



# 7705 Service Aggregation Router Gen 2

Release 25.10.R1

## Layer 2 Services and EVPN Guide

---

3HE 21569 AAAC TQZZA 01

Edition: 01

October 2025

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

---

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. Use this document as agreed. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice. At all times, the controlling version is the one available on Nokia's site.

No part of this document may be modified.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

© 2025 Nokia.

# Table of contents

|   |           |
|---|-----------|
| <b>List of tables.....</b>  | <b>14</b> |
| <b>List of figures.....</b>   | <b>15</b> |
| <b>1 Getting started.....</b>   | <b>21</b> |
| 1.1 About this guide.....   | 21        |
| 1.2 Platforms and terminology.....                                    | 21        |
| 1.3 Conventions.....  | 22        |
| 1.3.1 Precautionary and information messages.....                     | 22        |
| 1.3.2 Options or substeps in procedures and sequential workflows..... | 22        |
| <b>2 VLL services.....</b>  | <b>24</b> |
| 2.1 Ethernet pipe service.....  | 24        |
| 2.1.1 Epipe service overview.....                                     | 24        |
| 2.1.2 Epipe service pseudowire VLAN tag processing.....               | 24        |
| 2.1.3 Epipe up operational state configuration option.....            | 28        |
| 2.1.4 VLL CAC.....  | 29        |
| 2.1.5 MC-Ring and VLL.....  | 29        |
| 2.2 Pseudowire redundancy service models.....                         | 30        |
| 2.2.1 Redundant VLL service model.....                                | 30        |
| 2.2.2 T-LDP status notification handling rules.....                   | 32        |
| 2.2.2.1 Processing endpoint SAP active/standby status bits.....       | 32        |
| 2.2.2.2 Processing and merging.....                                   | 32        |
| 2.3 VLL using G.8031 protected Ethernet tunnels.....                  | 33        |
| 2.4 MPLS EL and hash label.....                                       | 34        |
| 2.5 BGP VPWS.....   | 34        |
| 2.5.1 Single-homed BGP VPWS.....                                      | 35        |
| 2.5.2 Dual-homed BGP VPWS.....  | 35        |
| 2.5.2.1 Single pseudowire example.....                                | 35        |
| 2.5.2.2 Active/standby pseudowire example.....                        | 36        |
| 2.5.3 BGP VPWS pseudowire switching.....                              | 37        |
| 2.5.4 Pseudowire signaling.....                                       | 38        |
| 2.5.5 BGP-VPWS with inter-AS model C.....                             | 41        |
| 2.5.6 BGP VPWS configuration procedure.....                           | 42        |

|          |  |           |
|----------|--|-----------|
| 2.5.7    | Use of pseudowire template for BGP VPWS.....                   | 42        |
| 2.5.8    | Use of endpoint for BGP VPWS.....                              | 44        |
| 2.6      | VLL service considerations.....                                | 44        |
| 2.6.1    | SDPs.....  | 44        |
| 2.6.1.1  | SDP statistics for VPLS and VLL services.....                  | 44        |
| 2.6.2    | SAP encapsulations and pseudowire types.....                   | 45        |
| 2.6.2.1  | QoS policies.....  | 45        |
| 2.6.2.2  | Filter policies.....   | 46        |
| 2.6.2.3  | MAC resources.....   | 46        |
| 2.7      | Configuring a VLL service using CLI.....                       | 46        |
| 2.7.1    | Common configuration tasks.....                                | 46        |
| 2.7.2    | Configuring VLL components.....                                | 46        |
| 2.7.2.1  | Creating an Epipe service.....                                 | 46        |
| 2.7.3    | Using spoke SDP control words.....                             | 52        |
| 2.7.4    | Same-fate Epipe VLANs access protection.....                   | 53        |
| 2.7.5    | Pseudowire configuration notes.....                            | 54        |
| 2.7.6    | Configuring two VLL paths terminating on T-PE2.....            | 56        |
| 2.7.7    | Configuring VLL resilience.....                                | 57        |
| 2.7.8    | Configuring VLL resilience for a switched pseudowire path..... | 58        |
| 2.7.9    | Configuring BGP VPWS.....                                      | 60        |
| 2.7.9.1  | Single-homed BGP VPWS.....                                     | 60        |
| 2.7.9.2  | Dual-homed BGP VPWS.....                                       | 61        |
| 2.8      | Service management tasks.....                                  | 66        |
| 2.8.1    | Modifying Epipe service parameters.....                        | 66        |
| 2.8.2    | Disabling an Epipe service.....                                | 66        |
| 2.8.3    | Re-enabling an Epipe service.....                              | 67        |
| 2.8.4    | Deleting an Epipe service.....                                 | 67        |
| <b>3</b> | <b>Virtual Private LAN Service.....</b>                        | <b>68</b> |
| 3.1      | VPLS service overview.....                                     | 68        |
| 3.1.1    | VPLS packet walkthrough.....                                   | 68        |
| 3.2      | VPLS features.....   | 71        |
| 3.2.1    | VPLS enhancements.....   | 71        |
| 3.2.2    | VPLS over MPLS.....  | 72        |
| 3.2.3    | VPLS service pseudowire VLAN tag processing.....               | 72        |
| 3.2.4    | VPLS MAC learning and packet forwarding.....                   | 76        |



|          |  |     |
|----------|--|-----|
| 3.2.4.1  | MAC learning protection.....                                     | 77  |
| 3.2.4.2  | DEI in IEEE 802.1ad.....   | 78  |
| 3.2.5    | VPLS using G.8031 protected Ethernet tunnels.....                | 78  |
| 3.2.6    | Pseudowire control word.....                                     | 79  |
| 3.2.7    | Layer 2 forwarding table management.....                         | 79  |
| 3.2.7.1  | Selective MAC address learning.....                              | 79  |
| 3.2.7.2  | System FDB size.....   | 85  |
| 3.2.7.3  | Per-VPLS service FDB size.....                                   | 86  |
| 3.2.7.4  | System FDB size alarms.....                                      | 86  |
| 3.2.7.5  | Line card FDB size alarms.....                                   | 86  |
| 3.2.7.6  | Per VPLS FDB size alarms.....                                    | 86  |
| 3.2.7.7  | Local and remote aging timers.....                               | 87  |
| 3.2.7.8  | Disable MAC aging.....   | 87  |
| 3.2.7.9  | Disable MAC learning.....  | 87  |
| 3.2.7.10 | Unknown MAC discard.....   | 87  |
| 3.2.7.11 | VPLS and rate limiting.....                                      | 87  |
| 3.2.7.12 | MAC move.....  | 88  |
| 3.2.7.13 | Auto-learn MAC protect.....                                      | 88  |
| 3.2.8    | Split horizon SAP groups and split horizon spoke SDP groups..... | 92  |
| 3.2.9    | VPLS and STP.....  | 92  |
| 3.2.9.1  | Spanning Tree operating modes.....                               | 92  |
| 3.2.9.2  | Multiple Spanning Tree Protocol.....                             | 93  |
| 3.2.9.3  | MSTP for QinQ SAPs.....  | 94  |
| 3.2.9.4  | Provider MSTP.....   | 94  |
| 3.2.9.5  | Enhancements to the STP.....                                     | 95  |
| 3.2.10   | VPLS redundancy.....   | 96  |
| 3.2.10.1 | Spoke SDP redundancy for metro interconnection.....              | 96  |
| 3.2.10.2 | Spoke SDP-based redundant access.....                            | 97  |
| 3.2.10.3 | Inter-domain VPLS resiliency using multichassis endpoints.....   | 98  |
| 3.2.10.4 | Support for single chassis endpoint mechanisms.....              | 102 |
| 3.2.11   | VPLS access redundancy.....                                      | 104 |
| 3.2.11.1 | STP-based redundant access to VPLS.....                          | 105 |
| 3.2.11.2 | Redundant access to VPLS without STP.....                        | 105 |
| 3.2.12   | Object grouping and state monitoring.....                        | 105 |
| 3.2.12.1 | VPLS applicability — block on VPLS a failure.....                | 106 |
| 3.2.13   | MAC flush message processing.....                                | 107 |

|           |  |     |
|-----------|--|-----|
| 3.2.13.1  | Dual-homing to a VPLS service.....   | 108 |
| 3.2.13.2  | MC-Ring and VPLS.....  | 110 |
| 3.2.14    | ACL next-hop for VPLS.....   | 110 |
| 3.2.15    | SDP statistics for VPLS and VLL services.....                                      | 111 |
| 3.2.16    | BGP Auto-Discovery for LDP VPLS.....   | 111 |
| 3.2.16.1  | BGP AD overview.....   | 112 |
| 3.2.16.2  | Information model.....   | 112 |
| 3.2.16.3  | FEC element for T-LDP signaling.....   | 113 |
| 3.2.16.4  | BGP AD and T-LDP interaction.....  | 114 |
| 3.2.16.5  | SDP usage.....   | 115 |
| 3.2.16.6  | Automatic creation of SDPs.....  | 115 |
| 3.2.16.7  | Manually provisioned SDP.....  | 116 |
| 3.2.16.8  | Automatic instantiation of pseudowires (SDP bindings).....                         | 116 |
| 3.2.16.9  | Mixing statically configured and auto-discovered pseudowires in a VPLS.....        | 117 |
| 3.2.16.10 | Resiliency schemes.....  | 117 |
| 3.2.17    | BGP VPLS.....  | 117 |
| 3.2.17.1  | Pseudowire signaling details.....  | 118 |
| 3.2.17.2  | Supported VPLS features.....   | 121 |
| 3.2.18    | VCCV BFD support for VPLS services.....  | 121 |
| 3.2.19    | BGP multihoming for VPLS.....  | 122 |
| 3.2.19.1  | Information model and required extensions to L2VPN NLRI.....                       | 123 |
| 3.2.19.2  | Supported services and multihoming objects.....                                    | 124 |
| 3.2.19.3  | Blackhole avoidance.....   | 124 |
| 3.2.19.4  | BGP multihoming for VPLS inter-domain resiliency.....                              | 125 |
| 3.2.20    | Multicast-aware VPLS.....  | 126 |
| 3.2.20.1  | IGMP snooping for VPLS.....  | 126 |
| 3.2.20.2  | MLD snooping for VPLS.....   | 127 |
| 3.2.20.3  | PIM snooping for VPLS.....   | 127 |
| 3.2.20.4  | IPv6 multicast forwarding.....   | 129 |
| 3.2.20.5  | PIM and IGMP/MLD snooping interaction.....   | 131 |
| 3.2.20.6  | Multichassis synchronization for Layer 2 snooping states.....                      | 131 |
| 3.2.20.7  | VPLS multicast-aware high availability features.....                               | 134 |
| 3.2.21    | RSVP and LDP P2MP LSP for forwarding VPLS/B-VPLS BUM and IP multicast packets..... | 134 |
| 3.2.22    | MPLS EL and hash label.....  | 135 |
| 3.3       | Routed VPLS and I-VPLS.....  | 136 |

|         |  |     |
|---------|--|-----|
| 3.3.1   | IES or VPRN IP interface binding.....                          | 136 |
| 3.3.1.1 | Assigning a service name to a VPLS service.....                | 136 |
| 3.3.1.2 | Service binding requirements.....                              | 136 |
| 3.3.1.3 | Bound service name assignment.....                             | 137 |
| 3.3.1.4 | Binding a service name to an IP interface.....                 | 137 |
| 3.3.1.5 | Bound service deletion or service name removal.....            | 137 |
| 3.3.1.6 | IP interface attached VPLS service constraints.....            | 137 |
| 3.3.1.7 | IP interface and VPLS operational state coordination.....      | 138 |
| 3.3.2   | IP interface MTU and fragmentation.....                        | 138 |
| 3.3.2.1 | Unicast IP routing into a VPLS service.....                    | 138 |
| 3.3.3   | ARP and VPLS FDB interactions.....                             | 138 |
| 3.3.3.1 | R-VPLS specific ARP cache behavior.....                        | 139 |
| 3.3.4   | The allow-ip-int-bind VPLS flag.....                           | 140 |
| 3.3.4.1 | R-VPLS SAPs only supported on standard Ethernet ports.....     | 140 |
| 3.3.4.2 | LAG port membership constraints.....                           | 140 |
| 3.3.4.3 | R-VPLS feature restrictions.....                               | 140 |
| 3.3.5   | IPv4 and IPv6 multicast routing support.....                   | 141 |
| 3.3.6   | BGP-AD for R-VPLS support.....                                 | 143 |
| 3.3.7   | R-VPLS restrictions.....                                       | 143 |
| 3.3.7.1 | VPLS SAP ingress IP filter override.....                       | 144 |
| 3.3.7.2 | IP interface defined egress QoS reclassification.....          | 144 |
| 3.3.7.3 | Interface statistics collection.....                           | 144 |
| 3.3.7.4 | Remarking for VPLS and routed packets.....                     | 144 |
| 3.3.7.5 | IPv4 multicast routing.....                                    | 144 |
| 3.3.7.6 | R-VPLS supported routing-related protocols.....                | 145 |
| 3.3.7.7 | Spanning Tree and split horizon.....                           | 145 |
| 3.4     | VPLS service considerations.....                               | 145 |
| 3.4.1   | SAP encapsulations.....  | 145 |
| 3.4.2   | VLAN processing.....   | 146 |
| 3.4.3   | Ingress VLAN swapping.....                                     | 146 |
| 3.4.4   | Service auto-discovery using MVRP.....                         | 147 |
| 3.4.4.1 | Configure the MVRP infrastructure using an M-VPLS context..... | 148 |
| 3.4.4.2 | Instantiate related VLAN FDBs and trunks in MVRP scope.....    | 148 |
| 3.4.4.3 | MVRP activation of service connectivity.....                   | 149 |
| 3.4.4.4 | MVRP control plane.....  | 151 |
| 3.4.4.5 | STP-MVRP interaction.....                                      | 151 |

|          |  |            |
|----------|--|------------|
| 3.5      | Configuring a VPLS service using CLI.....                | 153        |
| 3.5.1    | Basic configuration.....                                 | 153        |
| 3.5.2    | Common configuration tasks.....                          | 155        |
| 3.5.3    | Configuring VPLS components.....                         | 155        |
| 3.5.3.1  | Creating a VPLS service.....                             | 155        |
| 3.5.3.2  | Enabling MAC move.....                                   | 156        |
| 3.5.3.3  | Configuring STP bridge parameters in a VPLS.....         | 157        |
| 3.5.3.4  | Configuring GSMP parameters.....                         | 161        |
| 3.5.3.5  | Configuring a VPLS SAP.....                              | 162        |
| 3.5.3.6  | Configuring SAP subscriber management parameters.....    | 170        |
| 3.5.3.7  | MSTP control over Ethernet tunnels.....                  | 171        |
| 3.5.3.8  | Configuring SDP bindings.....                            | 171        |
| 3.5.3.9  | Configuring overrides on service SAPs.....               | 172        |
| 3.5.4    | Configuring VPLS redundancy.....                         | 181        |
| 3.5.4.1  | Creating a management VPLS for SAP protection.....       | 181        |
| 3.5.4.2  | Creating a management VPLS for spoke SDP protection..... | 183        |
| 3.5.4.3  | Configuring load balancing with management VPLS.....     | 185        |
| 3.5.4.4  | Configuring selective MAC flush.....                     | 189        |
| 3.5.4.5  | Configuring multichassis endpoints.....                  | 189        |
| 3.5.5    | Configuring BGP AD.....                                  | 193        |
| 3.5.5.1  | Configuration steps.....                                 | 193        |
| 3.5.5.2  | LDP signaling.....                                       | 195        |
| 3.5.5.3  | Pseudowire template.....                                 | 196        |
| 3.5.6    | Configuring BGP VPLS.....                                | 197        |
| 3.5.6.1  | Configuring a VPLS management interface.....             | 199        |
| 3.5.7    | Configuring policy-based forwarding for DPI in VPLS..... | 199        |
| 3.6      | Service management tasks.....                            | 202        |
| 3.6.1    | Modifying VPLS service parameters.....                   | 202        |
| 3.6.2    | Modifying management VPLS parameters.....                | 202        |
| 3.6.3    | Deleting a management VPLS.....                          | 203        |
| 3.6.4    | Disabling a management VPLS.....                         | 203        |
| 3.6.5    | Deleting a VPLS service.....                             | 203        |
| 3.6.6    | Disabling a VPLS service.....                            | 204        |
| 3.6.7    | Re-enabling a VPLS service.....                          | 204        |
| <b>4</b> | <b>Layer 2 control protocols.....</b>                    | <b>205</b> |

|          |   |            |
|----------|---|------------|
| <b>5</b> | <b>Ethernet Virtual Private Networks.....</b>   | <b>209</b> |
| 5.1      | Overview of EVPN applications.....  | 209        |
| 5.1.1    | EVPN for VXLAN tunnels in a Layer 2 DGW (EVPN-VXLAN).....   | 209        |
| 5.1.2    | EVPN for VXLAN tunnels in a Layer 2 DC with integrated routing bridging connectivity on the DGW.....  | 211        |
| 5.1.3    | EVPN for VXLAN tunnels in a Layer 3 DC with integrated routing bridging connectivity among VPRNs..... | 211        |
| 5.1.4    | EVPN for VXLAN tunnels in a Layer 3 DC with EVPN-tunnel connectivity among VPRNs.....                 | 213        |
| 5.1.5    | EVPN for MPLS tunnels in E-LAN services.....  | 214        |
| 5.1.6    | EVPN for MPLS tunnels in E-Line services.....   | 215        |
| 5.1.7    | EVPN for MPLS tunnels in E-Tree services.....   | 215        |
| 5.2      | EVPN for VXLAN tunnels and cloud technologies.....  | 215        |
| 5.2.1    | VXLAN.....  | 215        |
| 5.2.1.1  | VXLAN ECMP and LAG.....   | 217        |
| 5.2.1.2  | VXLAN VPLS tag handling.....  | 218        |
| 5.2.1.3  | VXLAN MTU considerations.....   | 218        |
| 5.2.1.4  | VXLAN QoS.....  | 218        |
| 5.2.1.5  | VXLAN ping.....   | 219        |
| 5.2.1.6  | EVPN-VXLAN routed VPLS multicast routing support.....   | 223        |
| 5.2.1.7  | IGMP and MLD snooping on VXLAN.....   | 223        |
| 5.2.1.8  | PIM snooping on VXLAN.....  | 225        |
| 5.2.1.9  | Static VXLAN termination in Epipe services.....   | 225        |
| 5.2.1.10 | Static VXLAN termination in VPLS/R-VPLS services.....   | 227        |
| 5.2.1.11 | Non-system IPv4 and IPv6 VXLAN termination in VPLS, R-VPLS, and Epipe services.....                   | 228        |
| 5.2.2    | EVPN for overlay tunnels.....   | 232        |
| 5.2.2.1  | BGP-EVPN control plane for VXLAN overlay tunnels.....   | 233        |
| 5.2.2.2  | EVPN for VXLAN in VPLS services.....  | 237        |
| 5.2.2.3  | EVPN for VXLAN in R-VPLS services.....  | 241        |
| 5.2.2.4  | EVPN-VPWS for VXLAN tunnels.....  | 249        |
| 5.2.3    | Layer 2 multicast optimization for VXLAN (Assisted-Replication).....                                  | 263        |
| 5.2.3.1  | Replicator (AR-R) procedures.....   | 264        |
| 5.2.3.2  | Leaf (AR-L) procedures.....   | 265        |
| 5.2.3.3  | Assisted-Replication interaction with other VPLS features.....  | 268        |
| 5.2.4    | EVPN VXLAN multihoming.....   | 268        |

|          |  |     |
|----------|--|-----|
| 5.2.4.1  | Local bias for EVPN VXLAN multihoming.....                                   | 271 |
| 5.2.4.2  | Known limitations for local bias.....  | 273 |
| 5.2.4.3  | Non-system IPv4 and IPv6 VXLAN termination for EVPN VXLAN multihoming..      | 275 |
| 5.3      | EVPN for MPLS tunnels.....   | 275 |
| 5.3.1    | BGP-EVPN control plane for MPLS tunnels.....                                 | 276 |
| 5.3.2    | EVPN for MPLS tunnels in VPLS services.....                                  | 281 |
| 5.3.2.1  | EVPN and VPLS integration.....   | 286 |
| 5.3.2.2  | EVPN single-active multihoming and BGP-VPLS integration.....                 | 289 |
| 5.3.2.3  | Auto-derived route-distinguisher in services with multiple BGP families..... | 291 |
| 5.3.3    | P2MP mLDP tunnels for BUM traffic in EVPN-MPLS services.....                 | 291 |
| 5.3.4    | EVPN-VPWS for MPLS tunnels.....  | 294 |
| 5.3.4.1  | BGP-EVPN control plane for EVPN-VPWS.....                                    | 294 |
| 5.3.4.2  | EVPN for MPLS tunnels in Epipe services (EVPN-VPWS).....                     | 294 |
| 5.3.4.3  | EVPN-VPWS services with local-switching support.....                         | 296 |
| 5.3.4.4  | EVPN-VPWS FXC.....   | 302 |
| 5.3.5    | EVPN for MPLS tunnels in routed VPLS services.....                           | 320 |
| 5.3.5.1  | EVPN-MPLS multihoming and passive VRRP.....                                  | 320 |
| 5.3.6    | EVPN-MPLS routed VPLS multicast routing support.....                         | 323 |
| 5.3.7    | IGMP snooping in EVPN-MPLS.....  | 323 |
| 5.3.7.1  | Data-driven IGMP snooping synchronization with EVPN multihoming.....         | 324 |
| 5.3.8    | PIM snooping for IPv4 in EVPN-MPLS and PBB-EVPN services.....                | 327 |
| 5.3.8.1  | Data-driven PIM snooping for IPv4 synchronization with EVPN multihoming....  | 329 |
| 5.3.9    | MPLS EL and hash label.....  | 332 |
| 5.3.10   | Inter-AS Option B and Next-Hop-Self Route-Reflector for EVPN-MPLS.....       | 332 |
| 5.3.10.1 | Inter-AS Option B and VPN-NH-RR procedures on EVPN routes.....               | 334 |
| 5.3.10.2 | BUM traffic in inter-AS Option B and VPN-NH-RR networks.....                 | 335 |
| 5.3.11   | ECMP for EVPN-MPLS destinations.....   | 336 |
| 5.3.12   | IPv6 tunnel resolution for EVPN MPLS services.....                           | 336 |
| 5.4      | General EVPN topics.....   | 337 |
| 5.4.1    | ARP/ND snooping and proxy support.....                                       | 337 |
| 5.4.1.1  | Proxy-ARP/ND periodic refresh, unsolicited refresh and confirm-messages....  | 341 |
| 5.4.1.2  | Advertisement of Proxy-ARP/ND flags in EVPN.....                             | 342 |
| 5.4.1.3  | Proxy-ARP/ND and flag processing.....  | 343 |
| 5.4.1.4  | Proxy-ARP/ND mac-List for dynamic entries.....                               | 345 |
| 5.4.2    | BGP-EVPN MAC mobility.....   | 349 |
| 5.4.3    | BGP-EVPN MAC duplication.....  | 349 |

|          |  |     |
|----------|--|-----|
| 5.4.4    | Conditional static MAC and protection.....   | 351 |
| 5.4.5    | Auto-learn MAC protect and restricting protected source MACs.....  | 352 |
| 5.4.6    | Blackhole MAC and its application to proxy-ARP/proxy-ND duplicate detection.....                         | 354 |
| 5.4.7    | Blackhole MAC for EVPN loop detection.....   | 356 |
| 5.4.7.1  | Deterministic EVPN loop detection with trusted MACs.....   | 360 |
| 5.4.8    | CFM interaction with EVPN services.....  | 362 |
| 5.4.9    | Multi-instance EVPN: Two instances of different encapsulation in the same VPLS/R-VPLS/Epipe service..... | 363 |
| 5.4.9.1  | EVPN-VXLAN to EVPN-MPLS interworking.....  | 363 |
| 5.4.9.2  | EVPN-SRv6 to EVPN-MPLS or EVPN-VXLAN interworking.....   | 365 |
| 5.4.9.3  | BGP-EVPN routes in services configured with two BGP instances.....                                       | 372 |
| 5.4.9.4  | Anycast redundant solution for dual BGP-instance services.....   | 377 |
| 5.4.9.5  | Using P2MP mLDP in redundant anycast DCGWs.....  | 385 |
| 5.4.9.6  | I-ES solution for dual BGP instance services.....  | 386 |
| 5.4.10   | Multi-instance EVPN: Two instances of the same encapsulation in the same VPLS/R-VPLS service.....        | 394 |
| 5.4.10.1 | BGP-EVPN routes in multi-instance EVPN services with the same encapsulation.....                         | 396 |
| 5.4.10.2 | Anycast redundant solution for multi-instance EVPN services with the same encapsulation.....             | 397 |
| 5.4.10.3 | I-ES solution for dual BGP EVPN instance services with the same encapsulation.....                       | 398 |
| 5.4.11   | EVPN IP-VRF-to-IP-VRF models.....  | 401 |
| 5.4.11.1 | Interface-ful IP-VRF-to-IP-VRF with SBD IRB model.....   | 401 |
| 5.4.11.2 | Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model.....  | 403 |
| 5.4.11.3 | Interoperable interface-less IP-VRF-to-IP-VRF model (Ethernet encapsulation).....                        | 404 |
| 5.4.11.4 | Interface-less IP-VRF-to-IP-VRF model (IP encapsulation) for MPLS tunnels..                              | 406 |
| 5.4.11.5 | EVPN MAC/IP advertisement route install into the route table.....  | 408 |
| 5.4.12   | ARP-ND host routes for extended Layer 2 data centers.....  | 411 |
| 5.4.13   | EVPN host mobility procedures within the same R-VPLS service.....  | 413 |
| 5.4.13.1 | EVPN host mobility configuration.....  | 413 |
| 5.4.14   | BGP and EVPN route selection for EVPN routes.....  | 418 |
| 5.4.15   | LSP tagging for BGP next-hops or prefixes and BGP-LU.....  | 419 |
| 5.4.16   | Oper-groups interaction with EVPN services.....  | 420 |
| 5.4.16.1 | LAG-based LLF for EVPN-VPWS services.....  | 420 |
| 5.4.16.2 | Core isolation blackhole avoidance.....  | 422 |
| 5.4.16.3 | LAG or port standby signaling to the CE on non-DF EVPN PEs (single-active).                              | 423 |

|          |   |            |
|----------|---|------------|
| 5.4.16.4 | AC-influenced DF election capability on an ES with oper-group.....    | 424        |
| 5.4.17   | EVPN Layer-2 multicast (IGMP/MLD proxy).....                          | 425        |
| 5.4.18   | EVPN-VPWS PW headend functionality.....                               | 427        |
| 5.4.19   | Interaction of EVPN and other features.....                           | 433        |
| 5.4.19.1 | Interaction of EVPN-MPLS with existing VPLS features.....             | 433        |
| 5.4.19.2 | Interaction of EVPN-MPLS with existing VPRN or IES features.....      | 434        |
| 5.4.20   | Interaction of EVPN with BGP owners in the same VPRN service.....     | 434        |
| 5.4.20.1 | BGP path attribute propagation.....                                   | 436        |
| 5.4.20.2 | BGP D-PATH attribute for Layer 3 loop protection.....                 | 438        |
| 5.4.20.3 | Configuration examples.....   | 443        |
| 5.4.21   | Routing policies for BGP EVPN routes.....                             | 450        |
| 5.4.21.1 | Routing policies for BGP EVPN IP prefixes.....                        | 451        |
| 5.4.22   | EVPN Weighted ECMP for IP prefix routes.....                          | 453        |
| 5.4.23   | EVPN Sticky ECMP for IP prefix routes.....                            | 460        |
| 5.4.24   | EVPN VLAN-aware bundle mode for BGP-EVPN VPLS or R-VPLS services..... | 460        |
| 5.5      | Configuring an EVPN service with CLI.....                             | 464        |
| 5.5.1    | EVPN-MPLS configuration examples.....                                 | 464        |
| 5.5.1.1  | EVPN all-active multihoming example.....                              | 464        |
| 5.5.1.2  | EVPN single-active multihoming example.....                           | 466        |
| <b>6</b> | <b>7705 SAR Gen 2 pseudowire ports.....</b>                           | <b>468</b> |
| 6.1      | PW port list.....   | 468        |
| 6.2      | Failover times.....   | 468        |
| 6.3      | QoS.....  | 469        |
| 6.4      | PW port termination for various tunnel types.....                     | 470        |
| 6.4.1    | MPLS-based spoke SDP.....   | 471        |
| 6.4.1.1  | Provisioning.....   | 471        |
| 6.4.1.2  | Flex PW-port operational state for MPLS based spoke SDP.....          | 472        |
| 6.4.1.3  | Statistics.....   | 473        |
| 6.4.2    | L2oGRE-based spoke SDP.....   | 474        |
| 6.4.2.1  | Provisioning.....   | 474        |
| 6.4.2.2  | Flex PW-port operational state for L2oGRE-based spoke SDP.....        | 475        |
| 6.4.2.3  | Reassembly.....   | 475        |
| <b>7</b> | <b>Standards and protocol support.....</b>                            | <b>476</b> |
| 7.1      | Bidirectional Forwarding Detection (BFD).....                         | 476        |



---

|      |  |     |
|------|--|-----|
| 7.2  | Border Gateway Protocol (BGP).....                                 | 476 |
| 7.3  | Bridging and management.....                                       | 477 |
| 7.4  | Certificate management.....  | 478 |
| 7.5  | Ethernet VPN (EVPN).....   | 478 |
| 7.6  | gRPC Remote Procedure Calls (gRPC).....                            | 478 |
| 7.7  | Intermediate System to Intermediate System (IS-IS).....            | 479 |
| 7.8  | Internet Protocol (IP) general.....                                | 480 |
| 7.9  | Internet Protocol (IP) multicast.....                              | 481 |
| 7.10 | Internet Protocol (IP) version 4.....                              | 481 |
| 7.11 | Internet Protocol (IP) version 6.....                              | 482 |
| 7.12 | Internet Protocol Security (IPsec).....                            | 483 |
| 7.13 | Label Distribution Protocol (LDP).....                             | 484 |
| 7.14 | Multiprotocol Label Switching (MPLS).....                          | 484 |
| 7.15 | Network Address Translation (NAT).....                             | 485 |
| 7.16 | Network Configuration Protocol (NETCONF).....                      | 485 |
| 7.17 | Media Sanitization.....  | 485 |
| 7.18 | Open Shortest Path First (OSPF).....                               | 485 |
| 7.19 | Path Computation Element Protocol (PCEP).....                      | 486 |
| 7.20 | Pseudowire (PW).....   | 486 |
| 7.21 | Quality of Service (QoS).....                                      | 487 |
| 7.22 | Remote Authentication Dial In User Service (RADIUS).....           | 487 |
| 7.23 | Resource Reservation Protocol - Traffic Engineering (RSVP-TE)..... | 488 |
| 7.24 | Routing Information Protocol (RIP).....                            | 488 |
| 7.25 | Segment Routing (SR).....  | 488 |
| 7.26 | Simple Network Management Protocol (SNMP).....                     | 489 |
| 7.27 | Timing.....  | 491 |
| 7.28 | Two-Way Active Measurement Protocol (TWAMP).....                   | 491 |
| 7.29 | Virtual Private LAN Service (VPLS).....                            | 491 |
| 7.30 | Yet Another Next Generation (YANG).....                            | 491 |

# List of tables

Table 1: Platforms and terminology.....

