Ethernet Virtual Private Networks (EVPNs)

In This Chapter

This chapter provides information about Ethernet Virtual Private Networks (EVPNs), process overview, and implementation notes.

Topics in this chapter include:

- Overview on page 1031
- EVPN for VXLAN Tunnels in a Layer-2 DC GW (EVPN-VXLAN) on page 1032
- EVPN for VXLAN Tunnels in a Layer-2 DC with Integrated Routing Bridging Connectivity on the DC GW on page 1034
- EVPN for VXLAN Tunnels in a Layer 3 DC with Integrated Routing Bridging Connectivity among VPRNs on page 1035
- EVPN for VXLAN Tunnels in a Layer 3 DC with EVPN-Tunnel Connectivity among VPRNs on page 1037
- EVPN for MPLS Tunnels in ELAN Services on page 1039
- EVPN for PBB over MPLS Tunnels (PBB-EVPN) on page 1041
- VXLAN on page 1042
- BGP-EVPN Control Plane for VXLAN Overlay Tunnels on page 1053
- EVPN for VXLAN in VPLS Services on page 1057
- EVPN for VXLAN in R-VPLS Services on page 1062
- EVPN for VXLAN in IRB Backhaul R-VPLS Services and IP Prefixes on page 1064
- EVPN for VXLAN in EVPN Tunnel R-VPLS Services on page 1068
- DC GW integration with the Nuage Virtual Services Directory (VSD) on page 1073
- Fully-Dynamic VSD Integration Model on page 1086
- BGP-EVPN Control Plane for MPLS Tunnels on page 1095

In This Chapter

- EVPN for MPLS Tunnels in VPLS Services (EVPN-MPLS) on page 1100
- BGP-EVPN Control Plane for PBB-EVPN on page 1134
- PBB-EVPN for I-VPLS and PBB Epipe Services on page 1136
- PBB-EVPN Multi-Homing in I-VPLS and PBB Epipe Services on page 1142
- ARP/ND Snooping and Proxy Support on page 1155
- BGP-EVPN MAC-Mobility on page 1161
- BGP-EVPN MAC-Duplication on page 1162
- Conditional Static MAC and Protection on page 1164
- CFM Interaction with EVPN Services on page 1165
- DC GW Policy Based Forwarding/Routing to an EVPN ESI (Ethernet Segment Identifier) on page 1167
- BGP and EVPN Route Selection for EVPN Routes on page 1175
- Interaction of EVPN-VXLAN and EVPN-MPLS with Existing VPLS Features on page 1177
- Interaction of PBB-EVPN with Existing VPLS Features on page 1179
- Interaction of EVPN-VXLAN with Existing VPRN Features on page 1180
- Routing Policies for BGP EVPN IP Prefixes on page 1181

Overview

EVPN is an IETF technology per RFC7432, *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 ELAN services in a similar way to RFC4364 IP-VPNs, while supporting MAC learning within the control plane (distributed by MP-BGP), efficient multi-destination traffic delivery, and active-active multi-homing.

EVPN can be used as the control plane for different data plane encapsulations. Alcatel-Lucent's 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 (DC GW) 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 draft-ietf-bess-evpn-overlay.

- **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 RFC7432.

- **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 ELAN and ELINE services in the WAN.

  The PBB-EVPN functionality is standardized in draft-ietf-l2vpn-pbb-evpn.

The 7x50 SR/ESS/XRS EVPN VXLAN implementation is integrated in the Nuage Data Center architecture, where the 7x50 serves as the DC GW.

Refer to the Nuage Networks Virtualized Service Platform Guide for more information about the Nuage Networks architecture and products. The following sections describe the applications supported by EVPN in the 7x50 implementation.

EVPN for VXLAN Tunnels in a Layer-2 DC GW (EVPN-VXLAN)

Figure 110 shows the use of EVPN for VXLAN overlay tunnels on the 7x50 SR/ESS/XRS when it is used as a Layer-2 DC GW.

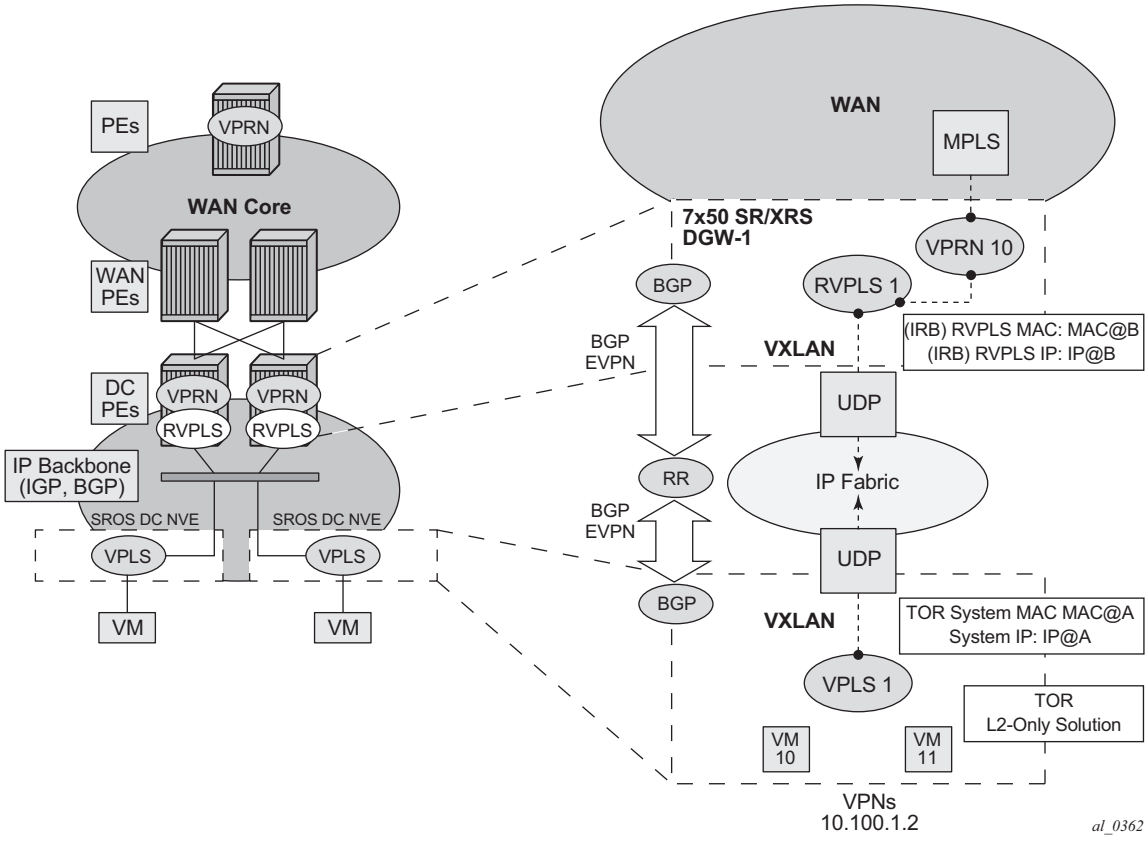

Figure 110: Layer-2 DC PE with VPLS to the WAN

DC providers require a DC GW 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 Alcatel-Lucent DC architecture (Nuage) uses EVPN and VXLAN as the control and data plane solutions for Layer-2 connectivity within the DC and so does the 7x50 SROS.

While EVPN VXLAN will be used within the DC, most service providers use VPLS and H-VPLS as the solution to extend Layer-2 VPN connectivity. Figure 110 shows the Layer-2 DC GW function on the 7x50, providing VXLAN connectivity to the DC and regular VPLS connectivity to the WAN.

The WAN connectivity will be 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 multi-homing resiliency through the use of BGP multi-homing.

EVPN for VXLAN Tunnels in a Layer-2 DC with Integrated Routing Bridging Connectivity on the DC GW

Figure 111 shows the use of EVPN for VXLAN overlay tunnels on the 7x50 SR/ESS/XRS, when the DC provides LAYER-2 connectivity and the DC GW can route the traffic to the WAN through an R-VPLS and linked VPRN.

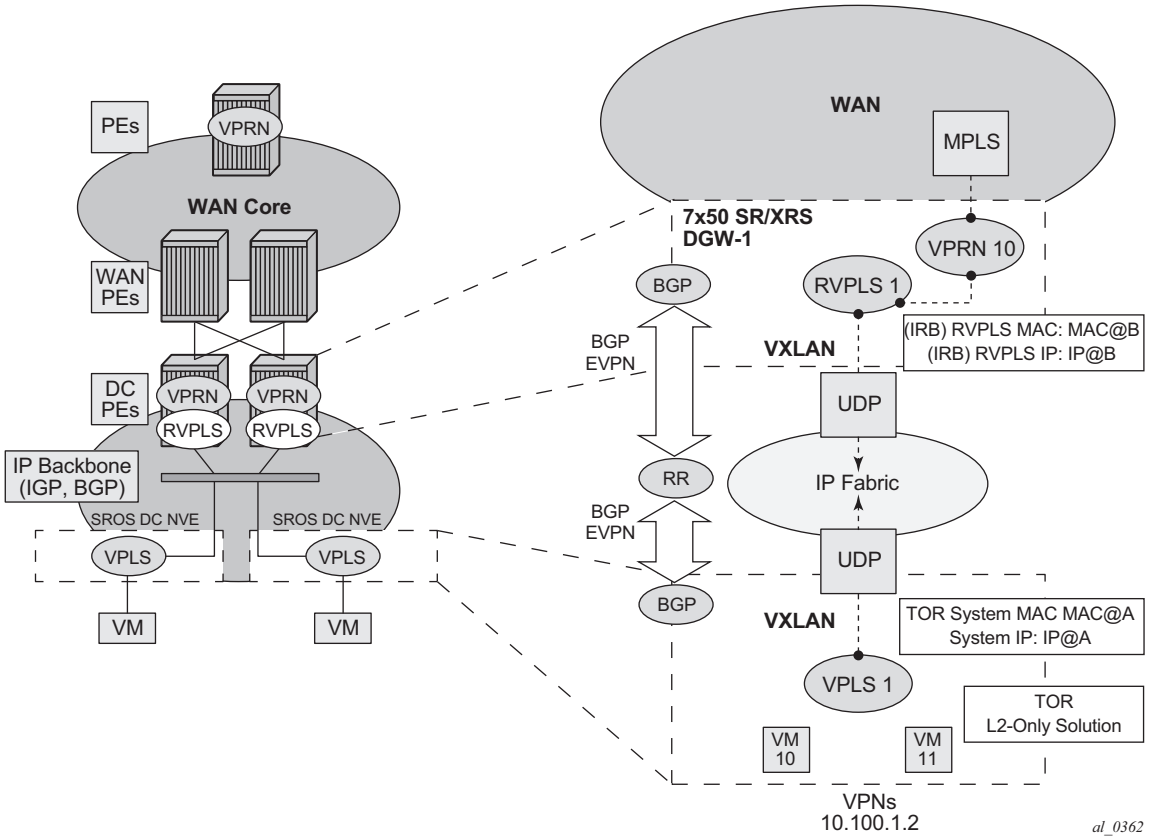

Figure 111: GW IRB on the DC PE for an L2 EVPN/VXLAN DC

In some cases, the DC GW must provide a Layer 3 default gateway function to all the hosts in a specified tenant subnet. In this case, the VXLAN data plane will be terminated in an R-VPLS on the DC GW, and connectivity to the WAN will be accomplished through regular VPRN connectivity. The 7x50 supports IPv4 and IPv6 interfaces as default gateways in this scenario.

EVPN for VXLAN Tunnels in a Layer 3 DC with Integrated Routing Bridging Connectivity among VPRNs

Figure 112 shows the use of EVPN for VXLAN tunnels on the 7x50 SR/ESS/XRS, when the DC provides distributed layer-3 connectivity to the DC tenants.

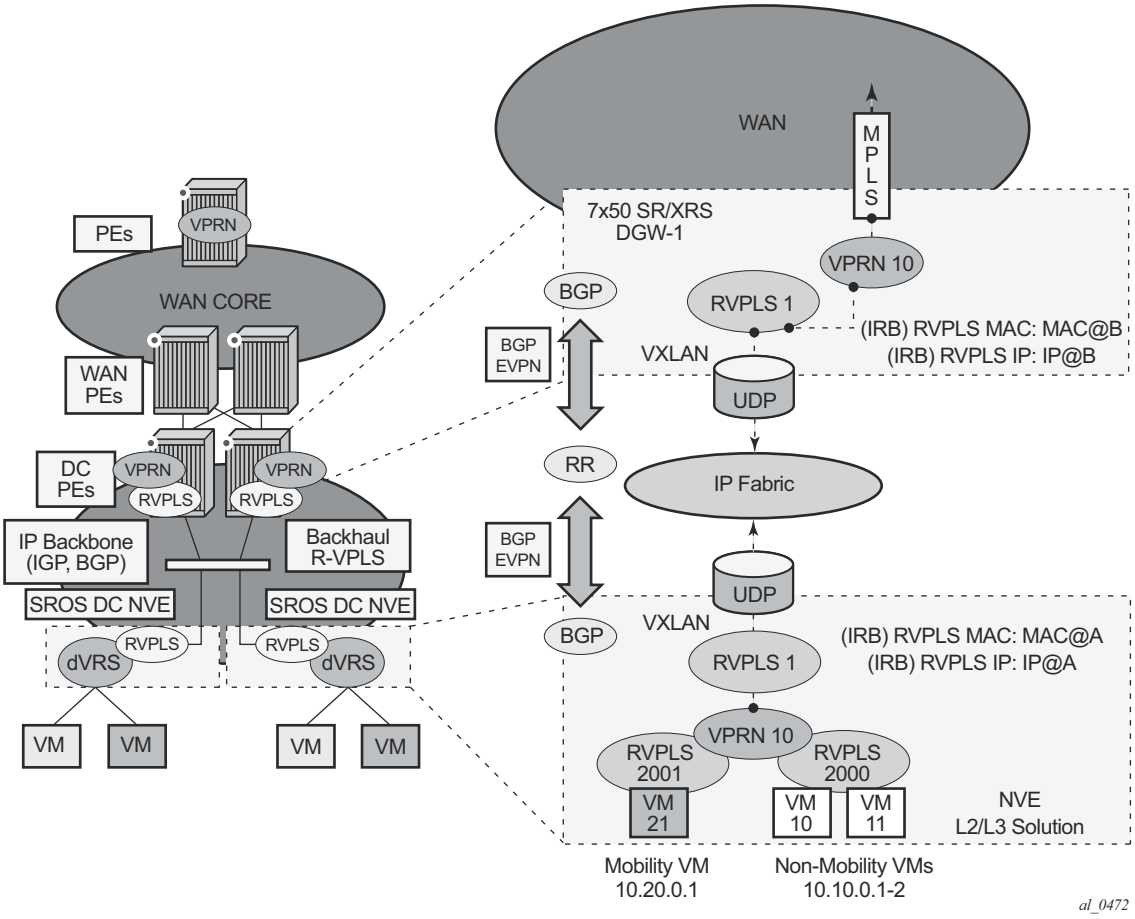

Figure 112: GW IRB on the DC PE for an L3 EVPN/VXLAN DC

Each tenant will have 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 112, 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 L3 connectivity to the IP-VPN in the WAN, a VPRN is defined in the DC GWs, which connects the tenant to the

WAN. That VPRN instance will be 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 L3 connectivity to the WAN without the need for defining all of the subnets in the DC GW.

The 7x50 DC GW supports this 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 EVPN and IP-VPN for IP prefixes is also fully supported.

EVPN for VXLAN Tunnels in a Layer 3 DC with EVPN-Tunnel Connectivity among VPRNs

Figure 113 shows the use of EVPN for VXLAN tunnels on the 7x50 SR/ESS/XRS, when the DC provides distributed layer-3 connectivity to the DC tenants and the VPRN instances are connected through EVPN tunnels.

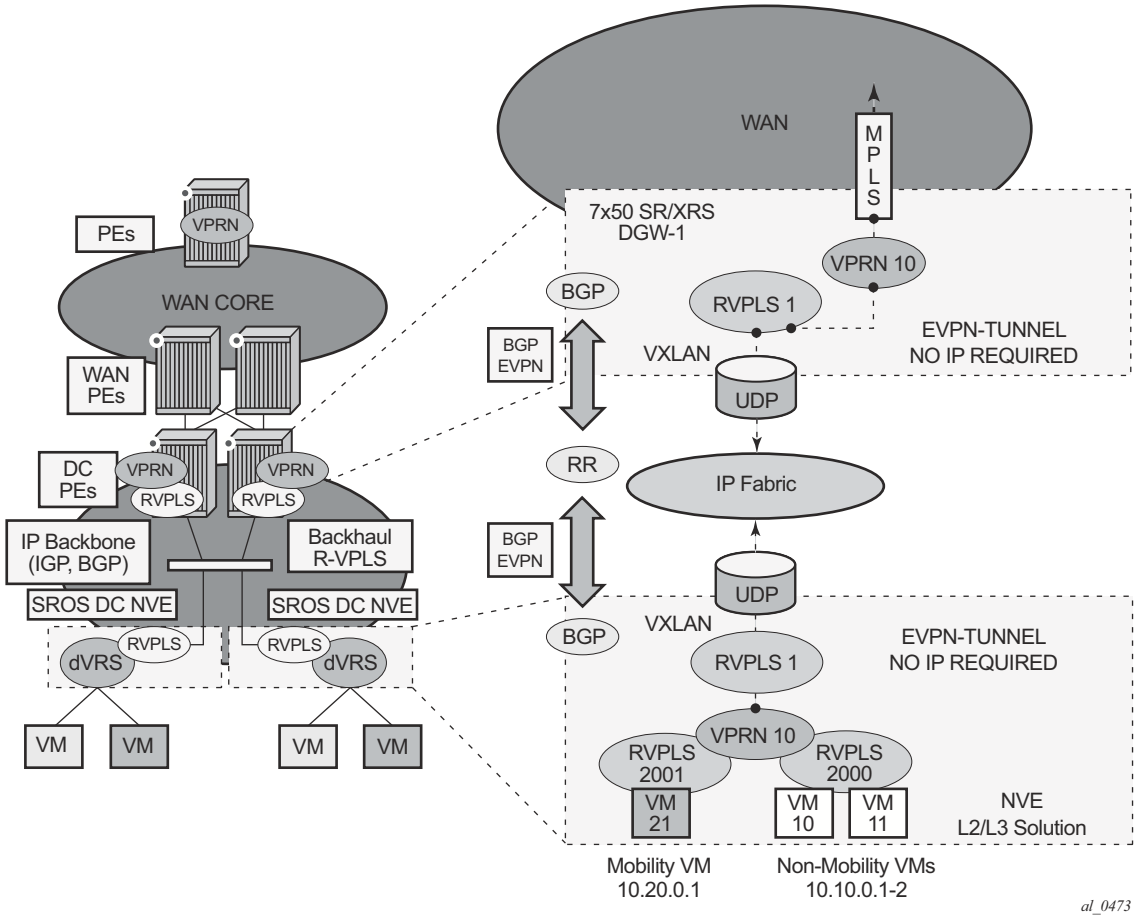

Figure 113: EVPN-Tunnel GW IRB on the DC PE for an L3 EVPN/VXLAN DC

The solution described in section EVPN for VXLAN Tunnels in a Layer 3 DC with Integrated Routing Bridging Connectivity among VPRNs on page 1035 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 7x50.

EVPN for MPLS Tunnels in ELAN Services

Figure 114 shows the use of EVPN for MPLS tunnels on the 7x50 SR/ESS/XRS. In this case, EVPN is used as the control plane for ELAN services in the WAN.

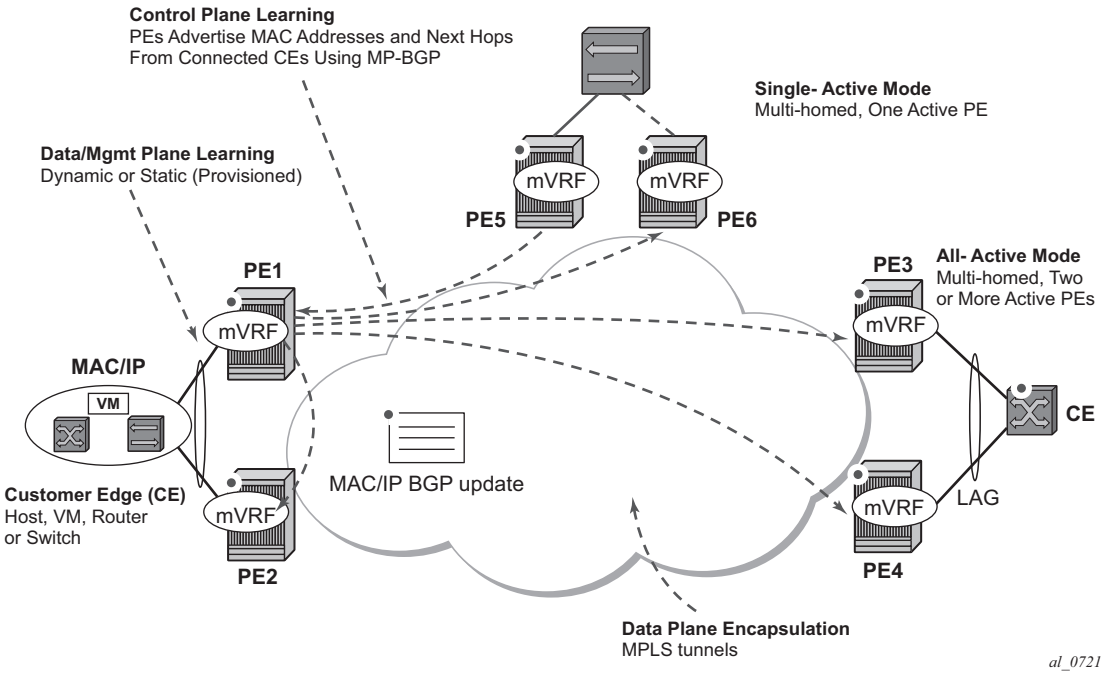

Figure 114: EVPN for MPLS in VPLS Services

EVPN-MPLS is standardized in RFC7432 as an L2VPN technology that can fill the gaps in VPLS for ELAN services. Besides the optimizations introduced by EVPN, a significant number of service providers offering ELAN services today are requesting EVPN for their multi-homing capabilities. EVPN supports all-active multi-homing (per-flow load-balancing multi-homing) in addition to single-active multi-homing (per-service load-balancing multi-homing).

EVPN is a standard-based technology that supports all-active multi-homing; and although VPLS already supports single-active multi-homing, EVPN's single-active multi-homing is also perceived as a superior technology due to its mass-withdrawal capabilities to speed up convergence in scaled environments.

As well as the superior multi-homing capabilities in EVPN, the technology also provides a number of significant benefits, such as:

- An IP-VPN-like operation and control for ELAN 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 7x50 SROS EVPN-MPLS implementation is compliant with RFC7432.

EVPN for PBB over MPLS Tunnels (PBB-EVPN)

Figure 115 shows the use of EVPN for MPLS tunnels on the 7x50 SR/ESS/XRS. In this case, EVPN is used as the control plane for ELAN services in the WAN.

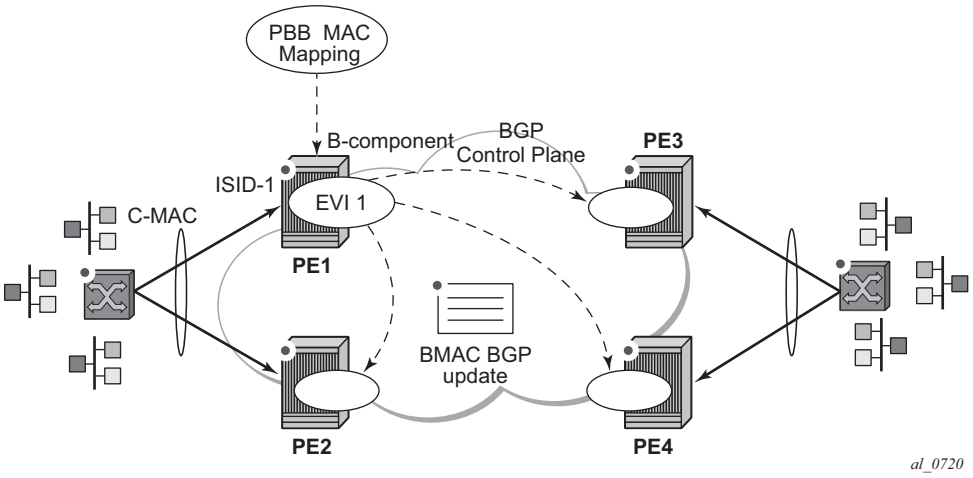

Figure 115: EVPN for PBB over MPLS

EVPN for PBB over MPLS (hereafter called PBB-EVPN) is specified in draft-ietf-l2vpn-pbb-evpn. 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 multi-homing capabilities).

