SMIv2*

RFC 2580, *Conformance Statements for SMIv2*

RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*

RFC 2819, *Remote Network Monitoring Management Information Base*

RFC 2856, *Textual Conventions for Additional High Capacity Data Types*

RFC 2863, *The Interfaces Group MIB*

RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*

RFC 2933, *Internet Group Management Protocol MIB*

RFC 3014, *Notification Log MIB*

RFC 3165, *Definitions of Managed Objects for the Delegation of Management Scripts*

RFC 3231, *Definitions of Managed Objects for Scheduling Management Operations*

RFC 3273, *Remote Network Monitoring Management Information Base for High Capacity Networks*

RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework*

RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*

RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*

RFC 3413, *Simple Network Management Protocol (SNMP) Applications*

RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*

RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP)* – SNMP over UDP over IPv4

RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

RFC 3419, *Textual Conventions for Transport Addresses*

RFC 3498, *Definitions of Managed Objects for Synchronous Optical Network (SONET) Linear Automatic Protection Switching (APS) Architectures*

RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

RFC 3592, *Definitions of Managed Objects for the Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) Interface Type*

RFC 3593, *Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals*

RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*

RFC 3637, *Definitions of Managed Objects for the Ethernet WAN Interface Sublayer*

RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

RFC 3895, *Definitions of Managed Objects for the DS1, E1, DS2, and E2 Interface Types*

RFC 3896, *Definitions of Managed Objects for the DS3/E3 Interface Type*

RFC 4001, *Textual Conventions for Internet Network Addresses*

RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*

RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*

RFC 4220, *Traffic Engineering Link Management Information Base*

RFC 4273, *Definitions of Managed Objects for BGP-4*

RFC 4292, *IP Forwarding Table MIB*

RFC 4293, *Management Information Base for the Internet Protocol (IP)*

RFC 4631, *Link Management Protocol (LMP) Management Information Base (MIB)*

RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*

RFC 7420, *Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module*

RFC 7630, *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*

SFLOW-MIB revision 200309240000Z, *sFlowMIB*

9.35 Timing

GR-1244-CORE Issue 3, *Clocks for the Synchronized Network: Common Generic Criteria*

GR-253-CORE Issue 3, *SONET Transport Systems: Common Generic Criteria*

IEEE 1588-2008, *IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems*

ITU-T G.781, *Synchronization layer functions*

ITU-T G.813, *Timing characteristics of SDH equipment slave clocks (SEC)*

ITU-T G.8261, *Timing and synchronization aspects in packet networks*

ITU-T G.8262, *Timing characteristics of synchronous Ethernet equipment slave clock (EEC)*

ITU-T G.8262.1, *Timing characteristics of an enhanced synchronous Ethernet equipment slave clock (eEEC)*

ITU-T G.8264, *Distribution of timing information through packet networks*

ITU-T G.8265.1, *Precision time protocol telecom profile for frequency synchronization*

ITU-T G.8275.1, *Precision time protocol telecom profile for phase/time synchronization with full timing support from the network*

RFC 3339, *Date and Time on the Internet: Timestamps*

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

9.36 Two-Way Active Measurement Protocol (TWAMP)

RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP)* – server, unauthenticated mode

RFC 5938, *Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)*

RFC 6038, *Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features*

RFC 8545, *Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP)* – TWAMP

RFC 8762, *Simple Two-Way Active Measurement Protocol* – unauthenticated

RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extensions* – unauthenticated

9.37 Virtual Private LAN Service (VPLS)

RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

RFC 5501, *Requirements for Multicast Support in Virtual Private LAN Services*

RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

RFC 7041, *Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging*

RFC 7117, *Multicast in Virtual Private LAN Service (VPLS)*

9.38 Voice and video

DVB BlueBook A86, *Transport of MPEG-2 TS Based DVB Services over IP Based Networks*

ETSI TS 101 329-5 Annex E, *QoS Measurement for VoIP - Method for determining an Equipment Impairment Factor using Passive Monitoring*

ITU-T G.1020 Appendix I, *Performance Parameter Definitions for Quality of Speech and other Voiceband Applications Utilizing IP Networks - Mean Absolute Packet Delay Variation & Markov Models*

ITU-T G.107, *The E Model - A computational model for use in planning*

ITU-T P.564, *Conformance testing for voice over IP transmission quality assessment models*

RFC 3550, *RTP: A Transport Protocol for Real-Time Applications* – Appendix A.8

RFC 4585, *Extended RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/AVPF)*

RFC 4588, *RTP Retransmission Payload Format*

9.39 Wireless Local Area Network (WLAN) gateway

3GPP TS 23.402, *Architecture enhancements for non-3GPP accesses* – S2a roaming based on GPRS