21

Table 2: Epipe spoke-SDP VLAN tag processing: ingress.....

26

Table 3: Epipe-spoke SDP VLAN tag processing: egress.....

26

Table 4: Supported SAP types.....

48

Table 5: VPLS mesh and spoke-SDP VLAN tag processing: ingress.....

73

Table 6: VPLS mesh and spoke-SDP VLAN tag processing: egress.....

74

Table 7: MAC address learning logic example.....

80

Table 8: BGP AD and T-LDP interaction key.....

115

Table 9: Ingress routed to VPLS next-hop behavior.....

139

Table 10: Egress R-VPLS next-hop behavior.....

140

Table 11: MSTP and MVRP interaction table.....

152

Table 12: Spoke SDP BPDU encapsulation states.....

179

Table 13: AR-R and AR-L routes and usage.....

235

Table 14: EVPN routes and usage.....

276

Table 15: Proxy-arp entry combinations.....

341

Table 16: Configuration steps for L2oGRE reassembly.....

475

# List of figures

|  |    |
|--|----|
| Figure 1: Epipe/VLL service.....   | 24 |
| Figure 2: MC-Ring in a combination with VLL service.....                       | 30 |
| Figure 3: Redundant VLL endpoint objects.....                                  | 31 |
| Figure 4: Single-homed BGP-VPWS example.....                                   | 35 |
| Figure 5: Dual-homed BGP VPWS with single pseudowire.....                      | 36 |
| Figure 6: Dual-homed BGP VPWS with active/standby pseudowires.....             | 37 |
| Figure 7: BGP VPWS update extended community format.....                       | 39 |
| Figure 8: Control flags.....   | 40 |
| Figure 9: BGP VPWS NLRI.....   | 40 |
| Figure 10: BGP VPWS NLRI TLV extension format.....                             | 41 |
| Figure 11: Circuit status vector TLV type.....                                 | 41 |
| Figure 12: SDP statistics for VPLS and VLL services.....                       | 45 |
| Figure 13: VLL resilience with pseudowire redundancy and switching.....        | 56 |
| Figure 14: VLL resilience.....   | 58 |
| Figure 15: VLL resilience with pseudowire switching.....                       | 58 |
| Figure 16: Single-homed BGP VPWS configuration example.....                    | 60 |
| Figure 17: Example of dual-homed BGP VPWS with single pseudowire.....          | 62 |
| Figure 18: Example of dual-homed BGP VPWS with active/standby pseudowires..... | 64 |
| Figure 19: VPLS service architecture.....                                      | 69 |
| Figure 20: Access port ingress packet format and lookup.....                   | 69 |
| Figure 21: Network port egress packet format and flooding.....                 | 70 |

---

|   |     |
|---|-----|
| Figure 22: Access port egress packet format and lookup.....       | 71  |
| Figure 23: MAC learning protection.....                           | 77  |
| Figure 24: DE bit in the 802.1ad S-TAG.....                       | 78  |
| Figure 25: MAC FDB entry allocation: global versus selective..... | 81  |
| Figure 26: Auto-learn-mac-protect operation.....                  | 90  |
| Figure 27: Auto-learn-mac-protect example.....                    | 91  |
| Figure 28: Access resiliency.....                                 | 94  |
| Figure 29: H-VPLS with spoke redundancy.....                      | 97  |
| Figure 30: HVPLS resiliency based on AS pseudowires.....          | 98  |
| Figure 31: Multichassis pseudowire endpoint for VPLS.....         | 99  |
| Figure 32: MC-EP in passive mode.....                             | 101 |
| Figure 33: MAC flush in the MC-EP solution.....                   | 102 |
| Figure 34: Dual-homed MTUs in two-tier hierarchy H-VPLS.....      | 105 |
| Figure 35: Dual-homed CE connection to VPLS.....                  | 109 |
| Figure 36: Application 1 diagram.....                             | 110 |
| Figure 37: SDP statistics for VPLS and VLL services.....          | 111 |
| Figure 38: BGP AD NLRI versus IP VPN NLRI.....                    | 112 |
| Figure 39: Generalized pseudowire-ID FEC element.....             | 113 |
| Figure 40: BGP AD and T-LDP interaction.....                      | 114 |
| Figure 41: BGP VPLS solution.....                                 | 117 |
| Figure 42: Layer 2 Info Extended Community attribute.....         | 119 |
| Figure 43: Control flags bit vector.....                          | 120 |
| Figure 44: BGP multihoming for VPLS.....                          | 123 |

---

|   |     |
|---|-----|
| Figure 45: BGP MH-NLRI for VPLS multihoming.....  | 123 |
| Figure 46: BGP MH used in an HVPLS topology.....  | 125 |
| Figure 47: IPv4/IPv6 multicast with a router VPLS service.....                                | 143 |
| Figure 48: Ingress VLAN swapping.....   | 146 |
| Figure 49: Infrastructure for MVRP exchanges.....   | 147 |
| Figure 50: Service instantiation with MVRP - QinQ to PBB example.....                         | 150 |
| Figure 51: SDPs — unidirectional tunnels.....   | 173 |
| Figure 52: Example configuration for protected VPLS SAP.....                                  | 182 |
| Figure 53: Example configuration for protected VPLS spoke SDP.....                            | 184 |
| Figure 54: Example configuration for load balancing across two protected VPLS spoke-SDPs..... | 185 |
| Figure 55: BGP AD configuration example.....  | 193 |
| Figure 56: BGP AD triggering LDP functions.....   | 195 |
| Figure 57: Show router LDP session output.....  | 195 |
| Figure 58: Show router LDP bindings FEC-type services.....                                    | 196 |
| Figure 59: PW-template-binding CLI syntax.....  | 197 |
| Figure 60: BGP VPLS example.....  | 198 |
| Figure 61: Policy-based forwarding for deep packet inspection.....                            | 200 |
| Figure 62: Layer 2 DC PE with VPLS to the WAN.....  | 210 |
| Figure 63: Gateway IRB on the DC PE for an L2 EVPN/VXLAN DC.....                              | 211 |
| Figure 64: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC.....                              | 212 |
| Figure 65: EVPN-tunnel gateway IRB on the DC PE for a Layer 3 EVPN/VXLAN DC.....              | 213 |
| Figure 66: EVPN for MPLS in VPLS services.....  | 214 |
| Figure 67: VXLAN frame format.....  | 216 |

---

|   |     |
|---|-----|
| Figure 68: EVPN-VXLAN required routes and communities.....        | 233 |
| Figure 69: PMSI attribute flags field for AR.....                 | 234 |
| Figure 70: EVPN route-type 5.....                                 | 236 |
| Figure 71: EVPN-VPWS BGP extensions.....                          | 250 |
| Figure 72: EVPN-MPLS VPWS.....                                    | 251 |
| Figure 73: A/S PW and MC-LAG support on EVPN-VPWS.....            | 253 |
| Figure 74: EVPN-VPWS single-active multihoming.....               | 255 |
| Figure 75: AR BM replication behavior for a BM packet.....        | 268 |
| Figure 76: EVPN multihoming for EVPN-VXLAN.....                   | 269 |
| Figure 77: EVPN-VXLAN multihoming with local bias.....            | 272 |
| Figure 78: EVPN-VXLAN multihoming and unknown unicast issues..... | 274 |
| Figure 79: Blackhole created by a remote SAP shutdown.....        | 275 |
| Figure 80: Composite p2mp mLDP and IR tunnels—PTA.....            | 278 |
| Figure 81: EVPN routes type 1 and 4.....                          | 279 |
| Figure 82: EVPN-VPLS integration.....                             | 287 |
| Figure 83: BGP-VPLS to EVPN integration and single-active MH..... | 289 |
| Figure 84: EVPN services with p2mp mLDP—control plane.....        | 293 |
| Figure 85: EVPN-VPWS endpoints example 1.....                     | 297 |
| Figure 86: EVPN-VPWS endpoints example 2.....                     | 299 |
| Figure 87: EVPN-VPWS endpoints example 3.....                     | 301 |
| Figure 88: Default FXC mode.....                                  | 308 |
| Figure 89: VLAN-aware bundle or VLAN-signaled FXC mode.....       | 313 |
| Figure 90: EVPN-MPLS multihoming in R-VPLS services.....          | 321 |

---

|   |     |
|---|-----|
| Figure 91: Data-driven IGMP snooping synchronization with EVPN multihoming.....             | 325 |
| Figure 92: Data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.....     | 330 |
| Figure 93: EVPN inter-AS Option B or VPN-NH-RR model.....                                   | 333 |
| Figure 94: VPN-NH-RR and ingress replication for BUM traffic.....                           | 335 |
| Figure 95: Proxy-ARP example usage in an EVPN network.....                                  | 338 |
| Figure 96: Format of EVPN ARP/ND extended community.....                                    | 342 |
| Figure 97: EVPN non-intrusive loop detection mechanism.....                                 | 360 |
| Figure 98: Multihomed anycast solution.....   | 377 |
| Figure 99: Multihomed anycast solution for Epipe services.....                              | 382 |
| Figure 100: Anycast multihoming and mLDP.....   | 385 |
| Figure 101: The Interconnect ES concept.....  | 387 |
| Figure 102: I-ES — single-active.....   | 392 |
| Figure 103: All-active multihoming and unknown unicast on the NDF.....                      | 393 |
| Figure 104: I-ES in dual EVPN-VXLAN services.....   | 399 |
| Figure 105: Interface-ful IP-VRF-to-IP-VRF with SBD IRB model.....                          | 402 |
| Figure 106: Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model.....               | 403 |
| Figure 107: Interface-less IP-VRF-to-IP-VRF model.....                                      | 404 |
| Figure 108: Interface-less IP-VRF-to-IP-VRF model for IP encapsulation in MPLS tunnels..... | 406 |
| Figure 109: Symmetric IRB model.....  | 408 |
| Figure 110: Extended Layer 2 data centers.....  | 412 |
| Figure 111: Host mobility within the same R-VPLS – initial phase.....                       | 414 |
| Figure 112: Host mobility within the same R-VPLS – move with GARP.....                      | 416 |
| Figure 113: Host mobility within the same R-VPLS – move with data packet.....               | 417 |

---

|  |     |
|--|-----|
| Figure 114: Host mobility within the same R-VPLS – silent host.....                                  | 418 |
| Figure 115: Link loss forwarding for EVPN-VPWS.....  | 420 |
| Figure 116: Core isolation blackhole avoidance.....  | 422 |
| Figure 117: LACP standby signaling from the non-DF.....  | 423 |
| Figure 118: SMET routes replace IGMP/MLD reports.....  | 425 |
| Figure 119: ES FPE-based PW port access using EVPN-VPWS.....   | 428 |
| Figure 120: ES FPE-based pw-port headend.....  | 430 |
| Figure 121: Different owners supported on the same VPRN.....   | 435 |
| Figure 122: BGP path attribute propagation when iff-attribute-uniform-propagation is configured..... | 437 |
| Figure 123: D-PATH attribute.....  | 438 |
| Figure 124: D-PATH attribute example.....  | 440 |
| Figure 125: D-PATH attribute example two.....  | 441 |
| Figure 126: Propagation of BGP path attributes for EVPN-IFF.....                                     | 444 |
| Figure 127: Use of D-PATH for Layer 3 DC gateway redundancy.....                                     | 448 |
| Figure 128: IP-VPN import and EVPN export BGP workflow.....  | 451 |
| Figure 129: EVPN import and I-VPN export BGP workflow.....   | 452 |
| Figure 130: Weighted ECMP for IP Prefix routes use case.....   | 453 |
| Figure 131: Provisioning MPLS-based spoke SDP termination on a flex PW port.....                     | 471 |
| Figure 132: PW SAP configuration example.....  | 472 |
| Figure 133: Provisioning L2oGRE spoke-SDP termination on a flex PW port.....                         | 474 |



# 1 Getting started

## 1.1 About this guide

This guide describes Layer 2 service and EVPN functionality provided by the 7705 SAR Gen 2 and presents examples to configure and implement various protocols and services.

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

Unless otherwise indicated, the topics and commands described in this guide apply only to the 7705 SAR Gen 2 platforms listed in [Platforms and terminology](#).

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



**Note:** Unless otherwise indicated, CLI commands, contexts, and configuration examples in this guide apply for both the classic CLI and the MD-CLI.

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

- *7705 SAR Gen 2 Classic CLI Command Reference Guide*
- *7705 SAR Gen 2 Clear, Monitor, Show, Tools CLI Command Reference Guide* (for both the MD-CLI and classic CLI)
- *7705 SAR Gen 2 MD-CLI Command Reference Guide*



**Note:** This guide generically covers Release 25.x.Rx content and may contain some content that will be released in later maintenance loads. See the *SR OS R25.x.Rx Software Release Notes*, part number 3HE 21562 000x TQZZA, for information about features supported in each load of the Release 25.x.Rx software. For a list of features and CLI commands that are present in SR OS but not supported on the 7705 SAR Gen 2 platforms, see "SR OS Features not Supported on SAR Gen 2" in the *SR OS R25.x.Rx Software Release Notes*.

## 1.2 Platforms and terminology



**Note:** Unless explicitly noted otherwise, this guide uses the terminology defined in the following table to collectively designate the specified platforms.

Table 1: Platforms and terminology

| Platform   | Collective platform designation |
|------------|---------------------------------|
| 7705 SAR-1 | 7705 SAR Gen 2                  |

## 1.3 Conventions

This section describes the general conventions used in this guide.

### 1.3.1 Precautionary and information messages

The following information symbols are used in the documentation.



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



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



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



**Note:** Note provides additional operational information.



**Tip:** Tip provides suggestions for use or best practices.

### 1.3.2 Options or substeps in procedures and sequential workflows

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

#### Example: Options in a procedure

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

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

#### Example: Substeps in a procedure

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

- b.** This is another substep.

## 2 VLL services

This chapter provides information about Virtual Leased Line (VLL) services and implementation notes.

### 2.1 Ethernet pipe service

This section provides information about the Ethernet pipe (Epipe) service and implementation notes.

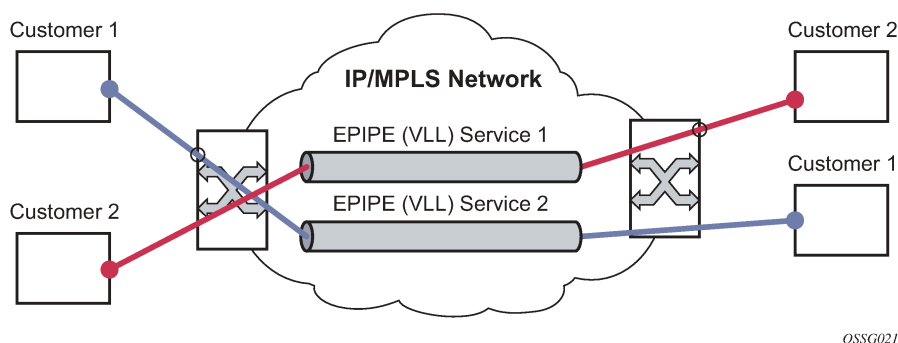
#### 2.1.1 Epipe service overview

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

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

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

Figure 1: Epipe/VLL service



#### 2.1.2 Epipe service pseudowire VLAN tag processing

Distributed Epipe services are connected using a pseudowire, which can be provisioned statically or dynamically and is represented in the system as a spoke-SDP. The spoke-SDP can be configured to process zero, one, or two VLAN tags as traffic is transmitted and received; see [Table 2: Epipe spoke-SDP](#)

[VLAN tag processing: ingress](#) and [Table 3: Epipe-spoke SDP VLAN tag processing: egress](#) for the ingress and egress tag processing. In the transmit direction, VLAN tags are added to the frame being sent. In the received direction, VLAN tags are removed from the frame being received. This is analogous to the SAP operations on a null, dot1q, and QinQ SAP.

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

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

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

- **zero VLAN tags processed**

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

- **one VLAN tag processed**

This requires one of the following configurations:

- **vc-type vlan** under the spoke-SDP or in the related PW template
- **vc-type ether** and **force-vlan-vc-forwarding** under the spoke-SDP or in the related PW template

- **two VLAN tags processed**

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

The PW template configuration provides support for BGP VPWS services.

The following restrictions apply to VLAN tag processing:

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

[Table 2: Epipe spoke-SDP VLAN tag processing: ingress](#) and [Table 3: Epipe-spoke SDP VLAN tag processing: egress](#) describe the VLAN tag processing with respect to the zero, one, and two VLAN tag configuration described for the VLAN identifiers, Ethertype, ingress QoS classification (dot1p or DE), and QoS propagation to the egress (which can be used for egress classification or to set the QoS information, or both, in the innermost egress VLAN tag).

Table 2: Epipe spoke-SDP VLAN tag processing: ingress

| Ingress (received on spoke-SDP)                     | Zero VLAN tags | One VLAN tag  | Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])   |
|---|----------------|---|---|
| VLAN identifiers                                    | —              | Ignored   | Both inner and outer ignored  |
| Ethertype (to determine the presence of a VLAN tag) | N/A            | 0x8100 or value configured under <b>sdp vlan-vc-etype</b> | Both inner and outer VLAN tags: 0x8100, or outer VLAN tag value configured under <b>sdp vlan-vc-etype</b> (inner VLAN tag value must be 0x8100)   |
| Ingress QoS (dot1p/DE) classification               | —              | Ignored   | Both inner and outer ignored  |
| QoS (dot1p/DE) propagation to egress                | Dot1p/DE=0     | Dot1p/DE taken from received VLAN tag                     | <p>Dot1p/DE taken as follows:</p> <ul style="list-style-type: none"> <li>If the egress encapsulation is a Dot1q SAP, Dot1p/DE bits are taken from the outer received VLAN tag</li> <li>If the egress encapsulation is QinQ SAP, the s-tag bits are taken from the outer received VLAN tag and the c-tag bits from the inner received VLAN tag</li> </ul> <p>The egress cannot be a spoke-sdp because <b>force-qinq-vc-forwarding</b> does not support multisegment PWs.</p> |

Table 3: Epipe-spoke SDP VLAN tag processing: egress

| Egress (sent on mesh or spoke-SDP)  | Zero VLAN tags | One VLAN tag   | Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])  |
|-------------------------------------|----------------|--|--|
| VLAN identifiers (set in VLAN tags) | —              | <p>The tag is derived from one of the following:</p> <ul style="list-style-type: none"> <li>the <b>vlan-vc-tag</b> value configured in PW template or under the spoke-SDP</li> <li>value from the inner tag received on a QinQ SAP or QinQ spoke-SDP</li> <li>value from the VLAN tag received on a dot1q SAP</li> </ul> | <p>The inner and outer VLAN tags are derived from one of the following:</p> <ul style="list-style-type: none"> <li>vlan-vc-tag value configured in PW template or under the spoke-SDP: <ul style="list-style-type: none"> <li>If c-tag-c-tag is configured, both inner and outer tags are taken</li> </ul> </li> </ul> |

| Egress (sent on mesh or spoke-SDP)       | Zero VLAN tags | One VLAN tag  | Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])   |
|--|----------------|---|---|
|  |                | <p>or spoke-SDP (with <b>vc-type vlan</b> or <b>force-vlan-vc-forwarding</b>)</p> <ul style="list-style-type: none"> <li>value from the outer tag received on a qtag.* SAP</li> <li>0 if there is no service delimiting VLAN tag at the ingress SAP or spoke-SDP</li> </ul> | <p>from the vlan-vc-tag value</p> <ul style="list-style-type: none"> <li>If s-tag-c-tag is configured, only the s-tag value is taken from vlan-vc-tag</li> <li>value from the inner tag received on a QinQ SAP for the c-tag-c-tag option and value from outer/inner tag received on a QinQ SAP for the s-tag-c-tag configuration option</li> <li>value from the VLAN tag received on a dot1q SAP for the c-tag-c-tag option and value from the VLAN tag for the outer tag and zero for the inner tag</li> <li>value from the outer tag received on a qtag.* SAP for the c-tag-c-tag option and value from the VLAN tag for the outer tag and zero for the inner tag</li> <li>value 0 if there is no service delimiting VLAN tag at the ingress SAP or spoke-SDP Ethertype (set in VLAN tags).</li> </ul> |
| Ethertype (set in VLAN tags)             | —              | 0x8100 or value configured under <b>sdp vlan-vc-etype</b>   | Both inner and outer VLAN tags: 0x8100, or outer VLAN tag value configured under <b>sdp vlan-vc-etype</b> (inner VLAN tag value is 0x8100)  |
| Egress QoS (dot1p/DE) (set in VLAN tags) | —              | <p>The tag taken from the innermost ingress service delimiting tag can be one of the following:</p> <ul style="list-style-type: none"> <li>The inner tag received on a QinQ SAP or QinQ spoke-SDP</li> </ul>  | <p>Inner and outer dot1p/DE:</p> <p>If <b>c-tag-c-tag</b> is configured, the inner and outer dot1p/DE bits are both taken from the innermost ingress service delimiting tag. It can be one of the following:</p>  |

| Egress (sent on mesh or spoke-SDP) | Zero VLAN tags | One VLAN tag  | Two VLAN tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])   |
|------------------------------------|----------------|---|---|
|                                    |                | <ul style="list-style-type: none"> <li>value from the VLAN tag received on a dot1q SAP or spoke-SDP (with <b>vc-type vlan</b> or <b>force-vlan-vc-forwarding</b>)</li> <li>value from the outer tag received on a qtag.* SAP</li> </ul> | <ul style="list-style-type: none"> <li>inner tag received on a QinQ SAP</li> <li>value from the VLAN tag received on a dot1q SAP</li> <li>value from the outer tag received on a qtag.* SAP</li> <li>value 0 if there is no service delimiting VLAN tag at the ingress SAP</li> </ul>   |
|                                    |                | 0 if there is no service delimiting VLAN tag at the ingress SAP or spoke-SDP<br>Note that neither the inner nor outer dot1p/DE values can be explicitly set.  | <p>If <b>s-tag-c-tag</b> is configured, the inner and outer dot1p/DE bits are taken from the inner and outer ingress service delimiting tag (respectively). They can be:</p> <ul style="list-style-type: none"> <li>inner and outer tags received on a QinQ SAP</li> <li>value from the VLAN tag received on a dot1q SAP for the outer tag and zero for the inner tag</li> <li>value from the outer tag received on a qtag.* SAP for the outer tag and zero for the inner tag</li> <li>value 0 if there is no service delimiting VLAN tag at the ingress SAP</li> </ul> <p>Note that neither the inner nor outer dot1p/DE values can be explicitly set.</p> |

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

### 2.1.3 Epipe up operational state configuration option

By default, the operational state of the Epipe is tied to the state of the two connections that comprise the Epipe. If either of the connections in the Epipe are operationally down, the Epipe service that contains that connection is also operationally down. The operator can configure a single SAP within an Epipe that does not affect the operational state of that Epipe, using the optional **ignore-oper-state** command. Within an Epipe, if a SAP that includes this optional command becomes operationally down, the operational state



of the Epipe does not transition to down. The operational state of the Epipe remains up. This does not change that the SAP is down and no traffic can transit an operationally down SAP. Removing and adding this command on the fly evaluates the operational state of the service, based on the SAPs and the addition or deletion of this command.

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

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

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

## 2.1.4 VLL CAC

The VLL Connection Admission Control (CAC) is supported for the 7705 SAR Gen 2 and provides a method to administratively account for the bandwidth used by VLL services inside an SDP that consists of RSVP LSPs.

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

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

## 2.1.5 MC-Ring and VLL

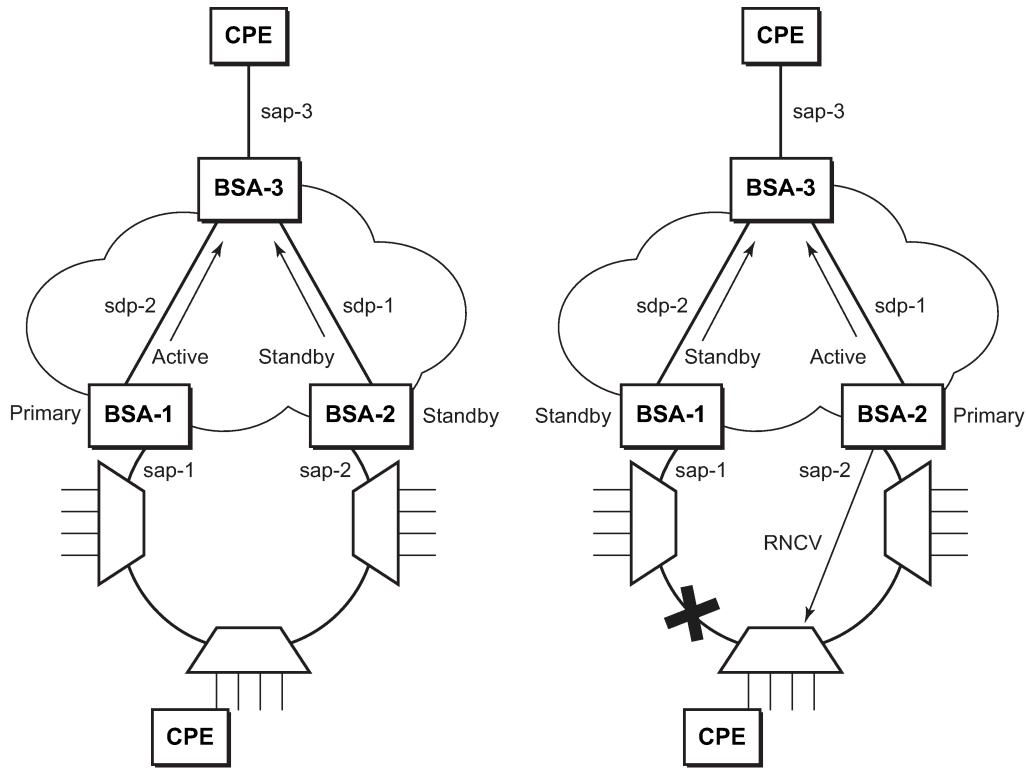
To support redundant VLL access in ring configurations, the multichassis ring (MC-Ring) feature is applicable to VLL SAPs. A conceptual drawing of the operation is shown in [Figure 2: MC-Ring in a combination with VLL service](#). The specific CPE that is connected behind the ring node has access to both BSAs through the same VLAN provisioned in all ring nodes. There are two SAPs (with the same VLAN) provisioned on both nodes.

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

The mapping of individual SAPs to the ring nodes is done statically through CLI provisioning. To keep the convergence time to a minimum, MAC learning must be disabled on the ring node so all CPE originated

traffic is sent in both directions. If the status is operationally down on the SAP on the standby BSA, that part of the traffic is blocked and not forwarded to the remote site.

Figure 2: MC-Ring in a combination with VLL service



OSSG174

## 2.2 Pseudowire redundancy service models

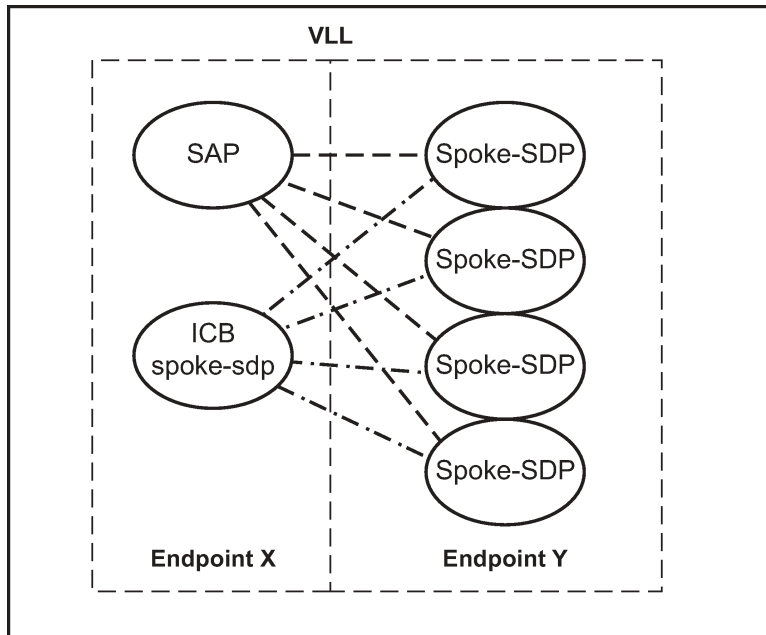
This section describes the MC-LAG and pseudowire redundancy scenarios and the algorithm used to select the active transmit object in a VLL endpoint.

### 2.2.1 Redundant VLL service model

To implement pseudowire redundancy, a VLL service accommodates more than a single object on the SAP side and on the spoke-SDP side.