PBB-EVPN is a combination of 802.1ah PBB and RFC7432 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 BMACs and setup per-ISID flooding trees for I-VPLS services. The learning of the CMACs, either on local SAPs/SDP-bindings or associated with remote BMACs, is still performed in the data plane. Only the learning of BMACs in the B-VPLS is performed through BGP.

The 7x50 SROS PBB-EVPN implementation supports PBB-EVPN for I-VPLS and PBB-Epipe services, including single-active and all-active multi-homing.

VXLAN

The 7x50 SROS and Nuage solution for DC supports VXLAN (Virtual eXtensible Local Area Network) overlay tunnels as per RFC7348.

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 116 shows an example of the VXLAN encapsulation supported by the Alcatel-Lucent's implementation.

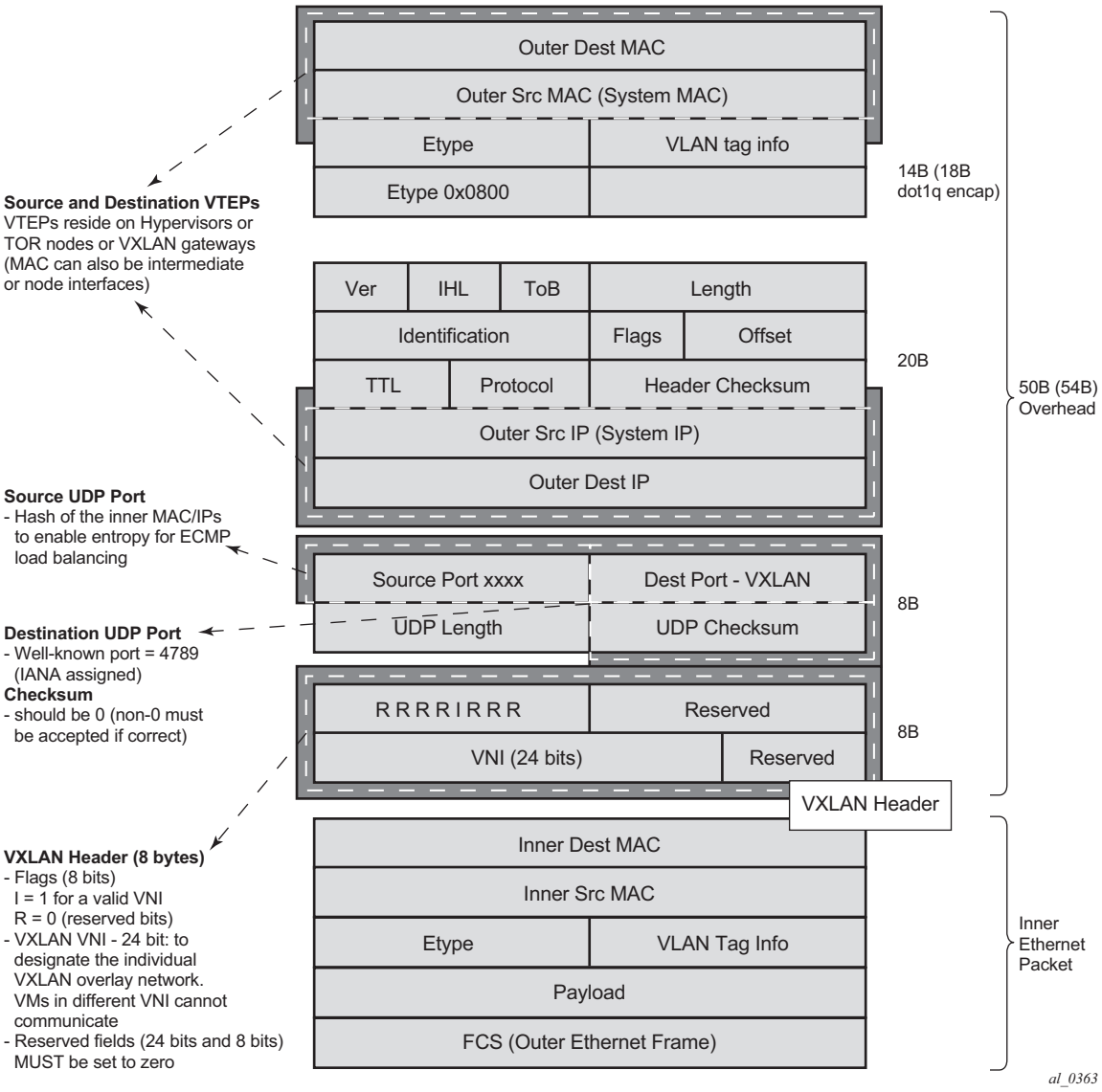

Figure 116: VXLAN Frame Format

As shown in Figure 116, 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 7x50 does at ingress. This will create an "entropy" value that can be used by the core DC nodes for load balancing on ECMP paths.
  - The checksum will be set to zero.
- Outer IP and Ethernet headers (34 or 38 bytes)
  - The source IP and source MAC will identify the source VTEP. That is, these fields will be populated with the PE's system IP and chassis MAC address.
    Note — The source MAC address will be changed on all the IP hops along the path, as is usual in regular IP routing.
  - The destination IP will identify the remote VTEP (remote system IP) and will be the result of the destination MAC lookup in the service Forwarding Database (FDB).
    Note — All remote MACs will be learned by the EVPN BGP and associated with a remote VTEP address and VNI.

Some considerations related to the support of VXLAN on the 7x50 are:

- VXLAN is only supported on network or hybrid ports with null or dot1q encapsulation.
- VXLAN is supported on Ethernet/LAG and POS/APS.
- Only IPv4 unicast addresses are supported as VTEPs.
- Only System IP addresses are supported, as VTEPs, for originating and terminating VXLAN tunnels.

VXLAN ECMP and LAG

The DC GW 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 will be based on the packet contents.
- ECMP for VXLAN is not supported on R-VPLS services.
- In both cases where ECMP is not supported, each VXLAN binding is tied to a single (different) ECMP path, so in a normal deployment with a reasonable number of remote VTEPs, there should be a fair distribution of the traffic across the paths.
- LAG spraying based on the packet hash is supported in all the cases (VPLS unicast, VPLS BUM, and R-VPLS).

VXLAN VPLS Tag Handling

The following describes the behavior on the 7x50 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 7x50 will not add any inner VLAN tag on top of the VXLAN encapsulation, and at the ingress network port, the 7x50 will ignore any VLAN tag received and will consider it as part of the payload.

VXLAN MTU Considerations

For VXLAN VPLS services, the network port MTU must be at least 50 Bytes (54 Bytest 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 will be discarded) and not on VXLAN termination (any VXLAN ingress packet will make it to the egress SAP regardless of the configured service-mtu).

Note—The 7x50 will never fragment or reassemble VXLAN packets. In addition, the 7x50 always sets the DF (Do not Fragment) flag in the VXLAN outer IP header.

VXLAN QoS

VXLAN is a network port encapsulation; therefore, the QoS settings for VXLAN are controlled from the network QoS policies:

- The ingress network QoS policy is used to classify the VXLAN packets based on the outer dot1p (if present, then the DSCP, to yield a FC/profile.
- QoS control of BUM traffic received on VXLAN bindings 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.
- On egress, because the VXLAN adds a new IPv4 header, and the DSCP will be always marked based on the egress network qos policy, there is no need to specify “remarking” in the policy to mark the DSCP.

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 will determine 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 will also encode 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 remainder of the variables are optional.

The VXLAN PDU will be encapsulated in the appropriate transport header and forwarded within the overlay to the appropriate VTEP termination. The VXLAN router alert (RA) bit will be set to prevent forwarding OAM PDU beyond the terminating VTEP. Since 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 will default 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 will not be learned in the local layer 2 forwarding databases.

The echo responder will terminate the VXLAN OAM frame, and will take 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 chose to override this default by changing the **reply-mode** to **overlay**. The overlay return mode will force the responder to use the VTEP connection representing the source IP and source VTEP. If a return overlay is not available, the echo response will be 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 will only validate 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 will select 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 will not be reflected to the source. The operator can change this behavior by including **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 will be 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. In these examples, the service identifier for the VTEP source is 600; the IP Address of the terminating VTEP is 1.1.1.31; the destination VNI on the terminating VTEP is 31.

```
oam vxlan-ping test-id 1 service 600 dest-vni 31 outer-ip-destination 1.1.1.31 interval
0.1 send-count 10
vxlan-ping destination vxlan-id 31 ip-address 1.1.1.31 reply-mode udp interval 0.1s count
```

```
10

! ! ! ! ! ! ! ! ! !
---- vxlan-id 31 ip-address 1.1.1.31 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 = 0.912ms, avg = 1.355ms, max = 2.332ms, stddev = 0.425ms
round-trip-delay min = 0.679ms, avg = 0.949ms, max = 1.587ms, stddev = 0.264ms


oam vxlan-ping test-id 2 service 600 dest-vni 31 outer-ip-destination 1.1.1.31 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 pad-
ding 2000 reflect-pad fc nc profile out

vxlan-ping destination vxlan-id 31 ip-address 1.1.1.31 reply-mode overlay interval 1s
count 20
===============================================================================================
==============================
rc=1 Malformed Echo Request Received, rc=2 Overlay Segment Not Present, rc=3 Overlay Seg-
ment Not Operational, rc=4 Ok
===============================================================================================
==============================

2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=1 ttl=255 rtt-time=0.722ms fwd-
time=0.000ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=2 ttl=255 rtt-time=0.750ms fwd-
time=1.508ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=3 ttl=255 rtt-time=0.974ms fwd-
time=0.588ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=4 ttl=255 rtt-time=1.714ms fwd-
time=0.819ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=5 ttl=255 rtt-time=0.799ms fwd-
time=1.776ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=6 ttl=255 rtt-time=0.892ms fwd-
time=0.000ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=7 ttl=255 rtt-time=0.843ms fwd-
time=1.560ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=8 ttl=255 rtt-time=0.825ms fwd-
time=1.253ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=9 ttl=255 rtt-time=0.958ms fwd-
time=0.000ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=10 ttl=255 rtt-time=0.963ms fwd-
time=1.673ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=11 ttl=255 rtt-time=0.929ms fwd-
time=1.697ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=12 ttl=255 rtt-time=0.973ms fwd-
time=1.362ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=13 ttl=255 rtt-time=0.813ms fwd-
time=0.000ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=14 ttl=255 rtt-time=0.887ms fwd-
time=1.676ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=15 ttl=255 rtt-time=1.119ms fwd-
time=0.000ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=16 ttl=255 rtt-time=1.017ms fwd-
time=1.887ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=17 ttl=255 rtt-time=0.873ms fwd-
```

```
time=1.746ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=18 ttl=255 rtt-time=1.105ms fwd-
time=0.000ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=19 ttl=255 rtt-time=0.909ms fwd-
time=1.484ms. rc=4
2132 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=20 ttl=255 rtt-time=0.906ms fwd-
time=1.849ms. rc=4

---- vxlan-id 31 ip-address 1.1.1.31 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 = 0.000ms, avg = 0.951ms, max = 1.887ms, stddev = 0.887ms
round-trip-delay min = 0.722ms, avg = 0.948ms, max = 1.714ms, stddev = 0.202ms


oam vxlan-ping test-id 1 service 600 dest-vni 31 outer-ip-destination 1.1.1.31 send-count
10 end-system 00:00:00:00:00:01 interval 0.1

vxlan-ping destination vxlan-id 31 ip-address 1.1.1.31 reply-mode udp end-system
00:00:00:00:00:01 interval 0.1s count 10
1 1 1 1 1 1 1 1 1 1
---- vxlan-id 31 ip-address 1.1.1.31 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
   10 end-system present(1), 0 end-system not present(2)
forward-delay min = 0.000ms, avg = 0.000ms, max = 0.316ms, stddev = 0.520ms
round-trip-delay min = 0.704ms, avg = 0.855ms, max = 1.151ms, stddev = 0.121ms


oam vxlan-ping test-id 1 service 600 dest-vni 31 outer-ip-destination 1.1.1.31 send-count
10 end-system 00:00:00:00:00:01

vxlan-ping destination vxlan-id 31 ip-address 1.1.1.31 reply-mode udp end-system
00:00:00:00:00:01 interval 1s count 10
================================================================================
==============================
rc=1 Malformed Echo Request Received, rc=2 Overlay Segment Not Present, rc=3 Overlay Seg-
ment Not Operational, rc=4 Ok
mac=1 End System Present, mac=2 End System Not Present
================================================================================
==============================

92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=1 ttl=255 rtt-time=0.753ms fwd-time=1.240ms.
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=2 ttl=255 rtt-time=0.785ms fwd-time=0.000ms.
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=3 ttl=255 rtt-time=1.425ms fwd-time=2.759ms.
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=4 ttl=255 rtt-time=1.657ms fwd-time=1.659ms.
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=5 ttl=255 rtt-time=0.650ms fwd-time=0.982ms.
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=6 ttl=255 rtt-time=0.894ms fwd-time=0.464ms.
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=7 ttl=255 rtt-time=0.839ms fwd-time=0.581ms.
```

```
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=8 ttl=255 rtt-time=0.714ms fwd-time=0.995ms.
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=9 ttl=255 rtt-time=0.798ms fwd-time=0.881ms.
rc=4 mac=1
92 bytes from vxlan-id 31 1.1.1.31: vxlan_seq=10 ttl=255 rtt-time=0.839ms fwd-
time=1.068ms. rc=4 mac=1

---- vxlan-id 31 ip-address 1.1.1.31 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
   10 end-system present, 0 end-system not present
forward-delay min = 0.000ms, avg = 0.978ms, max = 2.759ms, stddev = 0.865ms
round-trip-delay min = 0.650ms, avg = 0.935ms, max = 1.657ms, stddev = 0.314ms
```

IGMP-Snooping on VXLAN

The delivery of IP Multicast in VXLAN services can be optimized with IGMP-snooping. IGMP-snooping is supported in EVPN-VXLAN VPLS services. When enabled, IGMP reports will be snooped on SAPs/SDP-bindings, but also on VXLAN bindings, to create/modify entries in the MFIB for the VPLS service.

The following must be considered when configuring IGMP-snooping in EVPN-VXLAN VPLS services:

- There is an additional configuration command to enable IGMP-snooping on VXLAN: config>service>vpls>igmp-snooping no shutdown will enable the feature in the VPLS service.
- The VXLAN bindings only support basic IGMP-snooping functionality. Features configurable under SAPs or SDP-bindings are not available for VXLAN. Since there is no specific IGMP-snooping settings for VXLAN bindings (static mrouters or send-queries, and so on.), a specified VXLAN binding will only become a dynamic mrouter when it receives IGMP queries and will add a specified multicast group to the MFIB when it receives an IGMP report for that group.
- The corresponding show/clear service id igmp-snooping commands are also available for VXLAN bindings. The following CLI commands show how the system displays IGMP-snooping information and statistics on VXLAN bindings:

```
*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 232.0.0.1
-------------------------------------------------------------------------------
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 232.0.0.2
-------------------------------------------------------------------------------
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/Sdp Id               Svc Id   Fwd/Blk
-------------------------------------------------------------------------------
*               *                     sap:1/1/1:1              Local    Fwd
*               232.0.0.1             sap:1/1/1:1              Local    Fwd
                                      vxlan:192.0.2.72/1       Local    Fwd
10.0.0.232      232.0.0.2             sap:1/1/1:1              Local    Fwd
                                      vxlan:192.0.2.72/1       Local    Fwd
-------------------------------------------------------------------------------
Number of entries: 3
===============================================================================
```

BGP-EVPN Control Plane for VXLAN Overlay Tunnels

The draft-ietf-bess-evpn-overlay describes EVPN as the control plane for overlay-based networks. The 7x50 supports a subset of the routes and features described in RFC7432 that are required for the DC GW function. In particular, EVPN-specific multi-homing capabilities are not supported. However, multi-homing can be supported by using regular BGP multi-homing based on the L2VPN BGP address family.

Figure 117 shows the EVPN MP-BGP NLRI, required attributes and extended communities, and two route types supported for the DC GW Layer 2 applications:

- route type 3 – Inclusive Multicast Ethernet Tag route
- route type 2 – MAC/IP advertisement route

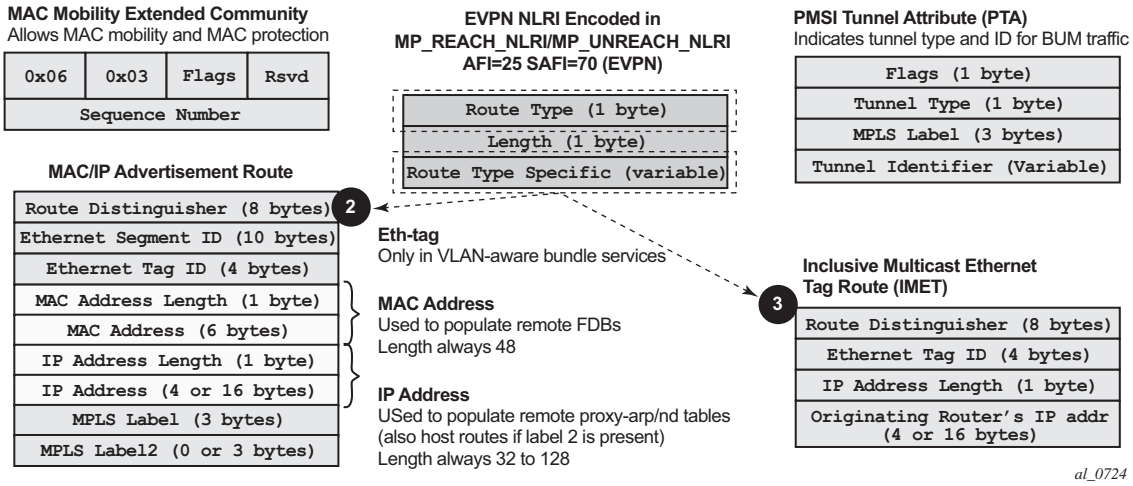

Figure 117: EVPN-VXLAN Required Routes and Communities

EVPN Route Type 3 – Inclusive Multicast Ethernet Tag Route

Route type 3 is used for setting up the flooding tree (BUM flooding) for a specified VPLS service within the data center. The received inclusive multicast routes will add entries to the VPLS flood list in the 7x50. Only ingress replication is supported over VXLAN.

A route type 3 is generated from the 7x50 per VPLS service as soon as the service is operationally UP and uses the following fields and values:

- Route Distinguisher: 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: 0.
- IP address length: Always 32.
- Originating router's IP address: Carries the system address (IPv4 only).
- PMSI attribute:
  - Tunnel type = Ingress replication (6).
  - Flags = Leaf not required.
  - MPLS label = Carries the VNI configured in the VPLS service. Only one VNI can be configured per VPLS service.
  - Tunnel end-point = Equal to the originating IP address.

EVPN Route Type 2 – MAC/IP Advertisement Route

The 7x50 will generate this route type for advertising MAC addresses. The 7x50 will generate MAC advertisement routes for the following:

- Learned MACs on SAPs or sdp-bindings – if mac-advertisement is enabled.
- Conditional static MACs – if mac-advertisement is enabled.
- 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 7x50 uses the following fields and values:

- Route Distinguisher: 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.
- Ethernet Tag ID: 0.
- MAC address length: Always 48.
- MAC Address:
  - It will be 00:00:00:00:00:00 for the Unknown MAC route address.
  - It will be different from 00:…:00 for the rest of the advertised MACs.
- IP address and IP address length:
  - It will be the IP address associated with the MAC being advertised with a length of 32 (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 will have IPL=0 (IP length) and the IP will be omitted.
  - When received, any IPL value not equal to zero, 32, or 128 will make discard the route.

- MPLS Label 1: Carries the VNI configured in the VPLS service. Only one VNI can be configured per VPLS.
- MPLS Label 2: 0.
- MAC Mobility extended community: 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 is used in an IRB backhaul R-VPLS that connects all the VPRN instances for a specified tenant and there is a need to advertise IP prefixes in EVPN, a separate route type is used: route-type 5 IP prefix route.

EVPN Route Type 5 – IP Prefix Route

Figure 118 shows the IP prefix route or route-type 5.

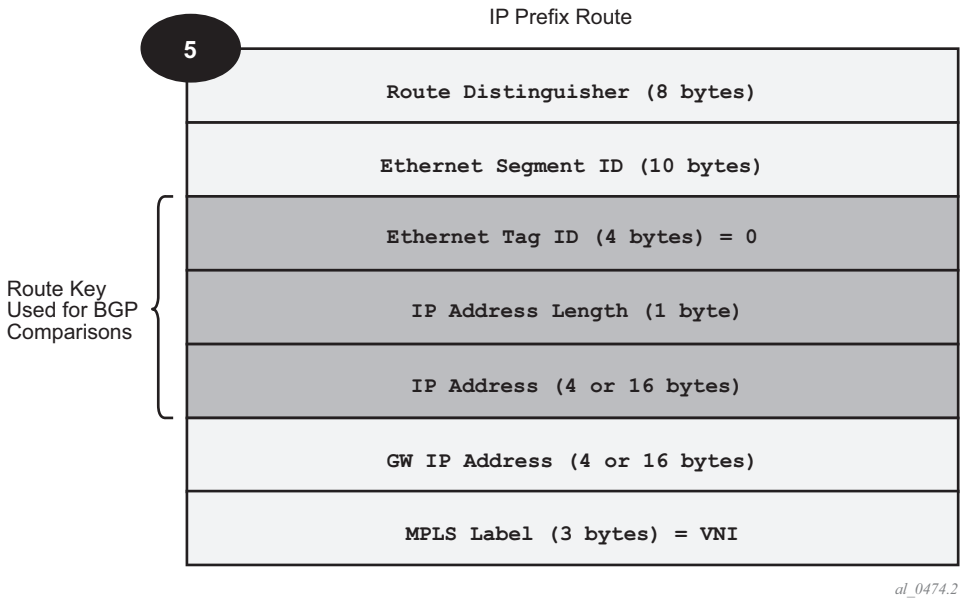

Figure 118: EVPN Route-Type 5

The 7x50 will generate this route type for advertising IP prefixes in EVPN. The 7x50 will generate 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 7x50 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.
- GW IP address: Can carry two different values:
  - If different from zero, the route-type 5 will carry 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 will be sent along with a MAC next-hop extended community that will carry 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 will be sent along with the RFC5512 tunnel encapsulation extended community, with the tunnel type value set to VXLAN.

EVPN for VXLAN in VPLS Services

The EVPN-VXLAN service is designed around the current VPLS objects and the additional VXLAN construct.

Figure 110 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 7x50 where a VPLS instance exists for that particular tenant. Within the DC, the tenant will have 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 DC GW and the DC NVEs will be connected by VXLAN bindings. BGP-EVPN will provide the required control plane for such VXLAN connectivity.

The DC GW 7x50s will be configured with a VPLS per tenant that will provide the VXLAN connectivity to the Nuage VPLS instances. On the 7x50, each tenant VPLS instance will be 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 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
                    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:

- **no unknown-mac-route** and **mac-advertisement** (default option) — The 7x50 will advertise new learned MACs (on the SAPs or sdp-bindings) or new conditional static MACs.

- **unknown-mac-route** and **no mac-advertisement —** The 7x50 will only advertise an unknown-mac-route as long as the service is operationally UP (if no BGP-MH site is configured in the service) or the 7x50 is the DF (if BGP-MH is configured in the service).
- **unknown-mac-route** and **mac-advertisement —** The 7x50 will advertise new learned MACs, conditional static MACs, and the unknown-mac-route. The unknown-mac-route will only be advertised under the preceding described conditions.

Other parameters related to EVPN or VXLAN are:

- Mac duplication parameters
- vxlan vni: Defines the VNI that the 7x50 will use in the EVPN routes generated for the VPLS service.