9.40 Yet Another Next Generation (YANG)

RFC 6991, *Common YANG Data Types*

RFC 7950, *The YANG 1.1 Data Modeling Language*

RFC 7951, *JSON Encoding of Data Modeled with YANG*

9.41 Yet Another Next Generation (YANG) OpenConfig Modules

openconfig-aaa.yang version 0.4.0, *OpenConfig AAA Module*

openconfig-aaa-radius.yang version 0.3.0, *OpenConfig AAA RADIUS Module*

openconfig-aaa-tacacs.yang version 0.3.0, *OpenConfig AAA TACACS+ Module*

openconfig-acl.yang version 1.0.0, *OpenConfig ACL Module*

openconfig-bfd.yang version 0.2.2, *OpenConfig BFD Module*

openconfig-bgp.yang version 6.1.0, *OpenConfig BGP Module*

openconfig-bgp-common.yang version 6.0.0, *OpenConfig BGP Common Module*

openconfig-bgp-common-multiprotocol.yang version 6.0.0, *OpenConfig BGP Common Multiprotocol Module*

openconfig-bgp-common-structure.yang version 6.0.0, *OpenConfig BGP Common Structure Module*

openconfig-bgp-global.yang version 6.0.0 *OpenConfig BGP Global Module*

openconfig-bgp-neighbor.yang version 6.1.0, *OpenConfig BGP Neighbor Module*

openconfig-bgp-peer-group.yang version 6.1.0, *OpenConfig BGP Peer Group Module*

openconfig-bgp-policy.yang version 4.0.1, *OpenConfig BGP Policy Module*

openconfig-if-aggregate.yang version 2.0.0, *OpenConfig Interfaces Aggregated Module*

openconfig-if-ethernet.yang version 2.0.0, *OpenConfig Interfaces Ethernet Module*

openconfig-if-ip.yang version 2.0.0, *OpenConfig Interfaces IP Module*

openconfig-if-ip-ext.yang version 2.0.0, *OpenConfig Interfaces IP Extensions Module*

openconfig-igmp.yang version 0.2.0, *OpenConfig IGMP Module*

openconfig-interfaces.yang version 2.0.0, *OpenConfig Interfaces Module*

openconfig-isis.yang version 0.3.2, *OpenConfig IS-IS Module*

openconfig-isis-policy.yang version 0.3.2, *OpenConfig IS-IS Policy Module*

openconfig-isis-routing.yang version 0.3.2, *OpenConfig IS-IS Routing Module*

openconfig-lacp.yang version 1.1.0, *OpenConfig LACP Module*

openconfig-lldp.yang version 0.1.0, *OpenConfig LLDP Module*

openconfig-local-routing.yang version 1.2.0, *OpenConfig Local Routing Module*

openconfig-mpls.yang version 2.3.0, *OpenConfig MPLS Module*

openconfig-mpls-ldp.yang version 3.0.2, *OpenConfig MPLS LDP Module*

openconfig-mpls-rsvp.yang version 2.3.0, *OpenConfig MPLS RSVP Module*

openconfig-mpls-te.yang version 2.3.0, *OpenConfig MPLS TE Module*

openconfig-network-instance.yang version 1.1.0, *OpenConfig Network Instance Module*

openconfig-network-instance-l3.yang version 0.11.1, *OpenConfig L3 Network Instance Module* – static routes

openconfig-packet-match.yang version 1.0.0, *OpenConfig Packet Match Module*

openconfig-pim.yang version 0.2.0 *OpenConfig PIM Module*

openconfig-platform.yang version 0.15.0, *OpenConfig Platform Module*

openconfig-platform-fan.yang version 0.1.1, *OpenConfig Platform Fan Module*

openconfig-platform-linecard.yang version 0.1.2, *OpenConfig Platform Linecard Module*

openconfig-platform-port.yang version 0.4.2, *OpenConfig Port Module*

openconfig-platform-transceiver.yang version 0.9.0, *OpenConfig Transceiver Module*

openconfig-procmon.yang version 0.4.0, *OpenConfig Process Monitoring Module*

openconfig-relay-agent.yang version 0.1.0, *OpenConfig Relay Agent Module*

openconfig-routing-policy.yang version 3.0.0, *OpenConfig Routing Policy Module*

openconfig-rsvp-sr-ext.yang version 0.1.0, *OpenConfig RSVP-TE and SR Extensions Module*

openconfig-system.yang version 0.10.1, *OpenConfig System Module*

openconfig-system-grpc.yang version 1.0.0, *OpenConfig System gRPC Module*

openconfig-system-logging.yang version 0.3.1, *OpenConfig System Logging Module*

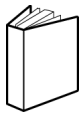
openconfig-system-terminal.yang version 0.3.0, *OpenConfig System Terminal Module*

openconfig-telemetry.yang version 0.5.0, *OpenConfig Telemetry Module*

openconfig-terminal-device.yang version 1.9.0, *OpenConfig Terminal Optics Device Module*

openconfig-vlan.yang version 2.0.0, *OpenConfig VLAN Module*

Customer document and product support



Customer documentation

Customer documentation welcome page

Technical support

Product support portal

Documentation feedback

Customer documentation feedback