The following figure shows the model for a redundant VLL service based on the concept of endpoints.

Figure 3: Redundant VLL endpoint objects



OSSG211

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

To add more objects, create up to two explicitly named endpoints per VLL service. The endpoint name is locally significant to the VLL service. In the preceding figure, endpoints are referred to as endpoint X and endpoint Y.

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

- **SAP**

A maximum of only one SAP per VLL endpoint is supported.

- **primary spoke-SDP**

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

- **secondary spoke-SDP**

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

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

This special pseudowire is used for MC-LAG and pseudowire redundancy applications. Forwarding between ICBs is blocked on the same node. At the time this endpoint object is created, the user must explicitly indicate that the spoke-SDP is an ICB; however, a few scenarios are possible where the user

can configure the spoke-SDP as an ICB or as a regular spoke-SDP on a specified node. The CLI for those cases indicate both options.

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

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

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

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

## 2.2.2 T-LDP status notification handling rules

Using [Figure 3: Redundant VLL endpoint objects](#) as a reference, this section describes the rules for generating, processing, and merging T-LDP status notifications in a VLL service with endpoints. Any allowed combination of objects, as specified in [Redundant VLL service model](#), can be used on endpoints X and Y.

This section uses the specific combination of objects shown in [Figure 3: Redundant VLL endpoint objects](#) as an example to describe the more general rules.

### 2.2.2.1 Processing endpoint SAP active/standby status bits

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

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

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

### 2.2.2.2 Processing and merging

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

If the SAP in endpoint X transitions locally to the down state or receives a SAP down notification via the SAP-specific OAM signal, the node must send T-LDP SAP down status bits on endpoint Y ICB spoke SDPs only. Ethernet SAP does not support the SAP OAM protocol. No other SAP types can exist on the same endpoint as an ICB spoke SDP because a non-Ethernet SAP cannot be part of an MC-LAG instance.

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

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

If any or all of the following are true for all objects in endpoint X, the node must send status bits of SAP down over all endpoint Y spoke SDPs, including the ICB:

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

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

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

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

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

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

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

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

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

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

An Ethernet SAP does not support signaling status notifications.

## 2.3 VLL using G.8031 protected Ethernet tunnels

The use of MPLS tunnels provides the 7705 SAR Gen 2 a way to scale the core while offering fast failover times using MPLS FRR. In environments where Ethernet services are deployed using native Ethernet backbones, Ethernet tunnels are provided to achieve the same fast failover times as in the MPLS FRR case.

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

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

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

For further information, see the *7705 SAR Gen 2 Services Overview Guide*.

## 2.4 MPLS EL and hash label

The router supports the MPLS entropy label (EL) (RFC 6790) and the Flow Aware Transport (FAT) label, known as the hash label (RFC 6391). These labels allow LSR nodes in a network to load-balance labeled packets in a more granular way than by hashing on the standard label stack. See the *7705 SAR Gen 2 MPLS Guide* for more information.

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

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

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

For BGP-VPWS spoke SDPs, configure it using the following command:

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

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

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

## 2.5 BGP VPWS

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

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

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

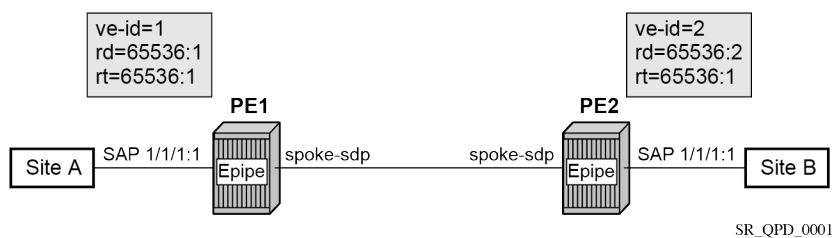
Inter-AS model C and dual-homing are supported.

## 2.5.1 Single-homed BGP VPWS

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

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

Figure 4: Single-homed BGP-VPWS example



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

## 2.5.2 Dual-homed BGP VPWS

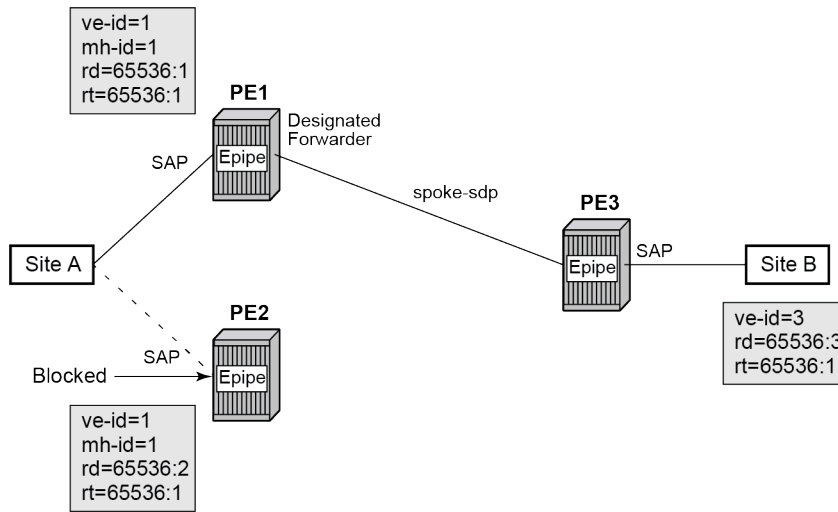
A BGP-VPWS service can benefit from dual-homing, as described in IETF Draft *draft-ietf-bess-vpls-multihoming-01*. When using dual-homing, two PEs connect to a site, with one PE being the designated forwarder (DF) for the site and the other blocking its connection to the site. On failure of the active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

### 2.5.2.1 Single pseudowire example

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

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

Figure 5: Dual-homed BGP VPWS with single pseudowire



SR\_QPD\_0002

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

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

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

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

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

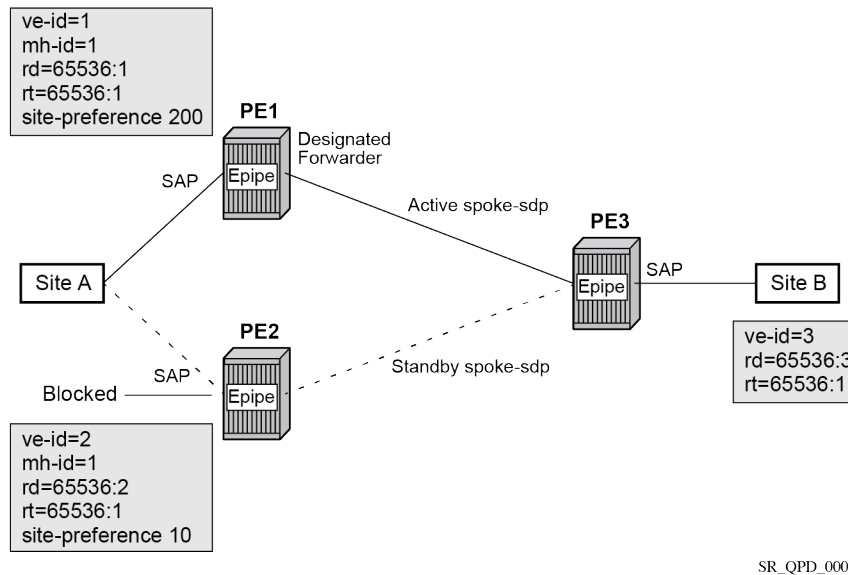
### 2.5.2.2 Active/standby pseudowire example

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

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



Figure 6: Dual-homed BGP VPWS with active/standby pseudowires



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

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

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

### 2.5.3 BGP VPWS pseudowire switching

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

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

The following configuration is required:

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

**Example:** BGP VPWS service configured to allow pseudowire switching

```
configure
```

```

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

```

## 2.5.4 Pseudowire signaling

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

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

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

*Figure 7: BGP VPWS update extended community format*

|                                    |
|------------------------------------|
| Extended Community Type (2 Octets) |
| Encaps Type (1 Octet)              |
| Control Flags (1 Octet)            |
| Layer-2 MTU (2 Octets)             |
| VPLS Preference (2 Octets)         |

*L2\_Guide\_42*

- **extended community type**

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

- **encaps type**

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

- **control flags**

This is control information concerning the pseudowires, see [Figure 8: Control flags](#) for more information.

- **Layer 2 MTU**

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

- **VPLS preference**

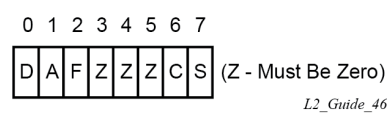
VPLS preference has a default value of zero for BGP-VPWS updates sent by the system, indicating that it is not in use. If the **site-preference** is configured, its value is used for the VPLS preference and is also used in the local designated forwarder election.

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

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

The following figure shows the pseudowire control flags.

Figure 8: Control flags



The following bits in the Control Flags are defined:

- D

Access circuit down indicator from IETF Draft *draft-kothari-l2vpn-auto-site-id-01*. D is 1 if all access circuits are down, otherwise D is 0.
- A

Automatic site ID allocation, which is not supported. This is ignored on receipt and set to 0 on sending.
- F

MAC flush indicator. This is not supported because it relates to a VPLS service. This is set to 0 and ignored on receipt.
- C

Presence of a control word. Control word usage is supported. When this is set to 1, packets are sent and are expected to be received with a control word. When this is set to 0, packets are sent and are expected to be received without a control word (by default).
- S

Sequenced delivery. Sequenced delivery is not supported. This is set to 0 on sending (no sequenced delivery) and, if a non-zero value is received (indicating sequenced delivery required), the pseudowire is not created.

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

Figure 9: BGP VPWS NLRI

|                                  |
|----------------------------------|
| Length (2 Octets)                |
| Route Distinguisher (8 Octets)   |
| VE ID (2 Octets)                 |
| VE Block Offset (2 Octets)       |
| VE Block Size (2 Octets)         |
| Label Base (3 Octets)            |
| Circuit Status Vector (4 Octets) |

L2\_Guide\_43

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

The circuit status vector is encoded as a TLV, as shown in [Figure 10: BGP VPWS NLRI TLV extension format](#) and [Figure 11: Circuit status vector TLV type](#).

Figure 10: BGP VPWS NLRI TLV extension format

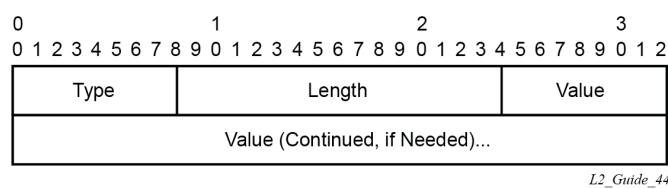


Figure 11: Circuit status vector TLV type

| TLV Type | Description           |
|----------|-----------------------|
| 1        | Circuit Status Vector |

L2\_Guide\_45

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

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

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

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

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

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

2.5.5 BGP-VPWS with inter-AS model C

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

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

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

## 2.5.6 BGP VPWS configuration procedure

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

- configuring the BGP Route Distinguisher, Route Target  
The updates are accepted into the service only if they contain the configured import route-target.
- configuring a binding to the pseudowire template  
The multiple pseudowire template bindings can be configured with their associated route-targets used to control which is applied.
- configuring the SAP or static GRE tunnel
- configuring the name of the local VE and its associated VE ID
- configuring the name of the remote VE and its associated VE ID
- for a dual-homed PE:
  - enabling the site
  - configuring the site with non-zero site-preference
- for a remote PE, configuring up to two remote VE names and associated VE IDs
- enabling BGP VPWS

## 2.5.7 Use of pseudowire template for BGP VPWS

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

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

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

The same value must be used by the remote BGP VPWS instance to ensure that the related pseudowire comes up.

- Layer 2 MTU is derived from the service VPLS **service-mtu** parameter.  
The same value must be used by the remote BGP VPWS instance to ensure that the related pseudowire comes up.
- Control Flag C can be 0 or 1, depending on the setting of the **control-word** parameter in the PW template 0.
- Control Flag S is always 0.

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

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

To interoperate with existing implementations, if the received MTU value = 0, the MTU negotiation does not take place; the related pseudowire is set up ignoring the MTU.

- If the value of the S flag is not zero, the pseudowire is not created.

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

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

For more information about this command, see the *7705 SAR Gen 2 Services Overview Guide*.

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

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

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

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

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

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

## 2.5.8 Use of endpoint for BGP VPWS

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

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

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

## 2.6 VLL service considerations

This section describes the general service features and any special capabilities or considerations as they relate to VLL services.

### 2.6.1 SDPs

The most basic SDPs must include the following:

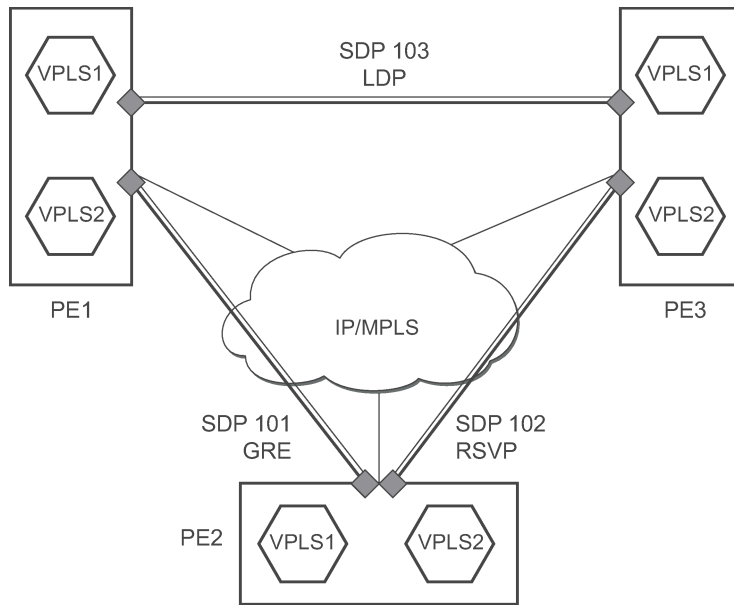
- a locally unique SDP identification (ID) number
- the system IP address of the originating and far-end routers
- an SDP encapsulation type, either GRE or MPLS

#### 2.6.1.1 SDP statistics for VPLS and VLL services

The three-node network in [Figure 12: SDP statistics for VPLS and VLL services](#) shows two MPLS SDPs and one GRE SDP defined between the nodes. These SDPs connect VPLS1 and VPLS2 instances that are defined in the three nodes. With this feature, the operator has local CLI-based and SNMP-based statistics collection for each VC used in the SDPs. This allows for traffic management of tunnel usage by the different services and with aggregation the total tunnel usage.



Figure 12: SDP statistics for VPLS and VLL services



OSSG208

## 2.6.2 SAP encapsulations and pseudowire types

The Epipe service is designed to carry Ethernet frame payloads, so it can provide connectivity between any two SAPs that pass Ethernet frames.

The following SAP encapsulations are supported on the Epipe service:

- Ethernet null
- Ethernet dot1q
- QinQ

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

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

### 2.6.2.1 QoS policies

When applied to Epipe services, service ingress QoS policies only create the unicast queues defined in the policy. The multipoint queues are not created on the service.

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

### 2.6.2.2 Filter policies

Epipe and lpipe services can have a single filter policy associated on both ingress and egress. Both MAC and IP filter policies can be used on Epipe services.

### 2.6.2.3 MAC resources

Epipe services are point-to-point Layer 2 VPNs capable of carrying any Ethernet payloads. Although an Epipe is a Layer 2 service, the 7705 SAR Gen 2 Epipe implementation does not perform any MAC learning on the service, so Epipe services do not consume any MAC hardware resources.

## 2.7 Configuring a VLL service using CLI

This section provides information to configure VLL services using the CLI.

### 2.7.1 Common configuration tasks

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

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

### 2.7.2 Configuring VLL components

This section provides VLL configuration examples for the VLL services.

#### 2.7.2.1 Creating an Epipe service

Use the following CLI syntax to create an Epipe service.

CLI syntax:

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

The following example shows an Epipe configuration:

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

### 2.7.2.1.1 Configuring Epipe SAP parameters

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

Use the following CLI syntax to create:

- [Local Epipe SAPs](#)
- [Distributed Epipe SAPs](#)

CLI syntax:

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

#### 2.7.2.1.1.1 Local Epipe SAPs

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

By default, QoS policy ID 1 is applied to ingress and egress service SAPs. Existing filter policies or other existing QoS policies can be associated with service SAPs on ingress and egress ports. [Table 4: Supported SAP types](#) shows supported SAP types.

Table 4: Supported SAP types

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

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

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

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

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

config>service>epipe# sap 1/1/3:0 create
config>service>epipe>sap# ingress
config>service>epipe>sap>ingress# qos 555
config>service>epipe>sap>ingress# filter ip 1
config>service>epipe>sap>ingress# exit
config>service>epipe>sap# egress
config>service>epipe>sap>egress# qos 627
config>service>epipe>sap>egress# scheduler-policy alpha
config>service>epipe>sap>egress# exit
config>service>epipe>sap# no shutdown
config>service>epipe>sap# exit
```

The following example shows the local Epipe configuration:

```
A:ALA-1>config>service# info
-----
...
    epipe 500 customer 5 vpn 500 create
      description "Local epipe service"
      sap 1/1/2:0 create
        ingress
```

```

        qos 20
        filter ip 1
    exit
    egress
        scheduler-policy "test1"
        qos 20
    exit
exit
sap 1/1/3:0 create
    ingress
        qos 555
        filter ip 1
    exit
    egress
        scheduler-policy "alpha"
        qos 627
    exit
exit
no shutdown
exit
-----
A:ALA-1>config>service#

```

### 2.7.2.1.2 Distributed Epipe SAPs

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

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

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

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

For SDP configuration information, see the *7705 SAR Gen 2 Services Overview Guide*. For SDP binding information, see [Configuring SDP bindings](#).

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

```

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

```

```

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

```

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

```

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

A:ALA-2>config>service# info
-----
...
    epipe 5500 customer 5 vpn 5500 create
        description "Distributed epipe service to west coast"
        sap 441/1/4:550 create
            ingress
                qos 654
                filter ip 1020
            exit
            egress
                scheduler-policy "test1"
                qos 432
                filter ip 6
            exit
        exit
    exit
...
-----
A:ALA-2>config>service#

```

### 2.7.2.1.2.1 Configuring ingress and egress SAP parameters

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

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

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

The following example shows the SAP ingress and egress parameters:

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

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

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

### 2.7.2.1.3 Configuring SDP bindings

VPLS provides scaling and operational advantages. A hierarchical configuration eliminates the need for a full mesh of VCs between participating devices. Hierarchy is achieved by enhancing the base VPLS core mesh of VCs with access VCs (spoke) to form two tiers. Spoke SDPs are generally created between Layer 2 switches and placed at the MTU. The PE routers are placed at the service provider's Point of Presence (POP). Signaling and replication overhead on all devices is considerably reduced.

A spoke SDP is treated like the equivalent of a traditional bridge port where flooded traffic received on the spoke SDP is replicated on all other "ports" (other spoke and mesh SDPs or SAPs) and not transmitted on the port it was received (unless a split horizon group was defined on the spoke SDP; see section [Configuring VPLS spoke SDPs with split horizon](#)).

A spoke SDP connects a VPLS service between two sites and, in its simplest form, could be a single tunnel LSP. A set of ingress and egress VC labels are exchanged for each VPLS service instance to be transported over this LSP. The PE routers at each end treat this as a virtual spoke connection for the VPLS service in the same way as the PE-MTU connections. This architecture minimizes the signaling overhead and avoids a full mesh of VCs and LSPs between the two metro networks.

A mesh SDP bound to a service is logically treated like a single bridge "port" for flooded traffic where flooded traffic received on any mesh SDP on the service is replicated to other "ports" (spoke SDPs and SAPs) and not transmitted on any mesh SDPs.

A VC-ID can be specified with the SDP-ID. The VC-ID is used instead of a label to identify a virtual circuit. The VC-ID is significant between peer SRs on the same hierarchical level. The value of a VC-ID is conceptually independent from the value of the label or any other datalink-specific information of the VC.

[Figure 51: SDPs — unidirectional tunnels](#) shows an example of a distributed VPLS service configuration of spoke and mesh SDPs (unidirectional tunnels) between routers and MTUs.

### 2.7.3 Using spoke SDP control words

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

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

When the control word is enabled, VCCV packets also include the VCCV control word. In that case, the VCCV CC type 1 (OAM CW) is signaled in the VCCV parameter in the FEC. If the control word is disabled on the spoke-SDP, the Router Alert label is used. In that case, VCCV CC type 2 is signaled. For a multisegment pseudowire (MS-PW), the CC type 1 is the only type supported; therefore, the control word must be enabled on the spoke SDP to be able to use VCCV-ping and VCCV-trace.

The following example shows a spoke SDP control word configuration.



## Example

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

### 2.7.4 Same-fate Epipe VLANs access protection

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

```
*A:node-2>config>eth-tunnel 1
-----
description "Protection is APS"
protection-type 8031_1tol
ethernet
    mac 00:11:11:11:11:12
    encap-type dot1q
exit
ccm-hold-time down 5 up 10 // 50 ms down, 1 second up
path 1
    member 1/1/1
    control-tag 5 // primary control vlan 5
    precedence primary
    eth-cfm
        mep 2 domain 1 association 1
        ccm-enable
        control-mep
        no shutdown
    exit
exit
no shutdown
exit
path 2
    member 2/1/1
    control-tag 105 //secondary control vlan 105
    eth-cfm
        mep 2 domain 1 association 2
        ccm-enable
```

```

        control-mep
        no shutdown
    exit
    exit
    no shutdown
    exit
    no shutdown
-----
# Configure Ethernet tunnel SAPs
-----
*A:node-2>config>service epipe 10 customer 5 create
    sap eth-tunnel-1 create // Uses control tags from the Ethernet tunnel port
    description "g8031-protected access ctl/data SAP for eth-tunnel 1"

    exit
    no shutdown
-----
*A:node-2>config>service epipe 11 customer 5 create
    sap eth-tunnel-1:1 create
    description "g8031-protected access same-fate SAP for eth-tunnel 1"

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

```

## 2.7.5 Pseudowire configuration notes

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

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

```

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

```

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

CLI syntax:

```

config>service# epipe service-id [customer customer-id] [vpn vpn-id] [vc-switching]
description description-string

```

```
spoke-sdp sdp-id:vc-id
```

CLI syntax:

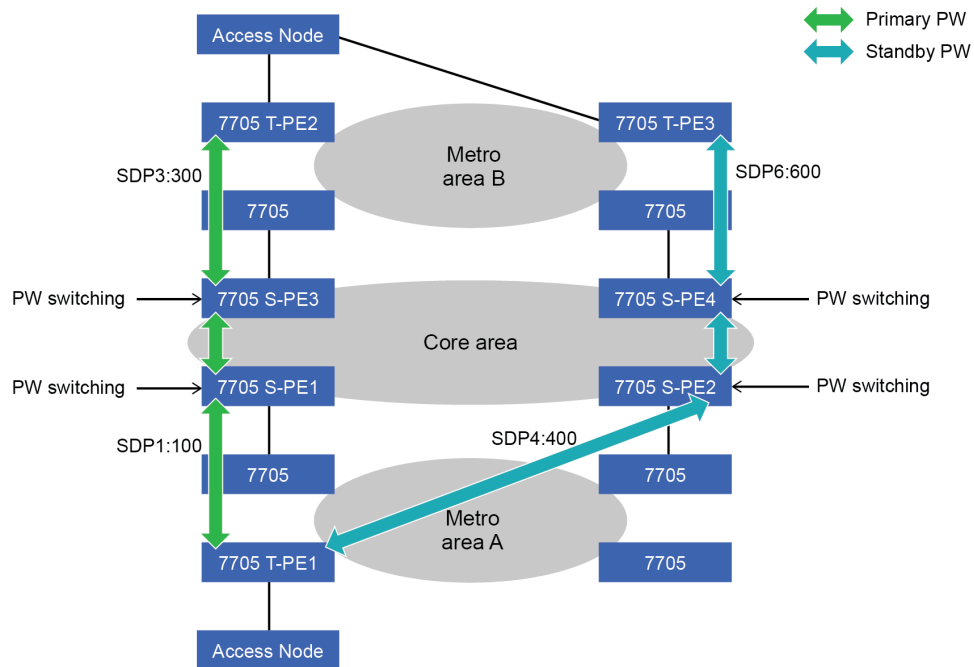
```
config>service# ipipe service-id [customer customer-id][vpn vpn-id] [vc-switching]
description description-string
spoke-sdp sdp-id:vc-id
```

The following example shows configurations for each service:

```
*A:ALA-48>config>service# info
-----
...
    epipe 107 customer 1 vpn 107 vc-switching create
        description "Default epipe description for service id 107"
        spoke-sdp 3:8 create
        exit
        spoke-sdp 6:207 create
        exit
        no shutdown
    exit
...
    ipipe 108 customer 1 vpn 108 vc-switching create
        description "Default ipipe description for service id 108"
        spoke-sdp 3:9 create
        exit
        spoke-sdp 6:208 create
        exit
        no shutdown
    exit
...
```

## 2.7.6 Configuring two VLL paths terminating on T-PE2

Figure 13: VLL resilience with pseudowire redundancy and switching



sw4315

### T-PE1

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

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

### T-PE2

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

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

```

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

```

### S-PE1

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

The following example shows the configuration:

```

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

```

### S-PE2

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

The following example shows the configuration:

```

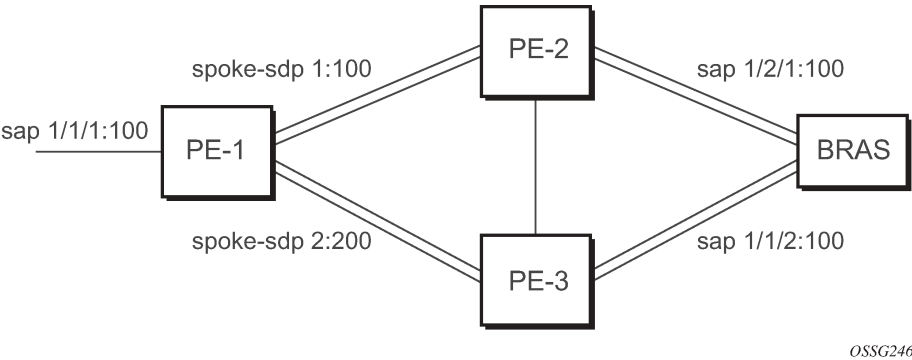
*A:ALA-S-PE2>config>service>epipe# info
-----
...
    spoke-sdp 2:200 create
    exit
    spoke-sdp 3:300 create
    exit
    no shutdown
-----
*A:ALA-S-PE2>config>service>epipe#

```

## 2.7.7 Configuring VLL resilience

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

Figure 14: VLL resilience



PE-1

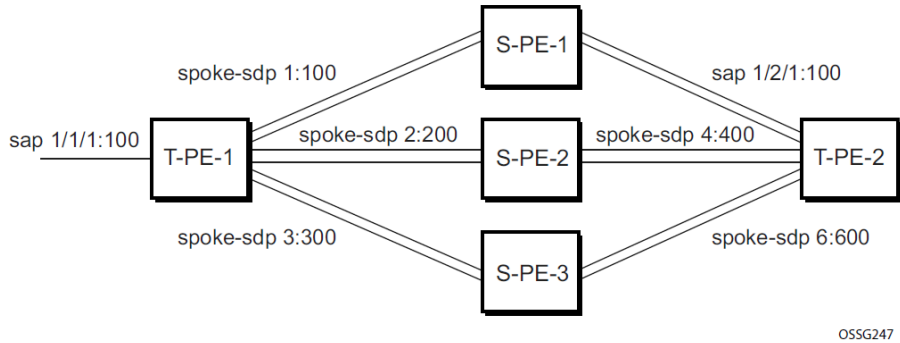
The following example shows the configuration on PE-1:

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

2.7.8 Configuring VLL resilience for a switched pseudowire path

The following figure shows an example of VLL resilience with pseudowire switching.

Figure 15: VLL resilience with pseudowire switching



T-PE-1

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

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

## T-PE-2

The following is an example configuration output on T-PE-2.