After the VPLS is configured and operationally UP, the 7x50 will send/receive Inclusive Multicast Ethernet Tag routes, and a full-mesh of VXLAN connections will be automatically created. These VXLAN "auto-bindings" can be characterized as follows:

- The VXLAN auto-bindings 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:

```
*A:DGW# show service id 1 vxlan
===============================================================================
VPLS VXLAN, Ingress VXLAN Network Id: 1
===============================================================================
Egress VTEP, VNI
===============================================================================
VTEP Address           Egress VNI     Num. MACs    Mcast   Oper State   L2 PBR
-------------------------------------------------------------------------------
192.0.2.71             1              1            Yes      Up           No
192.0.0.72             1              1            Yes      Up           No
-------------------------------------------------------------------------------
Number of Egress VTEP, VNI : 2
-------------------------------------------------------------------------------
===============================================================================
```

- 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 will not be marked as "used" and the respective entry in *show service id vxlan* will be removed.

After the flooding domain is setup, the 7x50s and DC NVEs start advertising MAC addresses, and the 7x50s 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 will be 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** will not be installed in the FDB (therefore, will not show up in the show service id x fdb detail command).

- While the 7x50 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 will 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:                   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:                   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
===============================================================================
```

Resiliency and BGP Multi-Homing

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 is provided by BGP-Multi-homing.

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 multi-homing (control plane for the multi-homed 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 DC GWs, VXLAN binding as well as a fec129 spoke-sdp binding will be established between the two DGWs, creating a loop. To avoid creating a loop, the 7x50 will allow the establishment of an EVPN VXLAN binding and an sdp-binding to the same far-end, but the sdp-binding will be kept operationally down. Only the VXLAN binding will be operationally up.

Use of the unknown-mac-route

This section describes the behavior of the EVPN-VXLAN service in the 7x50 when the unknown-mac-route and BGP-MH are configured at the same time.

The use of E-VPN, as the control plane of NVO networks in the DC, provides a significant number of benefits as described in 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 E-VPN 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 7x50 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 DC GW, 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 will 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 7x50 can be configured to generate and advertise the unknown-mac-route, the 7x50 will never honor the unknown-mac-route and will flood to the TLS-flood list when an unknown-unicast packet arrives at an ingress SAP/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 will only be 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 DC GW is Designated Forwarder (DF) for the site. In case of BGP-MH failover, the unknown-mac-route will be withdrawn by the former DF and advertised by the new DF.

EVPN for VXLAN in R-VPLS Services

Figure 111 shows a DC with a Layer-2 service that carries the traffic for a tenant who extends a subnet within the DC, while the DC GW is the default gateway for all the hosts in the subnet. The DC GW function is carried out by the 7x50 where an R-VPLS instance exists for that particular tenant. Within the DC, the tenant will have 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 will be based on existing IP-VPN features.

In this model, the DC GW 7x50s will be configured with a R-VPLS (bound to the VPRN that provides the WAN connectivity) per tenant that will provide 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 7x50:

- The VPRN will be configured with an interface bound to the backhaul R-VPLS. That interface will be 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 will be configured with the BGP, BGP-EVPN and VXLAN (VNI) parameters.

On the Nuage VSGs and NVEs:

- Regular VPLS service model with BGP EVPN and VXLAN parameters.

Other considerations:

- Route-type 2 routes with MACs and IPs will be advertised. Some considerations about MAC+IP and ARP/ND entries are:
  - The 7750 SR will advertise 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 master.
    In both cases, the MACs will be 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 will all be advertised in routes type 2, as static MACs.
  - The 7750 SR will process 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 will be tagged as “EVPN”:

```
A:PE73# show router 2 arp
===============================================================================
ARP Table (Service: 2)
```

```
===============================================================================
IP Address      MAC Address       Expiry    Type   Interface
-------------------------------------------------------------------------------
10.10.10.70     d8:46:ff:ff:ff:3e 00h00m00s Evp[I] local
10.10.10.71     d8:47:ff:ff:ff:3e 00h00m00s Evp[I] local
10.10.10.73     d8:49:ff:ff:ff:3e 00h00m00s Oth[I] local
-------------------------------------------------------------------------------
No. of ARP Entries: 3
===============================================================================
```

- When a VPLS containing proxy-ARP/ND entries is bound to a VPRN (allow-ip-int-bind) all the proxy-ARP/ND entries are moved to the VPRN ARP/ND table. ARP/ND entries will be also moved to proxy-ARP/ND entries if the VPLS is unbound.
- EVPN will not program EVPN-received ARP/ND entries if the receiving VPRN has no IP addresses for the same subnet. The entries will be 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 will bring 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 will be generated.
- Redundancy will be handled by VRRP. The 7750 SR master will advertise vMAC and vIP, as discussed, including the mac mobility extended community and the sticky bit.

EVPN for VXLAN in IRB Backhaul R-VPLS Services and IP Prefixes

Figure 112 shows a Layer 3 DC model, where a VPRN is defined in the DC GWs, connecting the tenant to the WAN. That VPRN instance will be connected to the VPRNs in the NVEs by means of an IRB backhaul R-VPLS. Since the IRB backhaul R-VPLS provides connectivity only to all the IRB interfaces and the DC GW 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 will send IP prefixes in EVPN that will be received by the DC GW 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.

Note — 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
            auto-bind-tunnel
                resolution-filter
                resolution-filter gre ldp rsvp
            vrf-target target:65000:500
            interface "evi-502" create
                address 20.20.20.72/24
```

```
                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 customer 1 create
            allow-ip-int-bind
            vxlan 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
                    no shutdown
                exit
            exit
            service-name "evpn-vxlan-501"
            no shutdown
        exit
vpls 502 customer 1 create
            allow-ip-int-bind
            vxlan 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
                    no shutdown
                exit
            exit
            service-name "evpn-vxlan-502"
            no shutdown
        exit
```

When the above commands are enabled, the 7x50 will:

- 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 7x50, the prefix included in the route-type 5 (for example, 10.0.0.0/24), will be added to the VPRN routing table with a next-hop equal to the GW 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 7x50 receives a packet from the WAN to the 10.0.0.0/24 subnet, the IP lookup on the VPRN routing table will yield 192.0.0.1 as the next-hop. That next-hop will be 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 will advertise 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
-------------------------------------------------------------------------------
20.20.20.0/24                                   Local   Local     01d11h10m  0
       evi-502                                                     0
20.20.20.71/32                                  Remote  BGP EVPN  00h02m26s  169
       10.10.10.71                                                 0
156.10.10.0/24                                  Remote  Static    00h00m05s  5
       10.10.10.71                                                 1
172.16.0.1/32                                   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 will be preferred.

- When the same route-type 5 prefix is received from different GW 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 will be advertised in a route type 2 as soon as IPv6 is enabled on the VPRN R-VPLS interface.
- Routes type 5 for IPv6 prefixes will be 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 will be used as GW IP. The "first IPv6 address" refers to the first one on the list of IPv6 addresses shown via show router <id> interface <interface> IPv6 or via SNMP.

  The rest of the addresses will be advertised only in MAC-IP routes (Route Type 2) but not used as GW IP for IPv6 prefix routes.

EVPN for VXLAN in EVPN Tunnel R-VPLS Services

Figure 113 shows an L3 connectivity model that optimizes the solution described in EVPN for VXLAN in IRB Backhaul R-VPLS Services and IP Prefixes on page 1064. 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 don't 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>interface>vpls <*name*> evpn-tunnel**.

Note — The **evpn-tunnel** can be enabled independently of **ip-route-advertisement**, however, no route-type 5 advertisements will be sent or processed in that case.

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
    auto-bind-tunnel
        resolution-filter
        resolution-filter gre ldp rsvp
    vrf-target target:65000:500
    interface "evi-504" create
        vpls "evpn-vxlan-504"
            evpn-tunnel
        exit
    exit
    no shutdown
exit
vpls 504 customer 1 create
    allow-ip-int-bind
    vxlan 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
            no shutdown
        exit
    exit
    service-name "evpn-vxlan-504"
    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 detai**l 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 GW IP address. If a packet is received from the WAN side and the IP lookup hits an entry for which the GW IP (IP next-hop) does not have an active ARP entry, the system will use ARP to get a MAC. If ARP is resolved but the MAC is unknown in the FDB table, the system will flood into the TLS multicast list. Routes type 5 can be checked in the routing table with the **show>router>route-table** command and the **show>router>fib** command.
- 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 will look 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 will be 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
-------------------------------------------------------------------------------
20.20.20.72/32                                  Remote  BGP EVPN  00h23m50s  169
       10.10.10.72                                                     0
30.30.30.0/24                                   Remote  BGP EVPN  01d11h30m  169
       evi-504 (ET-d8:45:ff:00:01:35)                                  0
156.10.10.0/24                                  Remote  BGP VPN   00h20m52s  170
       192.0.0.69 (tunneled)                                           0
200.1.0.0/16                                    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 will always be 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
-------------------------------------------------------------------------------
300::/64                                        Local   Local     00h01m19s  0
       int-PE-71-CE-1                                               0
500::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 500::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         : 500::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
==============================================================================
```

DC GW integration with the Nuage Virtual Services Directory (VSD)

The Nuage VSD (Virtual Services Directory) provides automation in the Nuage DC. The VSD is a programmable policy and analytics engine. It provides a flexible and hierarchical network policy framework that enables IT administrators to define and enforce resource policies.

The VSD contains a multi-tenant service directory that supports role-based administration of users, computing, and network resources. The VSD also manages network resource assignments such as IP addresses and ACLs.

To communicate with the Nuage controllers and gateways (including the 7x50 DC GW), VSD uses an XMPP (eXtensible Messaging and Presence Protocol) communication channel. The 7x50 can receive service parameters from the Nuage VSD through XMPP and add them to the existing VPRN/VPLS service configuration.

Note — The service must be pre-provisioned in the 7x50 using the CLI, SNMP, or other supported interfaces. The VSD will only push a limited number of parameters into the configuration. This 7x50 – VSD integration model is known as a Static-Dynamic provisioning model, because only a few parameters are dynamically pushed by VSD, as opposed to a Fully Dynamic model, where the entire service can be created dynamically by VSD.

The 7x50 – VSD integration comprises the following building blocks:

- An XMPP interface to the DC XMPP server, through which the 7x50 can discover the Data Center Nuage VSDs and select a specified VSD for each VPLS/VPRN service.
- The configuration of **vsd-domains** on those services where VSD will dynamically provision parameters. As part of the static provisioning of a service, the user will configure a domain name (that will be used between VSD and 7750 SR) using a new CLI command **vsd-domain** *name*. Any parameters sent by the VSD for an existing service will contain the **vsd-domain**. Based on that tag, the 7x50 will add the required configuration changes to the correct service.
- The dynamic provisioning of parameters in the following four use-cases:
  - L2-DOMAIN: To attach a service at the gateway to a Layer-2 (Ethernet) domain in the data center with no routing at the gateway, a VPLS service should be associated with a **vsd-domain** of type **l2-domain**. When the appropriate configuration for the domain is present/added at the VSD, the VSD will dynamically add the VXLAN VNI and BGP export and import route-targets to exchange DC EVPN routes with the VPLS service.
  - L2-DOMAIN-IRB: To attach a service at the gateway to a Layer-2 (Ethernet) domain in the data center with routing at the gateway, an R-VPLS service should be associated with a **vsd-domain** of type **l2-domain-irb**. When the appropriate

configuration for the domain is present/added at the VSD, the VSD will dynamically add the VXLAN VNI and BGP export and import route-targets to exchange DC EVPN routes with the R-VPLS service.

- VRF-GRE: To attach a service at the gateway to a layer 3 domain (with GRE transport) in the data center, a VPRN service should be associated with a **vsd-domain** of type **vrf-gre**. When the appropriate configuration for the domain is present/added at the VSD, the VSD will dynamically add the BGP export and import route-targets to exchange DC IP VPN routes with the VPRN service.
- VRF-VXLAN: To attach a service at the gateway to a layer 3 domain (with VXLAN transport) in the data center, an R-VPLS service (linked to an EVPN-tunnel with ip-route-advertisement enabled) should be associated with a **vsd-domain** of type **vrf-vxlan**. When the appropriate configuration for the domain is present/added at the VSD, the VSD will dynamically add the VXLAN VNI and BGP export and import route-targets to exchange DC EVPN routes with the backhaul R-VPLS connected to the data center VPRN service.

These building blocks are described in more detail in the following subsections.

XMPP Interface on the DC GW

The Extensible Messaging and Presence Protocol is an open technology for real-time communication using XML (Extensible Markup Language) as the base format for exchanging information. The XMPP provides a way to send small pieces of XML from one entity to another in close to real time.

In a Nuage DC, an XMPP ejabberd server will have an interface to the Nuage VSD as well as the Nuage VSC/VSG and the 7x50 DC GW.

Figure 119 shows the basic XMPP architecture in the data center. While a single XMPP server is represented in the diagram, XMPP allows for easy server clustering and performs message replication to the cluster. It is similar to how BGP can scale and replicate the messages through the use of route reflectors.

Also the VSD is represented as a single server, but a cluster of VSD servers (using the same data base) will be a very common configuration in a DC.

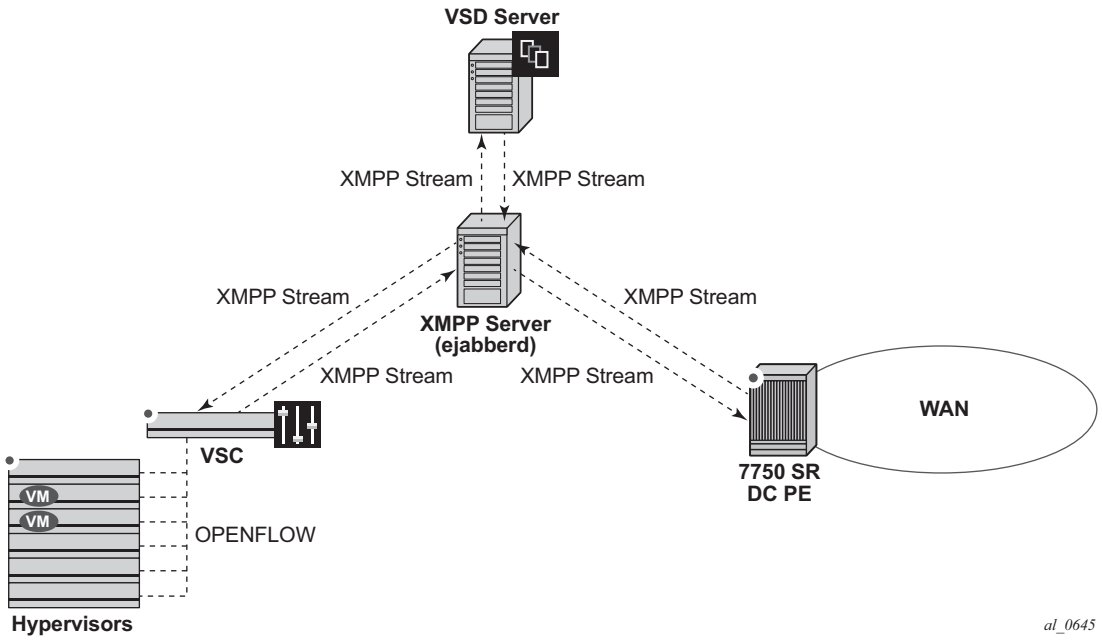

Figure 119: Basic XMPP Architecture

In the Nuage solution, each XMPP client, including the 7x50 SR, is referred to with a JID (JabberID) in the following format: username@xmppserver.domain. The xmppserver.domain points to the XMPP Server.

To enable the XMPP interface on the 7x50, the following command must be added to indicate to which XMPP server address the DC GW has to register, as well as the 7x50's JID:

```
A:Dut-C# configure system xmpp server
  - no server <xmpp-server-name>
  - server <xmpp-server-name> [domain-name <fqdn>] [username <user-name>]
    [password <password>] [create]
 <xmpp-server-name>    : [32 chars max]
 <fqdn>                : [256 chars max]
 <user-name>           : [32 chars max]
 <password>            : [32 chars max]
 <create>              : keyword - mandatory while creating an entry.
 [no] shutdown         - Administratively enable or disable XMPP server
```

Where:

- [domain-name <fqdn>] is the domain portion of the JID.
- <user-name> and <password> is the username:password portion of the JID of the 7x50 acting as an XMPP client. Plain/MD5/anonymous authentication is supported.

- The user can choose not to configure the username portion of the JID. In that case, an in-band registration will be attempted, using the chassis MAC as username.
- When the xmpp server is properly configured and **no shutdown,** the 7750 SR will try to establish a TCP session with the XMPP server through the management interface first. If it fails to establish communication, the 7750 SR will use an in-band communication and will use its system IP as source IP address. **Shutdown** will not remove the dynamic configs in all the services. No server will remove all the dynamic configs in all the services.
- Only one xmpp server can be configured.

Note—The DNS must be configured on the 7x50 so that the XMPP server name can be resolved. XMPP relies on the Domain Name System (DNS) to provide the underlying structure for addressing, instead of using raw IP addresses. The DNS is configured using the following bof commands: **bof primary-dns**, **bof secondary-dns**, **bof dns-domain**.

After the XMPP server is properly configured, the 7x50 can generate or receive XMPP stanza elements, such as presence and IQ (Information/Query) messages. IQ messages are used between the VSD and the 7x50 to request and receive configuration parameters. The status of the XMPP communication channel can be checked with the following command:

```
Dut# show system xmpp server "vsd1-hy"

===============================================================================
XMPP Server Table
===============================================================================
XMPP FQDN          : vsd1-hy.alu.us
XMPP Admin User    : csproot
XMPP Oper User     : csproot
State Lst Chg Since: 0d 02:56:44        State              : Functional
Admin State        : Up                 Connection Mode    : outOfBand
Auth Type          : md5
IQ Tx.             : 47                 IQ Rx.             : 47
IQ Error           : 0                  IQ Timed Out       : 0
IQ Min. Rtt        : 0 ms               IQ Max. Rtt        : 180 ms
IQ Ack Rcvd.       : 47
Push Updates Rcvd  : 1                  VSD list Upd Rcvd  : 12
Msg Tx.            : 27                 Msg Rx.            : 27
Msg Ack. Rx.       : 27                 Msg Error          : 0
Msg Min. Rtt       : 0 ms               Msg Max. Rtt       : 180 ms
Sub Tx.            : 1                  UnSub Tx.          : 0
Msg Timed Out      : 0

===============================================================================
```

In addition to the XMPP server, the 7x50 must be configured with a VSD **system-id** that uniquely identifies the 7x50 in the VSD:

```
*B:Dut>config>system>vsd# info
----------------------------------------------
                system-id "SR12U-46-PE"
----------------------------------------------
```

After the above configuration is complete, the 7x50 will subscribe to a VSD XMPP PubSub node to discover the available VSD servers. Then, the 7x50 will be discovered in the VSD UIs. On the 7x50, the available VSD servers can be shown with the following command.

```
B:Dut#show system xmpp vsd

===============================================================================
Virtual Services Directory Table
===============================================================================
Id User Name                    Uptime             Status
-------------------------------------------------------------------------------
1  cna@vsd1-hy.alu-srpm.us/nua* 0d 00:44:36        Available
-------------------------------------------------------------------------------
No. of VSD's: 1
===============================================================================
* indicates that the corresponding row element may have been truncated.
*B:Dut#show system xmpp vsd 1

===============================================================================
VSD Server Table
===============================================================================
VSD User Name      : cna@vsd1-hy.alu-srpm.us/nuage
Uptime             : 0d 00:44:39        Status             : Available
Msg Tx.            : 16                 Msg Rx.            : 10
Msg Ack. Rx.       : 4                  Msg Error          : 6
Msg TimedOut       : 0                  Msg MinRtt         : 80 ms
Msg MaxRtt         : 240 ms

===============================================================================
```

Overview of the Static-Dynamic VSD Integration Model

In the Static-Dynamic integration model, the DC and DC GW management entities can be the same or different. The DC GW operator will provision the required VPRN and VPLS services with all the parameters needed for the connectivity to the WAN. VSD will only push the required parameters so that those WAN services can be attached to the existing DC domains.

Figure 120 shows the workflow for the attachment of the WAN services defined on the DC GW to the DC domains.

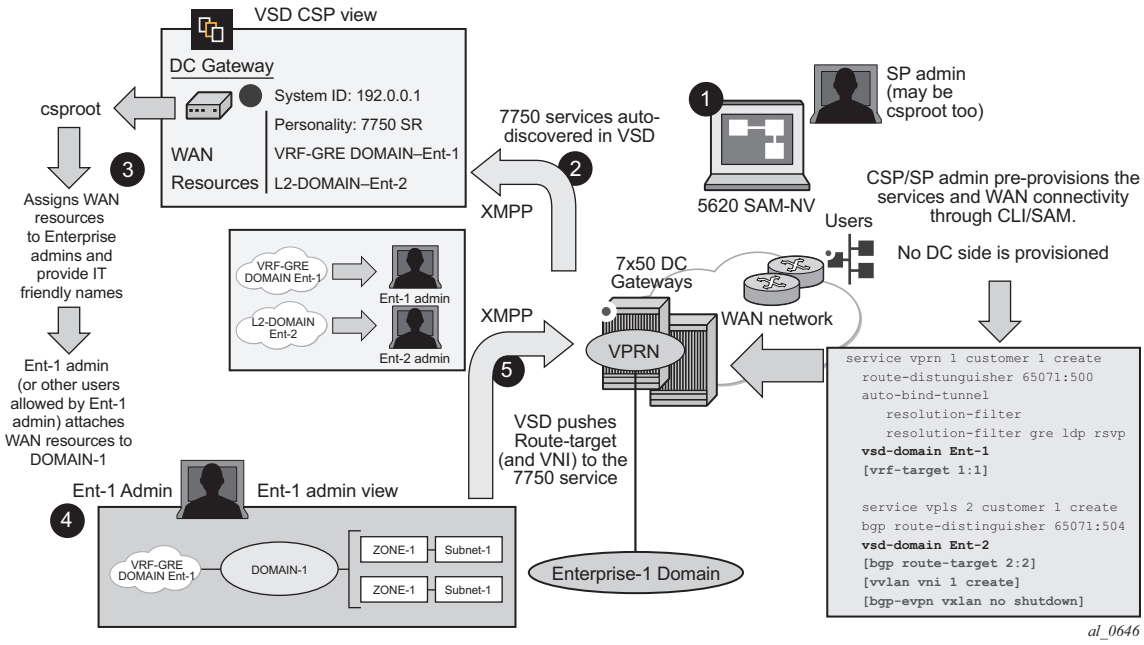

Figure 120: WAN Services Attachment Workflow

The Static-Dynamic VSD integration model can be summarized in the steps shown in Figure 120 and described as follows:

Step 1

The WAN or SP (Service Provider) administrator (which can be also the DC or Cloud Service Provider administrator) provisions the WAN services with all the parameters required for the connectivity to the WAN. This configuration is performed through the regular management

interfaces, for example, CLI or SNMP. In the example above, there are two services created by the SP:

- VPRN 1 – associated with **vsd-domain Ent-1**, which is a VRF-GRE domain.
- VPLS 2 – associated with **vsd-domain Ent-2***,* which is an L2-DOMAIN.

Note — The parameters between brackets "[..]" are not configured at this step. They will be pushed by the VSD through XMPP.

Step 2

The 7x50 communicates with the VSD through the XMPP channel and lets VSD know about its presence and available domains: Ent-1 and Ent-2. In the VSD's User Interface (UI), the 7x50 will show up as DC GW with its System ID, personality (for example, 7x50) and the available WAN resources, that is, **vsd-domains** Ent-1 and Ent-2.

Step 3

At VSD, the Cloud Service Provider administrator will assign the available WAN resources to Enterprises defined in VSD. In this example, VRF-GRE Ent-1 will be assigned to Enterprise-1 and L2-DOMAIN Ent-2 to Entrprise-2.

Step 4

Each Enterprise administrator will have visibility of their own assigned WAN resource and will attach it to an existing DC Domain, assuming that both the DC domain and WAN resource are compatible. For instance, a VRF-GRE domain can only be attached to an L3 domain in the DC that uses GRE as transport.

Step 5

When the Enterprise administrator attaches the WAN resource to the DC domain, VSD will send the required configuration parameters to the DC GW through the XMPP channel:

- In the case of the VRF-GRE domain, VSD will only send the **vrf-target** required for the service attachment to the DC domain.
- In the case of the L2-DOMAIN, VSD will send the **route-target** (in the **service>bgp** or **vsi-import/export** contexts) as well as the **vxlan vni** and the **bgp-evpn vxlan no shutdown** commands.

WAN resources can also be detached from the DC domains.

VSD-Domains and Association to Static-Dynamic Services

In the Static-Dynamic integration model, VSD can only provision certain parameters in VPLS and/or VPRN services. When VSD and the DC GW exchange XMPP messages for a specified service, they use **vsd-domains** to identify those services. A **vsd-domain** is a tag that will be used by the 7x50 and the VSD to correlate a configuration to a service. When redundant DC GWs are available, the **vsd-domain** for the same service can have the same or a different name in the two redundant DC GWs.

There are four different types of **vsd-domains** that can be configured in the 7x50:

- L2-DOMAIN – it will be associated with a VPLS service in the 7x50 and, in VSD, it will be attached to an existing Nuage L2-DOMAIN. This type of domain will be used for extending Layer-2 subnets to the WAN connected to the DC GW.
- L2-DOMAIN-IRB – it will be associated with a R-VPLS service in the 7x50 and, in VSD, it will be attached to an existing Nuage L2-DOMAIN. In this case, the DC GW will be the default gateway for all the VMs and hosts in the Nuage L2-DOMAIN.
- VRF-GRE – this domain type will be associated with a VPRN service in the 7x50 that uses GRE tunnels and MP-BGP VPN-IPv4 to provide connectivity to the DC. In VSD, it will be attached to an existing Nuage L3-DOMAIN, when GRE is configured as tunnel-type for L3-DOMAINs.
- VRF-VXLAN – this domain type will be associated with a 7x50 R-VPLS service (connected to a VPRN with an evpn-tunnel VPLS interface) that uses VXLAN tunnels and EVPN to provide connectivity to the DC. In VSD, it will be attached to an existing Nuage L3-DOMAIN, when VXLAN is configured as the tunnel-type for L3-DOMAINs.

The domains will be configured in the **config>service#** context and assigned to each service.