```
*A:ALA-49>config>service>epipe# info
-----
    endpoint "x" create
    exit
    endpoint "y" create
        revert-time 100
    exit
    spoke-sdp 4:400 endpoint "y" create
        precedence primary
    exit
    spoke-sdp 5:500 endpoint "y" create
        precedence 1
    exit
    spoke-sdp 6:600 endpoint "y" create
        precedence 1
    exit
    no shutdown
-----
*A:ALA-49>config>service>epipe#
```

## S-PE-1

The following is an example configuration output on S-PE-1.

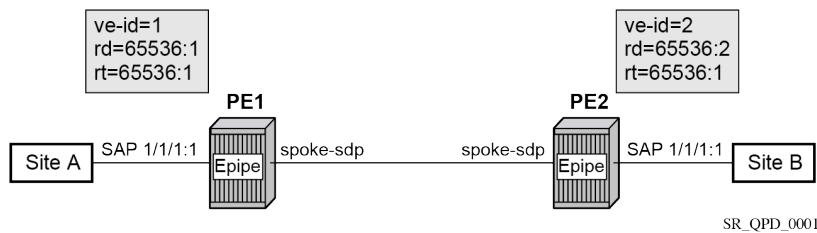
```
*A:ALA-50>config>service>epipe# info
-----
...
    spoke-sdp 1:100 create
    exit
    spoke-sdp 4:400 create
    exit
    no shutdown
-----
*A:ALA-49>config>service>epipe#
```

## 2.7.9 Configuring BGP VPWS

### 2.7.9.1 Single-homed BGP VPWS

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

Figure 16: Single-homed BGP VPWS configuration example



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

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

#### Example: PE1 configuration

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

#### Example: PE2 configuration

```
pw-template 1 create
```



```

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

```

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

```
show service l2-route-table bgp-vpws detail
```

### Output example

```

=====
Services: L2 Bgp-Vpws Route Information - Summary
=====
Svc Id       : 1
VeId         : 2
PW Temp Id   : 1
RD           : *65536:2
Next Hop     : 10.1.1.2
State (D-Bit) : up(0)
Path MTU     : 1514
Control Word : 0
Seq Delivery : 0
Status       : active
Tx Status    : active
CSV          : 0
Preference   : 0
Sdp Bind Id  : 17407:4294967295
=====
A:PE1#

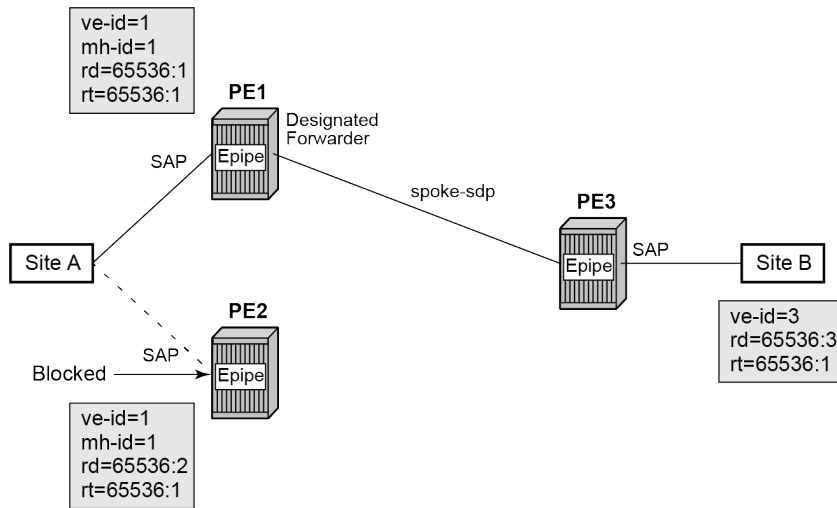
```

## 2.7.9.2 Dual-homed BGP VPWS

### Single pseudowire example:

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

Figure 17: Example of dual-homed BGP VPWS with single pseudowire



SR\_QPD\_0002

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

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

The following examples show the BGP VPWS configuration on each PE.

### Example: Dual-homed BGP VPWS configuration with single pseudowires on PE1

```

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

```

```
no shutdown
exit
```

### Example: Dual-homed BGP VPWS configuration with single pseudowires on PE2

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

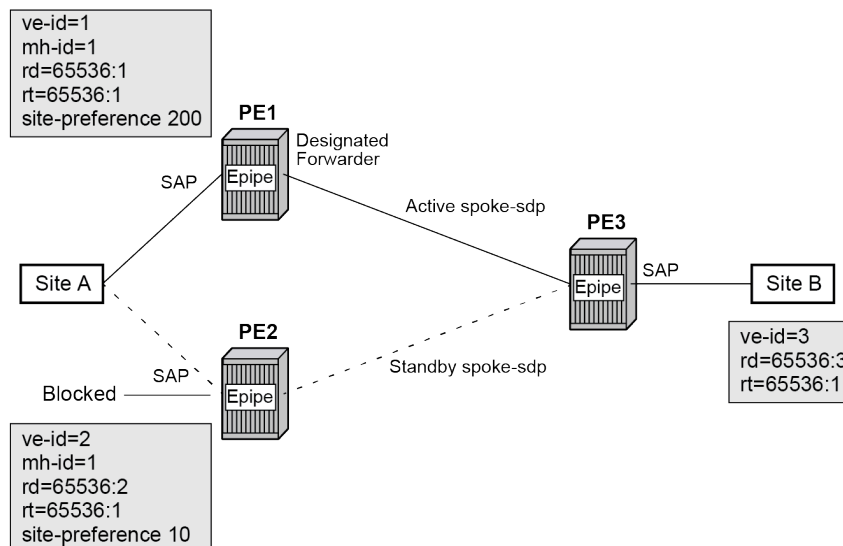
### Example: Dual-homed BGP VPWS configuration with single pseudowires on PE3

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

### Active/standby pseudowire example:

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

Figure 18: Example of dual-homed BGP VPWS with active/standby pseudowires



SR\_QPD\_0003

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

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

The following examples show the BGP VPWS configurations on each PE.

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

```

pw-template 1 create
exit
epipe 1 customer 1 create
  bgp
    route-distinguisher 65536:1
    route-target export target:65536:1 import target:65536:1
    pw-template-binding 1
  exit
exit
bgp-vpws
  ve-name PE1
  ve-id 1
  exit
  remote-ve-name PE3
  ve-id 3
  exit

```

```

        no shutdown
    exit
    sap 1/1/1:1 create
    exit
    site "siteA" create
        site-id 1
        sap 1/1/1:1
        boot-timer 20
        site-activation-timer 5
        site-preference 200
        no shutdown
    exit
    no shutdown
exit

```

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

```

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

```

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

```

pw-template 1 create
exit
epipe 1 customer 1 create
    bgp
        route-distinguisher 65536:3
        route-target export target:65536:1 import target:65536:1
        pw-template-binding 1
    exit
exit
bgp-vpws

```

```

    ve-name PE3
    ve-id 3
  exit
  remote-ve-name PE1
  ve-id 1
  exit
  remote-ve-name PE2
  ve-id 2
  exit
  no shutdown
exit
sap 1/1/1:1 create
exit
no shutdown
exit

```

## 2.8 Service management tasks

This section describes the VLL service management tasks.

### 2.8.1 Modifying Epipe service parameters

Use the following syntax to add an accounting policy to an existing SAP.

```

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

```

#### Example: SAP configuration output

```

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

```

### 2.8.2 Disabling an Epipe service

Use the following syntax to shut down an Epipe service without deleting the service parameters.

```

config>service> epipe service-id
shutdown

```

```

config>service# epipe 2

```

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

### 2.8.3 Re-enabling an Epipe service

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

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

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

### 2.8.4 Deleting an Epipe service

#### About this task

Perform the following steps to delete an Epipe service.

#### Procedure

- Step 1.** Shut down the SAP and SDP.
- Step 2.** Delete the SAP and SDP.
- Step 3.** Shut down the service.
- Step 4.** Use the following CLI syntax to delete the Epipe service:

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

#### Example

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

## 3 Virtual Private LAN Service

This chapter provides information about the Virtual Private LAN Service (VPLS), process overview, configuration tasks, and implementation notes.

### 3.1 VPLS service overview

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

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

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

Other VPLS advantages include the following.

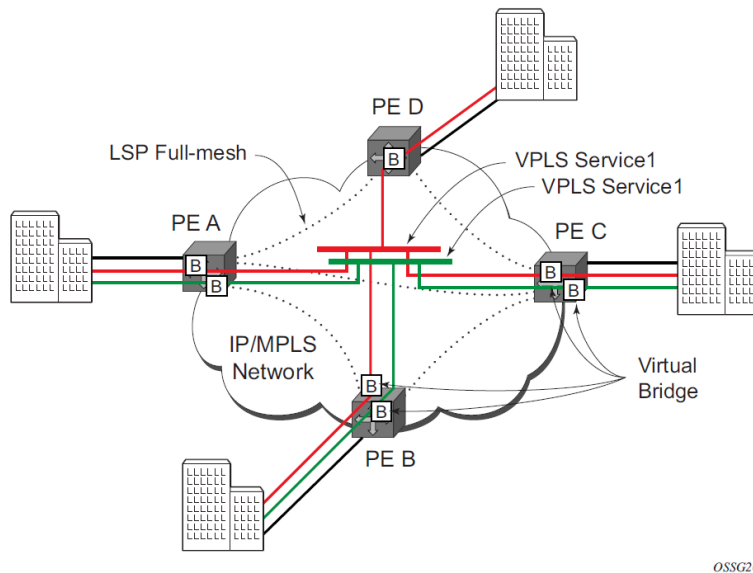
- VPLS is a transparent, protocol-independent service.
- There is no Layer 2 protocol conversion between LAN and WAN technologies.
- There is no need to design, manage, configure, and maintain separate WAN access equipment, which eliminates the need to train personnel on WAN technologies.

#### 3.1.1 VPLS packet walkthrough

This section provides an example of VPLS processing of a customer packet sent across the network, shown in [Figure 19: VPLS service architecture](#), from site-A, which is connected to PE-Router-A, to site-B, which is connected to PE-Router-C.



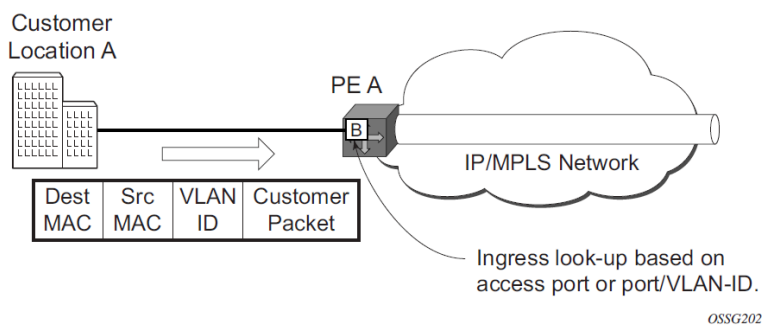
Figure 19: VPLS service architecture



1. PE-Router-A (shown in [Figure 20: Access port ingress packet format and lookup](#))

- a. Service packets arriving at PE-Router-A are associated with a VPLS service instance based on the combination of the physical port and the IEEE 802.1Q-tag (VLAN-ID) in the packet.

Figure 20: Access port ingress packet format and lookup



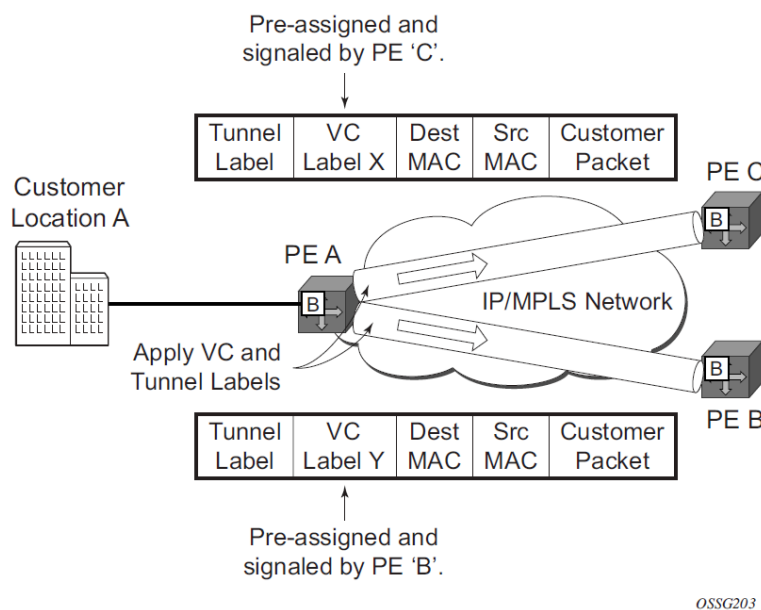
- b. PE-Router-A learns the source MAC address in the packet and creates an entry in the FDB table that associates the MAC address with the service access point (SAP) on which it was received.
- c. The destination MAC address in the packet is looked up in the FDB table for the VPLS instance. There are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

For a known MAC address, see [Figure 21: Network port egress packet format and flooding](#) and proceed to 1.d.

For an unknown MAC address, see [Figure 21: Network port egress packet format and flooding](#) and proceed to 1.f.

- d. If the destination MAC address has already been learned by PE-Router-A, an existing entry in the FDB table identifies the far-end PE-router and the service VC-label (inner label) to be used before sending the packet to far-end PE-Router-C.
- e. PE-Router-A chooses a transport LSP to send the customer packets to PE-Router-C. The customer packet is sent on this LSP after the IEEE 802.1Q-tag is updated as per the associated SAP qtag-manipulation configuration and service, and the service VC-label (inner label) and the transport label (outer label) are added to the packet.
- f. If the destination MAC address has not been learned, PE-Router-A floods the packet to both PE-Router-B and PE-Router-C, which are participating in the service, by using the VC-labels that each PE-Router previously signaled for the VPLS instance. The packet is not sent to PE-Router-D because this VPLS service does not exist on that PE-router.

Figure 21: Network port egress packet format and flooding



## 2. Core Router Switching

All the core routers (P routers in IETF nomenclature) between PE-Router-A and PE-Router-B and PE-Router-C are Label Switch Routers (LSRs) that switch the packet based on the transport (outer) label of the packet until the packet arrives at the far-end PE-Router. All core routers are unaware that this traffic is associated with a VPLS service.

## 3. PE-Router-C

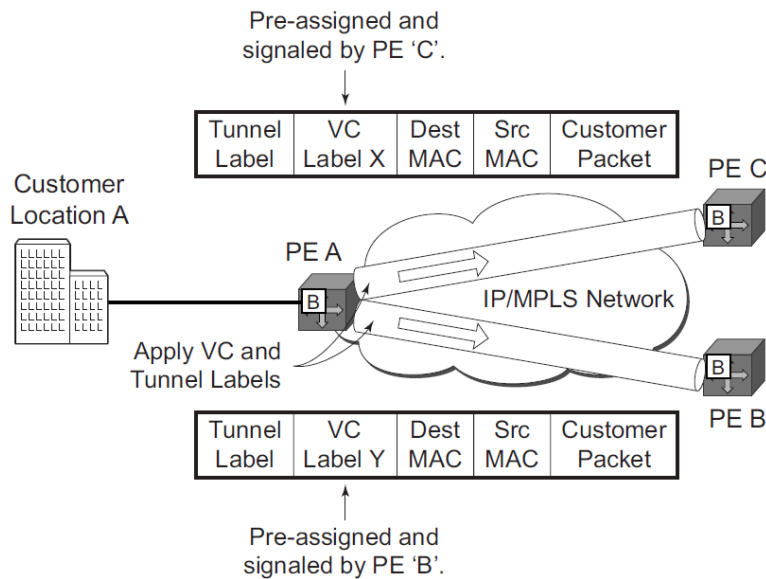
- a. PE-Router-C strips the transport label of the received packet to reveal the inner VC-label. The VC-label identifies the VPLS service instance to which the packet belongs.
- b. PE-Router-C learns the source MAC address in the packet and creates an entry in the FDB table that associates the MAC address with PE-Router-A and the VC-label that PE-Router-A signaled it for the VPLS service on which the packet was received.
- c. The destination MAC address in the packet is looked up in the FDB table for the VPLS instance. Again, there are two possibilities: either the destination MAC address has already been learned

(known MAC address) or the destination MAC address has not been learned on the access side of PE-Router-C (unknown MAC address).

- d. For a known MAC address ([Figure 22: Access port egress packet format and lookup](#)):

If the destination MAC address has been learned by PE-Router-C, an existing entry in the FDB table identifies the local access port and the IEEE 802.1Q-tag to be added before sending the packet to customer Location-C. The egress Q-tag may be different from the ingress Q-tag.

Figure 22: Access port egress packet format and lookup



OSSG204

## 3.2 VPLS features

This section provides information about VPLS features.

### 3.2.1 VPLS enhancements

The Nokia VPLS implementation includes several enhancements beyond basic VPN connectivity. The following VPLS features can be configured individually for each VPLS service instance:

- extensive MAC and IP filter support (up to Layer 4). Filters can be applied on a per-SAP basis
- FDB management features on a per service-level basis, including:
  - configurable FDB size limit
  - FDB size alarms
  - MAC learning disable
  - discard unknown
  - separate aging timers for locally and remotely learned MAC addresses

- ingress rate limiting for broadcast, multicast, and destination unknown flooding on a per SAP basis
- implementation of Spanning Tree Protocol (STP) parameters on a per-VPLS, per-SAP, and per spoke-SDP basis
- a split horizon group on a per-SAP and per spoke-SDP basis
- optional SAP or spoke-SDP redundancy to protect against node failure
- DHCP snooping and anti-spoofing on a per-SAP and per-SDP basis
- IGMP snooping on a per-SAP and per-SDP basis

### 3.2.2 VPLS over MPLS

The VPLS architecture proposed in RFC 4762, *Virtual Private LAN Services Using LDP Signaling* specifies the use of provider edge (PE) that is capable of learning, bridging, and replication on a per-VPLS basis. The PE routers that participate in the service are connected using MPLS Label Switched Path (LSP) tunnels in a full mesh composed of mesh SDPs or based on an LSP hierarchy (Hierarchical VPLS (H-VPLS)) composed of mesh SDPs and spoke-SDPs.

Multiple VPLS services can be offered over the same set of LSP tunnels. Signaling specified in RFC 4905 is used to negotiate a set of ingress and egress VC labels on a per-service basis. The VC labels are used by the PE routers for de-multiplexing traffic arriving from different VPLS services over the same set of LSP tunnels.

VPLS is provided over MPLS by:

- connecting bridging-capable provider edge routers with a full mesh of MPLS LSP tunnels
- negotiating per-service VC labels using draft-Martini encapsulation
- replicating unknown and broadcast traffic in a service domain
- enabling MAC learning over tunnel and access ports (see [VPLS MAC learning and packet forwarding](#))
- using a separate FDB per VPLS service

### 3.2.3 VPLS service pseudowire VLAN tag processing

VPLS services can be connected using pseudowires that can be provisioned statically or dynamically and are represented in the system as either a mesh or a spoke-SDP. The mesh and spoke-SDP can be configured to process zero, one, or two VLAN tags as traffic is transmitted and received. In the transmit direction, VLAN tags are added to the frame being sent, and in the received direction, VLAN tags are removed from the frame being received. This is analogous to the SAP operations on a null, dot1q, and QinQ SAP.

The system expects a symmetrical configuration with its peer; specifically, it expects to remove the same number of VLAN tags from received traffic as it adds to transmitted traffic. When removing VLAN tags from a mesh or spoke-SDP, the system attempts to remove the configured number of VLAN tags (see the following configuration information); if fewer tags are found, the system removes the VLAN tags found and forwards the resulting packet. As some of the related configuration parameters are local and not communicated in the signaling plane, an asymmetrical behavior cannot always be detected and so cannot be blocked. With an asymmetrical behavior, protocol extractions do not necessarily function as they would with a symmetrical configuration, resulting in an unexpected operation.

The VLAN tag processing is configured as follows on a mesh or spoke-SDP in a VPLS service:

- **zero VLAN tags processed**

VPLS Service Pseudowire VLAN Tag Processing. This requires the configuration of **vc-type ether** under the mesh-SDP or spoke-SDP, or in the related PW template.

- **one VLAN tag processed**

This requires one of the following configurations:

- **vc-type vlan** under the mesh-SDP or spoke-SDP, or in the related PW template
- **vc-type ether** and **force-vlan-vc-forwarding** under the mesh-SDP or spoke-SDP, or in the related PW template

- **two VLAN tags processed**

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

The PW template configuration provides support for BGP VPLS services and LDP VPLS services using BGP autodiscovery.

The following restrictions apply to VLAN tag processing:

- The configuration of **vc-type vlan** and **force-vlan-vc-forwarding** is mutually exclusive.
- BGP VPLS services operate in a mode equivalent to **vc-type ether**; consequently, the configuration of **vc-type vlan** in a PW template for a BGP VPLS service is ignored.
- **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** can be configured with the mesh-SDP or spoke-SDP signaled as either **vc-type ether** or **vc-type vlan**.
- The following are not supported with **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** configured under the mesh-SDP or spoke-SDP, or in the related PW template:
  - Routed, E-Tree, or PBB VPLS services (including B-VPLS and I-VPLS)
  - L2PT termination on QinQ mesh-SDP or spoke-SDPs
  - IGMP/MLD/PIM snooping within the VPLS service
  - force-vlan-vc-forwarding under the same spoke-SDP or PW template
  - Eth-CFM LM tests

Table 5: VPLS mesh and spoke-SDP VLAN tag processing: ingress and Table 6: VPLS mesh and spoke-SDP VLAN tag processing: egress describe the VLAN tag processing with respect to the zero, one, and two VLAN tag configuration described for the VLAN identifiers, Ethertype, ingress QoS classification (dot1p/DE), and QoS propagation to the egress (which can be used for egress classification or to set the QoS information, or both, in the innermost egress VLAN tag).

Table 5: VPLS mesh and spoke-SDP VLAN tag processing: ingress

| Ingress (received on mesh or spoke-SDP)             | Zero VLAN tags | One VLAN tag  | Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])             |
|---|----------------|---|---|
| VLAN identifiers                                    | —              | Ignored   | Both inner and outer ignored  |
| Ethertype (to determine the presence of a VLAN tag) | —              | 0x8100 or value configured under <b>sdp vlan-vc-etype</b> | Both inner and outer VLAN tags: 0x8100, or outer VLAN tag value configured under <b>sdp</b> |

| Ingress (received on mesh or spoke-SDP) | Zero VLAN tags | One VLAN tag                          | Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])   |
|---|----------------|---------------------------------------|---|
|   |                |                                       | <b>vlan-vc-etype</b> (inner VLAN tag value must be 0x8100)  |
| Ingress QoS (dot1p/DE) classification   | —              | Ignored                               | Both inner and outer ignored  |
| QoS (dot1p/DE) propagation to egress    | Dot1p/DE=0     | Dot1p/DE taken from received VLAN tag | Dot1p/DE taken as follows: <ul style="list-style-type: none"> <li>If the egress encapsulation is a Dot1q SAP, Dot1p/DE bits are taken from the outer received VLAN tag.</li> <li>If the egress encapsulation is QinQ SAP, the S-tag bits are taken from the outer received VLAN tag and the c-tag bits from the inner received VLAN tag.</li> </ul> |

Table 6: VPLS mesh and spoke-SDP VLAN tag processing: egress

| Egress (sent on mesh or spoke-SDP)  | Zero VLAN tags | One VLAN tag   | Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])  |
|-------------------------------------|----------------|--|--|
| VLAN identifiers (set in VLAN tags) | —              | <p>For one VLAN tag, one of the following applies:</p> <ul style="list-style-type: none"> <li>the <b>vlan-vc-tag</b> value configured in PW template or value under the mesh/spoke-SDP</li> <li>value from the inner tag received on a QinQ SAP or QinQ mesh/spoke-SDP</li> <li>value from the VLAN tag received on a dot1q SAP or mesh/spoke-SDP (with <b>vc-type vlan</b> or <b>force-vlan-vc-forwarding</b>)</li> <li>value from the outer tag received on a qtag.* SAP</li> <li>0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP</li> </ul> | <p>The inner and outer VLAN tags are derived from one of the following:</p> <ul style="list-style-type: none"> <li>vlan-vc-tag value configured in PW template or under the mesh/spoke-SDP: <ul style="list-style-type: none"> <li>If c-tag-c-tag is configured, both inner and outer tags are taken from the vlan-vc-tag value.</li> <li>If s-tag-c-tag is configured, only the s-tag value is taken from vlan-vc-tag.</li> </ul> </li> <li>value from the inner tag received on a QinQ SAP or QinQ mesh/spoke-SDP for the c-tag-c-tag option and value from outer/inner tag received on a QinQ SAP or QinQ mesh/spoke-SDP for</li> </ul> |

| Egress (sent on mesh or spoke-SDP)       | Zero VLAN tags | One VLAN tag  | Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])  |
|--|----------------|---|--|
|  |                |   | <p>the s-tag-c-tag configuration option</p> <ul style="list-style-type: none"> <li>value from the VLAN tag received on a dot1q SAP or mesh/spoke-SDP (with vc-type vlan or force-vlan-vc-forwarding) for the c-tag-c-tag option and value from the VLAN tag for the outer tag and zero for the inner tag</li> </ul>  |
|  |                |   | <ul style="list-style-type: none"> <li>value from the outer tag received on a qtag.* SAP for the c-tag-c-tag option and value from the VLAN tag for the outer tag and zero for the inner tag</li> <li>value 0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP Ethertype (set in VLAN tags)</li> </ul>  |
| Ethertype (set in VLAN tags)             | —              | 0x8100 or value configured under <b>sdp vlan-vc-etype</b>   | Both inner and outer VLAN tags: 0x8100, or outer VLAN tag value configured under <b>sdp vlan-vc-etype</b> (inner VLAN tag value is 0x8100)   |
| Egress QoS (dot1p/DE) (set in VLAN tags) | —              | <p>Taken from the innermost ingress service delimiting tag, one of the following applies:</p> <ul style="list-style-type: none"> <li>the inner tag received on a QinQ SAP or QinQ mesh/spoke-SDP</li> <li>value from the VLAN tag received on a dot1q SAP or mesh/spoke-SDP (with <b>vc-type</b> vlan or <b>force-vlan-vc-forwarding</b>)</li> <li>value from the outer tag received on a qtag.* SAP</li> </ul> | <p>Inner and outer dot1p/DE:</p> <p>If <b>c-tag-c-tag</b> is configured, the inner and outer dot1p/DE bits are both taken from the innermost ingress service delimiting tag. It can be one of the following:</p> <ul style="list-style-type: none"> <li>inner tag received on a QinQ SAP</li> <li>value from the VLAN tag received on a dot1q SAP or spoke-SDP (with <b>vc-type</b> vlan or <b>force-vlan-vc-forwarding</b>)</li> <li>value from the outer tag received on a qtag.* SAP</li> </ul> |

| Egress (sent on mesh or spoke-SDP)       | Zero VLAN tags | One VLAN tag  | Two VLAN Tags (enabled by force-qinq-vc-forwarding [c-tag-c-tag   s-tag-c-tag])  |
|--|----------------|---|--|
| Egress QoS (dot1p/DE) (set in VLAN tags) | —              | 0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP<br>Note that neither the inner nor outer dot1p/DE values can be explicitly set. | 0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP<br>If <b>s-tag-c-tag</b> is configured, the inner and outer dot1p/DE bits are taken from the inner and outer ingress service delimiting tag (respectively). They can be: <ul style="list-style-type: none"> <li>inner and outer tags received on a QinQ SAP or QinQ mesh/spoke-SDP</li> <li>value from the VLAN tag received on a dot1q SAP or mesh/spoke-SDP (with <b>vc-type vlan</b> or <b>force-vlan-vc-forwarding</b>) for the outer tag and zero for the inner tag</li> <li>value from the outer tag received on a qtag.* SAP for the outer tag and zero for the inner tag</li> <li>value 0 if there is no service delimiting VLAN tag at the ingress SAP or mesh/spoke-SDP</li> </ul> |

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

### 3.2.4 VPLS MAC learning and packet forwarding

The 7705 SAR Gen 2 performs the packet replication required for broadcast and multicast traffic across the bridged domain. MAC address learning is performed by the router to reduce the amount of unknown destination MAC address flooding.

The router learns the source MAC addresses of the traffic arriving on their access and network ports.

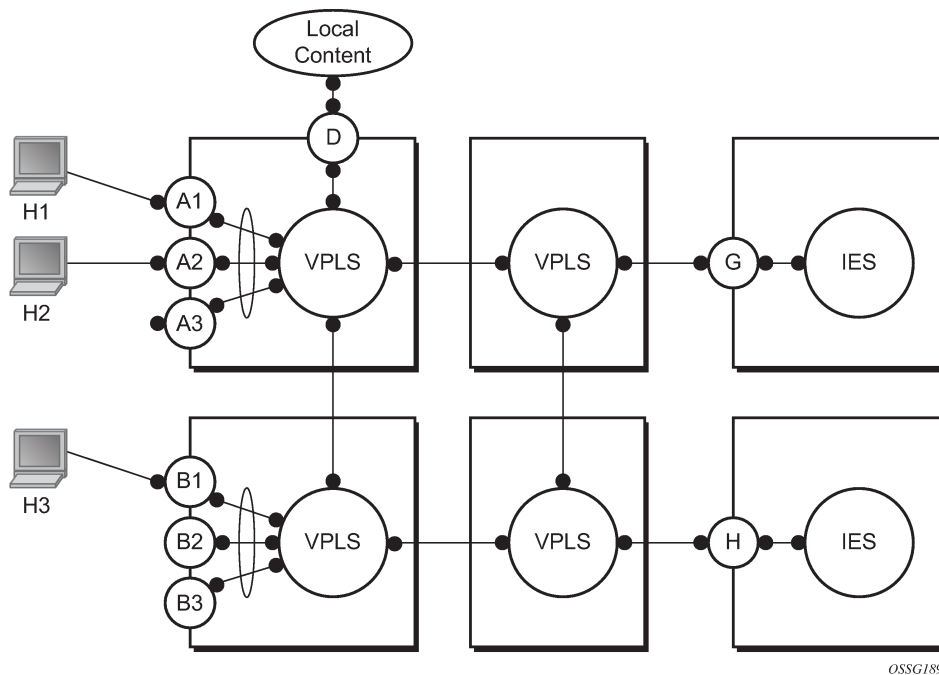
Each router maintains an FDB for each VPLS service instance and learned MAC addresses are populated in the FDB table of the service. All traffic is switched based on MAC addresses and forwarded between all objects in the VPLS service. Unknown destination packets (for example, the destination MAC address has not been learned) are forwarded on all objects to all participating nodes for that service until the target station responds and the MAC address is learned by the routers associated with that service.



### 3.2.4.1 MAC learning protection

In a Layer 2 environment, subscribers or customers connected to SAPs A or B can create a denial of service attack by sending packets sourcing the gateway MAC address. This moves the learned gateway MAC from the uplink SDP/SAP to the subscriber's or customer's SAP causing all communication to the gateway to be disrupted. If local content is attached to the same VPLS (D), a similar attack can be launched against it. Communication between subscribers or customers is also disallowed but split horizon is not sufficient in the topology shown in [Figure 23: MAC learning protection](#).

Figure 23: MAC learning protection



The 7705 SAR Gen 2 enables MAC learning protection capability for SAPs and SDPs. With this mechanism, forwarding and learning rules apply to the non-protected SAPs. Assume hosts H1, H2, and H3 ([Figure 23: MAC learning protection](#)) are non-protected while IES interfaces G and H are protected. When a frame arrives at a protected SAP/SDP, the MAC is learned as usual. When a frame arrives from a non-protected SAP or SDP, the frame must be dropped if the source MAC address is protected and the MAC address is not relearned. The system allows only packets with a protected MAC destination address.

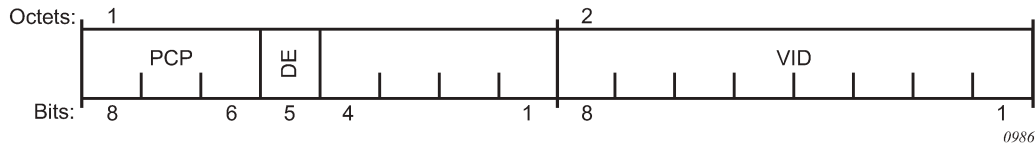
The system can be configured statically. The addresses of all protected MACs are configured. Only the IP address can be included and use a dynamic mechanism to resolve the MAC address (**cpe-ping**). All protected MACs in all VPLS instances in the network must be configured.

To eliminate the ability of a subscriber or customer to cause a DoS attack, the node restricts the learning of protected MAC addresses based on a statically defined list. Also, the destination MAC address is checked against the protected MAC list to verify that a packet entering a restricted SAP has a protected MAC as a destination.

### 3.2.4.2 DEI in IEEE 802.1ad

The IEEE 802.1ad-2005 standard allows drop eligibility to be conveyed separately from priority in Service VLAN TAGs (S-TAGs) so that all of the previously introduced traffic types can be marked as drop eligible. The S-TAG has a new format where the priority and discard eligibility parameters are conveyed in the 3-bit Priority Code Point (PCP) field and, respectively, in the DE bit (Figure 24: DE bit in the 802.1ad S-TAG).

Figure 24: DE bit in the 802.1ad S-TAG



The DE bit allows the S-TAG to convey eight forwarding classes/distinct emission priorities, each with a drop eligible indication.

When the DE bit is set to 0 (DE=FALSE), the related packet is not discarded eligible. This is the case for the packets that are within the CIR limits and must be prioritized in case of congestion. If the DEI is not used or backwards compliance is required, the DE bit should be set to zero on transmission and ignored on reception.

When the DE bit is set to 1 (DE=TRUE), the related packet is discarded eligible. This is the case for the packets that are sent above the CIR limit (but below the PIR). In case of congestion, these packets are the first ones to be dropped.

### 3.2.5 VPLS using G.8031 protected Ethernet tunnels

The use of MPLS tunnels provides a way to scale the core while offering fast failover times using MPLS FRR. In environments where Ethernet services are deployed using native Ethernet backbones, Ethernet tunnels are provided to achieve the same fast failover times as in the MPLS FRR case. There are still service provider environments where Ethernet services are deployed using native Ethernet backbones.

The Nokia VPLS implementation offers the capability to use core Ethernet tunnels compliant with ITU-T G.8031 specification to achieve 50 ms resiliency for backbone failures. This is required to comply with the stringent SLAs provided by service providers in the current competitive environment. The implementation also allows a LAG-emulating Ethernet tunnel providing a complimentary native Ethernet E-LAN capability. The LAG-emulating Ethernet tunnels and G.8031 protected Ethernet tunnels operate independently. For more information, see the *7705 SAR Gen 2 Services Overview Guide, "LAG Emulation using Ethernet Tunnels"*.

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

When configuring VPLS and B-VPLS using Ethernet tunnels, the services are very similar.

For examples, see the *IEEE 802.1ah PBB Guide*.

### 3.2.6 Pseudowire control word

The **control-word** command enables the use of the control word individually on each mesh SDP or spoke-SDP. By default, the control word is disabled. When the control word is enabled, all VPLS packets, including the BPDU frames, are encapsulated with the control word. The Targeted LDP (T-LDP) control plane behavior is the same as the control word for VLL services. The configuration for the two directions of the Ethernet pseudowire should match.

### 3.2.7 Layer 2 forwarding table management

The following sections describe VPLS features related to management of the FDB.

#### 3.2.7.1 Selective MAC address learning

Source MAC addresses are learned in a VPLS service by default with an entry allocated in the FDB for each address on all line cards. Therefore, all MAC addresses are considered to be global. This operation can be modified so that the line card allocation of some MAC addresses is selective, based on where the service has a configured object.

An example of the advantage of selective MAC address learning is for services to benefit from the higher MAC address scale of some line cards (particularly for network interfaces used by mesh or spoke-SDPs, EVPN-VXLAN tunnels, and EVPN-MPLS destinations) while using lower MAC address scale cards for the SAPs.

Selective MAC addresses are those learned locally and dynamically in the datapath (displayed in the **show** output with type "L") or by EVPN (displayed in the **show** output with type "Evpn", excluding those with the sticky bit set, which are displayed with type "EvpnS"). An exception is when a MAC address configured as a conditional static MAC address is learned dynamically on an object other than its monitored object; this can be displayed with type "L" or "Evpn" but is learned as global because of the conditional static MAC configuration.

Selective MAC addresses have FDB entries allocated on line cards where the service has a configured object. When a MAC address is learned, it is allocated an FDB entry on all line cards on which the service has a SAP configured (for LAG or Ethernet tunnel SAPs, the MAC address is allocated an FDB entry on all line cards on which that LAG or Ethernet tunnel has configured ports) and on all line cards that have a network interface port if the service is configured with VXLAN, EVPN-MPLS, or a mesh or spoke-SDP.

When using selective learning in an I-VPLS service, the learned C-MACs are allocated FDB entries on all the line cards where the I-VPLS service has a configured object and on the line cards on which the associated B-VPLS has a configured object. When using selective learning in a VPLS service with **allow-ip-intf-bind** configured (for it to become an R-VPLS), FDB entries are allocated on all line cards on which there is an IES or VPRN interface.

If a new configured object is added to a service and there are sufficient MAC FDB resources available on the new line cards, the selective MAC addresses present in the service are allocated on the new line cards. Otherwise, if any of the selective MAC addresses currently learned in the service cannot be allocated an FDB entry on the new line cards, those MAC addresses are deleted from all line cards. Such a deletion increments the FailedMacCmplxMapUpdts statistic displayed in the **tools dump service vpls-fdb-stats** output.

When the set of configured objects changes for a service using selective learning, the system must reallocate its FDB entries accordingly, which can cause FDB entry "allocate" or "free" operations to become

pending temporarily. The pending operations can be displayed using the **tools dump service id fdb** command.

When a global MAC address is to be learned, there must be a free FDB entry in the service and system FDBs and on all line cards in the system for it to be accepted. When a selective MAC address is to be learned, there must be a free FDB entry in the service and system FDBs and on all line cards where the service has a configured object for it to be accepted.

To demonstrate the selective MAC address learning logic, consider the following:

- a system has three line cards: 1, 2, and 3
- two VPLS services are configured on the system:
  - VPLS 1 having learned MAC addresses M1, M2, and M3 and has configured SAPs 1/1/1 and 2/1/1
  - VPLS 2 having learned MAC addresses M4, M5, and M6 and has configured SAPs 2/1/2 and 3/1/1

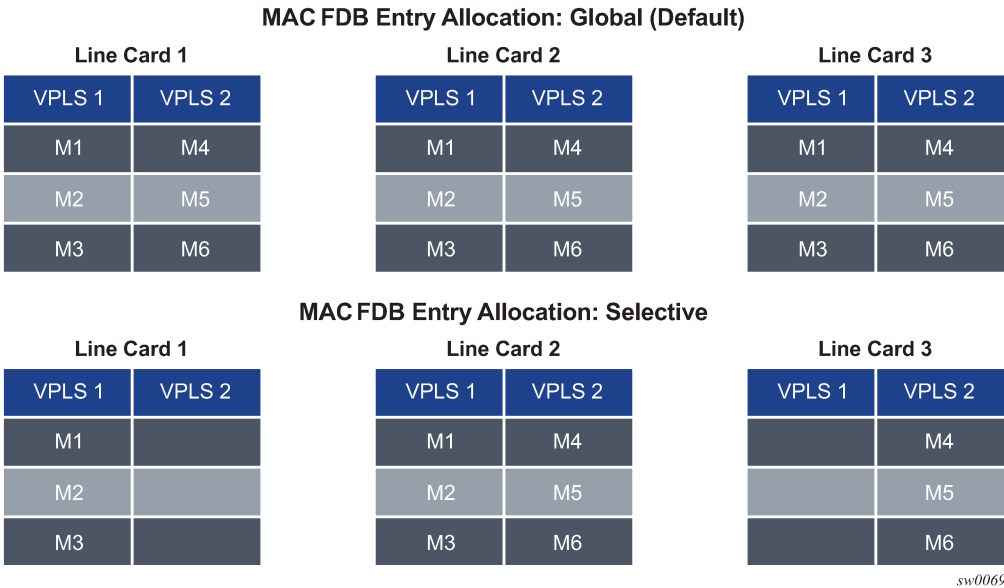
This is shown in [Table 7: MAC address learning logic example](#) .

Table 7: MAC address learning logic example

|       | Learned MAC addresses | Configured SAPs        |
|-------|-----------------------|------------------------|
| VPLS1 | M1, M2, M3            | SAP 1/1/1<br>SAP 2/1/1 |
| VPLS2 | M4, M5, M6            | SAP 2/1/2<br>SAP 3/1/1 |

[Figure 25: MAC FDB entry allocation: global versus selective](#) shows the FDB entry allocation when the MAC addresses are global and when they are selective. Notice that in the selective case, all MAC addresses are allocated FDB entries on line card 2, but line card 1 and 3 only have FDB entries allocated for services VPLS 1 and VPLS 2, respectively.

Figure 25: MAC FDB entry allocation: global versus selective



Selective MAC address learning can be enabled as follows within any VPLS service, except for B-VPLS and R-VPLS services:

```
configure
  service
    vpls <service-id> create
      [no] selective-learned-fdb
```

Enabling selective MAC address learning has no effect on single line card systems.

When selective learning is enabled or disabled in a VPLS service, the system may need to reallocate FDB entries; this can cause temporary pending FDB entry allocate or free operations. The pending operations can be displayed using the **tools dump service id fdb** command.

3.2.7.1.1 Example operational information

The **show** and **tools dump** command output can display the global and selective MAC addresses along with the MAC address limits and the number of allocated and free MAC-address FDB entries. The **show** output displays the system and card FDB usage, while the **tools** output displays the FDB per service with respect to MAC addresses and cards.

The configuration for the following output is similar to the simple example above:

- the system has three line cards: 1, 2, and 5
- the system has two VPLS services:
  - VPLS 1 is an EVPN-MPLS service with a SAP on 5/1/1:1 and uses a network interface on 5/1/5.
  - VPLS 2 has two SAPs on 2/1/1:2 and 2/1/2:2.

The first output shows the default where all MAC addresses are global. The second enables selective learning in the two VPLS services.

### 3.2.7.1.1.1 Global MAC address learning only (default)

By default, VPLS 1 and 2 are not configured for selective learning, so all MAC addresses are global:

```
*A:PE1# show service id [1,2] fdb | match expression ", Service|Sel Learned FDB"
Forwarding Database, Service 1
Sel Learned FDB : Disabled
Forwarding Database, Service 2
Sel Learned FDB : Disabled
*A:PE1#
```

Traffic is sent into the services, resulting in the following MAC addresses being learned:

```
*A:PE1# show service fdb-mac
=====
Service Forwarding Database
=====
ServId   MAC                Source-Identifier      Type      Last Change
-----
1         00:00:00:00:01:01  sap:5/1/1:1           L/0       01/31/17 08:44:37
1         00:00:00:00:01:02  sap:5/1/1:1           L/0       01/31/17 08:44:37
1         00:00:00:00:01:03  eMpls:                EvpnS     01/31/17 08:41:38
                        10.251.72.58:262142
                        P
1         00:00:00:00:01:04  eMpls:                EvpnS     01/31/17 08:41:38
                        10.251.72.58:262142
                        P
2         00:00:00:00:02:01  sap:2/1/2:2           L/0       01/31/17 08:44:37
2         00:00:00:00:02:02  sap:2/1/2:2           L/0       01/31/17 08:44:37
2         00:00:00:02:02:03  sap:2/1/1:2           L/0       01/31/17 08:44:37
2         00:00:00:02:02:04  sap:2/1/1:2           L/0       01/31/17 08:44:37
-----
No. of Entries: 8
-----
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static Lf=Leaf
=====
*A:PE1#
```

A total of eight MAC addresses are learned. There are two MAC addresses learned locally on SAP 5/1/1:1 in service VPLS 1 (type "L"), and another two MAC addresses learned using EVPN with the sticky bit set, also in service VPLS 1 (type "EvpnS"). A further two sets of two MAC addresses are learned on SAP 2/1/1:2 and 2/1/2:2 in service VPLS 2 (type "L").

The system and line card FDB usage is shown as follows:

```
*A:PE1# show service system fdb-usage
=====
FDB Usage
=====
System
-----
Limit:      511999
Allocated:  8
Free:       511991
Global:     8
-----
Line Cards
-----
Card        Selective    Allocated    Limit        Free
```

```

-----
1          0          8          511999          511991
2          0          8          511999          511991
5          0          8          511999          511991
-----
=====
*A:PE1#

```

The system MAC address limit is 511999, of which eight are allocated, and the rest are free. All eight MAC addresses are global and are allocated on cards 1, 2, and 5. There are no selective MAC addresses. This output can be reduced to specific line cards by specifying the card's slot ID as a parameter to the command.

To see the MAC address information per service, **tools dump** commands can be used, as follows for VPLS 1. The following output displays the card status:

```

*A:PE1# tools dump service id 1 fdb card-status
=====
VPLS FDB Card Status at 01/31/2017 08:44:38
=====
Card          Allocated          PendAlloc          PendFree
-----
1              4              0              0
2              4              0              0
5              4              0              0
=====
*A:PE1#

```

All of the line cards have four FDB entries allocated in VPLS 1. The "PendAlloc" and "PendFree" columns show the number of pending MAC address allocate and free operations, which are all zero.

The following output displays the MAC address status for VPLS 1:

```

*A:PE1# tools dump service id 1 fdb mac-status
=====
VPLS FDB MAC status at 01/31/2017 08:44:38
=====
MAC Address          Type          Status : Card list
-----
00:00:00:00:01:01    Global        Allocated : All
00:00:00:00:01:02    Global        Allocated : All
00:00:00:00:01:03    Global        Allocated : All
00:00:00:00:01:04    Global        Allocated : All
=====
*A:PE1#

```

The type and card list for each MAC address in VPLS 1 is displayed. VPLS 1 has learned four MAC addresses: the two local MAC addresses on SAP 5/1/1:1 and the two EvpnS MAC addresses. Each MAC address has an FDB entry allocated on all line cards. This output can be further reduced by optionally including a specified MAC address, a specific card, and the operational pending state.

### 3.2.7.1.1.2 Selective and global MAC address learning

Selective MAC address learning is now enabled in VPLS 1 and VPLS 2, as follows:

```

*A:PE1# show service id [1,2] fdb | match expression ", Service|Sel Learned FDB"
Forwarding Database, Service 1
Sel Learned FDB : Enabled

```

```
Forwarding Database, Service 2
Sel Learned FDB : Enabled
*A:PE1#
```

The MAC addresses learned are the same, with the same traffic being sent; however, there are now selective MAC addresses that are allocated FDB entries on different line cards.

The system and line card FDB usage is as follows:

```
*A:PE1# show service system fdb-usage
=====
FDB Usage
=====
System
-----
Limit:      511999
Allocated:  8
Free:       511991
Global:     2
-----
Line Cards
-----
Card        Selective    Allocated    Limit        Free
-----
1           0             2            511999       511997
2           4             6            511999       511993
5           2             4            511999       511995
-----
=====
*A:PE1#
```

The system MAC address limit and allocated numbers have not changed but now there are only two global MAC addresses; these are the two EvpnS MAC addresses.

There are two FDB entries allocated on card 1, which are the global MAC addresses; there are no services or network interfaces configured on card 1, so the FDB entries allocated are for the global MAC addresses.

Card 2 has six FDB entries allocated in total: two for the global MAC addresses plus four for the selective MAC addresses in VPLS 2 (these are the two sets of two local MAC addresses in VPLS 2 on SAP 2/1/1:2 and 2/1/2:2).

Card 5 has four FDB entries allocated in total: two for the global MAC addresses plus two for the selective MAC addresses in VPLS 1 (these are the two local MAC addresses in VPLS 1 on SAP 5/1/1:1).

This output can be reduced to specific line cards by specifying the card's slot ID as a parameter to the command.

To see the MAC address information per service, **tools dump** commands can be used for VPLS 1.

The following output displays the card status:

```
*A:PE1# tools dump service id 1 fdb card-status
=====
VPLS FDB Card Status at 01/31/2017 08:44:39
=====
Card        Allocated    PendAlloc    PendFree
-----
1           2            0            0
2           2            0            0
5           4            0            0
=====
*A:PE1#
```



There are two FDB entries allocated on line card 1, two on line card 2, and four on line card 5. The "PendAlloc" and "PendFree" columns are all zeros.

The following output displays the MAC address status for VPLS 1:

```
*A:PE1# tools dump service id 1 fdb mac-status
=====
VPLS FDB MAC status at 01/31/2017 08:44:39
=====
MAC Address      Type              Status : Card list
-----
00:00:00:00:01:01  Select           Allocated : 5
00:00:00:00:01:02  Select           Allocated : 5
00:00:00:00:01:03  Global           Allocated : All
00:00:00:00:01:04  Global           Allocated : All
=====
*A:PE1#
```

The type and card list for each MAC address in VPLS 1 is displayed. VPLS 1 has learned four MAC addresses: the two local MAC addresses on SAP 5/1/1:1 and the two EvpnS MAC addresses. The local MAC addresses are selective and have FDB entries allocated only on card 5. The global MAC addresses are allocated on all line cards. This output can be further reduced by optionally including a specified MAC address, a specific card, and the operational pending state.

### 3.2.7.2 System FDB size

The system FDB table size is configurable as follows:

```
configure
  service
    system
      fdb-table-size table-size
```

where table-size can have values in the range from 255999 to 2047999 (2000k).

The default, minimum, and maximum values for the table size are dependent on the chassis type. To support more than 500k MAC addresses, the CPMs provisioned in the system must have at least 16 GB memory. The maximum system FDB table size also limits the maximum FDB table size of any card within the system.

The actual achievable maximum number of MAC addresses depends on the MAC address scale supported by the active cards and whether selective learning is enabled.

If an attempt is made to configure the system FDB table size such that:

- the new size is greater than or equal to the current number of allocated FDB entries, the command succeeds and the new system FDB table size is used
- the new size is less than the number of allocated FDB entries, the command fails with an error message. In this case, the user is expected to reduce the current FDB usage (for example, by deleting statically configured MAC addresses, shutting down EVPN, clearing learned MACs, and so on) to lower the number of allocated MAC addresses in the FDB so that it does not exceed the system FDB table size being configured.

The logic when attempting a rollback is similar; however, when rolling back to a configuration where the system FDB table size is smaller than the current system FDB table size, the system flushes all learned MAC addresses (by performing a **shutdown** then **no shutdown** in all VPLS services) to allow the rollback to continue.

The system FDB table size can be larger than some of the line card FDB sizes, resulting in the possibility that the current number of allocated global MAC addresses is larger than the maximum FDB size supported on some line cards. When a new line card is provisioned, the system checks whether the line card's FDB can accommodate all of the currently allocated global MAC addresses. If it can, then the provisioning succeeds; if it cannot, then the provisioning fails and an error is reported. If the provisioning fails, the number of global MACs allocated must be reduced in the system to a number that the new line card can accommodate, then the **card-type** must be reprovisioned.

### 3.2.7.3 Per-VPLS service FDB size

The following MAC table management features are available for each instance of a SAP or spoke-SDP within a particular VPLS service instance.

MAC FDB size limits allow users to specify the maximum number of MAC FDB entries that are learned locally for a SAP or remotely for a spoke-SDP. If the configured limit is reached, no new addresses is learned from the SAP or spoke-SDP until at least one FDB entry is aged out or cleared.

- When the limit is reached on a SAP or spoke-SDP, packets with unknown source MAC addresses are still forwarded (this default behavior can be changed by configuration). By default, if the destination MAC address is known, it is forwarded based on the FDB, and if the destination MAC address is unknown, it is flooded. Alternatively, if discard unknown is enabled at the VPLS service level, any packets from unknown source MAC addresses are discarded at the SAP.
- The log event SAP MAC Limit Reached is generated when the limit is reached. When the condition is cleared, the log event SAP MAC Limit Reached Condition Cleared is generated.
- Disable learning allows users to disable the dynamic learning function on a SAP or a spoke-SDP of a VPLS service instance.
- Disable aging allows users to turn off aging for learned MAC addresses on a SAP or a spoke-SDP of a VPLS service instance.

### 3.2.7.4 System FDB size alarms

High and low watermark alarms give warning when the system MAC FDB usage is high. An alarm is generated when the number of FDB entries allocated in the system FDB reaches 95% of the total system FDB table size and is cleared when it reduces to 90% of the system FDB table size. These percentages are not configurable.

### 3.2.7.5 Line card FDB size alarms

High and low watermark alarms give warning when a line card's MAC FDB usage is high. An alarm is generated when the number of FDB entries allocated in a line card FDB reaches 95% of its maximum FDB table size and is cleared when it reduces to 90% of its maximum FDB table size. These percentages are not configurable.

### 3.2.7.6 Per VPLS FDB size alarms

The size of the VPLS FDB can be configured with a low watermark and a high watermark, expressed as a percentage of the total FDB size limit. If the actual FDB size grows above the configured high watermark

percentage, an alarm is generated. If the FDB size falls below the configured low watermark percentage, the alarm is cleared by the system.

### 3.2.7.7 Local and remote aging timers

Like a Layer 2 switch, learned MACs within a VPLS instance can be aged out if no packets are sourced from the MAC address for a specified period of time (the aging time). In each VPLS service instance, there are independent aging timers for locally learned MAC and remotely learned MAC entries in the FDB. A local MAC address is a MAC address associated with a SAP because it ingresses on a SAP. A remote MAC address is a MAC address received by an SDP from another router for the VPLS instance.

The local-age timer for the VPLS instance specifies the aging time for locally learned MAC addresses, and the remote-age timer specifies the aging time for remotely learned MAC addresses.

In general, the remote-age timer is set to a longer period than the local-age timer to reduce the amount of flooding required for unknown destination MAC addresses. The aging mechanism is considered a low priority process. In most situations, the aging out of MAC addresses happens within tens of seconds beyond the age time. However, it, can take up to two times their respective age timer to be aged out.

### 3.2.7.8 Disable MAC aging

The MAC aging timers can be disabled, which prevents any learned MAC entries from being aged out of the FDB. When aging is disabled, it is still possible to manually delete or flush learned MAC entries.

Aging can be disabled for learned MAC addresses on a SAP or a spoke-SDP of a VPLS service instance.

### 3.2.7.9 Disable MAC learning

When MAC learning is disabled for a service, new source MAC addresses are not entered in the VPLS FDB, whether the MAC address is local or remote. MAC learning can be disabled for individual SAPs or spoke-SDPs.

### 3.2.7.10 Unknown MAC discard

Unknown MAC discard is a feature that discards all packets that ingress the service where the destination MAC address is not in the FDB. The normal behavior is to flood these packets to all endpoints in the service.

Unknown MAC discard can be used with the disable MAC learning and disable MAC aging options to create a fixed set of MAC addresses allowed to ingress and traverse the service.

### 3.2.7.11 VPLS and rate limiting

Traffic that is normally flooded throughout the VPLS can be rate limited on SAP ingress through the use of service ingress QoS policies. In a service ingress QoS policy, individual queues can be defined per forwarding class to provide shaping of broadcast traffic, MAC multicast traffic, and unknown destination MAC traffic.

### 3.2.7.12 MAC move

The MAC move feature is useful to protect against undetected loops in a VPLS topology as well as the presence of duplicate MACs in a VPLS service.

If two clients in the VPLS have the same MAC address, the VPLS experiences a high relearn rate for the MAC. When MAC move is enabled, the 7705 SAR Gen 2 shuts down the SAP or spoke-SDP and creates an alarm event when the threshold is exceeded.

MAC move allows sequential order port blocking. By configuration, some VPLS ports can be configured as "non-blockable", which allows a simple level of control of which ports are being blocked during loop occurrence. There are two sophisticated control mechanisms that allow blocking of ports in a sequential order:

1. Configuration capabilities to group VPLS ports and to define the order in which they should be blocked
2. Criteria defining when individual groups should be blocked

For the first control mechanism, configuration CLI is extended by definition of "primary" and "secondary" ports. Per default, all VPLS ports are considered "tertiary" ports unless they are explicitly declared primary or secondary. The order of blocking always follows a strict order starting from tertiary to secondary, and then primary.

The definition of criteria for the second control mechanism is the number of periods during which the specified relearn rate has been exceeded. The mechanism is based on the cumulative factor for every group of ports. Tertiary VPLS ports are blocked if the relearn rate exceeds the configured threshold during one period, while secondary ports are blocked only when relearn rates are exceeded during two consecutive periods, and primary ports when exceeded during three consecutive periods. The retry timeout period must be larger than the period before blocking the highest priority port so that the retry timeout sufficiently spans across the period required to block all ports in sequence. The period before blocking the highest priority port is the cumulative factor of the highest configured port multiplied by 5 seconds (the retry timeout can be configured through the CLI).

### 3.2.7.13 Auto-learn MAC protect

This section provides information about the auto-learn MAC protect and **restrict-protected-src discard-frame** features.

VPLS solutions usually involve learning MAC addresses so that traffic can be forwarded to the correct SAP or SDP. If a MAC address is learned on the wrong SAP or SDP, traffic is redirected away from its intended destination. This could occur as a result of a misconfiguration, a problem in the network, or by a malicious source creating a DoS attack, and is applicable to any type of VPLS network; for example, mobile backhaul or residential service delivery networks. The auto-learn MAC protect feature safeguards against the possibility of MAC addresses being learned on the wrong SAP or SDP.

This feature automatically protects source MAC addresses that have been learned on a SAP or SDP (spoke or mesh) and prevents frames with the same protected source MAC address from entering a different SAP or SDP (spoke or mesh).

This solution has low operational complexity and is complementary to features such as MAC move and MAC pinning, but has the advantage that MAC moves are not seen. If a MAC is initially learned on the wrong SAP or SDP, the operator can clear the MAC from the MAC FDB so it can be relearned on the correct SAP or SDP.

Two separate commands provide the configuration flexibility of separating the identification (learning) function from the application of the restriction (discard).

The **auto-learn-mac-protect** and **restrict-protected-src** commands allow the following functions:

- the ability to enable the automatic protection of a learned MAC using the **auto-learn-mac-protect** command under a SAP, or spoke or mesh SDP, or SHG context
- the ability to discard frames associated with automatically protected MACs instead of shutting down the entire SAP or SDP, as with the **restrict-protected-src** feature. This is enabled using a **restrict-protected-src discard-frame** command in the SAP, or spoke or mesh SDP, or split horizon group (SHG) context. An optimized alarm mechanism is used to generate alarms related to these discards. The frequency of alarm generation is fixed to be, at most, one alarm per MAC address per forwarding complex per 10 minutes in a VPLS service.

If the **auto-learn-mac-protect** or **restrict-protected-src discard-frame** feature is configured under an SHG, the operation applies only to SAPs in the SHG, not to spoke-SDPs in the SHG. If required, these parameters can also be enabled explicitly under specific SAPs or spoke-SDPs within the SHG.

Applying or removing **auto-learn-mac-protect** or **restrict-protected-src discard-frame** to or from a SAP, spoke or mesh SDP, or SHG, clears the MACs on the related objects. For the SHG, this results in clearing the MACs only on the SAPs within the SHG.

The use of **restrict-protected-src discard-frame** and both the **restrict-protected-src [alarm-only]** command and with the configuration of manually protected MAC addresses, using the **mac-protect** command, within a specified VPLS are mutually exclusive.

The following rules apply to the state changes of protected MACs:

- Automatically learned protected MACs are subject to normal removal, aging (unless disabled), and flushing, at which time the associated entries are removed from the FDB.
- Automatically learned protected MACs can move from their learned SAP or spoke or mesh SDP only if they enter a SAP or spoke or mesh SDP without the **restrict-protected-src** command enabled.

If a MAC address legitimately moves between SAPs or spoke or mesh SDPs after it is automatically protected on a specified SAP or spoke or mesh SDP (thereby causing discards when received on the new SAP or spoke or mesh SDP), the operator must manually clear the MAC from the FDB for it to be learned in the new location.

MAC addresses that are manually created (using **static-mac**, **static-host** with a MAC address specified, or **oam mac-populate**) are not protected even if they are configured on a SAP or spoke or mesh SDP that has the **auto-learn-mac-protect** command enabled on it. Also, the MAC address associated with an R-VPLS IP interface is protected within its VPLS service such that frames received with this MAC address as the source address are discarded (this is not based on the auto-learn MAC protect function). However, VRRP MAC addresses associated with an R-VPLS IP interface are not protected, either in this way or using the auto-learn MAC protect function.

MAC addresses that are dynamically created (learned, using **static-host** with no MAC address specified, or **lease-populate**) are protected when the MAC address is learned on a SAP or spoke or mesh SDP that has the **auto-learn-mac-protect** command enabled on it.

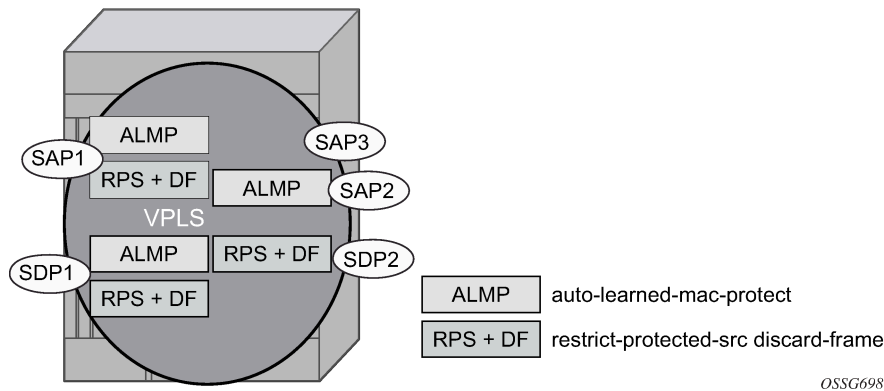
The actions of the following features are performed in the order listed.

1. Restrict protected SRC
2. MAC pinning
3. MAC move

### 3.2.7.13.1 Operation

The following figure shows a specific configuration using **auto-learn-mac-protect** and **restrict-protected-src discard-frame** to describe their operation.

Figure 26: Auto-learn-mac-protect operation



A VPLS service is configured with SAP1 and SDP1 connecting to access devices and SAP2, SAP3, and SDP2 connecting to the core of the network. The **auto-learn-mac-protect** feature is enabled on SAP1, SAP3, and SDP1, and **restrict-protected-src discard-frame** is enabled on SAP1, SDP1, and SDP2. The following series of events describes the details of the functionality.

Assume that the FDB is empty at the start of each sequence.

#### Sequence 1

1. A frame with source MAC A enters SAP1. MAC A is learned on SAP1, and MAC-A/SAP1 is protected because of the presence of **auto-learn-mac-protect** on SAP1.
2. All subsequent frames with source MAC A entering SAP1 are forwarded into the VPLS.
3. Frames with source MAC A enter either SDP1 or SDP2. These frames are discarded, and an alarm indicating MAC A and SDP1/SDP2 is initiated because of the presence of **restrict-protected-src discard-frame** on SDP1/SDP2.
4. The preceding continues, with MAC-A/SAP1 protected in the FDB until MAC A on SAP1 is removed from the FDB.

#### Sequence 2

1. A frame with source MAC A enters SAP1. MAC A is learned on SAP1, and MAC-A/SAP1 is protected because of the presence of **auto-learn-mac-protect** on SAP1.
2. A frame with source MAC A enters SAP2. Because **restrict-protected-src** is not enabled on SAP2, MAC A is relearned on SAP2 (but not protected), replacing the MAC-A/SAP1 entry in the FDB.
3. All subsequent frames with source MAC A entering SAP2 are forwarded into the VPLS. This is because **restrict-protected-src** is not enabled on SAP2 and **auto-learn-mac-protect** is not enabled on SAP2, so the FDB is not changed.
4. A frame with source MAC A enters SAP1, MAC A is relearned on SAP1, and MAC-A/SAP1 is protected because of the presence of **auto-learn-mac-protect** on SAP1.

**Sequence 3**

1. A frame with source MAC A enters SDP2. MAC A is learned on SDP2, but is not protected because **auto-learn-mac-protect** is not enabled on SDP2.
2. A frame with source MAC A enters SDP1. and MAC A is relearned on SDP1 because previously it was not protected. Consequently, MAC-A/SDP1 is protected because of the presence of **auto-learn-mac-protect** on SDP1.

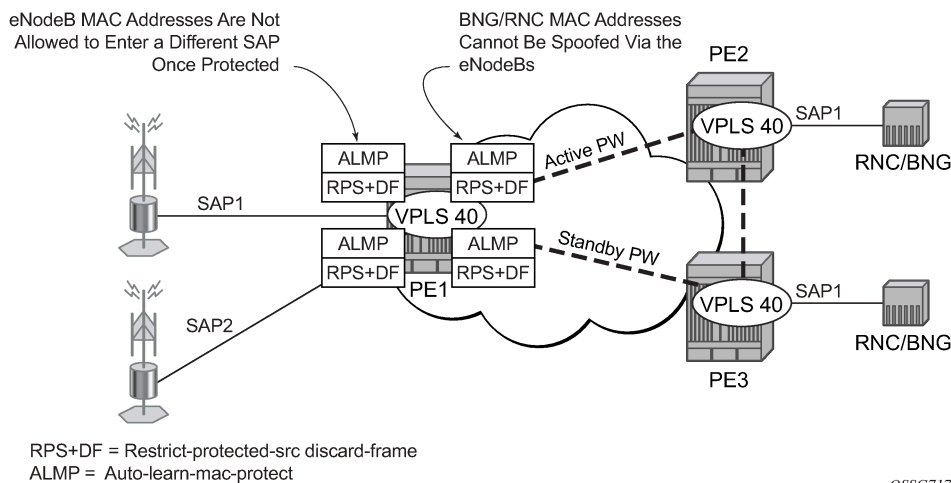
**Sequence 4**

1. A frame with source MAC A enters SAP1, MAC A is learned on SAP1, and MAC-A/SAP1 is protected because of the presence of **auto-learn-mac-protect** on SAP1.
2. A frame with source MAC A enters SAP3. Because **restrict-protected-src** is not enabled on SAP3, MAC A is relearned on SAP3 and the MAC-A/SAP1 entry is removed from the FDB. MAC-A/SAP3 is added as protected to the FDB (because **auto-learn-mac-protect** is enabled on SAP3).
3. All subsequent frames with source MAC A entering SAP3 are forwarded into the VPLS.
4. A frame with source MAC A enters SAP1. These frames are discarded, and an alarm indicating MAC A and SAP1 is initiated because of the presence of the **restrict-protected-src discard-frame** on SAP1.

**Example: Use in a mobile backhaul network**

The following figure shows a possible configuration using **auto-learn-mac-protect** and **restrict-protected-src discard-frame** in a mobile backhaul network, with the focus on PE1.

*Figure 27: Auto-learn-mac-protect example*



To protect the MAC addresses of the BNG/RNCs on PE1, the **auto-learn-mac-protect** command is enabled on the pseudowires connecting PE1 to PE2 and PE3. Enabling the **restrict-protected-src discard-frame** command on the SAPs toward the eNodeBs prevents frames with the source MAC addresses of the BNG/RNCs from entering PE1 from the eNodeBs.

The MAC addresses of the eNodeBs are protected in two ways. In addition to the preceding commands, enabling the **auto-learn-mac-protect** command on the SAPs toward the eNodeBs prevents the MAC addresses of the eNodeBs being learned on the wrong eNodeB SAP. Enabling the **restrict-protected-src discard-frame** command on the pseudowires connecting PE1 to PE2 and PE3 protects the eNodeB MAC addresses from being learned on the pseudowires. This may occur if the MAC addresses are incorrectly injected into VPLS 40 on PE2/PE3 from another eNodeB aggregation PE.



The preceding configuration is equally applicable to other Layer 2 VPLS-based aggregation networks; for example, to business or residential service networks.

### 3.2.8 Split horizon SAP groups and split horizon spoke SDP groups

Within the context of VPLS services, a loop-free topology within a fully meshed VPLS core is achieved by applying a split horizon forwarding concept that packets received from a mesh SDP are never forwarded to other mesh SDPs within the same service. The advantage of this approach is that no protocol is required to detect loops within the VPLS core network.

In applications such as DSL aggregation, it is useful to extend this split horizon concept also to groups of SAPs and spoke-SDPs. This extension is referred to as a split horizon SAP group or residential bridging. Traffic arriving on an SAP or a spoke-SDP, or both, within a split horizon group is not copied to other SAPs and spoke-SDPs in the same split horizon group (but is copied to SAPs/spoke-SDPs in other split horizon groups if these exist within the same VPLS).

### 3.2.9 VPLS and STP

The Nokia VPLS service provides a bridged or switched Ethernet Layer 2 network. Equipment connected to SAPs forward Ethernet packets into the VPLS service. The 7705 SAR Gen 2 participating in the service learns where the customer MAC addresses reside, on ingress SAPs or ingress SDPs.

Unknown destinations, broadcasts, and multicasts are flooded to all other SAPs in the service. If SAPs are connected together, either through misconfiguration or for redundancy purposes, loops can form and flooded packets can keep flowing through the network. The Nokia implementation of the STP is designed to remove these loops from the VPLS topology. This is done by putting one or several SAPs or spoke-SDPs, or both, in the discarding state.

The Nokia implementation of the STP incorporates some modifications to make the operational characteristics of VPLS more effective.

The STP instance parameters allow a balance between resiliency and speed of convergence extremes. Modifying particular parameters can affect the behavior. For information about command usage, descriptions, and CLI syntax, see [Configuring a VPLS service using CLI](#).

#### 3.2.9.1 Spanning Tree operating modes

Per VPLS instance, a preferred STP variant can be configured. The STP variants supported are:

- **rstp**  
Rapid Spanning Tree Protocol (RSTP) compliant with IEEE 802.1D-2004 - default mode
- **dot1w**  
Compliant with IEEE 802.1w
- **comp-dot1w**  
Operation as in RSTP but backwards compatible with IEEE 802.1w (this mode allows interoperability with some MTU types)
- **mstp**



Compliant with the Multiple Spanning Tree Protocol specified in IEEE 802.1Q-REV/D5.0-09/2005. This mode of operation is only supported in a Management VPLS (MVPLS).

While the 7705 SAR Gen 2 initially uses the mode configured for the VPLS, it dynamically falls back (on a per-SAP basis) to STP (IEEE 802.1D-1998) based on the detection of a BPDU of a different format. A trap or log entry is generated for every change in spanning tree variant.

Some older 802.1w compliant RSTP implementations may have problems with some of the features added in the 802.1D-2004 standard. Interworking with these older systems is improved with the comp-dot1w mode. The differences between the RSTP mode and the comp-dot1w mode are as follows.

- The RSTP mode implements the improved convergence over shared media feature; for example, RSTP transitions from discarding to forwarding in 4 seconds when operating over shared media. The comp-dot1w mode does not implement this 802.1D-2004 improvement and transitions conform to 802.1w in 30 seconds (both modes implement fast convergence over point-to-point links).
- In the RSTP mode, the transmitted BPDUs contain the port's designated priority vector (DPV) (conforms to 802.1D-2004). Older implementations may be confused by the DPV in a BPDU and may fail to recognize an agreement BPDU correctly. This would result in a slow transition to a forwarding state (30 seconds). For this reason, in the comp-dot1w mode, these BPDUs contain the port's port priority vector (conforms to 802.1w).

The 7705 SAR Gen 2 supports two BPDU encapsulation formats, and can dynamically switch between the following supported formats (on a per-SAP basis):

- IEEE 802.1D STP
- Cisco PVST

### 3.2.9.2 Multiple Spanning Tree Protocol

Multiple Spanning Tree Protocol (MSTP) extends the concept of the IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) by allowing grouping and associating VLANs to Multiple Spanning Tree Instances (MSTI). Each MSTI can have its own topology, which provides architecture enabling load balancing by providing multiple forwarding paths. At the same time, the number of STP instances running in the network is significantly reduced as compared to Per VLAN STP (PVST) mode of operation. Network fault tolerance is also improved because a failure in one instance (forwarding path) does not affect other instances.

The Nokia implementation of M-VPLS is used to group different VPLS instances under one RSTP instance. Introducing MSTP into the M-VPLS allows the following:

- interoperation with traditional Layer 2 switches in an access network
- provides an effective solution for dual-homing of many business Layer 2 VPNs into a provider network

#### 3.2.9.2.1 Redundancy access to VPLS

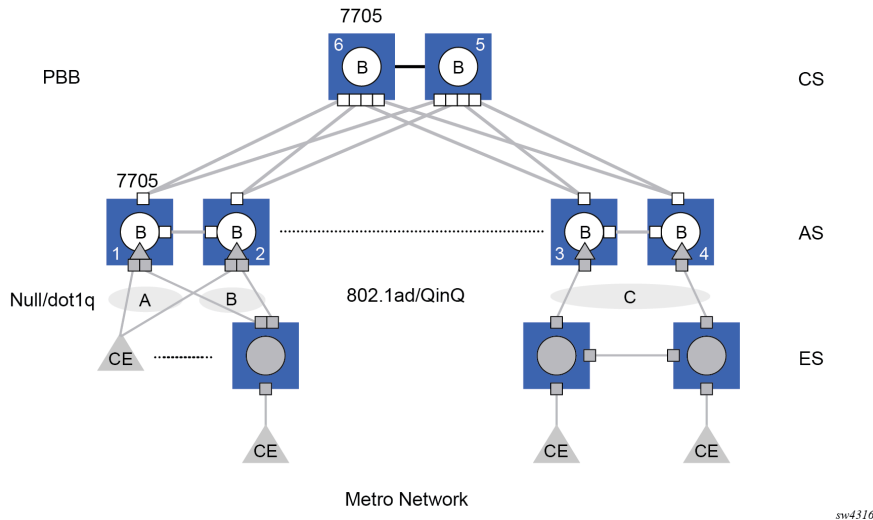
The GigE MAN portion of the network is implemented with traditional switches. Using MSTP running on individual switches facilitates redundancy in this part of the network. To provide dual homing of all VPLS services accessing from this part of the network, the VPLS PEs must participate in MSTP.

This can be achieved by configuring M-VPLS on VPLS-PEs (only PEs directly connected to the GigE MAN network), then assigning different managed-VLAN ranges to different MSTP instances. Typically, the M-VPLS would have SAPs with null encapsulations (to receive, send, and transmit MSTP BPDUs) and a mesh SDP to interconnect a pair of VPLS PEs.

Different access scenarios are displayed in [Figure 28: Access resiliency](#) as an example of network diagrams dually connected to the PBB PE:

|                      |   |
|----------------------|---|
| <b>Access Type A</b> | Source devices connected by null or dot1q SAPs        |
| <b>Access Type B</b> | One QinQ switch connected by QinQ/801ad SAPs          |
| <b>Access Type C</b> | Two or more ES devices connected by QinQ/802.1ad SAPs |

Figure 28: Access resiliency



The following mechanisms are supported for the I-VPLS:

- STP/RSTP can be used for all access types.
- M-VPLS with MSTP can be used as-is, for access type A; MSTP is required for access type B and C.
- LAG and MC-LAG can be used for access type A and B.
- Split-horizon-group does not require residential.

PBB I-VPLS inherits current STP configurations from the regular VPLS and M-VPLS.

### 3.2.9.3 MSTP for QinQ SAPs

MSTP runs in a M-VPLS context and can control SAPs from source VPLS instances. QinQ SAPs are supported. The outer tag is considered by MSTP as part of VLAN range control.

### 3.2.9.4 Provider MSTP

Provider MSTP is specified in IEEE-802.1ad-2005. It uses a provider bridge group address instead of a regular bridge group address used by STP, RSTP, and MSTP BPDUs. This allows for implicit separation of source and provider control planes.

The 802.1ad access network sends PBB PE P-MSTP BPDUs using the specified MAC address and also works over QinQ interfaces. P-MSTP mode is used in PBBN for core resiliency and loop avoidance.

Similar to regular MSTP, the STP mode (for example, PMSTP) is only supported in VPLS services where the m-VPLS flag is configured.

#### 3.2.9.4.1 MSTP general principles

MSTP represents a modification of RSTP that allows the grouping of different VLANs into multiple MSTIs. To enable different devices to participate in MSTIs, they must be consistently configured. A collection of interconnected devices that have the same MST configuration (region-name, revision, and VLAN-to-instance assignment) comprises an MST region.

There is no limit to the number of regions in the network, but every region can support a maximum of 16 MSTIs. Instance 0 is a special instance for a region, known as the Internal Spanning Tree (IST) instance. All other instances are numbered from 1 to 4094. IST is the only spanning-tree instance that sends and receives BPDUs (typically, BPDUs are untagged). All other spanning-tree instance information is included in MSTP records (M-records), which are encapsulated within MSTP BPDUs. This means that a single BPDU carries information for multiple MSTIs, which reduces overhead of the protocol.

Any MSTI is local to an MSTP region and completely independent from an MSTI in other MST regions. Two redundantly connected MST regions use only a single path for all traffic flows (no load balancing between MST regions or between MST and SST region).

Traditional Layer 2 switches running MSTP protocol assign all VLANs to the IST instance per default. The operator may then "re-assign" individual VLANs to a specified MSTI by configuring per VLAN assignment. This means that a PE can be considered as a part of the same MST region only if the VLAN assignment to IST and MSTIs is identical to the one of Layer 2 switches in the access network.

#### 3.2.9.4.2 MSTP in the 7705 SAR Gen 2

7705 SAR Gen 2 platforms use a concept of M-VPLS to group different SAPs under a single STP instance. The VLAN range covering SAPs to be managed by a specified M-VPLS is declared under a specific M-VPLS SAP definition. MSTP mode-of-operation is only supported in an M-VPLS.

When running MSTP, by default, all VLANs are mapped to the CIST. At the VPLS level, VLANs can be assigned to specific MSTIs. When running RSTP, the operator must explicitly indicate, per SAP, which VLANs are managed by that SAP.

#### 3.2.9.5 Enhancements to the STP

To interconnect PE devices across the backbone, service tunnels (SDPs) are used. These service tunnels are shared among multiple VPLS instances. The Nokia implementation of the STP incorporates some enhancements to make the operational characteristics of VPLS more effective. The implementation of STP on the router is modified to guarantee that service tunnels are not blocked in any circumstance without imposing artificial restrictions on the placement of the root bridge within the network. The modifications introduced are fully compliant with the 802.1D-2004 STP specification.

When running MSTP, spoke-SDPs cannot be configured. Also, ensure that all bridges connected by mesh SDPs are in the same region. If not, the mesh is prevented from becoming active (trap is generated).

To achieve this, all mesh SDPs are dynamically configured as either root ports or designated ports. The PE devices participating in each VPLS mesh determine (using the root path cost learned as part of the normal protocol exchange) which of the devices is closest to the root of the network. This PE device is internally

designated as the primary bridge for the VPLS mesh. As a result of this, all network ports on the primary bridges are assigned the designated port role and therefore remain in the forwarding state.

The second part of the solution ensures that the remaining PE devices participating in the STP instance see the SDP ports as a lower-cost path to the root instead of a path that is external to the mesh. Internal to the PE nodes participating in the mesh, the SDPs are treated as zero cost paths toward the primary bridge. As a consequence, the paths through the mesh are seen as lower cost than any alternative and the PE node designates the network port as the root port. This ensures that network ports always remain in forwarding state.

A combination of the preceding features ensure that network ports are never blocked and maintain interoperability with bridges external to the mesh that are running STP instances.

### 3.2.9.5.1 BPDU translation

VPLS networks are typically used to interconnect different customer sites using different access technologies such as Ethernet and bridged-encapsulated ATM PVCs. Typically, different Layer 2 devices can support different types of STP, even if they are from the same vendor. In some cases, it is necessary to provide BPDU translation to provide an interoperable e2e solution.

To address these network designs, BPDU format translation is supported on the 7705 SAR Gen 2. If enabled on a specified SAP or spoke-SDP, the system intercepts all BPDUs destined for that interface and perform required format translation such as STP-to-PVST or the other way around.

Similarly, BPDU interception and redirection to the CPM is performed only at ingress, meaning that as soon as at least one port within a specified VPLS service has BPDU translation enabled, all BPDUs received on any of the VPLS ports are redirected to the CPM.

BPDU translation requires all encapsulation actions that the datapath would perform for a specified outgoing port (such as adding VLAN tags depending on the outer SAP and the SDP encapsulation type) and adding or removing all the required VLAN information in a BPDU payload.

This feature can be enabled on a SAP only if STP is disabled in the context of the specified VPLS service.

## 3.2.10 VPLS redundancy

The VPLS standard (RFC 4762, *Virtual Private LAN Services Using LDP Signaling*) includes provisions for H-VPLS, using point-to-point spoke-SDPs. Two applications have been identified for spoke-SDPs:

- connect Multi-Tenant Units (MTUs) to PEs in a metro area network
- interconnect VPLS nodes of two networks

In both applications, the spoke-SDPs serve to improve the scalability of VPLS. While node redundancy is implicit in non-hierarchical VPLS services (using a full mesh of SDPs between PEs), node redundancy for spoke-SDPs needs to be provided separately.

Nokia routers have implemented special features for improving the resilience of hierarchical VPLS instances, in both MTU and inter-metro applications.

### 3.2.10.1 Spoke SDP redundancy for metro interconnection

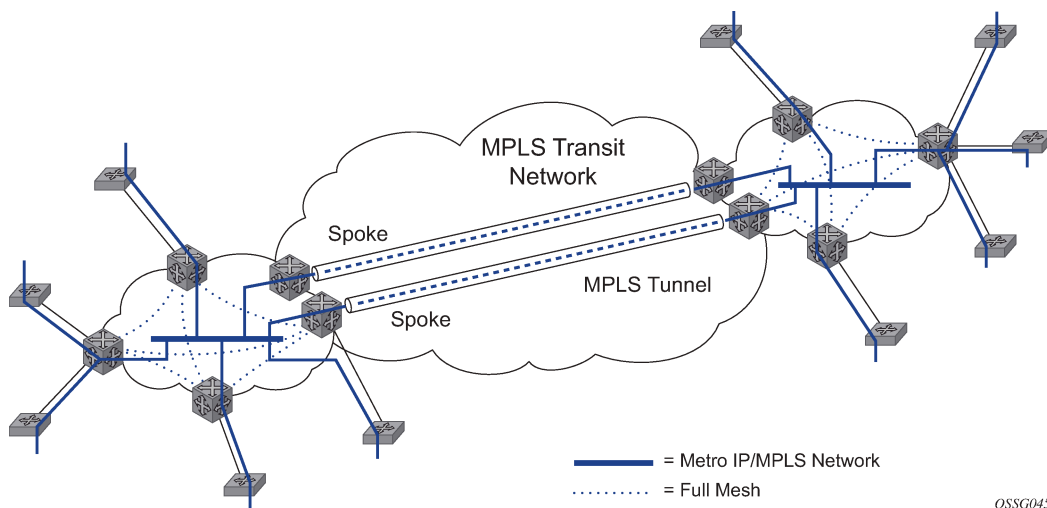
When two or more meshed VPLS instances are interconnected by redundant spoke-SDPs (as shown in [Figure 29: H-VPLS with spoke redundancy](#)), a loop in the topology results. To remove such a loop from the

topology, STP can be run over the SDPs (links) that form the loop, such that one of the SDPs is blocked. As running STP in each and every VPLS in this topology is not efficient, the node includes functionality that can associate a number of VPLSs with a single STP instance running over the redundant-SDPs. Node redundancy is therefore achieved by running STP in one VPLS, and applying the conclusions of this STP to the other VPLS services. The VPLS instance running STP is referred to as the “management VPLS” or M-VPLS.

In the case of a failure of the active node, STP on the management VPLS in the standby node changes the link states from disabled to active. The standby node then broadcasts a MAC flush LDP control message in each of the protected VPLS instances, so that the address of the newly active node can be re-learned by all PEs in the VPLS.

It is possible to configure two management VPLS services, where both VPLS services have different active spokes (this is achieved by changing the path cost in STP). By associating different user VPLSs with the two management VPLS services, load balancing across the spokes can be achieved.

Figure 29: H-VPLS with spoke redundancy



### 3.2.10.2 Spoke SDP-based redundant access

This feature provides the ability to have a node deployed as MTUs (Multi-tenant Unit Switches) to be multi-homed for VPLS to multiple routers deployed as PEs without requiring the use of M-VPLS.

In the configuration example shown in [Figure 29: H-VPLS with spoke redundancy](#), the MTUs have spoke-SDPs to two PE devices. One is designated as the primary and one as the secondary spoke-SDP. This is based on a precedence value associated with each spoke.

The secondary spoke is in a blocking state (both on receive and transmit) as long as the primary spoke is available. When the primary spoke becomes unavailable (because of the link failure, PEs failure, and so on), the MTUs immediately switch traffic to the backup spoke and start receiving traffic from the standby spoke. Optional revertive operation (with configurable switch-back delay) is supported. Forced manual switchover is also supported.

To speed up the convergence time during a switchover, MAC flush is configured. The MTUs generates a MAC flush message over the newly unblocked spoke when a spoke change occurs. As a result, the PEs receiving the MAC flush flushes all MACs associated with the impacted VPLS service instance and forward the MAC flush to the other PEs in the VPLS network if **propagate-mac-flush** is enabled.

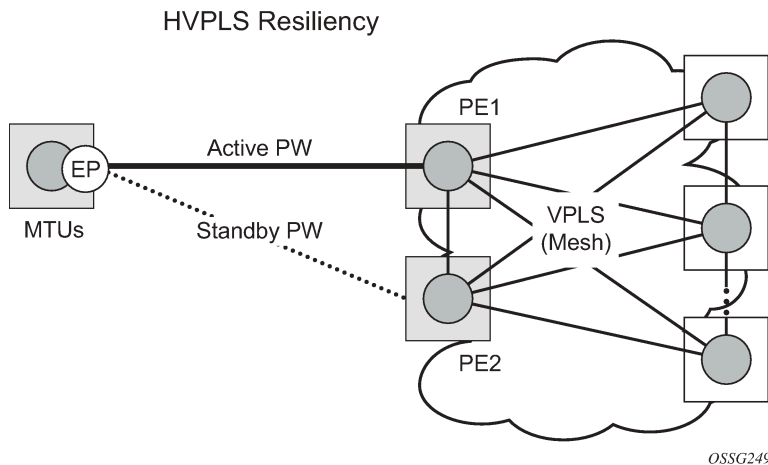
### 3.2.10.3 Inter-domain VPLS resiliency using multichassis endpoints

Inter-domain VPLS refers to a VPLS deployment where sites may be located in different domains. An example of inter-domain deployment can be where different metro domains are interconnected over a Wide Area Network (Metro1-WAN-Metro2) or where sites are located in different autonomous systems (AS1-ASBRs-AS2).

Multichassis endpoint (MC-EP) provides an alternate solution that does not require RSTP at the gateway VPLS PEs while still using pseudowires to interconnect the VPLS instances located in the two domains. It is supported in both VPLS and PBB-VPLS on the B-VPLS side.

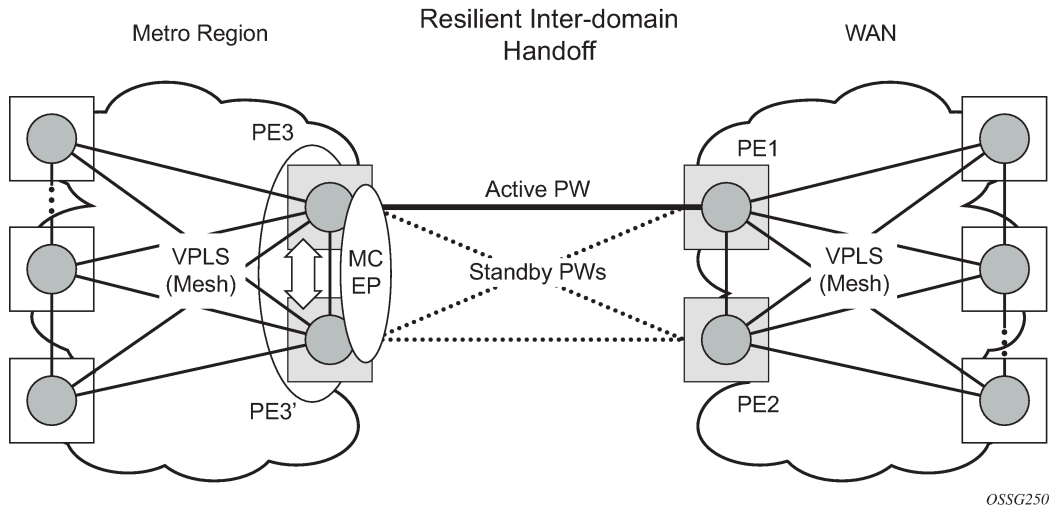
MC-EP expands the single chassis endpoint based on active/standby pseudowires for VPLS, shown in [Figure 30: HVPLS resiliency based on AS pseudowires](#).

*Figure 30: HVPLS resiliency based on AS pseudowires*



The active/standby pseudowire solution is appropriate for the scenario when only one VPLS PE (MTUs) needs to be dual-homed to two core PEs (PE1 and PE2). When multiple VPLS domains need to be interconnected, the above solution provides a single point of failure at the MTU-s. The example shown in [Figure 31: Multichassis pseudowire endpoint for VPLS](#) can be used.

Figure 31: Multichassis pseudowire endpoint for VPLS



The two gateway pairs, PE3-PE3' and PE1-PE2, are interconnected using a full mesh of four pseudowires out of which only one pseudowire is active at any time.

The concept of pseudowire endpoint for VPLS provides multi-chassis resiliency controlled by the MC-EP pair, PE3-PE3' in this example. This scenario, referred to as multi-chassis pseudowire endpoint for VPLS, provides a way to group pseudowires distributed between PE3 and PE3 chassis in a virtual endpoint that can be mapped to a VPLS instance.

The MC-EP inter-chassis protocol is used to ensure configuration and status synchronization of the pseudowires that belong to the same MC-EP group on PE3 and PE3'. Based on the information received from the peer shelf and the local configuration, the master shelf decides on which pseudowire becomes active.

The MC-EP solution is built around the following components:

- Multichassis protocol used to perform the following functions:
  - Selection of master chassis.
  - Synchronization of the pseudowire configuration and status.
  - Fast detection of peer failure or communication loss between MC-EP peers using either centralized BFD, if configured, or its own keep-alive mechanism.
- T-LDP signaling of pseudowire status informs the remote PEs about the choices made by the MC-EP pair.
- Pseudowire data plane is represented by the four pseudowires inter-connecting the gateway PEs.
  - Only one of the pseudowires is activated based on the primary/secondary, preference configuration, and pseudowire status. In case of a tie, the pseudowire located on the master chassis is chosen.
  - The rest of the pseudowires are blocked locally on the MC-EP pair and on the remote PEs as long as they implement the pseudowire active/standby status.



### 3.2.10.3.1 Fast detection of peer failure using BFD

Although the MC-EP protocol has its own keep-alive mechanisms, sharing a common mechanism for failure detection with other protocols (for example, BGP, RSVP-TE) scales better. MC-EP can be configured to use the centralized BFD mechanism.

Similar to other protocols, MC-EP registers with BFD if the **bfd-enable** command is active under the **config>redundancy>multi-chassis>peer>mc-ep** context. As soon as the MC-EP application is activated using no shutdown, it tries to open a new BFD session or register automatically with an existing one. The source-ip configuration under redundancy multi-chassis peer-ip is used to determine the local interface while the peer-ip is used as the destination IP for the BFD session. After MC-EP registers with an active BFD session, it uses it for fast detection of MC-EP peer failure. If BFD registration or BFD initialization fails, the MC-EP keeps using its own keep-alive mechanism and it sends a trap to the NMS signaling the failure to register with/open a BFD session.

To minimize operational mistakes and wrong peer interpretation for the loss of BFD session, the following additional rules are enforced when the MC-EP is registering with a BFD session:

- Only the centralized BFD sessions using system or loopback IP interfaces (source-ip parameter) are accepted in order for MC-EP to minimize the false indication of peer loss.
- If the BFD session associated with MC-EP protocol is using a system/loopback interface, the following actions are not allowed under the interface: IP address change, "shutdown", "no bfd" commands. If one of these actions is required under the interface, the operator needs to disable BFD using one of the following procedures:
  - The **no bfd-enable** command in the **config>redundancy>multi-chassis>peer>mc-ep** context.



**Note:** This is the recommended procedure.

- The **shutdown** command in the **config>redundancy>multi-chassis>peer>mc-ep** or from under **config>redundancy>multi-chassis>peer** contexts.

MC-EP keep-alives are still exchanged for the following reasons:

- As a backup; if the BFD session does not come up or is disabled, the MC-EP protocol uses its own keep-alives for failure detection.
- To ensure the database is cleared if the remote MC-EP peer is shut down or misconfigured (each x seconds; one second suggested as default).

If MC-EP de-registers with BFD using the **no bfd-enable** command, the following processing steps occur:



**Note:** There should be no pseudowire status change during this process.

1. The local peer indicates to the MC-EP peer that the local BFD is being disabled using the MC-EP peer-config-TLV fields ([BFD local: BFD remote]). This is done to avoid the wrong interpretation of the BFD session loss.
2. The remote peer acknowledges reception indicating through the same peer-config-TLV fields that it is de-registering with the BFD session.
3. Both MC-EP peers de-register and use only keep-alives for failure detection.

Traps are sent when the status of the monitoring of the MC-EP session through BFD changes in the following instances:



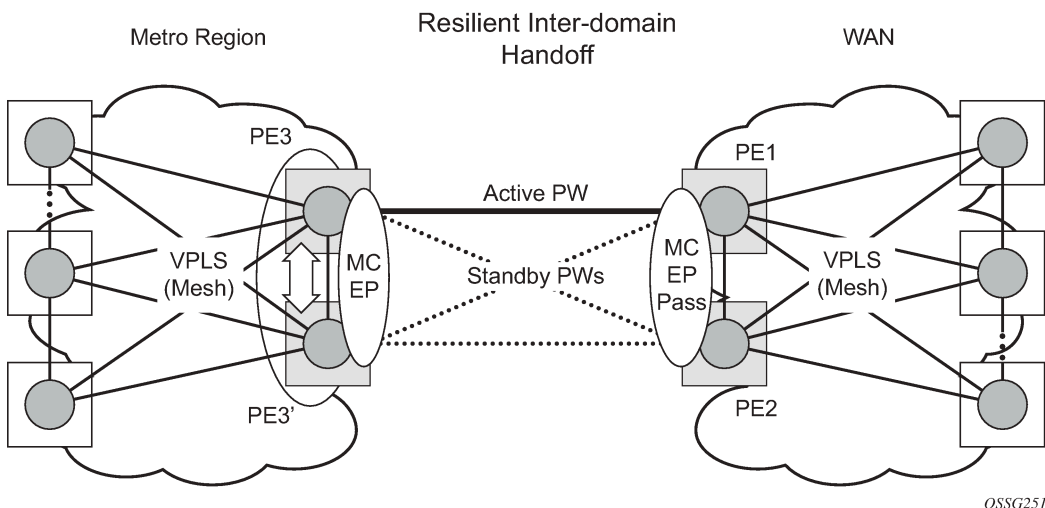
- When red/mc/peer is no shutdown and BFD is not enabled, a notification is sent indicating BFD is not monitoring the MC-EP peering session.
- When BFD changes to open, a notification is sent indicating BFD is monitoring the MC-EP peering session.
- When BFD changes to down/close, a notification is sent indicating BFD is not monitoring the MC-EP peering session.

### 3.2.10.3.2 MC-EP passive mode

The MC-EP mechanisms are built to minimize the possibility of loops. It is possible that human error could create loops through the VPLS service. One way to prevent loops is to enable the MAC move feature in the gateway PEs (PE3, PE3', PE1, and PE2).

An MC-EP passive mode can also be used on the second PE pair, PE1 and PE2, as a second layer of protection to prevent any loops from occurring if the operator introduces operational errors on the MC-EP PE3, PE3' pair. An example is shown in [Figure 32: MC-EP in passive mode](#).

Figure 32: MC-EP in passive mode



OSSG251

When in passive mode, the MC-EP peers stay dormant as long as one active pseudowire is signaled from the remote end. If more than one pseudowire belonging to the passive MC-EP becomes active, the PE1 and PE2 pair applies the MC-EP selection algorithm to select the best choice and blocks all others. No signaling is sent to the remote pair to avoid flip-flop behavior. A trap is generated each time MC-EP in passive mode activates. Every occurrence of this kind of trap should be analyzed by the operator as it is an indication of possible misconfiguration on the remote (active) MC-EP peering.

For the MC-EP passive mode to work, the pseudowire status signaling for active/standby pseudowires should be enabled. This requires the following CLI configurations:

For the remote MC-EP PE3, PE3' pair:

```
config>service>vpls>endpoint no suppress-standby-signaling
```

When MC-EP passive mode is enabled on the PE1 and PE2 pair, the following command is always enabled internally, regardless of the actual configuration:

```
config>service>vpls>endpoint no ignore-standby-signaling
```

### 3.2.10.4 Support for single chassis endpoint mechanisms

In cases of SC-EP, there is a consistency check to ensure that the configuration of the member pseudowires is the same. For example, mac-pining, mac-limit, and ignore standby signaling must be the same. In the MC-EP case, there is no consistency check between the member endpoints located on different chassis. The user must carefully verify the configuration of the two endpoints to ensure consistency.

The following rules apply for suppress-standby-signaling and ignore-standby parameters:

- Regular MC-EP mode (non-passive) follows the suppress-standby-signaling and ignore-standby settings from the related endpoint configuration.
- For MC-EP configured in passive mode, the following settings are used, regardless of previous configuration: **suppress-standby-sig** and **no ignore-standby-sig**. It is expected that when passive mode is used at one side, the regular MC-EP side activates signaling with **no suppress-stdby-sig**.
- When passive mode is configured in just one of the nodes in the MC-EP peering, the other node is forced to change to passive mode. A trap is sent to the operator to signal the wrong configuration.

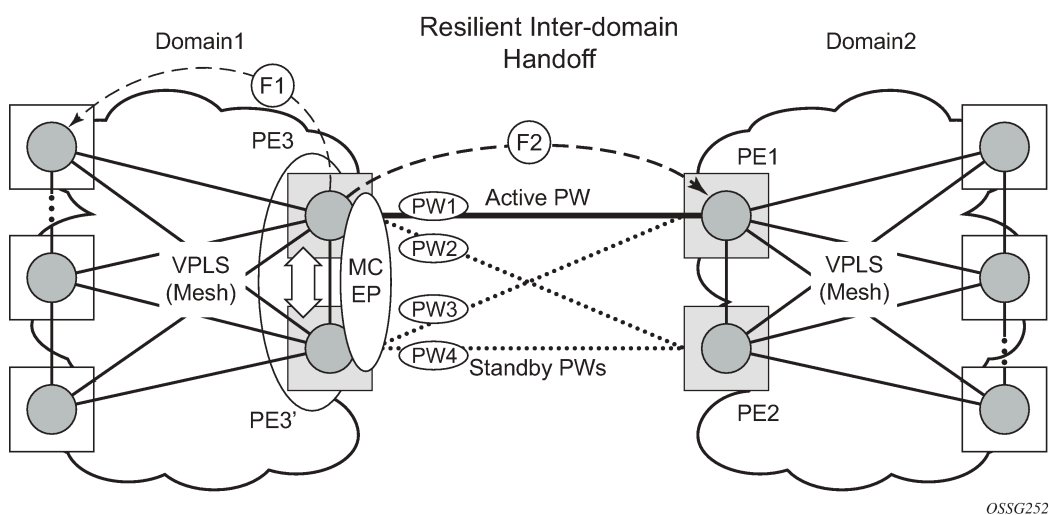
This section also describes how the main mechanisms used for single chassis endpoint are adapted for the MC-EP solution.

#### 3.2.10.4.1 MAC flush support in MC-EP

In an MC-EP scenario, failure of a pseudowire or gateway PE determines activation of one of the next best pseudowires in the MC-EP group. This section describes the MAC flush procedures that can be applied to ensure blackhole avoidance.

**Figure 33: MAC flush in the MC-EP solution** shows a pair of PE gateways (PE3 and PE3') running MC-EP toward PE1 and PE2 where F1 and F2 are used to indicate the possible direction of the MAC flush, signaled using T-LDP MAC withdraw message. PE1 and PE2 can only use regular VPLS pseudowires and do not have to use an MC-EP or a regular pseudowire endpoint.

*Figure 33: MAC flush in the MC-EP solution*



Regular MAC flush behavior applies for the LDP MAC withdraw sent over the T-LDP sessions associated with the active pseudowire in the MC-EP; for example, PE3 to PE1. That includes any Topology Change Notification (TCN) events or failures associated with SAPs or pseudowires not associated with the MC-EP.

The following MAC flush behaviors apply to changes in the MC-EP pseudowire selection:

- If the local PW2 becomes active on PE3:
  - On PE3, the MACs mapped to PW1 are moved to PW2.
  - A T-LDP flush-all-but-mine message is sent toward PE2 in the F2 direction and is propagated by PE2 in the local VPLS mesh.
  - No MAC flush is sent in the F1 direction from PE3.
- If one of the pseudowires on the pair PE3 becomes active; for example, PW4:
  - On PE3, the MACs mapped to PW1 are flushed, the same as for a regular endpoint.
  - PE3 must be configured with **send-flush-on-failure** to send a T-LDP flush-all-from-me message toward VPLS mesh in the F1 direction.
  - PE3 sends a T-LDP flush-all-but-mine message toward PE2 in the F2 direction, which is propagated by PE2 in the local VPLS mesh. When MC-EP is in passive mode and the first spoke becomes active, a no MAC flush-all-but-mine message is generated.

### 3.2.10.4.2 Block-on-mesh-failure support in MC-EP scenario

The following rules describe how the block-on-mesh-failure operates with the MC-EP solution (see [Figure 33: MAC flush in the MC-EP solution](#)):

- If PE3 does not have any forwarding path toward Domain1 mesh, it should block both PW1 and PW2 and inform PE3 so one of its pseudowires can be activated.
- To allow the use of block-on-mesh-failure for MC-EP, a block-on-mesh-failure parameter can be specified in the **config>service>vpls>endpoint** context with the following rules:
  - The default is **no block-on-mesh-failure** to allow for easy migration.
  - For a spoke-SDP to be added under an endpoint, the setting for its **block-on-mesh-failure** parameter must be in synchronization with the endpoint parameter.
  - After the spoke-SDP is added to an endpoint, the configuration of its **block-on-mesh-failure** parameter is disabled. A change in endpoint configuration for the **block-on-mesh-failure** parameter is propagated to the individual spoke-SDP configuration.
  - When a spoke-SDP is removed from the endpoint group, it inherits the last configuration from the endpoint parameter.
  - Adding an MC-EP under the related endpoint configuration does not affect the above behavior.

Before Release 7.0, the **block-on-mesh-failure** command could not be enabled under **config>service>vpls>endpoint** context. For a spoke-SDP to be added to an (single-chassis) endpoint, its **block-on-mesh-failure** had to be disabled (**config>service>vpls>spoke-sdp>no block-on-mesh-failure**). Then, the configuration of **block-on-mesh-failure** under a spoke-SDP is blocked.

- If **block-on-mesh-failure** is enabled on PE1 and PE2, these PEs signal pseudowire standby status toward the MC-EP PE pair. PE3 and PE3 should consider the pseudowire status signaling from remote PE1 and PE2 when making the selection of the active pseudowire.

### 3.2.10.4.3 Support for force spoke SDP in MC-EP

In a regular (single chassis) endpoint scenario, the following command can be used to force a specific SDP binding (pseudowire) to become active:

**tools perform service id *service-id* endpoint force**

In the MC-EP case, this command has a similar effect when there is a single forced SDP binding in an MC-EP. The forced SDP binding (pseudowire) is selected as active.

However, when the command is run at the same time as both MC-EP PEs, when the endpoints belong to the same MC-EP, the regular MC-EP selection algorithm (for example, the operational status ⇒ precedence value) is applied to determine the winner.

### 3.2.10.4.4 Revertive behavior for primary pseudowires in an MC-EP

For a single-chassis endpoint, a revert-time command is provided under the VPLS endpoint.

In a regular endpoint, the revert-time setting affects just the pseudowire defined as primary (precedence 0). For a failure of the primary pseudowire followed by restoration, the revert-timer is started. After it expires, the primary pseudowire takes the active role in the endpoint. This behavior does not apply for the case when both pseudowires are defined as secondary; that is, if the active secondary pseudowire fails and is restored, it stays in standby until a configuration change or a force command occurs.

In the MC-EP case, the revertive behavior is supported for pseudowire defined as primary (precedence 0). The following rules apply:

- The revert-time setting under each individual endpoint control the behavior of the local primary pseudowire if one is configured under the local endpoint.
- The secondary pseudowires behave as in the regular endpoint case.

## 3.2.11 VPLS access redundancy

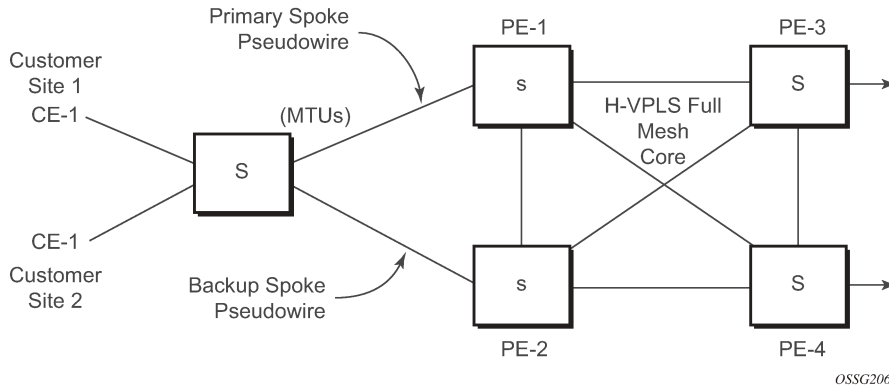
A second application of hierarchical VPLS is using MTUs that are not MPLS-enabled that must have Ethernet links to the closest PE node. To protect against failure of the PE node, an MTU can be dual-homed and have two SAPs on two PE nodes.

There are several mechanisms that can be used to resolve a loop in an access circuit; however, from an operations perspective, they can be subdivided into two groups:

- STP-based access, with or without M-VPLS.
- Non-STP based access using mechanisms such as MC-LAG, MC-APS, MC-Ring.

### 3.2.11.1 STP-based redundant access to VPLS

Figure 34: Dual-homed MTUs in two-tier hierarchy H-VPLS



In the configuration shown in [Figure 34: Dual-homed MTUs in two-tier hierarchy H-VPLS](#), STP is activated on the MTU and two PEs to resolve a potential loop. STP only needs to run in a single VPLS instance, and the results of the STP calculations are applied to all VPLSs on the link.

In this configuration, the scope of the STP domain is limited to MTU and PEs, while any topology change needs to be propagated in the whole VPLS domain including mesh SDPs. This is done by using so-called MAC-flush messages defined by RFC 4762. In the case of STP as an loop resolution mechanism, every TCN received in the context of an STP instance is translated into an LDP-MAC address withdrawal message (also referred to as a MAC-flush message) requesting to clear all FDB entries except the ones learned from the originating PE. Such messages are sent to all PE peers connected through SDPs (mesh and spoke) in the context of VPLS services, which are managed by the specified STP instance.

### 3.2.11.2 Redundant access to VPLS without STP

The Nokia implementation also includes alternative methods for providing a redundant access to Layer 2 services, such as MC-LAG, MC-APS, or MC-Ring. Also in this case, the topology change event needs to be propagated into the VPLS topology to provide fast convergence. The topology change propagation and its corresponding MAC flush processing in a VPLS service without STP is described in [Dual-homing to a VPLS service](#).

### 3.2.12 Object grouping and state monitoring

This feature introduces a generic operational group object that associates different service endpoints (pseudowires, SAPs, IP interfaces) located in the same or in different service instances.

The operational group status is derived from the status of the individual components, using specific rules specific to the application using the feature. A number of other service entities, the monitoring objects, can be configured to monitor the operational group status and to perform specific actions as a result of status transitions. For example, if the operational group goes down, the monitoring objects are brought down.

### 3.2.12.1 VPLS applicability — block on VPLS a failure

This feature is used in VPLS to enhance the existing BGP MH solution by providing a block-on-group failure function similar to the block-on-mesh failure feature implemented for LDP VPLS. On the PE selected as the Designated Forwarder (DF), if the rest of the VPLS endpoints fail (pseudowire spokes/pseudowire mesh or SAPs, or both), there is no path forward for the frames sent to the MH site selected as DF. The status of the VPLS endpoints, other than the MH site, is reflected by bringing down or up the objects associated with the MH site.

Support for the feature is provided initially in VPLS and B-VPLS instance types for LDP VPLS, with or without BGP-AD and for BGP VPLS. The following objects may be placed as components of an operational group: BGP VPLS pseudowires, SAPs, spoke-pseudowire, BGP-AD pseudowires. The following objects are supported as monitoring objects: BGP MH site, individual SAP, spoke-pseudowire.

The following rules apply:

- An object can only belong to one group at a time.
- An object that is part of a group cannot monitor the status of any group.
- An object that monitors the status of a group cannot be part of any group.
- An operational group may contain any combination of member types: SAP, spoke-pseudowire, BGP-AD, or BGP VPLS pseudowires.
- An operational group may contain members from different VPLS service instances.
- Objects from different services may monitor the operational group.
- The operational group feature may coexist in parallel with the block-on-mesh failure feature, provided they are running in different VPLS instances.

Perform the following steps to enable the block-on-mesh failure feature in a VPLS scenario.

1. Identify a set of objects whose forwarding state should be considered as a whole group, and then group them under an operational group using the **oper-group** command.
2. Associate other existing objects (clients) with the **oper-group** command using the **monitor-group** command; its forwarding state is derived from the related operational group state.

The status of the operational group (oper-group) is dictated by the status of one or more members, according to the following rules:

- The oper-group goes down if all the objects in the oper-group go down; the oper-group comes up if at least one of the components is up.
- An object in the oper-group is considered down if it is not forwarding traffic in at least one direction. That could be because the operational state is down, or the direction is blocked through resiliency mechanisms.
- If an oper-group is configured but no members are specified, its status is considered up. As soon as the first object is configured, the status of the oper-group is dictated by the status of the provisioned members.
- For BGP-AD or BGP VPLS pseudowires associated with the oper-group (under the **config>service-vpls>bgp>pw-template-binding** context), the status of the oper-group is down as long as the pseudowire members are not instantiated (auto-discovered and signaled).

A configuration can be described for the case of a BGP VPLS mesh used to interconnect different customer locations. For example, if a customer edge (CE) device is dual-homed to two PEs using BGP MH, the following configuration steps apply:

1. The bgp-vpls-mesh oper-group is created.
2. The BGP VPLS mesh is added to the bgp-vpls-mesh group through the pseudowire template used to create the BGP VPLS mesh.
3. The BGP MH site defined for the access endpoint is associated with the bgp-vpls-mesh group; its status from now on is influenced by the status of the BGP VPLS mesh.

### Example

The following is an example configuration for this case.