```
# configure service vsd domain
  - domain <name> [type {l2-domain|vrf-gre|vrf-vxlan|l2-domain-irb}] [create]
  - no domain <name>
 <name>               : [32 chars max]
 <create>             : keyword
 [no] description     - Set VSD Domain Description
 [no] shutdown        - Administratively enable/disable the domain
```

VSD-Domain Type L2-DOMAIN

L2-DOMAIN VSD-domains will be associated with VPLS services configured without a **route-target** and **vxlan VNI**. VSD will configure the route-target and VNI in the 7x50 VPLS service. Some considerations related to L2-DOMAINs are:

- **ip-route-advertisement** and **allow-ip-int-bind** commands are not allowed in this type of domain. An example of configuration for an L2-DOMAIN association is shown below:

```
*B:Dut>config>service# info
...
      vsd
            domain nuage_501 type l2-domain create
                description "nuage_501_l2_domain"
                no shutdown
            exit
*B:Dut>config>service# vpls 501
*B:Dut>config>service>vpls# info
----------------------------------------------
            bgp
                route-distinguisher 192.0.2.2:52
                vsi-import "policy-1"
                vsi-export "policy-1"
            exit
            bgp-evpn
            exit
            sap 1/1/1:501 create
            exit
            spoke-sdp 10:501 create
                no shutdown
            exit
            vsd-domain "nuage_501"
            no shutdown
----------------------------------------------
```

- The VSD will push a dynamic **vxlan vni** and **route-target** that the 7x50 will add to the VPLS service. For the **route-targe**t, the system will check whether the VPLS service has a configured policy:
  - If there is **no vsi-import/export** policy, the received dynamic route-target will be added in the **vpls>bgp>** context, and will be used for all the BGP families in the service.
  - If there is a **vsi-import/export** policy, the dynamic route-target will be added to the policy, in an auto-created community that will be shown with the following format: "**_VSD_svc-id**". That community will be added to dynamically created entries 1000 and 2000 in the first policy configured in the service **vsi-import** and **vsi-export** commands. This allows the user to allocate entries before entries 1000 and 2000 in case other modifications have to be made (user entries would have an action next-entry). An example of the auto-generated entries is shown below:

```
*A:PE# show router policy "policy-1"
```

```
entry 900 # manual entry
    from
      as-path "null"
      family evpn
    exit
    action next-entry
      local-preference 500
    exit
exit
entry 1000 # automatic VSD-generated entry
    from
        community "_VSD_1"
        family evpn
    exit
    action accept
    exit
exit
entry 2000 # automatic VSD-generated entry
    from
        family evpn
    exit
    action accept
        community add "_VSD_1"
    exit
exit
```

VSD-Domain Type L2-DOMAIN-IRB

L2-DOMAIN-IRB VSD-domains will be associated with R-VPLS services configured without a static **route-target** and **vxlan VNI**. VSD will configure the dynamic route-target and VNI in the 7x50 VPLS service. The same considerations described for L2-DOMAINs apply to L2-DOMAIN-IRB domains with one exception: **allow-ip-int-bind** is now allowed.

VSD-Domain Type VRF-GRE

VRF-GRE VSD-domains will be associated with VPRN services configured without a static route-target. In this case, the VSD will push a route-target that the 7x50 will add to the VPRN service. The system will check whether the VPRN service has a configured policy:

- If there is no **vrf-import** policy, the received dynamic route-target will be added in the vprn> context.
- If there is a **vrf-import** policy, the dynamic route-target will be added to the policy, in an auto-created community that will be shown with the following format: "**VSD_svc-id**" in a similar way as in L2-DOMAINs.

  Note — In cases where a **vrf-import** policy is used, the user will provision the WAN **route-target** statically in a **vrf-export** policy. This **route-target** will also be used for the routes advertised to the DC.

An example of the auto-generated entry is shown below:

```
*A:PE# show router policy "policy-1"
    entry 1000 # automatic VSD-generated entry
        from
            community "_VSD_1"
            family vpn-ipv4
        exit
        action accept
        exit
    exit
```

VSD-Domain Type VRF-VXLAN

VRF-VXLAN VSD-domains will be associated with R-VPLS services configured without a static **route-target** and **vxlan VNI**. VSD will configure the dynamic route-target and VNI in the 7x50 VPLS service. Some considerations related to VRF-VXLAN domains are:

- **ip-route-advertisement**, **allow-ip-int-bind**, as well as the VPRN **evpn-tunnel** commands are now required for this type of VSD-domain. An example of configuration for a VRF-VXLAN association is shown below:

```
*A:Dut>config>service# info
<snip>
     vsd
            domain L3Domain-1 type vrf-vxlan create
                description "L3Domain-example"
                no shutdown
            exit
*A:Dut>config>service# vpls 20003
*A:Dut>config>service>vpls# info
----------------------------------------------
            allow-ip-int-bind
            bgp
                route-distinguisher 65000:20003
            exit
            bgp-evpn
                ip-route-advertisement
            exit
            stp
                shutdown
            exit
            service-name "vpls-20003"
            vsd-domain "L3Domain-1"
            no shutdown
----------------------------------------------
*A:sr7L2-47-PE4# configure service vprn 20002
*A:sr7L2-47-PE4>config>service>vprn# info
----------------------------------------------
            route-distinguisher 65000:20002
            auto-bind-tunnel
                resolution-filter
                resolution-filter gre ldp rsvp
            vrf-target target:10:10
            interface "toDC" create
                vpls "vpls-20003"
                    evpn-tunnel
                exit
            exit
            no shutdown
```

- The VSD will push a dynamic **vxlan VNI** and **route-target** that the 7x50 will add to the VPLS service. For the **route-target**, the system will check whether the VPLS service has a configured policy and will push the **route-target** either in the service context or the **vsi-import/export** policies, as described in the section for L2-DOMAINs.

The following commands help show the association between the 7x50 services and VSD-domains, as well as statistics and configuration errors sent/received to/from VSD.

```
*A:Dut# show service service-using vsd
==========================================
Services-using VSD Domain
==========================================
Svc Id     Domain
```

```
-----------------------------------------
501        nuage_501
200001     MyL2Domain
20003      MyL3Domain
-----------------------------------------
Number of services using VSD Domain: 3
=========================================
*A:Dut# show service vsd domain "MyL3Domain"

===============================================================================
VSD Information
===============================================================================
Name               : MyL3Domain
Description        : MyL3Domain-example
Type               : vrfVxlan                    Admin State   : inService
Last Error To Vsd  : (Not Specified)
Last Error From Vsd: (Not Specified)

Statistics
-------------------------------------------------------------------------------
Last Cfg Chg Evt   : 02/06/2015 01:28:30         Cfg Chg Evts  : 671
Last Cfg Update    : 02/06/2015 02:58:41         Cfg Upd Rcvd  : 3
Last Cfg Done      : 02/06/2015 02:58:41
Cfg Success        : 667                         Cfg Failed    : 0
===============================================================================

*A:Dut# show service vsd domain "MyL3Domain" association

============================================================
Service VSD Domain
============================================================
Svc Id      Svc Type  Domain Type    Domain Admin   Origin
------------------------------------------------------------
20003       vpls      vrfVxlan       inService      manual
------------------------------------------------------------
Number of entries: 1
============================================================
```

Fully-Dynamic VSD Integration Model

In the Static-Dynamic VSD integration model, the VPLS/VPRN service, as well as most of the parameters, must be provisioned "statically" through usual procedures (CLI, SNMP, and so on). VSD will "dynamically" send the parameters that are required for the attachment of the VPLS/VPRN service to the L2/L3 domain in the Data Center. In the Fully-Dynamic VSD integration model, the entire VPLS/VPRN service configuration will be dynamically driven from VSD and no static configuration is required. Through the existing XMPP interface, the VSD will provide the 7x50 with a handful of parameters that will be translated into a service configuration by a python-script. This python-script provides an intermediate abstraction layer between VSD and the 7x50, translating the VSD data model into a 7x50 CLI data model.

In this Fully-Dynamic integration model, the DC and DC GW management entities are usually the same. The DC GW operator will provision the required VPRN and VPLS services with all the parameters needed for the connectivity to the WAN and the DC. VSD will push the required parameters so that the 7x50 can create the service completely and get it attached to an existing DC domain.

The workflow of the Fully-Dynamic integration model is shown in Figure 121.

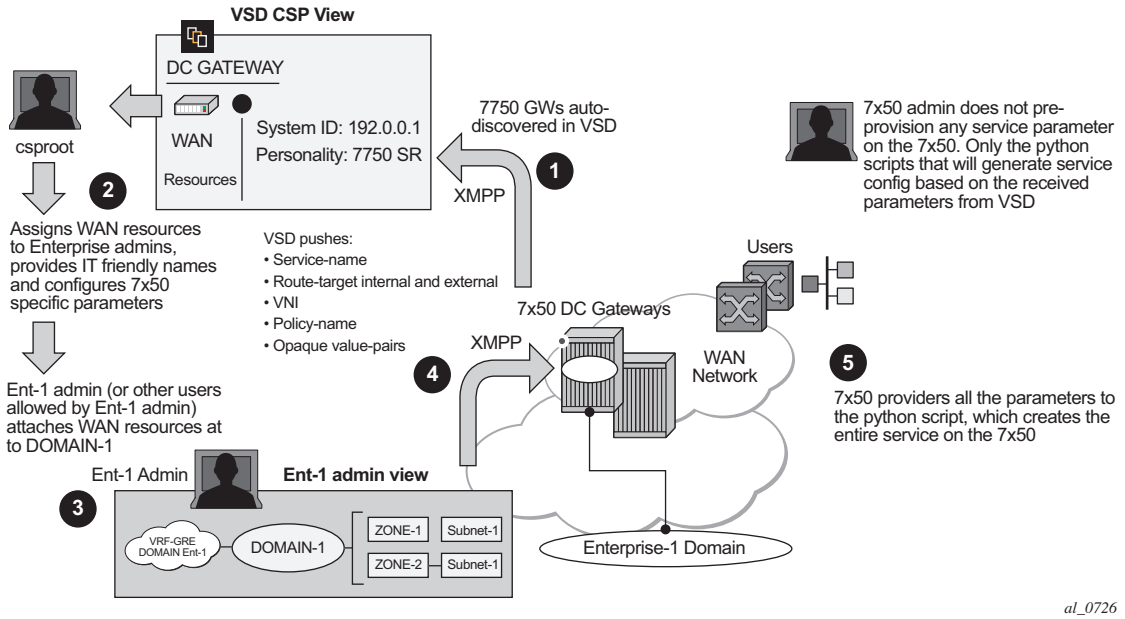

Figure 121: Fully-Dynamic VSD Integration Model Workflow

The Fully-Dynamic VSD integration model can be summarized in the steps shown in Figure 121 and described as follows:

Step 1

The 7x50 administrator only needs to provision parameters required for connectivity to the VSD, a s**ervice-range**, and configure the python script/policy in the system. Provisioning of service parameters is not required.

The **service-range** defines the service identifiers to include for VSD provisioned services and, once configured, they are protected from CLI changes. The vsd-policy defines the script to be used:

```
*A:PE1>config>python# info
----------------------------------------------
        python-script "l2-domain_services" create
            primary-url "ftp://1.1.1.1/cf2/l2domain_service.py"
            no shutdown
        exit
  python-policy "py-l2" create
            description "Python script to create L2 domains"
            vsd script "l2-domain_services"
        exit
----------------------------------------------

*A:PE1>config>service>vsd# info
----------------------------------------------
  service-range 64000 to 65000
----------------------------------------------
```

When the 7x50 boots up or the gateway configuration is changed, the 7x50 sends a message to the VSD indicating its capabilities:

- System-ID
- Name and gateway type

The VSD uses this information to register the 7x50 on its list of 7x50 GWs.

Once registered, the VSD and 7x50 exchange messages where the VSD communicates its list of service-names and their domain-type to the 7x50. Based on this list, the 7x50 sends an XMPP IQ message to request the configuration of a specified service.

The 7x50 will periodically audit the VSD and request a “DIFF” list of Full-Dynamic VSD domains. The VSD keeps a “DIFF” list of domains, that contains the Fully-Dynamic domain names for which the VSD has not received an IQ request from the 7x50 for a long time.

The 7x50 CLI user can also audit the VSD to get the DIFF list, or even the "FULL" list of all the domains in the VSD. The following command triggers this audit: **tools perform service vsd fd-domain-sync {full|diff}**.

Step 2

Concurrently at the VSD, the Cloud Service Provider administrator will assign WAN resources to Enterprises defined in the VSD. In this example, a VRF-GRE domain will be assigned to Enterprise-1.

Step 3

Each Enterprise administrator will have visibility of their own assigned WAN resource and will attach it to an existing DC Domain, assuming that both the DC domain and WAN resource are compatible. For instance, a VRF-GRE domain can only be attached to an L3 domain in the DC that uses GRE as transport.

Step 4

When the Enterprise administrator attaches the service requested by the 7x50 to the DC domain, the VSD will send the required configuration parameters for that service to the DC GW through the XMPP channel in an IQ Service message, including the following information:

- Service name and service type, where the type can be:
  - L2-DOMAIN
  - L2-DOMAIN-IRB
  - VRF-GRE
  - VRF-VXLAN
- Configuration type— Static (for Static-Dynamic model) or Dynamic (for Fully-Dynamic model).
- Internal route-target (RT-i) — Used to export/import the BGP routes sent/received from/to the DC route-reflector.
- External route-target (RT-e) — Used to export/import the BGP routes sent/received from/to the WAN route-reflector. The value can be the same as the RT-i.
- VNI (VXLAN Network Identifier) — Used to configure the EVPN-VXLAN VPLS service on the 7x50 (if the domain type is L2-DOMAIN, L2-DOMAIN-IRB, or VRF-VXLAN).

- Metadata — A collection of 'opaque' <key=value> pairs including the rest of the service parameters required for the service configuration at the 7x50.

  Note — The keys or values do not need to follow any specific format. The python script interprets and translates them into the 7x50 data model.
- Python-policy— Used by the 7x50 to find the Python script that will translate the VSD parameters into configuration.

Step 5

When the 7x50 receives the IQ Service message, it builds a string with all the parameters and passes it to the Python module. The Python module is responsible for creating and activating the service, and, therefore, provides connectivity between the tenant domain and the WAN.

Note — The python-script cannot access all the CLI commands and nodes in the system. A white-list of nodes and commands is built and Python will only have access to those nodes/commands. The list can be displayed using the following command: **tools dump service vsd-services command-list**.

In addition to the *white-list*, the user can further restrict the allowed CLI nodes to the VSD by using a separate CLI user for the XMPP interface, and associating that user to a profile where the commands are limited. The CLI user for the XMPP interface is configurable:

```
config>system>security>cli-script>authorization>
     vsd
[no] cli-user <username>
```

When the system executes a python-script for VSD commands, the *vsd cli-user* profile is checked to allow the resulting commands. A single CLI user is supported for VSD, therefore, the operator can configure a single 'profile' to restrict (even further than the *whitelist*) the CLI commands that can be executed by the VSD Python scripts.

No *cli-user* means that the system will not perform any authorization checks and the VSD scripts can access any commands that are supported in the *white-list*.

Python Script Implementation Details

A python-script provides an intermediate abstraction layer between VSD and the 7750, translating the VSD data model into the 7x50 CLI data model. VSD will use metadata key=value parameters to provision service specific attributes on the 7750. The XMPP messages get to the 7750 and are passed transparently to the Python module so that the CLI is generated and the service created.

Note — The CLI generated by the python-script is not saved in the configuration file and it is not displayed by the info commands when executed on the service contexts. Also the configuration generated by the python-script is protected and cannot be changed by a CLI user.

The following example shows how the python-script works for a VSD generated configuration:

- The following configuration is added to the 7750. In this case, *py-l2* is the python-policy received from VSD that will call the *l2domain_service.py* python script:

```
*A:PE1>config>python# info
----------------------------------------------
        python-script "l2-domain_services" create
            primary-url "ftp://1.1.1.1/cf2/l2domain_service.py"
            no shutdown
        exit
python-policy "py-l2" create
            description "Python script to create L2 domains"
            vsd script "l2-domain_services"
        exit
----------------------------------------------
*A:PE1>config>service>vsd# info
----------------------------------------------
  service-range 64000 to 65000
----------------------------------------------
```

- VSD will send metadata containing the service parameters. This opaque parameter string will be passed to the python script and is composed of tag=value pairs, with the following format:
- In addition, other information provided by the VSD, (domain, vni, rt-i, rt-e, and service type) is bundled with the metadata string and passed to the python script. For example:

```
{'rt': 'target:64000:64000', 'rte': 'target:64000:64000', 'domain': 'L2-DOMAIN-1', 'ser-
vicetype': 'L2DOMAIN', 'vni': '64000', 'metadata': 'rd=1:1, sap=1/1/10:3000'}
```

The user should consider the following:

- The python script is solely responsible for generating the configuration; no configuration aspects of the current Static-Dynamic model are used. The python script must be written in a manner similar to those used by RADIUS Dynamic Business Services. Currently, RADIUS Dynamic Business Services and the Fully-Dynamic VSD model are mutually exclusive, one or the other can operate on the same system, but not both at the same time.
- The following scripts must be defined in order to set up, modify, revert, and tear down the configuration for a service: *setup_script(), modify_script(), revert_script(),* and *teardown_script()*. These names must always be the same in all scripts. The *revert_script()* is only required if the *modify_script()* is defined, the latter being optional.

- When the configuration for a new domain name is received from the VSD, the metadata and the VSD parameters are concatenated into a single dictionary and *setup_script()* is called. Within the python script:
  - The VSD UI parameters are referenced as vsdParams['rt'], vsdParams ['domain'], and so on.
  - The metadata parameters are referenced as vsdParams['metadata'].
- When the startup script is executed, the **config>service>vsd>domain** is created outside the script context before running the actual script. The teardown script will remove the **vsd domain**.
  - If a specified *setup_script()* fails, the *teardown_script()* is invoked.
- When subsequent configuration messages are received from the VSD, the new parameter list is generated again from the VSD message and compared to the last parameter list that was successfully executed.
  - If the two strings are identical, no action is taken.
  - If there is a difference between the strings, the *modify_script()* function is called.
  - If the *modify_script()* fails, the *revert_script()* is invoked. The *teardown_script()* is invoked if the *revert_script()* fails or does not exist.
- The **python-policy** is always present in the attributes received from VSD; if the VSD user does not include a policy name, VSD will include 'default' as the python-policy. Hence, care must be taken to ensure that the 'default' policy is always configured in the 7750.
- If the scripts are incorrect, teardown and modify procedures could leave orphaned objects. An admin mode (**enable-vsd-config**) is available to enable an administrator to clean up these objects; it is strictly meant for cleaning orphaned objects only.

  Note — The CLI configured services cannot be modified when the user is in **enable-vsd-config** mode.
- Unless the CLI user enters the **enable-vsd-config** mode, changes of the dynamic created services are not allowed in the CLI. However, changes through SNMP are allowed.
- The command **tools perform service vsd evaluate-script** is introduced to allow the user to test their setup and to modify and tear down python scripts in a lab environment without the need to be connected to a VSD. The successful execution of the command for **action setup** will create a **vsd domain** and the corresponding configuration, just as the system would do when the parameters are received from VSD.

The following example shows the use of the **tools perform service vsd evaluate-script** command:

```
*A:PE1# tools perform service vsd evaluate-script domain-name "L2-DOMAIN-5" type l2-domain
action setup policy "py-l2" vni 64000 rt-i target:64000:64000 rt-e target:64000:64000
metadata "rd=1:1, sap=1/1/10:3000"

Success
```

The following example output shows a python-script that can set up or tear down L2-DOMAINs.

```
*A:PE1# show python python-script "l2-domain_services" source-in-use

===============================================================================
Python script "l2-domain_services"
===============================================================================
Admin state   : inService
Oper state    : inService
Primary URL   : ftp://1.1.1.1/timos86/cses-V71/cf2/l2domain_service.py
Secondary URL : (Not Specified)
Tertiary URL  : (Not Specified)
Active URL    : primary

-------------------------------------------------------------------------------
Source (dumped from memory)
-------------------------------------------------------------------------------
      1 from alc import dyn
      2
      3 def setup_script(vsdParams):
      4
      5     print ("These are the VSD params: " + str(vsdParams))
      6     servicetype = vsdParams.get('servicetype')
      7     vni = vsdParams.get('vni')
      8     metadata = vsdParams['metadata']
      9     print ("VSD metadata: " + str(metadata))
     10     metadata = dict(e.split('=') for e in metadata.split(','))
     11     print ("Modified metadata: " + str(metadata))
     12     vplsSvc_id = dyn.select_free_id("service-id")
     13     print ("this is the free svc id picked up by the system: " + vplsSvc_id)
     14
     15     if servicetype == "L2DOMAIN":
     16         rd = metadata['rd']
     17         sap_id = metadata[' sap']
     18         print ('servicetype, VPLS id, rd, sap:', servicetype, vplsSvc_id, rd,
sap_id)
     19         dyn.add_cli("""
     20           configure service
     21               vpls %(vplsSvc_id)s customer 1 create
     22                   description vpls%(vplsSvc_id)s
     23                   bgp
     24                       route-distinguisher %(rd)s
     25                       route-target %(rt)s
     26                   exit
     27                   vxlan vni %(vni)s create
     28                   exit
     29                   bgp-evpn
     30                       evi %(vplsSvc_id)s
     31                       vxlan
     32                           no shutdown
     33                       exit
     34                   exit
     35                   service-name evi%(vplsSvc_id)s
     36                   sap %(sap_id)s create
     37                   exit
     38                   no shutdown
     39                   exit
     40               exit
     41           exit
```

```
     42         """ % {'vplsSvc_id' : vplsSvc_id, 'vni' : vsdParams['vni'], 'rt' : vsdPar-
ams['rt'], 'rd' : metadata['rd'], 'sap_id' : sap_id})
     43         # L2DOMAIN returns setupParams: vplsSvc_id, servicetype, vni, sap
     44       return {'vplsSvc_id' : vplsSvc_id, 'servicetype' : servicetype, 'vni' : vni,
'sap_id' : sap_id}
     45
     46
     47 #--------------------------------------------------------------------------------
     48
     49 def teardown_script(setupParams):
     50     print ("These are the teardown_script setupParams: " + str(setupParams))
     51     servicetype = setupParams.get('servicetype')
     52     if servicetype == "L2DOMAIN":
     53       dyn.add_cli("""
     54         configure service
     55             vpls %(vplsSvc_id)s
     56                 no description
     57                 bgp-evpn
     58                     vxlan
     59                         shutdown
     60                     exit
     61                     no evi
     62                     exit
     63                 no vxlan vni %(vni)s
     64                 bgp
     65                    no route-distinguisher
     66                    no route-target
     67                 exit
     68                 no bgp
     69                 no bgp-evpn
     70                 sap %(sap_id)s
     71                    shutdown
     72                    exit
     73                 no sap %(sap_id)s
     74                 shutdown
     75                 exit
     76                 no vpls %(vplsSvc_id)s
     77             exit
     78         exit
     79       """ % {'vplsSvc_id' : setupParams['vplsSvc_id'], 'vni' : setupParams['vni'],
'sap_id' : setupParams['sap_id']})
     80       return setupParams
     81
     82
     83 d = { "script" : (setup_script, None, None, teardown_script) }
     84
     85 dyn.action(d)
================================================================================
```

For instance, assuming that the VSD sends the following:

```
{'rt': 'target:64000:64000', 'rte': 'target:64000:64000', 'domain': 'L2-DOMAIN-5', 'ser-
vicetype': 'L2DOMAIN', 'vni': '64000', 'metadata': 'rd=1:1, sap=1/1/10:3000 '}
```

The system will execute the setup script as follows:

```
123 2015/06/16 23:35:40.22 UTC MINOR: DEBUG #2001 Base dyn-script req=setup
"dyn-script req=setup: L2-DOMAIN-5
```

```
  state=init->waiting-for-setup
"

124 2015/06/16 23:35:40.22 UTC MINOR: DEBUG #2001 Base dyn-script req=setup
"dyn-script req=setup: L2-DOMAIN-5
  state=waiting-for-setup->generating-setup
"

125 2015/06/16 23:35:40.22 UTC MINOR: DEBUG #2001 Base Python Output
"Python Output: l2-domain_services
These are the VSD params: {'rt': 'target:64000:64000', 'rte': 'target:64000:6400
0', 'domain': '', 'servicetype': 'L2DOMAIN', 'vni': '64000', 'metadata': 'rd=1:1
, sap=1/1/10:3000 '}
VSD metadata: rd=1:1, sap=1/1/10:3000
Modified metadata: {'rd': '1:1', ' sap': '1/1/10:3000 '}
this is the free svc id picked up by the system: 64000
('servicetype, VPLS id, rd, sap:', 'L2DOMAIN', '64000', '1:1', '1/1/10:3000 ')
"
Success

126 2015/06/16 23:35:40.22 UTC MINOR: DEBUG #2001 Base Python Result
"Python Result: l2-domain_services
"

127 2015/06/16 23:35:40.22 UTC MINOR: DEBUG #2001 Base dyn-script req=setup
"dyn-script req=setup: L2-DOMAIN-5
  state=generating-setup->executing-setup
"

128 2015/06/16 23:35:40.22 UTC MINOR: DEBUG #2001 Base dyn-script cli 1/1
"dyn-script cli 1/1: script:L2-DOMAIN-5(cli 698 dict 0->126)

          configure service
              vpls 64000 customer 1 create
                  description vpls64000
                  bgp
                      route-distinguisher 1:1
                      route-target target:64000:64000
                  exit
                  vxlan vni 64000 create
                  exit
                  bgp-evpn
                      evi 64000
                      vxlan
                          no shutdown
                      exit
                  exit
                  service-name evi64000
                  sap 1/1/10:3000  create
                  exit
                  no shutdown
                  exit
              exit
          exit
        "

143 2015/06/16 23:35:40.23 UTC MINOR: DEBUG #2001 Base dyn-script req=setup
"dyn-script req=setup: L2-DOMAIN-5
  state=executing-setup->established"
```

BGP-EVPN Control Plane for MPLS Tunnels

Table 18 lists all the EVPN routes supported in 7x50 SROS and their usage in EVPN-VXLAN, EVPN-MPLS, and PBB-EVPN.
Note — Route type 1 is not required in PBB-EVPN as per draft-ietf-l2vpn-pbb-evpn.

Table 18: 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	-
Type 2 - MAC/IP Advertisement route	MAC/IP advertisement, IP advertisement for ARP resolu-tion	Y	Y	Y
Type 3 - Inclusive Multicast Ethernet Tag route	Flooding tree setup (BUM flooding)	Y	Y	Y
Type 4 - Ethernet Segment route	ES discovery and DF elec-tion	-	Y	Y
Type 5 - IP Prefix advertisement route	IP Routing	Y	-	-

RFC7432 describes the BGP-EVPN control plane for MPLS tunnels. If EVPN multi-homing 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 multi-homing is required, the Ethernet Segment and the Auto-Discovery routes are also needed.

The route fields and extended communities for route types 2 and 3 are shown in Figure 117. See “BGP-EVPN Control Plane for VXLAN Overlay Tunnels” on page 1053. 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 within the data center. The received inclusive multicast routes will add entries to the VPLS flood list in the 7x50. Only ingress replication is supported over MPLS tunnels. The following route values are used for EVPN-MPLS services:

- Route Distinguisher: 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: 0.
- IP address length: Always 32.
- Originating router's IP address: Carries the system address (IPv4 only).
- 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 will be the same label used in the MAC/IP routes for the same service unless bgp-evpn mpls ingress-replication-bum-label is configured in the service.
  - Tunnel end-point = Equal to the originating IP address.

EVPN Route Type 2 - MAC/IP Advertisement Route

The 7x50 generates this route type for advertising MAC addresses (and IP addresses if proxy-arp/nd is enabled). The 7x50 generates MAC advertisement routes for the following:

- Learned MACs on SAPs or sdp-bindings—if mac-advertisement is enabled.
- Conditional static MACs—if mac-advertisement is enabled.

  Note — The **unknown-mac-route** is not supported for EVPN-MPLS services.

The route type 2 generated by a 7x50 uses the following fields and values:

- Route Distinguisher: 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): Zero for MACs learned from single-homed CEs and different from zero for MACs learned from multi-homed CEs.
- Ethernet Tag ID: 0.
- MAC address length: Always 48.
- MAC Address learned or statically configured.

- IP address and IP address length:
  - It will be 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 will have IPL=0 (IP length) and the IP will be omitted.
  - When received, any IPL value not equal to zero, 32, or 128 will discard 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 will be 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: 0.
- The MAC Mobility extended community: 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 multi-homing is enabled in the system, two more routes are required. Figure 122 shows the fields in routes type 1 and 4 and their associated extended communities.

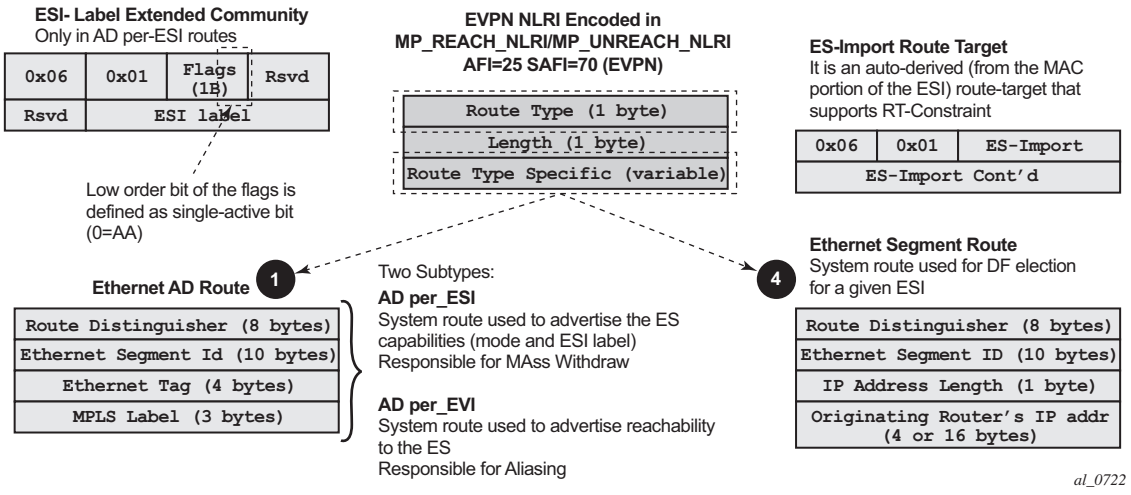

Figure 122: EVPN Routes Type 1 and 4

EVPN Route Type 1 - Ethernet Auto-discovery Route (AD route)

The 7x50 generates this route type for advertising for multi-homing 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 7x50 uses the following fields and values:

- Route Distinguisher: Taken from the system level RD or service level RD.
- Ethernet Segment Identifier (ESI): Will contain a 10-byte identifier as configured in the system for a specified **ethernet-segment**.
- Ethernet Tag ID: MAX-ET (0xFFFFFFFF). This value is reserved and used only for AD routes per ESI.
- MPLS label: 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 multi-homing split-horizon.
- Route-target extended community: Taken from the service level RT or an RT-set for the services defined on the ethernet-segment.

The Ethernet AD per-EVI route generated by a 7x50 uses the following fields and values:

- Route Distinguisher: Taken from the service level RD.
- Ethernet Segment Identifier (ESI): Will contain a 10-byte identifier as configured in the system for a specified ethernet-segment.
- Ethernet Tag ID: 0.
- MPLS label: Encodes the unicast label allocated for the service (high-order 20 bits).
- Route-target extended community: 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 - Ethernet Segment Route (ES route)

The 7x50 generates this route type for multi-homing ES discovery and DF (Designated Forwarder) election.

- Route Distinguisher: Taken from the service level RD.
- Ethernet Segment Identifier (ESI): Will contain a 10-byte identifier as configured in the system for a specified **ethernet-segment**.

- ES-import route-target community: The value 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).

RFC5512 - BGP Tunnel Encapsulation Extended Community

The following routes are sent with the RFC5512 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 7X50 processes the following BGP Tunnel Encapsulation tunnel values registered by IANA for RFC5512:

- VXLAN encapsulation: 8.
- MPLS encapsulation: 10.

Any other tunnel value will make the route 'treat-as-withdraw'.

If the encapsulation value is MPLS, the BGP will validate the high-order 20-bits of the label field, ignoring the low-order 4 bits. If the encapsulation is VXLAN, the BGP will take the entire 24-bit value encoded in the MPLS label field as the VNI.

If no RFC5512 encapsulation extended community is present in a received route, BGP will treat the route as MPLS or VXLAN-based configuration of the **config>router>bgp>neighbor# def-recv-evpn-encap [mpls|vxlan]** command.

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, LDP, RFC3107 BGP, Segment Routing IS-IS, or Segment Routing OSPF. As with VPRN services, the selection of the tunnel to be used in a VPLS service (with BGP-EVPN MPLS enabled) is based on the **auto-bind-tunnel** command.

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
  bgp-evpn
    evi 10
    vxlan
      shutdown
    mpls
      no shutdown
      auto-bind-tunnel resolution any
  sap 1/1/1:1 create
  exit
  spoke-sdp 1:1 create
```

The user will configure a **bgp-evpn** context where **vxlan** must be shutdown and **mpls no shutdown**. In addition to the **mpls no shutdown** 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. However, the user can configure some other options. The most relevant configuration options are described below.

evi {1..65535} — This EVPN identifier is unique in the system and will be used for the service-carving algorithm used for multi-homing (if configured) and auto-deriving route-target and route-distinguishers in the service. It can be used for EVPN-MPLS and EVPN-VXLAN services.

If this EVPN identifier is not specified, the value will be zero and no route-distinguisher or route-targets will be auto-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 auto-derived route-targets. The operational route-target for a service will be displayed by the **show service id x bgp** command.

When the **evi** is configured, a **config>service>vpls>bgp node** (even empty) is required to allow the user to see the correct information on the **show service id 1 bgp** and **show service system bgp-route-distinguisher** commands.

Although not mandatory, if no multi-homing is configured, the configuration of an **evi** is enforced for EVPN services with SAPs/SDP-bindings in an **ethernet-segment**. See the 'EVPN multi-homing' section for more information about **ethernet-segments**.

The following options are specific to EVPN-MPLS (and defined on **bgp-evpn>mpls**):

- **control-word:** Required as per RFC7432 to avoid frame disordering. The user can enable/disable it so that interoperability to other vendors can be guaranteed.
- **auto-bind-tunnel:** Allows the user to decide what type of MPLS transport tunnels will be used for a particular instance. The command will be used in the same way as it is used in VPRN services.

  Note — For bgp-evpn mpls, '**bgp**' is explicitly added to the **resolution-filter** in EVPN ('**bgp**' is implicit in VPRNs).
- **force-vlan-vc-forwarding:** This command will allow 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 — This command may be used in conjunction with the **sap ingress vlan-translation** command. If so, the configured translated vlan-id will be the vlan-id 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 will be zero.
- **split-horizon-group:** This command allows the association of a user-created split-horizon-group to all the EVPN-MPLS destinations. See the EVPN and VPLS integration section for more information.
- **ecmp:** When this command is set to a value greater than 1, aliasing is activated to the remote PEs that are defined in the same all-active multi-homing ethernet-segment. See the EVPN multi-homing section for more information.
- **ingress-replication-bum-label:** This command is only enabled when the user wants 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 multi-homing.

In addition to these options, the following bgp-evpn commands are also available for EVPN-MPLS services:

- **[no] mac-advertisement**
- **mac-duplication and settings**

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 will create:

- 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**. Hence, 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).

```
*A:Dut# 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
-------------------------------------------------------------------------------
===============================================================================

*A:Dut# show service id 1 fdb detail

===============================================================================
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
===============================================================================
```

EVPN and VPLS Integration

The 7x50 SROS EVPN implementation supports draft-ietf-bess-evpn-vpls-seamless-integ so that EVPN-MPLS and VPLS can be integrated into the same network and within the same service. Since EVPN will not be 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:

a) Systems with EVPN endpoints and sdp-bindings to the same far-end bring down the sdp-bindings.

- The 7x50 will allow the establishment of an EVPN endpoint and a sdp-binding to the same far-end but the sdp-binding will be kept operationally down. Only the EVPN endpoint will be operationally up. This is true for spoke-sdps (manual and BGP-AD) 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 will be 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 will be setup and the spoke-sdp will be 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 will be 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.

b) 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 (in which all the EVPN endpoints are) is still used, but it will not be possible to refer to it on saps/spoke-sdps.

c) The system disables the advertisement of MACs learned on spoke-sdps/saps that are part of an EVPN split-horizon-group.

- When the saps and/or spoke-sdps (manual or BGP-AD-discovered) are configured within the same **split-horizon-group** as the EVPN endpoints, MAC addresses will still be learned on them, but they will not be advertised in EVPN.
- The preceding statement is also true if proxy-ARP/ND is enabled and an IP->MAC pair is learned on a sap/sdp-binding that belongs to the EVPN **split-horizon-group**.
- The SAPs and/or spoke-SDPs added to an EVPN **split-horizon-group** should not be part of any EVPN multi-homed ES. If that happened, the PE would still advertise the AD per-EVI route for the SAP and/or spoke-SDP, attracting EVPN traffic that could not possibly be forwarded to that SAP and/or sdp-binding.
- 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 123 shows an example of EVPN-VPLS integration.

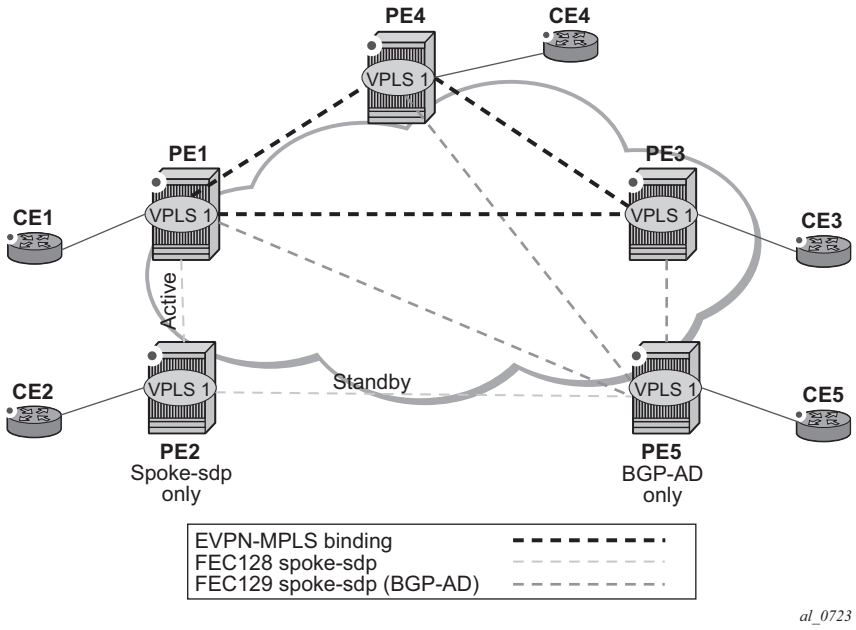

Figure 123: EVPN-VPLS Integration

An example CLI configuration for PE1, PE5, and PE2 is provided below.

```
*A:PE1>config>service# info
----------------------------------------------
pw-template 1 create
vpls 1 customer 1 create
  split-horizon-group "SHG-1" create
  bgp
    route-target target:65000:1
    pw-template-binding 1 split-horizon-group SHG-1
  bgp-ad
    no shutdown
    vpls-id 65000:1
  bgp-evpn
    evi 1
    mpls
      no shutdown
      split-horizon-group SHG-1
  spoke-sdp 12:1 create
  exit
  sap 1/1/1:1 create
  exit

*A:PE5>config>service# info
----------------------------------------------
pw-template 1 create
vpls 1 customer 1 create
  bgp
```

```
    route-target target:65000:1
    pw-template-binding 1 split-horizon-group SHG-1 # auto-created SHG
  bgp-ad
    no shutdown
    vpls-id 65000:1
  spoke-sdp 52:1 create
  exit

*A:PE2>config>service# info
----------------------------------------------
vpls 1 customer 1 create
  end-point CORE create
    no suppress-standby-signaling
  spoke-sdp 21:1 end-point CORE
    precedence primary
  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 will attempt to establish BGP-AD spoke-sdps, but they will be 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 will flood to all the active bindings.
  - When CE2 sends BUM, PE2 will send it to PE1 (active spoke-sdp) and PE1 will flood to all the bindings and saps.
  - When CE5 sends BUM, PE5 will flood to the three EVPN PEs. PE1 will flood to the active spoke-sdp and saps, never to the EVPN PEs because they are part of the same SHG.

Auto-Derived Route-Distinguisher (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 supported as well (not bgp-vpls in 13.0.R4). Note that a single RD is used per service and not per BGP family/protocol.

The following rules apply:

- The VPLS RD is selected based on the following precedence:
  - Manual RD or auto-rd always take precedence when configured.
  - If no manual/auto-rd configuration, the RD is derived from the **bgp-ad>vpls-id**.
  - If no manual/auto-rd/vpls-id configuration, the RD is derived from the **bgp-evpn>evi**.
  - If no manual/auto-rd/vpls-id/evi configuration, there will not be RD and the service will fail.
- The selected RD (see above rules) will be displayed by the **Oper Route Dist** field of the **show service id bgp** command.
- The service supports dynamic RD changes, for instance, the CLI allows the vpls-id be changed dynamically, even if it is used to auto-derive the service RD for **bgp-ad**, **bgp-vpls**, or **bgp-mh**.
  Note — When the RD changes, the active routes for that VPLS will be withdrawn and re-advertised with the new RD.
- If one of the mechanisms to derive the RD for a specified service is removed from the configuration, the system will select a new RD based on the above rules. For example, if the vpls-id is removed from the configuration, the routes will be withdrawn, the new RD selected from the evi, and the routes re-advertised with the new RD.

  Note — This reconfiguration will fail if the new RD already exists in a different VPLS/epipe.
- Because the **vpls-id** takes precedence over the evi when deriving the RD automatically, adding **evpn** to an existing **bgp-ad** service will not impact the existing RD - this is important to support **bgp-ad** to **evpn** migration.

EVPN Multi-Homing in VPLS Services

EVPN multi-homing 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) multi-homed to the EVPN PEs. An **ethernet-segment** is associated with port, LAG, or SDP objects and is shared by all the services defined on those objects.

Each **ethernet-segment** has a unique identifier called **esi** (Ethernet Segment Identifier) that is 10 bytes long and is manually configured in the 7x50.

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 multi-homing implementation in an EVPN-MPLS service.

EVPN All-Active Multi-Homing

As described in RFC7432, all-active multi-homing 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. LACP is optional.

Three different procedures are implemented in 7x50 SROS to provide all-active multi-homing for a specified ethernet-segment:

- DF (Designated Forwarder) election
- Split-horizon
- Aliasing

Figure 124 shows the need for DF election in all-active multi-homing.

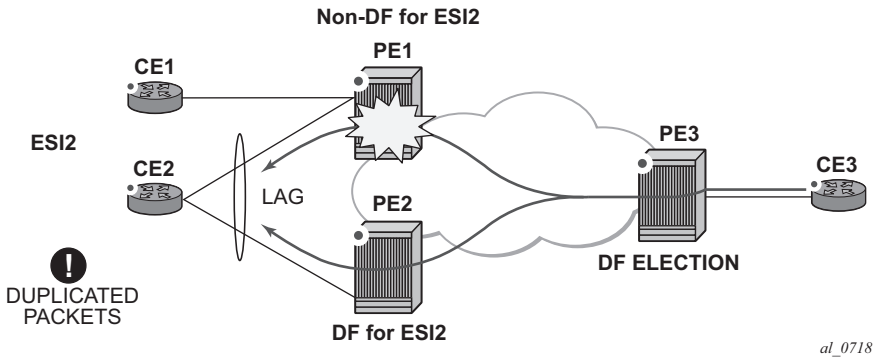

Figure 124: DF Election

The DF election in EVPN all-active multi-homing avoids duplicate packets on the multi-homed 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 will forward BUM traffic from the EVPN network toward the ES SAPs (the multi-homed CE). The non-DF PEs will not forward BUM traffic to the local ethernet-segment SAPs.

Note — 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 125 shows the EVPN split-horizon concept for all-active multi-homing.

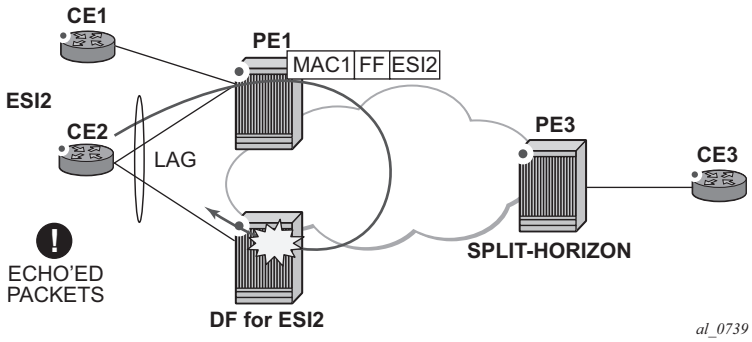

Figure 125: Split-Horizon

The EVPN split-horizon procedure ensures that the BUM traffic originated by the multi-homed 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) will send 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 will find the ESI label that will identify its local ethernet-segment ESI2. The BUM packet will be replicated to other local CEs but not to the ESI2 SAP.

Figure 126 shows the EVPN aliasing concept for all-active multi-homing.

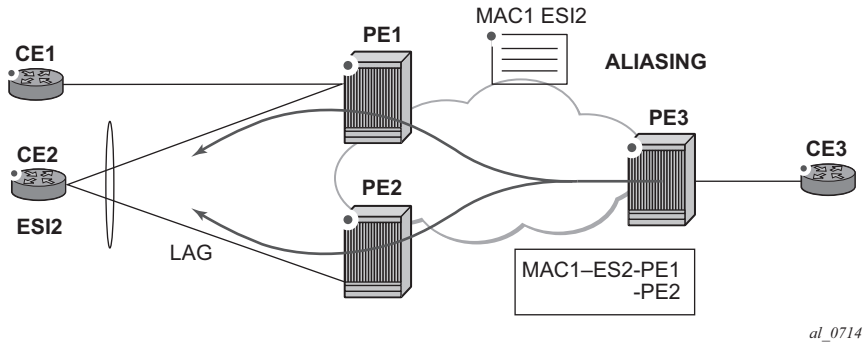

Figure 126: Aliasing

Because CE2 is multi-homed 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 will associate 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.

All-Active Multi-Homing Service Model

The following shows an example PE1 configuration that provides all-active multi-homing to the CE2 described in Figure 126.

```
*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 1.1.1.1:0
  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
      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 1.1.1.1:0
  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
    no shutdown
    auto-bind-tunnel resolution any
sap lag-1:1 create
exit
```

The preceding configuration will enable the all-active multi-homing 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 can be associated with the **ethernet-segment**. This LAG will be exclusively used for EVPN multi-homing. 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 and PBB-EVPN services. **Note** — The **source-bmac-lsb** attribute must be defined for PBB-EVPN (so that it will only be used in PBB-EVPN, and ignored by EVPN). Other than EVPN-MPLS and PBB-EVPN I-VPLS/Epipe services, no other Layer-2 services are allowed in the same **ethernet-segment** (regular VPLS or EVPN-VXLAN SAPs defined on the **ethernet-segment** will be kept operationally down).
- Only one sap per service can be part of the same **ethernet-segment**.

ES Discovery and DF Election Procedures

The ES (Ethernet Segment) discovery and DF election is implemented in three logical steps, as shown in Figure 127.

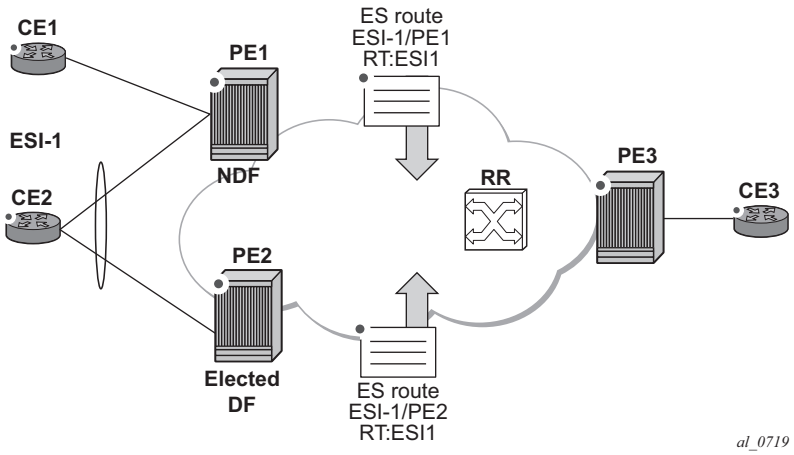

Figure 127: ES Discovery and DF Election

Step 1 - ES Advertisement and Discovery

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 will advertise an ES route for ESI-1. They will 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 will allow the RR to keep the dissemination of the ES routes under control.

In addition to the ES route, PE1 and PE2 will advertise AD per-ESI routes and AD per-EVI routes.

- AD per-ESI routes will 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 procedures.

Step 2 - DF Election

Once 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 7x50 SROS is **service-carving** (as per RFC7432). 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 will be 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 will only consider a specified remote IP address as candidate for the DF election algorithm for a specified service if, in addition to 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), will 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 will behave 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 due to "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 will allow 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 will NOT advertise ES routes until the boot timer expires. This will guarantee that the peer ES PEs don't 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 will run 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 127, PE1 and PE2 are configured with ESI-1. Given that V(10) mod N(2) = 0, PE1 will be 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 service <evi> to <evi> primary.**
  - The system will be the PE forwarding/multicasting traffic for the **evi** identifiers included as primary. The PE will be secondary (non-DF) for the non-specified **evis**.
  - 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 will remain non-DF irrespective of the primary status.
  - For example, as shown in Figure 127: if PE1 is configured with service-carving manual evi 1 to 100 primary and PE2 with service-carving manual evi 101 to 200 primary, then PE1 will be the primary PE for service VPLS 10 and PE2 the secondary PE.
- When service-carving is disabled, the lowest originator IP will win the election for a specified service and ESI:

config>service>system>bgp-evpn>ethernet-segment> 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
===============================================================================
```

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) will instruct the data path to remove the LAG SAP (associated with the ESI) from the default flooding list for BUM traffic. On PE1 and PE2, both LAG SAPs will 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 due to 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
===============================================================================
```

Note — When PE1 (non-DF) and PE2 (DF) exchange BUM packets for **evi 1**, all those packets will be sent including the ESI label at the bottom of the stack (in both directions). The ESI label being used 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
===============================================================================
```

Aliasing

Following the example in Figure 127, if the service configuration on PE3 has ECMP > 1, PE3 will add PE1 and PE2 to the list of next-hops for ESI-1. As soon as PE3 receives a MAC for ESI-1, it will start 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.

Note — 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 will perform 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
    vxlan
        shutdown
    exit
    mpls
        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
```

Network Failures and Convergence for All-Active Multi-Homing

Figure 128 shows the behavior on the remote PEs (PE3) when there is an **ethernet-segment** failure.

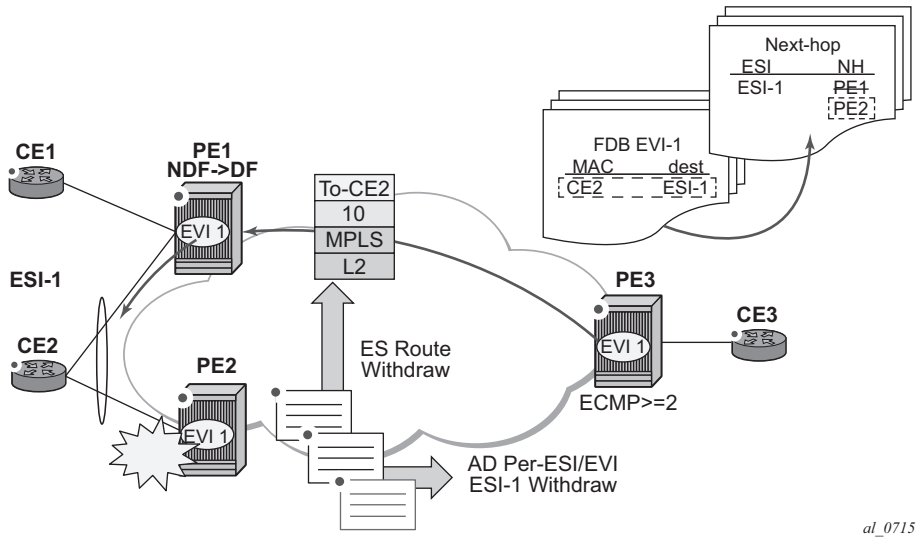

Figure 128: All-Active Multi-Homing ES Failure

The unicast traffic behavior on PE3 is as follows:

1. PE3 can only forward MAC DA = CE2 to both PE1 and PE2 when the MAC advertisement route 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 to CE2 to PE1 only. PE3 does not need to wait for the withdrawal of the individual MAC.
3. The same behavior would be followed if the failure had been at PE1.
4. If after (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).

1. Reception of ES route update (local ES shutdown/no shutdown or remote route)
2. New AD-ES route update/withdraw
3. New AD-EVI route update/withdraw
4. Local ES port/SAP/service shutdown
5. Service carving range change (affecting the evi)
6. Multi-homing mode change (single/all active to all/single-active)

Logical Failures on Ethernet Segments and Black-Holes

Be aware of the effects triggered by certain 'failure scenarios'; some of these scenarios are shown in Figure 129:

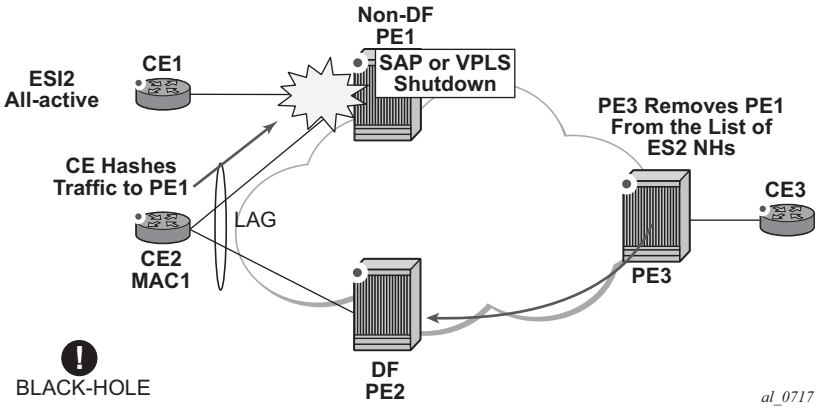

Figure 129: Black-hole Caused by SAP/SVC Shutdown

If an individual VPLS service is **shutdown** in PE1 (the example is also valid for PE2), the corresponding LAG SAP will go *oper-down*. This event will trigger the withdrawal of the AD per-EVI route for that particular SAP. PE3 will remove PE1 of its list of aliased next-hops and PE2 will take over as DF (if it was not the DF already). However, this will 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 will be unaffected, 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 will be brought operationally down (**StandbyforMHprotocol**) and so will 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 will remain operationally up.

Transient Issues Due to MAC Route Delays

Some situations may cause potential transient issues to occur. These are shown in Figure 130and explained below.

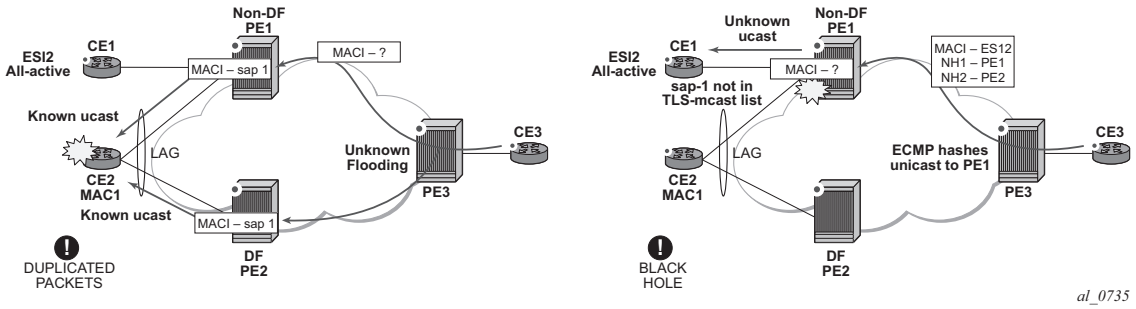

Figure 130: Transient Issues Caused by "slow" MAC Learning

Transient packet duplication caused by delay in PE3 to learn MAC1:

This scenario is illustrated by the diagram on the left in Figure 130. In an all-active multi-homing 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 might send duplicated packets to the CE.

In an all-active multi-homing 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 might 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 will know that the received packet is an unknown unicast packet, therefore, the NDF (PE1) will not send the packets to the CE and there will not be 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 black-hole caused by delay in PE1 to learn MAC1:

This case is illustrated by the diagram on the right in Figure 130. In an all-active multi-homing 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 will be black-holed.

As soon as PE1 learns MAC1, the black-hole will be resolved.

EVPN Single-Active Multi-Homing

The 7x50 SROS supports single-active multi-homing on access LAG SAPs, regular SAPs, and spoke-SDPs for a specified VPLS service. For LAG SAPs, the CE will be configured with a different LAG to each PE in the ethernet-segment (as opposed to a single LAG in an all-active multi-homing).

The following 7x50 SROS procedures support EVPN single-active multi-homing for a specified ethernet-segment:

- DF (Designated Forwarder) election

  As in all-active multi-homing, DF election in single-active multi-homing determines the forwarding for BUM traffic from the EVPN network to the ethernet-segment CE. Also, in single-active multi-homing, DF election also determines the forwarding of any traffic (unicast/BUM) and in any direction (to/from the CE).

- Backup PE

  In single-active multi-homing, 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 RFC7432 procedure is known as 'Backup PE' and is shown in Figure 131 for PE3.

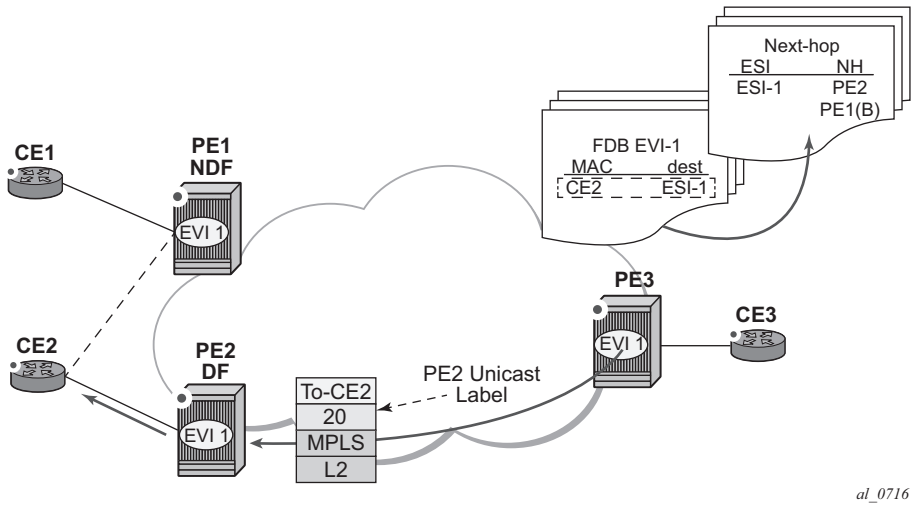

Figure 131: Backup PE

Single-Active Multi-Homing Service Model

The following shows an example of PE1 configuration that provides single-active multi-homing to CE2, as shown in Figure 131.

```
*A:PE1>config>service>system>bgp-evpn# info
----------------------------------------------
  route-distinguisher 1.1.1.1:0
  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
      no shutdown
      auto-bind-tunnel resolution any
  spoke-sdp 1:1 create
  exit
```

The PE2 example configuration for this secanrio is as follows:

```
*A:PE1>config>service>system>bgp-evpn# info
----------------------------------------------
  route-distinguisher 1.1.1.1:0
  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
      no shutdown
      auto-bind-tunnel resolution any
  spoke-sdp 2:1 create
  exit
```

In single-active multi-homing, the non-DF PEs for a specified ESI will block unicast and BUM traffic in both directions (upstream and downstream) on the object associated with the ESI. Other than that, single-active multi-homing is similar to all-active multi-homing with the following differences:

- The **ethernet-segment** will be configured for single-active: **service>system>bgp-evpn>ethernet-segment>multi-homing single-active**.
- The advertisement of the ESI-label in a per-ESI AD route is optional for **single-activ**e ethernet-segments. The user can control the no advertisement of the ESI label by using the following command: **service>system>bgp-evpn>ethernet-segment>multi-homing single-active no-esi-label**. By default, the ESI label is used for single-active ESs too.
- For single-active multi-homing, the ethernet-segment can be associated with a **port** and **sdp**, as well as a **lag-id**, as shown in Figure 131, where:
  - **port** would be used for single-active sap redundancy without the need for lag.
  - **sdp** would be used for single-active spoke-sdp redundancy.
  - **lag** would be 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 multi-homing, when the PE is non-DF for the service, the saps/spoke-sdps on the ethernet-segment will be down and show **StandByForMHProtocol** as the reason.
- From a service perspective, single-active multi-homing 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 multi-homed ports on the CE will be part of the different LAGs (a LAG per multi-homed PE will be used in the CE). The non-DF PE for each service can signal that the sap is oper-down if eth-cfm fault-propagation-enable {use-if-tlv|suspend-ccm} is configured.
  - **Regular Ethernet 802.1q/ad ports**
    In this case, the multi-homed ports on the CE/network will not be part of any LAG. Eth-cfm can also be used for non-DF indication to the multi-homed device/network.
  - **Active-standby PWs**
    In this case, the multi-homed 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 will make use of the LDP PW status bits to signal that the spoke-sdp is oper-down on the PE side.

ES and DF Election Procedures

In all-active multi-homing, the non-DF keeps the SAP up, although it removes it from the default flooding list. In the single-active multi-homing implementation the non-DF will bring the SAP/SDP-binding operationally down. Refer to the ES Discovery and DF Election Procedures on page 1112 for more information.

The following **show** commands display the status of the single-active ESI-7413 in the non-DF.
Note — The associated spoke-SDP is operationally down and it signals PW Status standby to the multi-homed 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
```

Backup PE Function

A remote PE (PE3 in Figure 131) will import the AD routes per ESI, where the single-active flag is set. PE3 will interpret 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 will be learned from a single PE, that is, the DF for that <ESI, EVI>.

The remote PE will install 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 will be the backup PE. Upon receiving an AD per-ES/per-EVI route withdrawal for the ESI from the primary PE, the PE3 will start 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 will not install any backup route in the data path. Upon receiving an AD per-ES/per-EVI route withdrawal for the ESI, it will flush the MACs associated with the ESI.

Network Failures and Convergence for Single-Active Multi-Homing

Figure 132 shows the remote PE (PE3) behavior when there is an ethernet-segment failure.

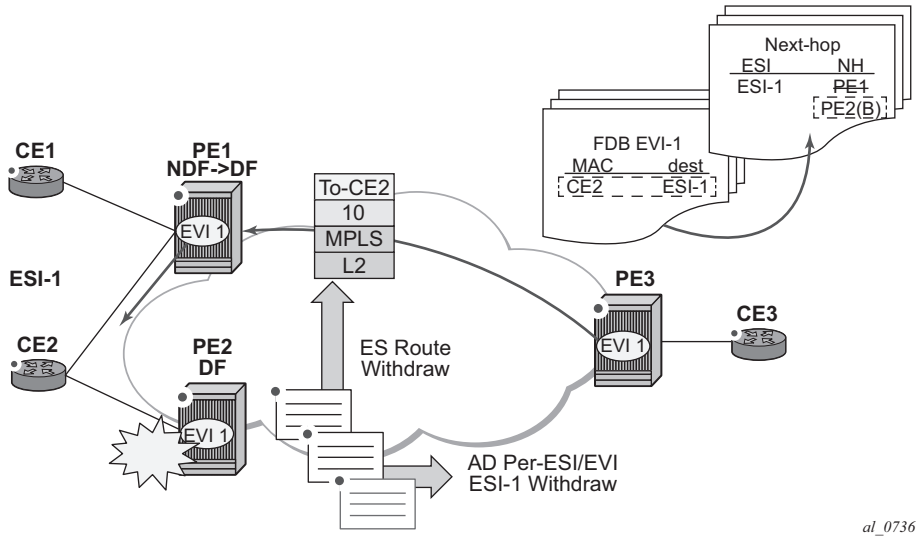

Figure 132: Single-Active Multi-Homing ES Failure

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 to CE2 to PE1 only (the backup PE). PE3 does not need to wait for the withdrawal of the individual MAC.
3. After the (2) PE2 withdraws its MAC advertisement route, PE3 will treat 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 multi-homing. In this case, the DF will forward traffic to CE2 once the **esi-activation-timer** expiration occurs (the timer kicks in when there is a transition from non-DF to DF).

Logical Failures on Ethernet Segments and Black-Holes

Be aware of the effects triggered by certain 'failure scenarios'; some of these scenarios are shown in Figure 133:

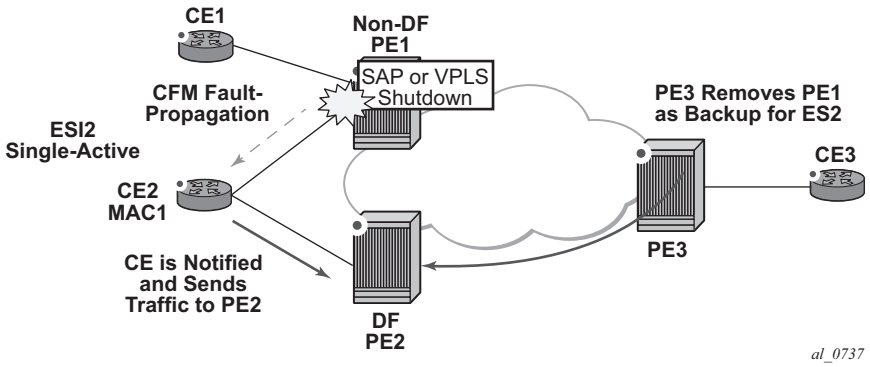

Figure 133: Single-Active Multi-Homing SAP/SDP/Service Shutdown

Common effects to consider are:

- If an individual VPLS service is administrative **shutdown** in PE1, the corresponding SAP/SDP-binding will go oper-down. This event will trigger the withdrawal of the AD per-EVI route for that particular SAP/SDP-binding. PE3 will apply its backup mechanisms and PE2 will take over as DF (if it was not the DF already). To signal the fault to CE2:
  - Eth-cfm must be used between the PEs and the multi-homed CE - if the connectivity is based on SAPs. A down MEP on the SAP going down will propagate the fault to a MEP on the CE. This network event will not produce any black-holing (in the all-active multi-homing case, this would partially black-hole the traffic).
  - PW status bits must be used between the PEs and the multi-homed CE - if the connectivity is based on pseudowires.
- The same result occurs if the ES SAP/SDP-binding is administrative **shutdown** instead of the service.

BGP-EVPN Control Plane for PBB-EVPN

PBB-EVPN uses a reduced subset of the routes and procedures described in RFC7432. The supported routes are:

- ES routes
- MAC/IP routes
- Inclusive Multicast Ethernet Tag routes.

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: Taken from the RD of the B-VPLS service within the BGP context. **Note** —The RD can be configured or derived from the **bgp-evpn evi** value.
- Ethernet Tag ID: Encodes the ISID for a specified I-VPLS.
- IP address length: Always 32.
- Originating router's IP address: Carries the system address (IPv4 only).
- PMSI attribute:
  - Tunnel type = Ingress replication (6).
  - Flags = Leaf no required.
  - MPLS label: Carries the MPLS label allocated for the service in the high-order 20 bits of the label field.
    Note — This label will be the same label used in the BMAC routes for the same B-VPLS service unless **bgp-evpn mpls ingress-replication-bum-label** is configured in the B-VPLS service.
  - Tunnel end-point = equal to the originating IP address.

EVPN Route Type 2 - MAC/IP Advertisement Route (or BMAC Routes)

The 7x50 will generate this route type for advertising BMAC 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-BMACs (**source-bmacs**) and dedicated-BMACs (**es-bmacs**).

The route type 2 generated by the 7x50 uses the following fields and values:

- Route Distinguisher—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):
  - ESI = 0 for the advertisement of source-bmac, es-bmacs, sap-bmacs, or sdp-bmacs if no multi-homing or single-active multi-homing is used.
  - ESI=MAX-ESI (0xFF..FF) in the advertisement of es-bmacs used for all-active multi-homing.
  - ESI different from zero or MAX-ESI for learned BMACs on B-SAPs/SDP-bindings if EVPN multi-homing is used on B-VPLS SAPs and SDP-bindings.
- Ethernet Tag ID: 0.
- MAC address length: Always 48.
- BMAC 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 will be 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 CMAC flush purposes if per ISID load balancing (single-active multi-homing) 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, CMAC flush through the withdrawal of the BMAC CANNOT be done (other ESIs are still working); therefore, the mac mobility extended community is used to signal CMAC 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 BMAC from the failing node.
  - es-bmacs will be advertised as static (sticky bit set).
  - Source-bmacs will be advertised as static MACs (sticky bit set). In the case of an update, if advertised to indicate that CMAC flush is needed, the mac mobility extended community will be added to the BMAC route including a higher sequence number (than the one previously advertised) in addition to the sticky bit.

EVPN Route Type 4 - Ethernet Segment Route

This route type is used for DF election as described in section BGP-EVPN Control Plane for MPLS Tunnels on page 1095.

Note — The EVPN route type 1—Ethernet Auto Discovery route is not used in PBB-EVPN.

PBB-EVPN for I-VPLS and PBB Epipe Services

The 7x50 SROS 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 134 shows the PBB-EVPN model in 7x50 SROS.

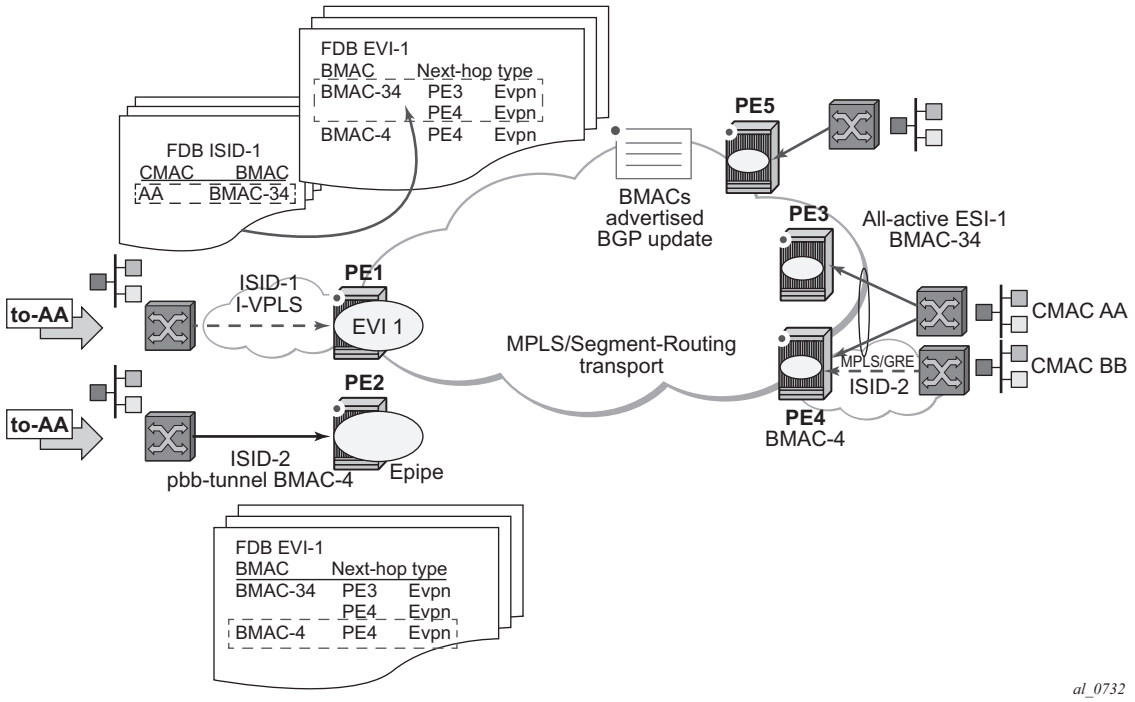

Figure 134: PBB-EVPN for I-VPLS and PFF Epipe Services

Each PE in the B-VPLS domain will advertise its **source-bmac** as either configured in **(b)vpls>pbb>source-bmac** or auto-derived from the chassis mac. The remote PEs will install the advertised BMACs 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 134, when a frame with MAC DA = AA gets to PE1, a mac lookup is performed on the I-VPLS FDB and BMAC-34 is found. A BMAC lookup on the B-VPLS FDB will yield 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 will be 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 will send the frame encapsulated into a PBB packet with BMAC DA = Group BMAC for the ISID. That packet will be distributed to all the PEs where the ISID is defined and will contain the EVPN label distributed by the Incusive Multicast routes for that ISID, in addition to the transport label.

For PBB-Epipes, all the traffic is sent in a unicast PBB packet to the BMAC 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 b-vpls create
  description "pbb-evpn-service"
  service-mtu 2000
  pbb
    source-bmac 00:00:00:00:00:03
  bgp
  bgp-evpn
    evi 1
    vxlan
      shutdown
    mpls
      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 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 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) on page 1100.

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** (See PBB-EVPN Multi-Homing in I-VPLS and PBB Epipe Services on page 1142 for more information)
  - **vpls>isid-policies**
  - **vpls>static-mac**
  - **vpls>sap/sdp-binding>static-isid**
  - **bgp-evpn>mac-advertisement -** this command will only have affect on the 'learned' BMACs on saps or sdp-bindings and not on the system BMAC 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**

Flood Containment for I-VPLS Services

In general, PBB technologies in 7x50 SROS 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 will create an MFIB per I-VPLS ISID on the B-VPLS instance. That MFIB supports SAP/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 135 shows how the MFIBs are populated in PBB-EVPN.

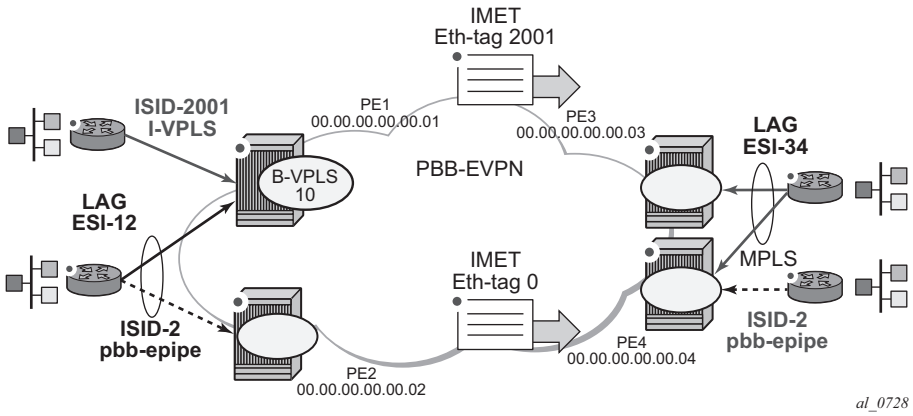

Figure 135: PBB-EVPN and I-VPLS Flooding Containment

When the B-VPLS 10 is enabled, PE1 will advertise as follows:

- A BMAC route containing PE1's system BMAC (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 will allow the remote B-VPLS 10 instances to add an entry for PE1 in the default multicast list.
  Note — The MPLS label that will be 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 will depend 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 will advertise as follows:

- An additional inclusive multicast route with Ethernet-tag = 2001. This will allow 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 will be 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 (7x50 SROS behavior).
- The B-VPLS 10 also allows the ISID flooding to legacy PBB networks via B-SAPs or B-SDPs. The legacy PBB network BMACs will be 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.

PBB-EVPN and PBB-VPLS Integration

The 7x50 SROS EVPN implementation supports draft-ietf-bess-evpn-vpls-seamless-integ 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 PBB-EVPN and PBB-VPLS Integration on page 1141 are also supported in B-VPLS services so that B-VPLS SAP/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.

PBB-EVPN Multi-Homing in I-VPLS and PBB Epipe Services

The 7x50 SROS PBB-EVPN implementation supports all-active and single-active multi-homing for I-VPLS and PBB Epipe services.

PBB-EVPN multi-homing reuses the **ethernet-segment** concept described in section 'EVPN Multi-Homing in VPLS Services on page 1108'. However, unlike EVPN-MPLS, PBB-EVPN does not use AD routes; it uses BMACs for split-horizon checks and aliasing.

System BMAC Assignment in PBB-EVPN

Draft-ietf-l2vpn-pbb-evpn describes two types of BMAC assignments that a PE can implement:

- Shared BMAC addresses that can be used for single-homed CEs and a number of multi-homed CEs connected to ethernet-segments.
- Dedicated BMAC addresses per ethernet-segment.

In this document and in 7x50 SROS 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**

BMAC selection and use depends on the multi-homing model; for single-active mode, the type of BMAC will impact the flooding in the network as follows:

- All-active multi-homing requires **es-bmacs**.
- Single-active multi-homing can use **es-bmacs** or **source-bmacs**.
  - The use of **source-bmacs** minimizes the number of BMACs being advertised but has a larger impact on CMAC flush upon ES failures.
  - The use of **es-bmacs** optimizes the CMAC flush upon ES failures at the expense of advertising more BMACs.

PBB-EVPN all-active multi-homing service model

Figure 136 shows the use of all-active multi-homing in the 7x50 SROS PBB-EVPN implementation.

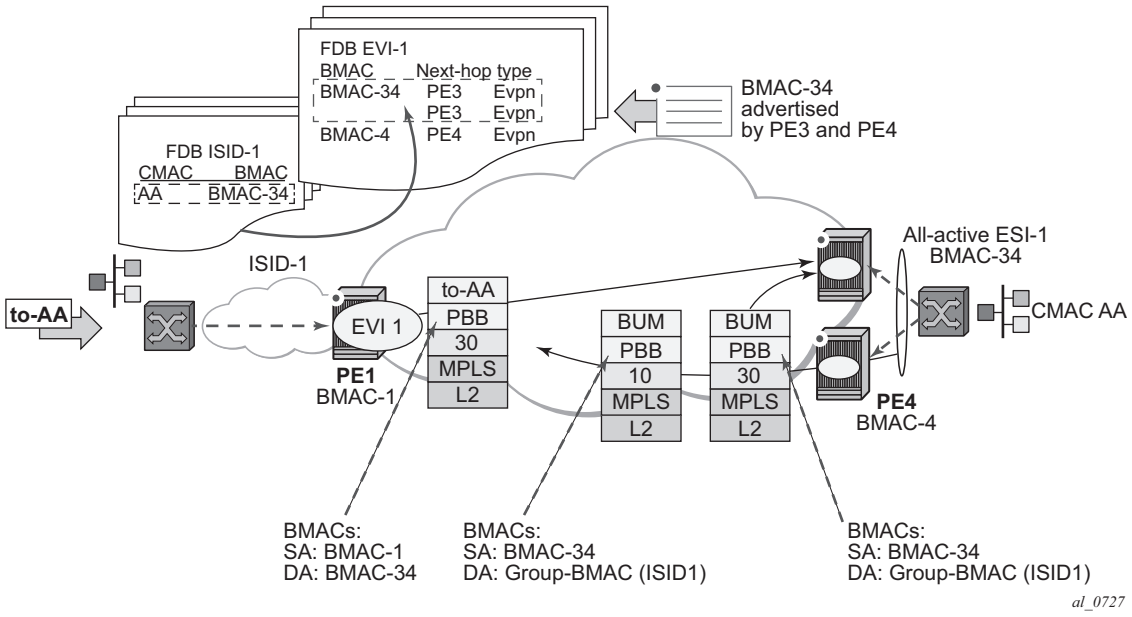

Figure 136: PBB-EVPN All-Active Multi-Homing

For example, the following shows the ESI-1 and all-active configuration in PE3 and PE4. As in EVPN-MPLS, all-active multi-homing is only possible if a LAG is used at the CE. All-active multi-homing uses es-bmacs, that is, each ESI will be assigned a dedicated BMAC. All the PEs part of the ES will source traffic using the same **es-bmac**.

In Figure 136 and the following configuration, the **es-bmac** used by PE3 and PE4 will be BMAC-34, i.e. 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 3.3.3.3:0
  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
  bgp-evpn
    evi 1
    mpls
      no shutdown
      ecmp 2
      auto-bind-tunnel resolution any
  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 4.4.4.4:0
  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
  bgp-evpn
    evi 1
    mpls
      no shutdown
      ecmp 2
      auto-bind-tunnel resolution any
  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 will enable the all-active multi-homing 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 multi-homing. 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-bmac-lsb <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 BMACs that can be associated with the **ethernet-segment**, i.e. **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 will be restricted by the CLI.
- If **service-carving** is configured in the ESI, the DF election algorithm will be 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).
- A **service-carving mode manual** option is added so that the user can control what PE is DF for a specified ISID. The PE will be 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 will only block 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 BMAC (as opposed to the ESI label in EVPN-MPLS). In Figure 136 PE3 sends packets using BMAC 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 BMAC sent by all the PEs part of the same ESI. See the following section for more information. In Figure 136 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, i.e. PE1, the configuration of the PBB-Epipe will point 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 multi-homing, Alcatel-Lucent 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 will forward the traffic to the CE.

Note — PBB-Epipe traffic always uses BMAC DA = unicast; therefore, the DF cannot check whether the inner frame is unknown unicast or not based on the group BMAC. Therefore, the use of an EVPN BUM label is highly recommended.

Aliasing for PBB-epipes with all-active multi-homing only works if shared-queuing or ingress policing is enabled on the ingress PE epipe. In any other case, the IOM will send the traffic to a single destination (no ECMP will be used in spite of the **bgp-evpn mpls ecmp** setting).

All-active multi-homed **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 136:

```
*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**, i.e. 00:00:00:00:00:34, has two associated next-hops, i.e. 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
-------------------------------------------------------------------------------
===============================================================================
```

Network failures and convergence for all-active multi-homing

ES failures are resolved by the PEs withdrawing the **es-bmac**. The remote PEs will 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 will be flushed from the B-VPLS FDB and all the CMACs associated with it will be flushed too.

The following events will trigger a withdrawal of the **es-bmac** and the corresponding next-hop update in the remote PEs:

- B-VPLS transition to oper-down status.

- Change of **pbb>source-bmac**.
- Change of **es-bmac** (or removal of **pbb use-es-bmac**).
- Ethernet-segment transition to oper-down status.

Note — Individual saps going oper-down in an ES will not generate any BGP withdrawal or indication so that the remote nodes can flush their CMACs. 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.

PBB-EVPN Single-Active Multi-Homing Service Model

In single-active multi-homing, the non-DF PEs for a specified ESI will block unicast and BUM traffic in both directions (upstream and downstream) on the object associated with the ESI. Other than that, single-active multi-homing will follow the same service model defined in the section 'PBB-EVPN all-active muti-homing service model' with the following differences:

- The **ethernet-segment** will be configured for **single-active**: **service>system>bgp-evpn>ethernet-segment>multi-homing single-active**.
- For single-active multi-homing, the **ethernet-segment** can be associated with a port and sdp, as well as a **lag**.
- From a service perspective, single-active multi-homing 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 multi-homing is supported for PBB-Epipes only in two-node scenarios with local switching.
- While all-active multi-homing only uses **es-bmac** assignment to the ES, single-active multi-homing 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 will be configured in the B-VPLS and the same procedures explained in PBB-EVPN all-active multi-homing service model on page 1143 will follow with one difference. While in all-active multi-homing all the PEs part of the ESI will source the PBB packets with the same source es-bmac, single-active multi-homing requires the use of a different **es-bmac** per PE.
  - A non-dedicated **source-bmac** can be used. In this case, the user will not configure **pbb>use-es-bmac** and the regular **source-bmac** will be 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 multi-homed ESIs has a different impact on CMAC flushing, as shown in Figure 137.

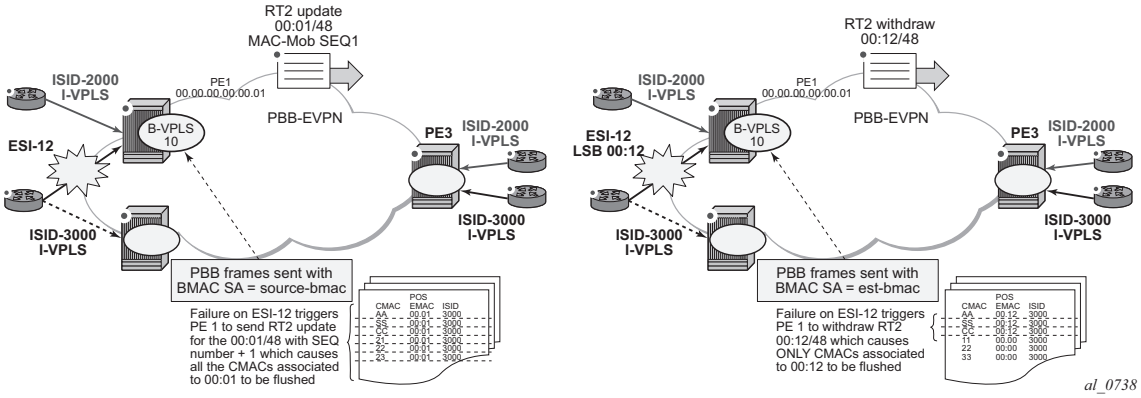

Figure 137: Source-Bmac Versus Es-Bmac CMAC Flushing

- If **es-bmacs** are used as shown in the representation on the right in Figure 137, a less-impacting CMAC flush is achieved, therefore, minimizing the flooding after ESI failures. In case of ESI failure, PE1 will withdraw the **es-bmac** 00:12 and the remote PE3 will only flush the CMACs associated with that **es-bmac** (only the CMACs behind the CE are flushed).
- If **source-bmacs** are used, as shown on the left-hand side of Figure Figure 137, in case of ES failure, a BGP update with higher sequence number will be issued by PE1 and the remote PE3 will flush all the CMACs associated with the **source-bmac**. Therefore, all the CMACs behind the PE's B-VPLS will be flushed, as opposed to only the CMACs 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 multi-homed ports on the CE will NOT be part of the same LAG. The non-DF PE for each service may signal that the LAG sap is oper-down by using **eth-cfm fault-propagation-enable {use-if-tlv|suspend-ccm}**.
  - Regular Ethernet 802.1q/ad ports: In this case, the multi-homed ports on the CE/network will not be part of any LAG. The non-DF PE for each service will signal that the sap is oper-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 will make use of the LDP PW status bits to signal that the spoke-sdp is standby at the PE side. Alcatel-Lucent recommends that the CE suppresses the signaling of PW status standby.

Network Failures and Convergence for Single-Active Multihoming

ESI failures are resolved depending on the BMAC address assignment chosen by the user:

- If the BMAC address assignment is based on the use of **es-bmacs**, DF and non-DFs will send the **es-bmac/ESI=0** for a specified ESI. Each PE will have 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 on a PE:
  - The PE will withdraw its **es-bmac** route triggering a mac-flush of all the CMACs associated with it in the remote PEs.
- If the BMAC address assignment is based on the use of **source-bmac**, DF and non-DFs will advertise their respective **source-bmacs**. In case of an ES failure:
  - The PE will re-advertise its **source-bmac** with a higher sequence number (the new DF will not readvertise its s**ource-bmac**).
  - The far-end PEs will interpret a **source-bmac** advertisement with a different sequence number as a flush-all-from-me message from the PE detecting the failure. They will flush all the CMACs associated with that BMAC in all the ISID services.

The following events will trigger a CMAC flush notification. A 'CMAC flush notification' means the withdrawal of a specified BMAC or the update of BMAC with a higher sequence number (SQN). Both BGP messages will make the remote PEs flush all the CMACs associated with the indicated BMAC:

- B-VPLS transition to oper-down status. This will trigger the withdrawal of the associated BMACs, irrespective of the **use-es-bmac** setting.
- Change of **pbb>source-bmac**. This will trigger the withdrawal and re-advertisement of the **source-bmac**, causing the corresponding CMAC flush in the remote PEs.
- Change of **es-bmac** (removal of **pbb use-es-bmac**). This will trigger the withdrawal of the **es-bmac** and re-advertisement of the new **es-bmac**.
- Ethernet-Segment (ES) transition to oper-down or admin-down status. This will trigger an **es-bmac** withdrawal (if **use-es-bmac** is used) or an update of the source-bmac with a higher SQN (if n**o use-es-bmac** is used).
- Service Carving Range change for the ES. This will trigger 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 will trigger 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 oper-down will not generate any BGP withdrawal or indication so that the remote nodes can flush their CMACs. 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.

PBB-Epipes and EVPN Multi-Homing

EVPN multi-homing 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 don't support spoke-sdps that are associated with EVPN Ethernet Segments (ES).
- PBB-EVPN Epipes support all-active EVPN multi-homing as long as no local-switching is required in the Epipe instance where the ES is defined.
- PBB-EVPN Epipes support single-active EVPN multi-homing only in a two-node case scenario.

Figure 138 shows the EVPN MH support in a three-node scenario.

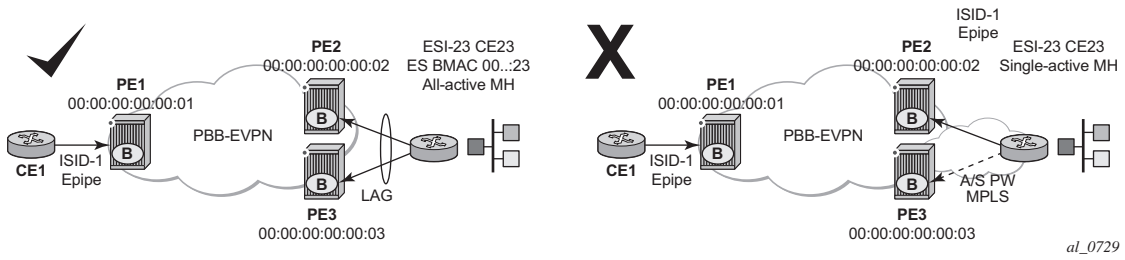

Figure 138: PBB-EVPN MH in a Three-Node Scenario

EVPN MH support in a three-node scenario has the following characteristics:

- All-active EVPN multi-homing is fully supported (diagram on the left in Figure 138). CE1 might also be multi-homed 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 BMAC.
- Single-active EVPN multi-homing is NOT supported in a three (or more)-node scenario (diagram on the right in Figure 138). Since PE1's Epipe **pbb-tunnel** can only point at a single remote BMAC and single-active multi-homing requires the use of separate BMACs on PE2 and PE3, the scenario is not possible and not supported irrespective of the ES association to port/LAG/sdps.
- Irrespective of the EVPN multi-homing type, the CLI prevents the user from adding a spoke-sdp to an Epipe, if the corresponding SDP is part of an ES.

Figure 139 shows the EVPN MH support in a two-node scenario.

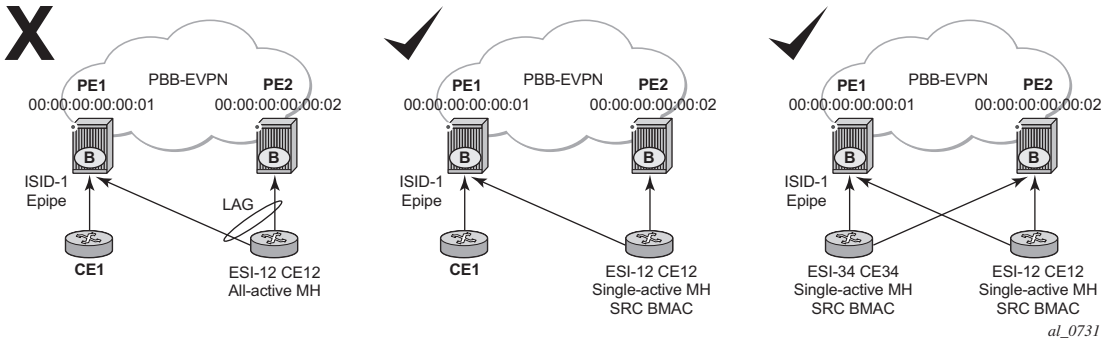

Figure 139: PBB-EVPN MH in a Two-Node Scenario

EVPN MH support in a two-node scenario has the following characteristics, as shown in Figure 139:

- All-active multi-homing is not supported for redundancy in this scenario because PE1's **pbb-tunnel** cannot point at a locally defined ES-BMAC. This is represented in the left-most scenario in Figure 139.
- Single-active multi-homing is supported for redundancy in a two-node three or four SAP scenario, as displayed by the two right-most scenarios in Figure 139).

  In these two cases, the Epipe **pbb-tunnel** will be configured with the source BMAC of the remote PE node.

  When two saps are active in the same Epipe, local-switching is used to exchange frames between the CEs.

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/nd no shutdown), the system will populate the corresponding proxy-ARP/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/SDP-binding will be intercepted and processed. ARP requests and Neighbor Solicitations will be answered by the system if the requested IP address is present in the proxy table.

Figure 140 shows an example of how proxy-ARP is used in an EVPN network. Proxy-ND would work in a similar way. Note that the MAC address notation in the diagram is shortened for readability.

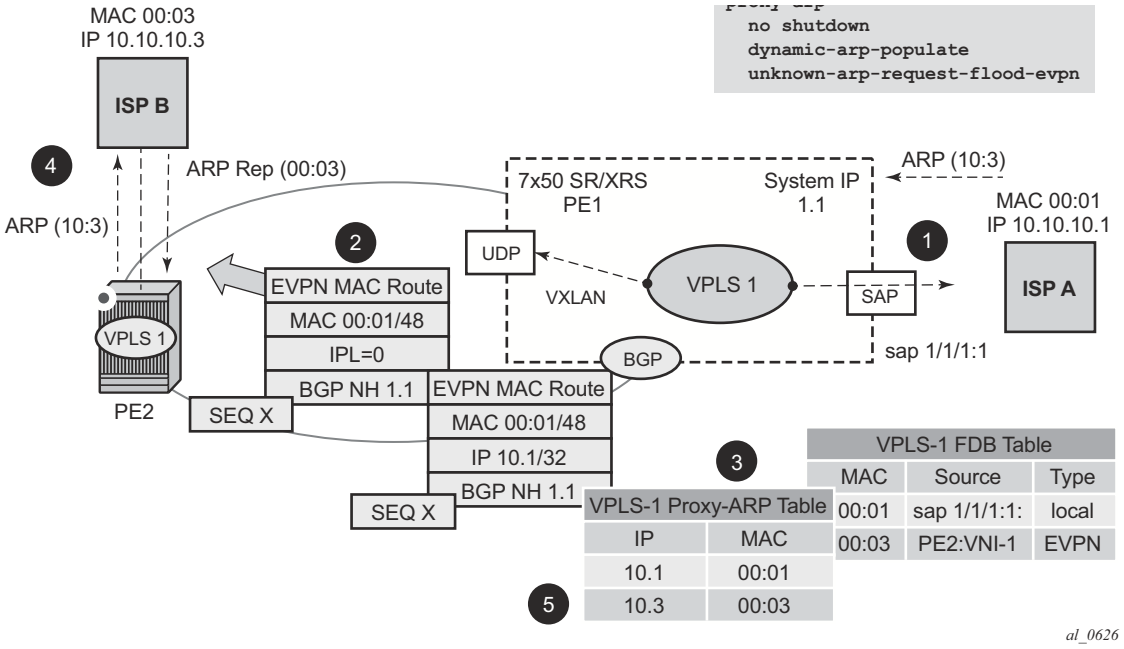

Figure 140: Proxy-ARP Example Usage in an EVNP Network

PE1 is configured as follows:

```
*A:PE1>config>service>vpls# info
----------------------------------------------
vxlan 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
               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 140 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 will be 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 will learn 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/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 will populate proxy-ARP/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 will be 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->MACentry requires the addition of the MAC address to the FDB as either learned or CStatic (conditional static mac) in order to become active (*Status —> active*).

- **age-time <60..86400>** (seconds)

  Specifies the aging timer per proxy-ARP/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..86400>** (seconds)

  If enabled, the system will send 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..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/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/ND offending entry's MAC is replaced by this <mac-address> and advertised in an unsolicited GARP/NA for local SAP/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 will be dropped. An ingress mac-filter has to be configured in order to drop traffic to the **anti-spoof-mac**.

Table 19 shows the combinations that will produce a **Status = Active** proxy-arp entry in the table. The system will only reply to proxy-ARP requests for active entries. Any other combination will result in a **Status = inActv** entry. If the service is not active, the proxy-arp entries will not be active either, irrespective of the FDB entries

NOTE—A static entry is active in the FDB even when the service is down.

Table 19: Proxy-arp Entry combinations

Proxy-arp Entry Type	FDB Entry Type (for the same MAC)
Dynamic	learned
Static	learned
Dynamic	CStatic/Static

Table 19: Proxy-arp Entry combinations

Proxy-arp Entry Type	FDB Entry Type (for the same MAC)
Static	CStatic/Static
EVPN	EVPN
Duplicate	—

When proxy-ARP/ND is enabled on services with all-active multi-homed ethernet-segments, a proxy-arp entry type 'EVPN' might be associated with a 'learned' FDB entry (because the CE can send traffic for the same MAC to all the multi-homed PEs in the ES). If that is the case, the entry will be inactive, as per Table 19.

Proxy-ARP/ND Periodic Refresh, Unsolicited Refresh and Confirm-Messages

When proxy-ARP/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, in order 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 will force the owners of the duplicate IP to reply and cause *dup-detect* to kick in.

Proxy-ND and the Router Flag in Neighbor Advertisement messages

RFC4861 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 use of the "R" flag in NA messages impacts how the hosts select their default gateways when sending packets "off-link". Therefore, it is important that the proxy-ND function on the 7x50 must meet one of the following criteria:

a. Either provide the appropriate R flag information in proxy-ND NA replies

b. Flood the received NA messages if it cannot provide the appropriate R flag when replying

Due to the use of the "R" flag, the procedure for learning proxy-ND entries and replying to NS messages differs from the procedures for proxy-ARP in IPv4: the router or host flag will be added to each entry, and that will determine the flag to use when responding to a NS.

Procedure to Add the R Flag to a Specified Entry

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 as per the command **[no] static <ip-address> <ieee-address> {host | router}**.
- EVPN entries are learned from BGP and the command **evpn-nd-advertise {host | router}** determines the R flag added to them.
- In addition, the **evpn-nd-advertise {host | router}** command will indicate what static and dynamic IP->MAC entries the system will advertise in EVPN. If **evpn-nd-advertise router** is configured, the system should flood the received unsolicited NA messages for hosts. This is controlled by the **[no] host-unsolicited-na-flood-evpn** command. The opposite is also recommended so that the **evpn-nd-advertise host** is configured with the **router-unsolicited-na-flood-evpn**.

BGP-EVPN MAC-Mobility

EVPN defines a mechanism to allow the smooth mobility of MAC addresses from an NVE to another NVE. The 7x50 supports this procedure as well as the MAC-mobility extended community in MAC advertisement routes as follows:

- The 7x50 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 multi-homing 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 multi-homing. 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.

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 7x50 supports 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 mis-configured 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 7x50 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 7x50 then alerts the operator with a trap message. The offending MAC address can be shown using the show service id x 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 7x50 stops sending and processing any BGP MAC advertisement routes for that MAC address until one of the following occurs:

a. The MAC is flushed due to a local event (sap/sdp-binding associated with the MAC fails) or the reception of a remote update with better SEQ number (due to a mac flush at the remote 7x50).

b. The retry <in-minutes> timer expires, which will flush the MAC and restart the process.

Note—The other 7x50s in the VPLS instance will forward the traffic for the duplicate MAC address to the 7x50 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

might 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
         no shutdown
       exit
```

Conditional Static MAC and Protection

The draft-ietf-bess-evpn-overlay 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 7x50, any conditional static mac defined in an EVPN-VXLAN VPLS service will be 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 will be discarded if its source MAC address matches one of these 'protected' MACs.

CFM Interaction with EVPN Services

Ethernet Connectivity & 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 & Diagnostics guides and is not repeated here.

EVPN provides powerful solution architectures. ETH-CFM is supported in the various layer 2 EVPN architectures. Since the destination layer 2 MAC address, unicast or multicast, is ETH-CFM tool dependant (ie. ETH-CC is sent as a 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 will track 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, there is no ETH-CFM functionality installed on the virtual connections. The ETH-CFM functions will only be installed on the supported SAP, spoke, and mesh connections.

When MPs are used in combination with EVPN multi-homing, the following must be considered:

- Behavior of operationally DOWN MEPs on SAPs/SDP-bindings with EVPN multi-homing:
  - All-active multi-homing: no ETH-CFM is expected to be used in this case, since the two (or more) SAPs/SDP-bindings on the PEs will be oper-up and active; however, the CE will have a single LAG and will respond as though it is connected to a single system. In addition to that, **cfm-mac-advertisement** can lead to traffic loops in all-active multi-homing.
  - Single-active multi-homing: DOWN MEPs defined on single-active ethernet-segment SAPs/SDP-bindings will not send any CCMs when the PE is non-DF for the ES and fault-propagation is configured. For single-active multi-homing, the behavior will be equivalent to MEPs defined on BGP-MH saps/binds.

- Behavior for UP MEPs on ES SAPs/SDP-bindings with EVPN multi-homing:
  - All-active multi-homing: 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 will be treated as unknown, and not sent to the non-DF SAP MEP).
  - Single-active multi-homing: UP MEPs should be able to send or receive CFM packets normally.
  - 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.

Due to the above considerations, the use of ETH-CFM in EVPN multi-homed SAPs/SDP-bindings is only recommended on operationally DOWN MEPs and single-active multi-homing. ETH-CFM is used in this case to notify the CE of the DF or non-DF status.

DC GW Policy Based Forwarding/Routing to an EVPN ESI (Ethernet Segment Identifier)

The Nuage VSP (Virtual Services Platform) 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 meeds 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 VSD (Virtual Services Directory). 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 (ESI) that is uniquely associated with the SF.

Refer to the Nuage VSP documentation for more information about the Nuage Service Chaining solution.

The 7x50 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.

Policy Based Forwarding in VPLS Services for Nuage Service Chaining Integration in L2-Domains

Figure 141 shows the 7x50 Service Chaining integration with the Nuage VSP on VPLS services. In this example, the DC GW, 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.

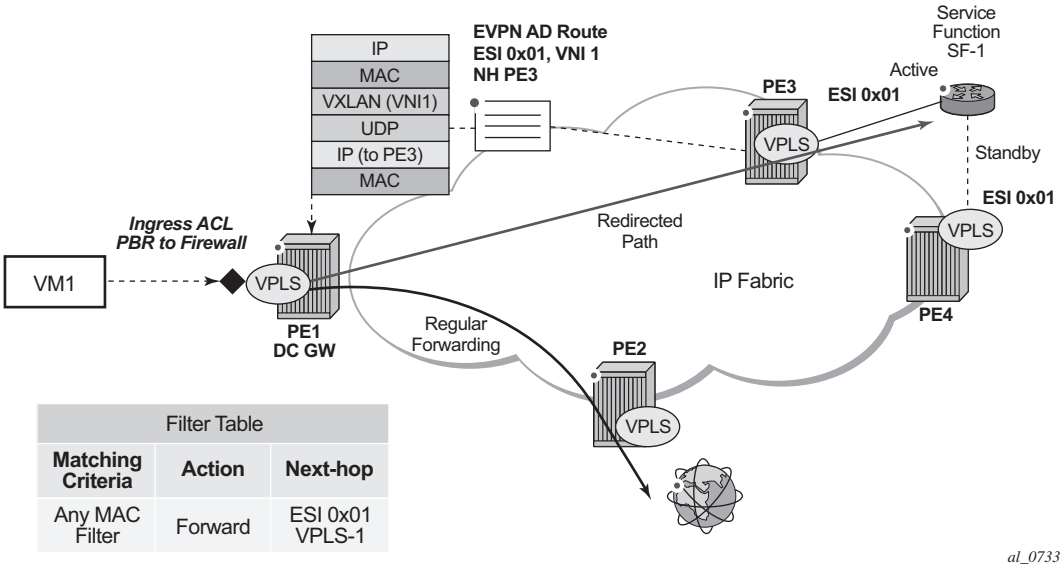

Figure 141: PBF to ESI Function

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 Ethernet Segment Identifier.

In the example shown in Figure 141, the PBF filter will redirect the matching packets to ESI 0x01 in VPLS-1

Note — Figure 141 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 will send 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) will 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:00:00:00:00:00:00:00:01 service-id 301
                exit
            exit
```

When the filter is properly applied to the VPLS service (VPLS-301 in this example), it will show '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:00:00:00:00:00:00:00:01
  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:00:00:00:00:00:00:00:01 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:00:00:00:00:00:00:00:01
===============================================================================
 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 will disappear and the filter will be inactive again. The user can control whether the matching packets are dropped or forwarded if the PBF target cannot be resolved by BGP.

Policy Based Routing in VPRN Services for Nuage Service Chaining Integration in L2-DOMAIN-IRB Domains

Figure 142 shows the 7x50 Service Chaining integration with the Nuage VSP on L2-DOMAIN-IRB domains. In this example, the DC GW, 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.

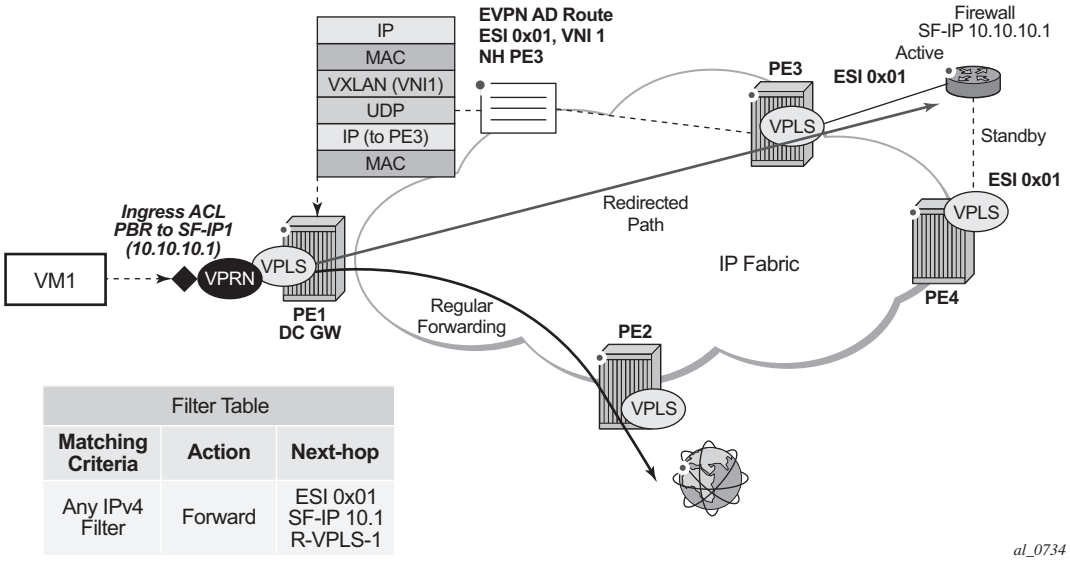

Figure 142: PBR to ESI Function

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/NETCONF management interfaces. The PBR target identifies first service function in the chain (ESI 0x01 in Figure 142, 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 will be, 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
----------------------------------------------
```

Note — 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 will be 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 will show '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     : 172.16.0.253/32
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/IP routes for the MAC (see section '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:                   EvpnS    06/15/15 21:55:27
                            192.0.2.69:301
301       d8:47:01:01:00:0a vxlan:                   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
===============================================================================
```

As for the Layer-2 case, if the AD route is withdrawn or the SF-IP ARP not resolved, the filter will be inactive again. The user can control whether the matching packets are dropped or forwarded if the PBF target cannot be resolved by BGP.

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 for the routes is 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 will hand over the routes to EVPN and the EVPN application will perform the selection.

EVPN and BGP selection criteria are described below.

- EVPN route selection for MAC routes: when two or more routes with the same mac-length/mac but different route key are received, BGP will hand the routes over to EVPN. EVPN will select the route based on the following tie-break order:

  1. Conditional static MACs (local protected MACs)
  2. EVPN ES PBR MACs (see ES PBR MAC routes below)
  3. EVPN static MACs (remote protected MACs)
  4. Data plane learned MACs (regular learning on saps/sdp-bindings)
  5. EVPN MACs with higher SEQ number
  6. Lowest IP (next-hop IP of the EVPN NLRI)
  7. Lowest eth-tag (that will be zero for MPLS and might be different from zero for VXLAN)
  8. Lowest RD

- 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 will be 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 doesn't 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 tie-break order above).
  - If there are two competing ES PBR MAC routes, then the selection goes through the rest of checks (Lowest IP > Lowest RD).
- 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).

- The BGP route selection for the rest of the EVPN routes: regular BGP selection is followed.

Note — In case BGP has to run an actual selection and a given (otherwise valid) EVPN route 'loses' to another EVPN route, the non-selected route will be displayed by the **show router BGP routes evpn x detail** command with a 'tie-breaker' reason.

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 **etree** or **m-vpls**, are not supported with either EVPN-VXLAN or EVPN-MPLS.
- In general, no 7x50-generated control packets will be sent to the EVPN destination bindings, except for ARP, VRRP, ping, BFD and Eth-CFM for EVPN-VXLAN, and proxy-ARP/ND confirm messages and Eth-CFM for EVPN-MPLS.
- **eth-cfm** (meps, vmeps, mips): This command can be configured and used in EVPN VPLS service objects (service, saps and sdp-bindings). Although **vmeps** can be configured and used to local saps and sdp-bindings, **eth-cfm** tests will not work through EVPN destination bindings.
- xSTP and M-VPLS services:
  - xSTP can be configured in bgp-evpn services. BPDUs will not be 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.
- **mac-move**—in **bgp-evpn** enabled VPLS services, **mac-move** can be used in **saps/sdp-bindings**; however, the MACs being learned through BGP-EVPN will not be 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—these will only work for data plane learned mac addresses.
- **mac-protect**—**mac-protect** cannot be used in conjunction with EVPN.

  Note—EVPN provides its own protection mechanism for static mac addresses.
- **provider-tunnel**—p2mp RSVP/mLDP LSPs are not supported in the bgp-evpn service. The configuration of the provider-tunnel is blocked.
- MAC OAM—any MAC OAM tool is not supported for **bgp-evpn** services, that is: **mac-ping**, **mac-trace**, **mac-populate**, **mac-purge,** and **cpe-ping**.
- EVPN multi-homing and BGP-MH cannot be enabled in the same VPLS service. BGP-MH can still be used for multi-homing as long as no ethernet-segments are configured in the service SAPs/SDP-bindings. There is no limitation in terms of 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.

- **Note**—SAPs/SDP-bindings that belong to a specified ES but are configured on non-bgp-evpn-mpls-enabled VPLS or Epipe services will be kept down with the **StandByForMHProtocol** flag.
- IGMP-snooping is not supported in VPLS (or I-VPLS) services when **bgp-evpn mpls** is enabled (in the service or the associated B-VPLS).
- CPE-ping is not supported on EVPN services but it is in PBB-EVPN services (including I-VPLS and PBB-Epipe). CPE-ping packets will not be sent over EVPN destinations. CPE-ping will only work on local active SAP/SDP-bindings in I-VPLS and PBB-Epipe services.
- Other commands not supported in conjunction with **bgp-evpn**:
  - bgp-vpls
  - Endpoints and attributes
  - Subscriber management commands under service, SAP, and sdp-binding interfaces
  - MLD/PIM-snooping and attributes
  - BPDU translation
  - L2PT termination
  - MAC-pinning
- Other commands not supported in conjunction with **bgp-evpn mpls** are:
  - VSD-domains
  - VXLAN cannot be not shutdown under bgp-evpn. Both bgp-evpn vxlan and bgp-evpn mpls are mutually exclusive
  - SPB configuration and attributes
  - allow-ip-int-bind (R-VPLS) and bgp-evpn ip-route-advertisement

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 on page 1177, 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).
- PBB-EVPN multi-homing and BGP-MH cannot be enabled in the same **i-vpls** service.
- **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/SDP-bindings at the same time.
- When PBB-epipes are used with PBB-EVPN multi-homing, spoke-SDPs are not supported on **ethernet-segments**.

Interaction of EVPN-VXLAN with Existing VPRN Features

When trying to enable existing VPRN features on interfaces linked to EVPN-VXLAN IRB backhaul or EVPN tunnel R-VPLS interfaces, consider the following:

- The following commands are not supported:
  - arp-populate
  - authentication-policy
- Dynamic routing protocols such as IS-IS, RIP, and OSPF are not supported.
- BFD is not supported on EVPN tunnel interfaces.

Routing Policies for BGP EVPN IP Prefixes

BGP routing policies are supported for IP prefixes imported or exported through BGP-EVPN.

When applying routing policies to control the distribution of prefixes between EVPN and IP-VPN, the user must consider that both families are completely separate as far as BGP is concerned and that when prefixes are imported in the VPRN routing table, the BGP attributes are lost to the other family. The use of route tags allows the controlled distribution of prefixes across the two families.

Figure 143 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.

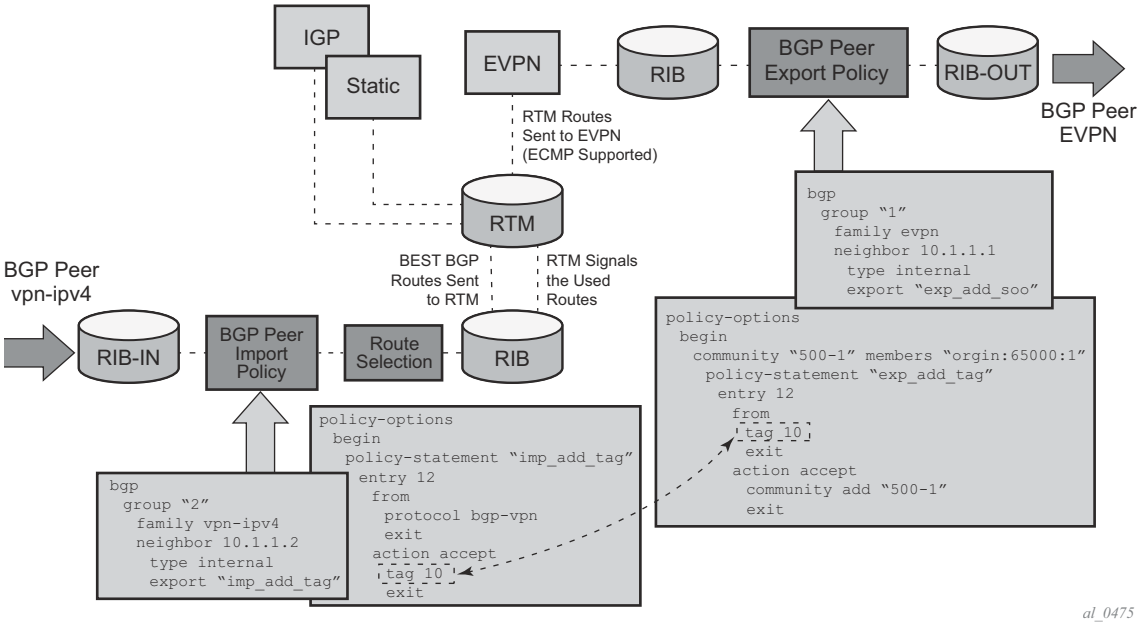

Figure 143: IP-VPN Import and EVPN Export BGP Workflow

Note — 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:

```
<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 144 shows an example of the reverse workflow: routes imported from EVPN and exported from RTM to BGP VPN-IPv4.

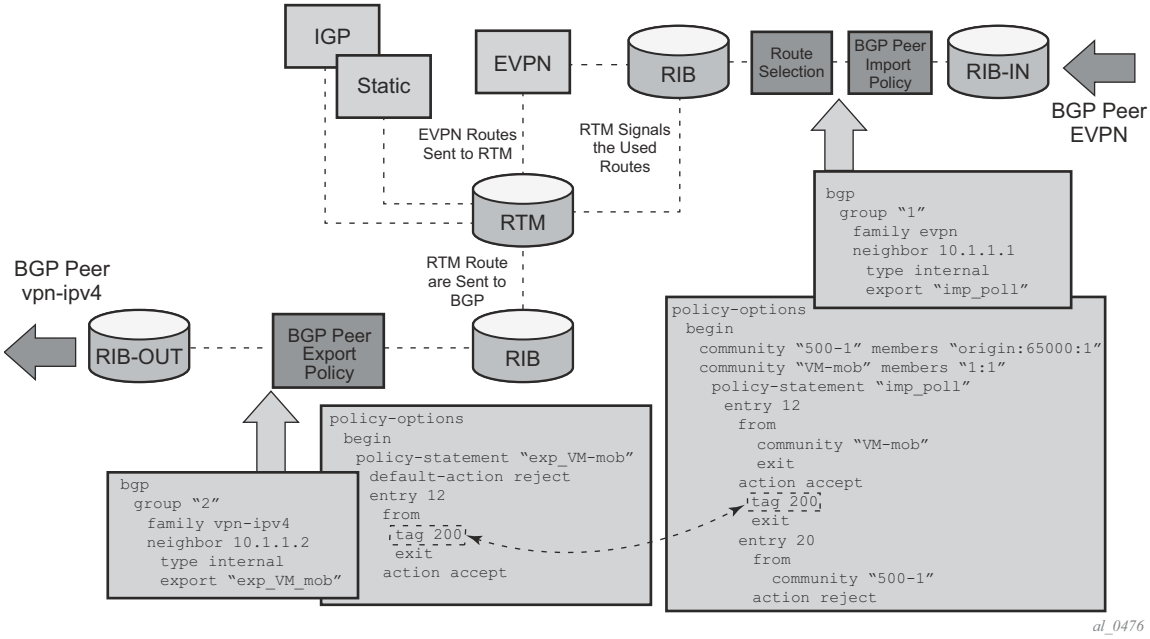

Figure 144: IEVPN Import and I-VPN Export BGP Workflow

Note — The preceding described behavior and the use of tags is also valid for vsi-import and vsi-export policies in the R-VPLS.

The policy behavior for EVPN ip-prefixes can then be summarized in the following statements.

- For EVPN prefix routes received and imported in RTM:
  - Policy entries can match on communities and add tags. This works at the peer level or at the vsi-import level.
  - Policy entries can match on *family evpn.*
- For exporting RTM to EVPN prefix routes:
  - Policy entries can match on tags and based on that, add communities, accept, or reject. This works at the peer level or the vsi-export level.

Policy entries can add tags for static-routes, RIP, OSPF, IS-IS, and BGP that can then be matched on the BGP peer export policy or vsi-export policy for EVPN prefix routes.