```
service>oper-group bgp-vpls-mesh-1 create
service>vpls>bgp>pw-template-binding> oper-group bgp-vpls-mesh-1
service>vpls>site> monitor-group bgp-vpls-mesh-1
```

### 3.2.13 MAC flush message processing

The previous sections described the operating principles of several redundancy mechanisms available in the context of a VPLS service. All of them rely on a MAC flush message as a tool to propagate topology change in the context of the VPLS. This section summarizes the basic rules to generate and process these messages.

The 7705 SAR Gen 2 supports two types of MAC flush message: flush-all-but-mine and flush-mine. The main difference between these messages is the type of action they signal.

The flush-all-but-mine message requests clearing all FDB entries learned from all other LDP peers except the originating PE. This type is also defined by RFC 4762 as an LDP MAC address withdrawal with an empty MAC address list.

The flush-all-mine message requests clearing all FDB entries learned from the originating PE. This means that this message has the opposite effect from the flush-all-but-mine message. This type is not included in the RFC 4762 definition and is implemented using a vendor-specific TLV.

The advantages and disadvantages of the individual message types are apparent from examples in the previous section. The description here focuses on summarizing actions taken upon reception of MAC flush messages and the conditions under which individual messages are generated.

The PE takes the following actions upon reception of MAC flush messages (regardless of the type).

1. Clears FDB entries of all indicated VPLS services conforming to the definition.
2. Propagates the message (preserving the type) to all LDP peers, if the **propagate-mac-flush** flag is enabled at the corresponding VPLS level.

The following conditions generate a flush-all-but-mine message.

- A flush-all-but-mine message is received from the LDP peer and the propagate-mac-flush flag is enabled. The message is sent to all LDP peers in the context of the VPLS service it was received.
- A TCN message in a context of the STP instance is received. The flush-all-but-mine message is sent to all LDP peers connected with spoke and mesh SDPs in the context of the VPLS service controlled by the STP instance (based on M-VPLS definition).

If all LDP peers are in the STP domain, that is, the M-VPLS and the uVPLS (user VPLS) both have the same topology, the router does not send any flush-all-but-mine message. If the router has uVPLS LDP peers outside the STP domain, the router sends flush-all-but-mine messages to all its uVPLS peers.





**Note:** The 7705 SAR Gen 2 does not send a withdrawal if the M-VPLS does not contain a mesh SDP. A mesh SDP must be configured in the M-VPLS to send withdrawals.

- A flush-all-but-mine message is generated when a switchover between spoke-SDPs of the same endpoint occurs. The message is sent to the LDP peer connected through the newly active spoke-SDP.

The following conditions generate a flush-mine message.

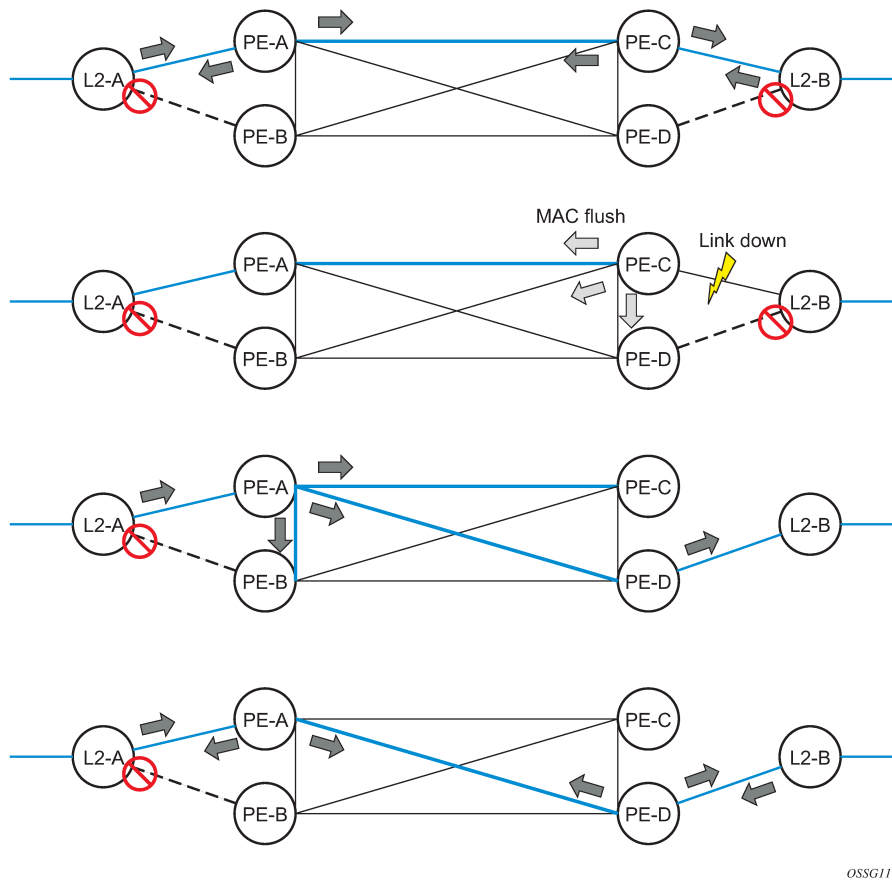
- The flush-mine message is received from the LDP peer and the propagate-mac-flush flag is enabled. The message is sent to all LDP peers in the context of the VPLS service it was received.
- The flush-mine message is generated when a SAP or SDP transitions from operationally up to an operationally down state and the send-flush-on-failure flag is enabled in the context of the specified VPLS service. The message is sent to all LDP peers connected in the context of the specified VPLS service. The send-flush-on-failure flag is blocked in M-VPLS and is only allowed to be configured in a VPLS service managed by M-VPLS. This is to prevent both messages being sent at the same time.
- The flush-mine message is generated when an MC-LAG SAP or MC-APS SAP transitions from an operationally up state to an operationally down state. The message is sent to all LDP peers connected in the context of the specified VPLS service.
- The flush-mine message is generated when an MC-Ring SAP transitions from operationally up to an operationally down state or when MC-Ring SAP transitions to slave state. The message is sent to all LDP peers connected in the context of the specified VPLS service.

### 3.2.13.1 Dual-homing to a VPLS service

The following figure shows a dual-homed connection to VPLS service (PE-A, PE-B, PE-C, PE-D) and operation in case of link failure (between PE-C and L2-B). Upon detection of a link failure, PE-C sends MAC-address-withdraw messages, which indicate to all LDP peers that they should flush all MAC addresses learned from PE-C. This leads to a broadcasting of packets addressing affected hosts and re-learning in case an alternative route exists.



Figure 35: Dual-homed CE connection to VPLS



OSSG117

The message described here is different from the message described in RFC 4762, *Virtual Private LAN Services Using LDP Signaling*. The difference is in the interpretation and action performed in the receiving PE. According to the standard definition, upon receipt of a MAC withdraw message, all MAC addresses, except the ones learned from the source PE, are flushed. This section specifies that all MAC addresses learned from the source are flushed. This message has been implemented as an LDP address withdraw message with vendor-specific type, length, and value (TLV), and is called the flush-mine message.

The RFC 4762 compliant message is used in VPLS services for recovering from failures in STP (Spanning Tree Protocol) topologies. The mechanism described in this section represents an alternative solution.

The advantage of this approach (as compared to STP-based methods) is that only the affected MAC addresses are flushed and not the full FDB. While this method does not provide a mechanism to secure alternative loop-free topology, the convergence time depends on the speed that the specified CE device opens an alternative link (L2-B switch in [Figure 35: Dual-homed CE connection to VPLS](#)) as well as on the speed that PE routers flush their FDB.

In addition, this mechanism is effective only if PE and CE are directly connected (no hub or bridge) as the mechanism reacts to the physical failure of the link.

### 3.2.13.2 MC-Ring and VPLS

The use of multichassis ring control in a combination with the plain VPLS SAP is supported by the FDB in individual ring nodes, in case the link (or ring node) failure cannot be cleared on the 7705 SAR Gen 2.

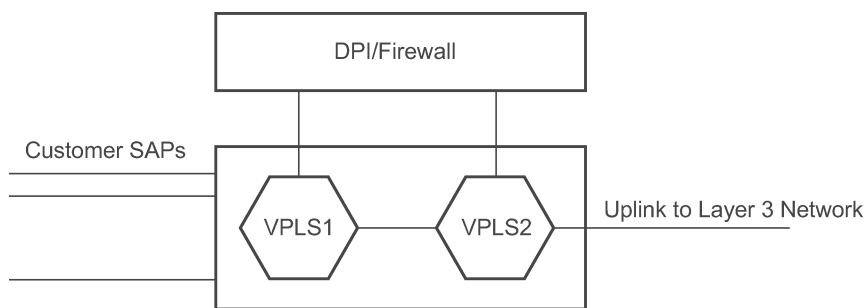
This combination is not easily blocked in the CLI. If configured, the combination may be functional but the switchover times are proportional to MAC aging in individual ring nodes or to the relearning rate, or both, because of the downstream traffic.

Redundant plain VPLS access in ring configurations, therefore, exclude corresponding SAPs from the multichassis ring operation. Configurations such as M-VPLS can be applied.

### 3.2.14 ACL next-hop for VPLS

The ACL next-hop for VPLS feature enables an ACL that has a forward to a SAP or SDP action specified to be used in a VPLS service to direct traffic with specific match criteria to a SAP or SDP. This allows traffic destined for the same gateway to be split and forwarded differently based on the ACL.

Figure 36: Application 1 diagram



OSSG207

Policy routing is a popular tool used to direct traffic in Layer 3 networks. As Layer 2 VPNs become more popular, especially in network aggregation, policy forwarding is required. Many providers are using methods such as DPI servers, transparent firewalls, or Intrusion Detection/Prevention Systems (IDS/IPS). Because these devices are bandwidth limited, providers want to limit traffic forwarded through them. In the setup shown in [Figure 36: Application 1 diagram](#), a mechanism is required to direct some traffic coming from a SAP to the DPI without learning, and other traffic coming from the same SAP directly to the gateway uplink-based learning.

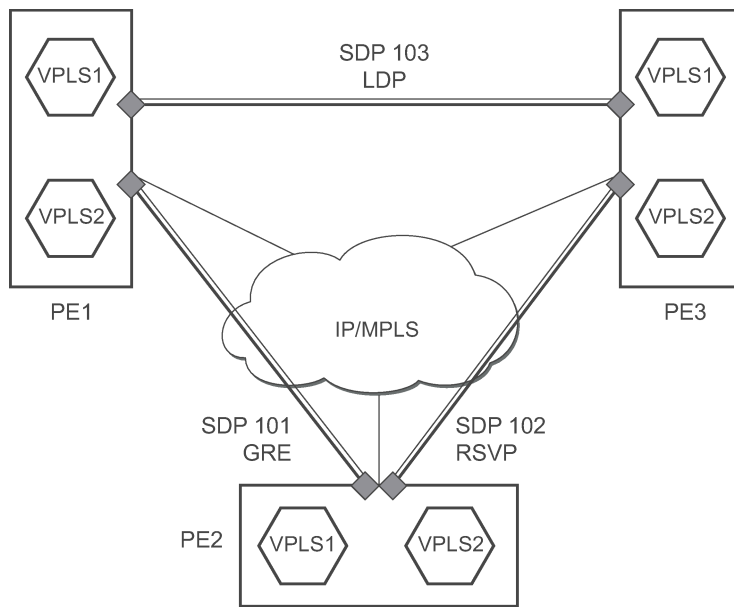
This feature allows the provider to create a filter that forwards packets to a specific SAP or SDP. The packets are then forwarded to the destination SAP regardless of learned destination. The SAP can either terminate a Layer 2 firewall, perform deep packet inspection (DPI) directly, or may be configured to be part of a cross-connect bridge into another service. This is useful when running the DPI remotely using VLLs. If an SDP is used, the provider can terminate it in a remote VPLS or VLL service where the firewall is connected. The filter can be configured under a SAP or SDP in a VPLS service. All packets (unicast, multicast, broadcast, and unknown) can be delivered to the destination SAP/SDP.

The filter may be associated with SAPs/SDPs belonging to a VPLS service only if all actions in the ACL forward to SAPs/SDPs that are within the context of that VPLS. Other services do not support this feature. An ACL that contains this feature is allowed, but the system drops any packet that matches an entry with this action.

### 3.2.15 SDP statistics for VPLS and VLL services

The simple three-node network in [Figure 37: SDP statistics for VPLS and VLL services](#) shows two MPLS SDPs and one GRE SDP defined between the nodes. These SDPs connect VPLS1 and VPLS2 instances that are defined in the three nodes. With this feature, the operator has local CLI-based as well as SNMP-based statistics collection for each VC used in the SDPs. This allows for traffic management of tunnel usage by the different services and with aggregation of the total tunnel usage.

Figure 37: SDP statistics for VPLS and VLL services



OSSG208

SDP statistics allow providers to bill customers on a per-SDP per-byte basis. This destination-based billing model can be used by providers with a variety of circuit types and have different costs associated with the circuits. An accounting file allows the collection of statistics in bulk.

### 3.2.16 BGP Auto-Discovery for LDP VPLS

BGP Auto-Discovery (BGP AD) for LDP VPLS is a framework for automatically discovering the endpoints of a Layer 2 VPN, using an operational model similar to that of an IP VPN. This model allows carriers to leverage existing network elements and functions, including but not limited to, route reflectors and BGP policies to control the VPLS topology.

BGP AD is a complement to targeted LDP, which is an established Layer 2 VPN signaling mechanism. BGP AD provides one-touch provisioning for LDP VPLS, where all related PEs are discovered automatically. The service provider may make use of existing BGP policies to regulate the exchanges between PEs in the same, or in different, autonomous system (AS) domains. The addition of BGP AD procedures does not require carriers to uproot their existing VPLS deployments or to change the signaling protocol.

3.2.16.1 BGP AD overview

The BGP protocol establishes neighbor relationships between configured peers. An open message is sent after the completion of the three-way TCP handshake. This open message contains information about the BGP peer sending the message. This message contains the Autonomous System Number (ASN), BGP version, timer information, and operational parameters, including capabilities. The capabilities of a peer are exchanged using two numerical values: the Address Family Identifier (AFI) and Subsequent Address Family Identifier (SAFI). These numbers are allocated by the Internet Assigned Numbers Authority (IANA). BGP AD uses AFI 65 (L2VPN) and SAFI 25 (BGP VPLS). For a complete list of allocations, see <http://www.iana.org/assignments/address-family-numbers> (for AFI) and <http://www.iana.org/assignments/safi-namespaces> (for SAFI).

3.2.16.2 Information model

Following the establishment of the peer relationship, the discovery process begins as soon as a new VPLS service instance is provisioned on the PE.

The following VPLS identifiers are used to indicate the VPLS membership and the individual VPLS instance:

- VPLS-ID**  
Membership information, unique network-wide identifier; the same value is assigned for all VPLS Switch Instances (VSIs) belonging to the same VPLS; encodable and carried as a BGP extended community in one of the following formats:
  - A two-octet AS-specific extended community
  - An IPv4 address-specific extended community
- VSI-ID**  
The unique identifier for each individual VSI, built by concatenating a route distinguisher (RD) with a 4-byte identifier (usually the system IP of the VPLS PE); encoded and carried in the corresponding BGP Network Layer Reachability Information (NLRI).

To advertise this information, BGP AD uses a simplified version of the BGP VPLS NLRI, where just the RD and the next four bytes are used to identify the VPLS instance. The Label Block and Label Size fields are not needed; T-LDP signals the service labels.

The format of the BGP AD NLRI is very similar to the one used for IP VPN, as shown in the following figure. The system IP may be used for the last four bytes of the VSI-ID, further simplifying the addressing and the provisioning process.

Figure 38: BGP AD NLRI versus IP VPN NLRI

| BGP AD NLRI Usage                 | IP VPN NLRI Usage                 |
|-----------------------------------|-----------------------------------|
| Route Distinguisher<br>(8 Octets) | Route Distinguisher<br>(8 Octets) |
| VSI id (4) (IP Prefix)            | IP Prefix                         |

The NLRI is exchanged between BGP peers, indicating how to reach prefixes. The NLRI is used in the Layer 2 VPN case to tell PE peers how to reach the VSI instead of specific prefixes. The advertisement

includes the BGP next-hop and a route target (RT). The BGP next-hop indicates the VSI location and is used in the next step to determine which signaling session is used for pseudowire signaling. The RT, also coded as an extended community, can be used to build a VPLS full mesh or an H-VPLS hierarchy through the use of BGP import or export policies.

BGP is only used to discover VPN endpoints and the corresponding far-end PEs; it is not used to signal the pseudowire labels. This task remains the responsibility of T-LDP.

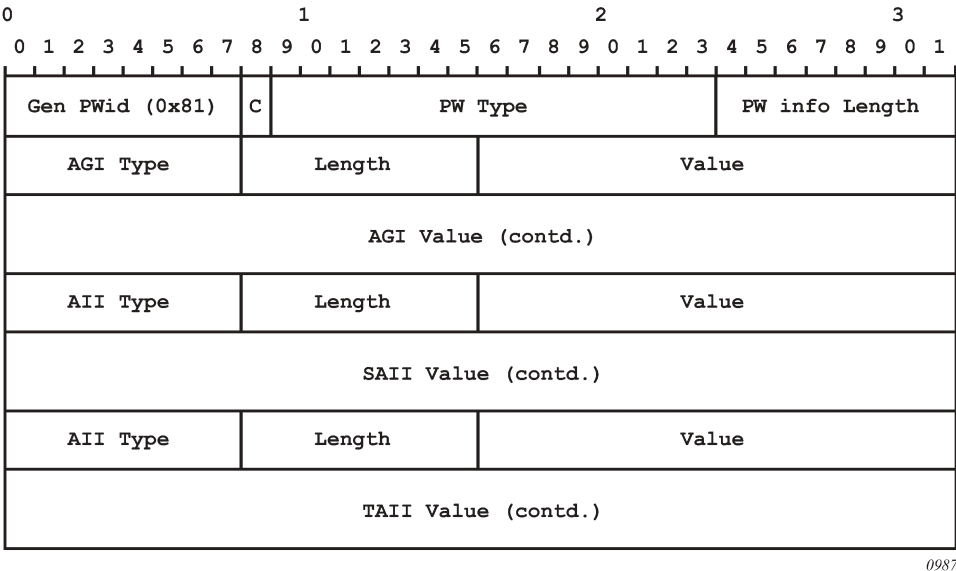
3.2.16.3 FEC element for T-LDP signaling

Two LDP FEC elements are defined in RFC 4447, *PW Setup & Maintenance Using LDP*. The original pseudowire-ID FEC element 128 (0x80) uses a 32-bit field to identify the virtual circuit ID and was used extensively in the initial VPWS and VPLS deployments. The simple format is easy to understand, but it does not provide the required information model for the BGP auto-discovery function. To support BGP AD and other new applications, a new Layer 2 FEC element, the generalized FEC element 129 (0x81) is required.

The generalized pseudowire-ID FEC element has been designed for auto-discovery applications. It provides a field, the Address Group Identifier (AGI), that is used to signal the membership information from the VPLS-ID. Separate address fields are provided for the source and target address associated with the VPLS endpoints, called the Source Attachment Individual Identifier (SAII) and Target Attachment Individual Identifier (TAII), respectively. These fields carry the VSI-ID values for the two instances that are to be connected through the signaled pseudowire.

The detailed format for FEC 129 is shown in the following figure.

Figure 39: Generalized pseudowire-ID FEC element



Each FEC field is designed as a sub-TLV equipped with its own type and length, providing support for new applications. The following FEC formats are used to accommodate the BGP AD information model.

- AGI (type 1) is identical in format and content to the BGP extended community attribute used to carry the VPLS-ID value.
- Source AII (type 1) is a 4-byte value used to carry the local VSI-ID (outgoing NLRI minus the RD).

- Target All (type 1) is a 4-byte value used to carry the remote VSI-ID (incoming NLRI minus the RD).

The user can configure the **adv-service-mtu** command to override the MTU value used in LDP signaling to the far-end of the pseudowire. If the **ignore-l2vpn-mtu-mismatch** command is not configured, the MTU value is also used to validate the value signaled by the far-end PE.

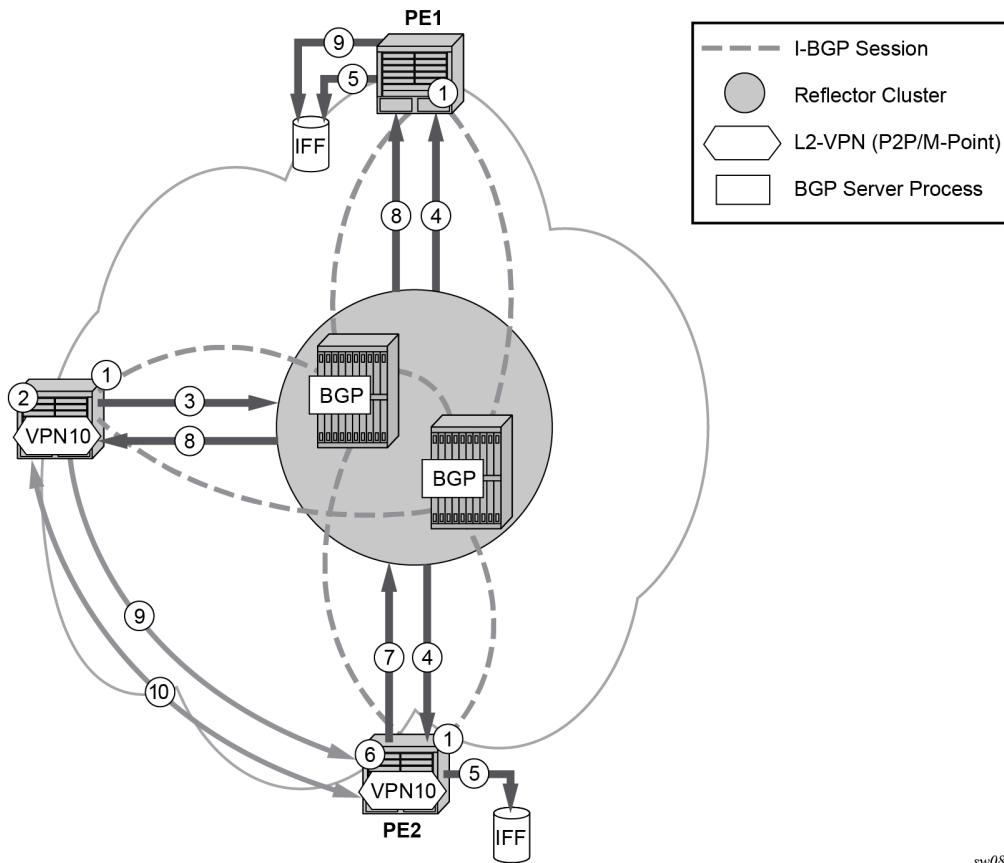
If the **ignore-l2vpn-mtu-mismatch** command is configured, the value of the MTU interface parameter received in the FEC element is not checked against the local service MTU, or against the MTU value signaled by the router. The router brings up the VPLS service regardless of any MTU mismatch.

### 3.2.16.4 BGP AD and T-LDP interaction

BGP is responsible for discovering the location of VSIs that share the same VPLS membership. LDP protocol is responsible for setting up the pseudowire infrastructure between the related VSIs by exchanging service-specific labels between them.

After the local VPLS information is provisioned in the local PE, the related PEs participating in the same VPLS are identified through BGP AD exchanges. A list of far-end PEs is generated and triggers the creation, if required, of the necessary T-LDP sessions to these PEs and the exchange of the service-specific VPN labels. The steps for the BGP AD discovery process and LDP session establishment and label exchange are shown in [Figure 40: BGP AD and T-LDP interaction](#) and described in [Table 8: BGP AD and T-LDP interaction key](#).

Figure 40: BGP AD and T-LDP interaction



sw0890

Table 8: BGP AD and T-LDP interaction key

| Key | Description   |
|-----|---|
| 1   | Establish I-BGP connectivity RR                                       |
| 2   | Configure VPN (10) on edge node (PE3)                                 |
| 3   | Announce VPN to RR using BGP AD                                       |
| 4   | Send membership update to each client of the cluster                  |
| 5   | LDP exchange or Inbound FEC Filtering (IFF) of non-match or VPLS down |
| 6   | Configure VPN (10) on edge node (PE2)                                 |
| 7   | Announce VPN to RR using BGP AD                                       |
| 8   | Send membership update to each client of the cluster                  |
| 9   | LDP exchange or IFF of non-match or VPLS down                         |
| 10  | Complete LDP bidirectional pseudowire establishment FEC 129           |

### 3.2.16.5 SDP usage

Service Access Points (SAPs) are linked to transport tunnels using service destination points (SDPs). The service architecture allows services to be abstracted from the transport network.

MPLS transport tunnels are signaled using the Resource Reservation Protocol (RSVP-TE) or by the Label Distribution Protocol (LDP). The capability to automatically create an SDP only exists for LDP-based transport tunnels. Using a manually provisioned SDP is available for both RSVP-TE and LDP transport tunnels. For more information about MPLS, LDP and RSVP, see the *7705 SAR Gen 2 MPLS Guide*.

GRE transport tunnels use GRE encapsulation and can be used with manually provisioned or auto created SDPs.

### 3.2.16.6 Automatic creation of SDPs

When BGP AD is used for LDP VPLS, with an LDP or GRE transport tunnel, there is no requirement to manually create an SDP. The LDP or GRE SDP can be automatically instantiated using the information advertised by BGP AD. This simplifies the configuration on the service node.

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

Enabling LDP on the IP interfaces connecting all nodes between the ingress and the egress builds transport tunnels based on the best IGP path. LDP bindings are automatically built and stored in the hardware. These entries contain an MPLS label pointing to the best next-hop along the best path toward the destination.

When two endpoints need to connect and no SDP exists, a new SDP is created automatically. New services added between two endpoints that already have an automatically-created SDP are immediately used. No new SDP is created. The far-end information is obtained from the BGP next-hop information in the NLRI. When services are withdrawn with a BGP\_Unreach\_NLRI, the automatically established SDP remains up as long as at least one service is connected between those endpoints. An automatically created SDP is removed and the resources released when the only or last service is removed.

The service provider has the option of associating the auto-discovered SDP with a split horizon group using the pw-template-binding option, to control the forwarding between pseudowires and to prevent Layer 2 service loops.

An auto-discovered SDP using a pw-template-binding without a split horizon group configured has similar traffic flooding behavior as a spoke-SDP.

### 3.2.16.7 Manually provisioned SDP

The carrier is required to manually provision the SDP if they create transport tunnels using RSVP-TE. Operators have the option to choose a manually configured SDP, if they use LDP as the tunnel signaling protocol. The functionality is the same regardless of the signaling protocol.

Creating a BGP AD enabled VPLS service on an ingress node with the manually provisioned SDP option causes the tunnel manager to search for an existing SDP that connects to the far-end PE. The far-end IP information is obtained from the BGP next-hop information in the NLRI. If a single SDP exists to that PE, it is used. If no SDP is established between the two endpoints, the service remains down until a manually configured SDP becomes active.

When multiple SDPs exist between two endpoints, the tunnel manager selects the appropriate SDP. The algorithm prefers SDPs with the best (lower) metric. If there are multiple SDPs with equal metrics, the operational states of the SDPs with the best metric is considered. If the operational states are the same, the SDP with the higher SDP-ID is used. If an SDP with a preferred metric is found with an operational state that is not active, the tunnel manager flags it as ineligible and restarts the algorithm.

### 3.2.16.8 Automatic instantiation of pseudowires (SDP bindings)

The choice of manual or auto-provisioned SDPs has limited impact on the amount of required provisioning. Most of the savings are achieved through the automatic instantiation of the pseudowire infrastructure (SDP bindings). This is achieved for every auto-discovered VSI through the use of the pseudowire template concept.

Each VPLS service that uses BGP AD contains the pw-template-binding option defining specific Layer 2 VPN parameters. This command references a PW template, which defines the pseudowire parameters. The same PW template may be referenced by multiple VPLS services. As a result, changes to these pseudowire templates have to be treated with caution as they may impact many customers simultaneously.

The Nokia implementation provides for safe handling of pseudowire templates. Changes to the pseudowire templates are not automatically propagated. Tools are provided to evaluate and distribute the changes. The following command is used to distribute changes to a PW template at the service level to one or all services that use that template:

```
tools perform service id 300 eval-pw-template 1 allow-service-impact
```

If the service ID is omitted, all services are updated. The type of change made to the PW template influences how the service is impacted.



- Adding or removing a **split-horizon-group** causes the router to destroy the original object and recreate using the new value.
- Changing parameters in the **vc-type {ether | vlan}** command requires LDP to re-signal the labels.

Both of these changes are service affecting. Other changes are not service affecting.

### 3.2.16.9 Mixing statically configured and auto-discovered pseudowires in a VPLS

The services implementation allows for manually provisioned and auto-discovered pseudowire (SDP bindings) to coexist in the same VPLS instance (for example, both FEC 128 and FEC 129 are supported). This allows for gradual introduction of auto-discovery into an existing VPLS deployment.

As FEC 128 and 129 represent different addressing schemes, it is important to make sure that only one is used at any time between the same two VPLS instances. Otherwise, both pseudowires may become active causing a loop that may adversely impact the correct functioning of the service. It is recommended that FEC 128 pseudowire be disabled as soon as the FEC129 addressing scheme is introduced in a portion of the network. Alternatively, RSTP may be used during the migration as a safety mechanism to provide additional protection against operational errors.

### 3.2.16.10 Resiliency schemes

The use of BGP AD on the network side or in the backbone does not affect the resiliency schemes Nokia has developed in the access network. This means that both MC-LAG and M-VPLS can still be used.

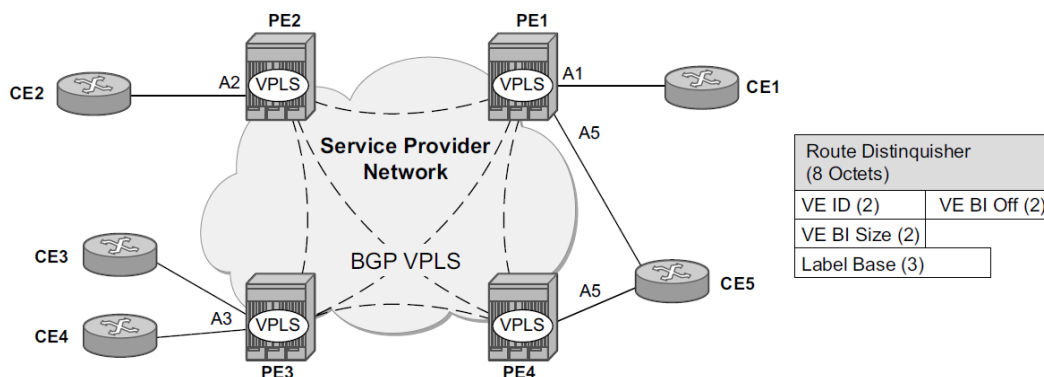
BGP AD may coexist with H-VPLS resiliency schemes (for example, dual-homed MTU devices to different PE-rs nodes) using existing methods (M-VPLS and statically configured active or standby pseudowire endpoint).

If provisioned SDPs are used by BGP AD, M-VPLS may be used to provide loop avoidance. However, it is not possible to auto-discover active or standby pseudowires and to instantiate the related endpoint.

### 3.2.17 BGP VPLS

The Nokia BGP VPLS solution, compliant with RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*, is described in this section.

Figure 41: BGP VPLS solution



OSSG488

The preceding figure shows the service representation for BGP VPLS mesh. The major BGP VPLS components and the deltas from LDP VPLS with BGP AD are as follows.

- The data plane is identical to the LDP VPLS solution; for example, VPLS instances interconnected by pseudowire mesh. Split-horizon groups may be used for loop avoidance between pseudowires.
- Addressing is based on the 2-byte VE ID assigned to the VPLS instance.  
BGP AD for LDP VPLS: 4-byte VSI-ID (system IP) identifies the VPLS instance.
- The target VPLS instance is identified by the Route Target (RT) contained in the MP-BGP advertisement (extended community attribute).  
BGP AD: a new MP-BGP extended community is used to identify the VPLS. RT is used for topology control.
- Auto-discovery is MP-BGP based; the same AFI, SAFI is used as for LDP VPLS BGP-AD.
  - The BGP VPLS updates are distinguished from the BGP AD updates based on the value of the NLRI prefix length: 17 bytes for BGP VPLS, 12 bytes for BGP AD.
  - BGP AD NLRI is shorter because there is no need to carry pseudowire label information as T-LDP does the pseudowire signaling for LDP VPLS.
- Pseudowire label signaling is MP-BGP based. Therefore, the BGP NLRI content also includes label-related information; for example, block offset, block size, and label base.
  - For LDP VPLS, T-LDP is used for signaling the pseudowire service label.
  - The Layer 2 extended community proposed in RFC 4761 is used to signal pseudowire characteristics; for example, VPLS status, control word, and sequencing.

### 3.2.17.1 Pseudowire signaling details

The pseudowire is set up using the following NLRI fields:

- **VE Block Offset (VBO)**  
VBO is used to define for each VE-ID set the NLRI is targeted, as follows:
  - $VBO = n * VBS + 1$ ; for  $VBS=8$ , this results in 1, 9, 17, 25, ...
  - Targeted Remote VE-IDs are from VBO to  $(VBO + VBS - 1)$
- **VE Block Size (VBS)**  
VBS defines how many contiguous pseudowire labels are reserved, starting with the Label Base.  
The Nokia implementation always uses a value of 8.
- **Label Base (LB)**  
LB is the local allocated label base. The next eight consecutive labels available are allocated for remote PEs.

This BGP update is telling the other PEs that accept the RT: "to reach me (VE-ID = x), use a pseudowire label of  $LB + VE-ID - VBO$  using the BGP NLRI for which  $VBO \leq \text{local VE-ID} < VBO + VBS$ ."

The following is an example of how this algorithm works, assuming PE1 has VE-ID 7 configured.

1. PE1 generates a Label Block of eight consecutive labels, starting with  $LB = 1000$ .
2. PE1 then sends BGP update with pseudowire information of ( $VBO = 1$ ,  $VBS=8$ ,  $LB=1000$ ) in the NLRI.
3. This pseudowire information is accepted by all participating PEs with VE-IDs from 1 to 8.

4. Each of the receiving PEs use the pseudowire label = LB + VE-ID - VBO to send traffic back to the originator PE. For example, VE-ID 2 uses pseudowire label 1001.

The following procedure applies assuming that VE-ID = 10 is configured in another PE4.

1. PE4 sends a BGP update with the new VE-ID in the network, which is received by all the other participating PEs, including PE1.
2. PE1, upon reception, generates another label block of 8 labels for the VBO = 9. For example, the initial PE creates new pseudowire signaling information of (VBO = 9, VBS = 8, LB = 3000) and inserts it in a new NLRI and BGP update that is sent in the network.
3. This new NLRI is used by the VE-ID from 9 to 16 to establish pseudowires back to the originator PE1. For example, PE4 with VE-ID 10 uses pseudowire label 3001 to send VPLS traffic back to PE1.
4. The PEs owning the set of VE-IDs from 1 to 8 ignore this NLRI.

In addition to the pseudowire label information, the "Layer2 Info Extended Community" attribute must be included in the BGP update for BGP VPLS to signal the attributes of all the pseudowires that converge toward the originator VPLS PE.

The following figure shows the format for the attribute.

*Figure 42: Layer 2 Info Extended Community attribute*

|                                    |
|------------------------------------|
| Extended Community Type (2 octets) |
| Encaps Type (1 octet)              |
| Control Flags (1 octet)            |
| Layer-2 MTU (2 octets)             |
| Reserved (2 octets)                |

sw0908

The following list describes the fields shown in the preceding figure:

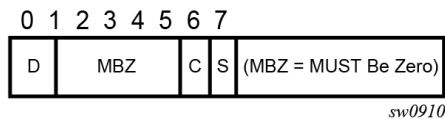
- **Extended Community Type**  
The value allocated by IANA for this attribute is 0x800A.
- **Encaps Type**  
Encapsulation type identifies the type of pseudowire encapsulation. The only value used by BGP VPLS is 19 (13 in HEX). This value identifies the encapsulation to be used for a pseudowire instantiated through BGP signaling, which is the same as the one used for Ethernet pseudowire type for regular VPLS. There is no support for an equivalent Ethernet VLAN pseudowire in BGP VPLS in BGP signaling.
- **Control Flags**  
The control information about the pseudowires (see [Figure 41: BGP VPLS solution](#)).  
[Figure 43: Control flags bit vector](#) shows the Control Flags bit vector.
- **Layer-2 MTU**  
The Maximum Transmission Unit to be used on the pseudowires.
- **Reserved**

This field is reserved and must be set to zero and ignored on reception except where it is used for VPLS preference.

For inter-AS, the preference information must be propagated between autonomous systems. Consequently, as the VPLS preference in a BGP-VPLS or BGP multihoming update extended community is zero, the local preference is copied by the egress ASBR into the VPLS preference field before sending the update to the EBGP peer. The adjacent ingress ASBR then copies the received VPLS preference into the local preference to prevent the update being considered malformed.

The following figure shows the detailed format for the Control Flags bit vector.

*Figure 43: Control flags bit vector*



The bits in the Control Flags are defined as follows:

- **S**  
Sequenced delivery of frames must or must not be used when sending VPLS packets to this PE, depending on whether S is 1 or 0, respectively.
- **C**  
A control word must or must not be present when sending VPLS packets to this PE, depending on whether C is 1 or 0, respectively. By default, Nokia implementation uses value 0.
- **MBZ**  
Must Be Zero bits, set to zero when sending and ignored when receiving.
- **D**  
The status of the whole VPLS instance (VSI); D=0 if Admin and Operational status are up, D=1 otherwise.

The events that set the D-bit to 1 to indicate VSI down status in a BGP update message sent out from a PE are as follows.

- Local VSI is shutdown administratively using the **configure service vpls shutdown** command.
- All the related endpoints (SAPs or LDP pseudowires) are down.
- There are no related endpoints (SAPs or LDP pseudowires) configured yet in the VSI. The intent is to save the core bandwidth by not establishing the BGP pseudowires to an empty VSI.
- Upon reception of a BGP update message with D-bit set to 1, all the receiving VPLS PEs must mark related pseudowires as down.

The following events do not set the D-bit to 1:

- **the local VSI is deleted**  
A BGP update with unreachable-NLRI is sent out. Upon reception, all remote VPLS PEs must remove the related pseudowires and BGP routes.
- **the local SDP goes down**  
Only the BGP pseudowires mapped to that SDP goes down. There is no BGP update sent.

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

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

### 3.2.17.2 Supported VPLS features

BGP VPLS supports a new type of pseudowire signaling based on MP-BGP. It makes use of VPLS and inherits all the existing Ethernet switching functions.

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

The following are some of the most important VPLS features that are also ported to BGP VPLS:

- VPLS data plane features (for example, FDB management, SAPs, LAG access, and BUM rate limiting)
- MPLS tunneling: LDP, LDP over RSVP-TE, RSVP-TE, GRE, and MP-BGP based on RFC 8277 (Inter-AS option C solution)



**Note:** Pre-provisioned SDPs must be configured when RSVP-signaled transport tunnels are used.

- H-VPLS topologies, hub and spoke traffic distribution
- coexistence with LDP VPLS (with or without BGP AD) in the same VPLS instance



**Note:** LDP and BGP signaling should operate in disjoint domains to simplify loop avoidance.

- Coexists with BGP-based multihoming
- BGP VPLS is supported as the control plane for B-VPLS
- Supports IGMP/PIM snooping for IPv4
- High Availability (HA)
- Ethernet service OAM toolset (IEEE 802.1ag and Y.1731)



**Note:** CPE ping, MAC trace, MAC ping, MAC populate, and MAC purge are not supported.

- Support for RSVP and LSP P2MP LSP for VPLS/B-VPLS BUM

### 3.2.18 VCCV BFD support for VPLS services

The SR OS supports RFC 5885, which specifies a method for carrying BFD in a pseudowire associated channel. For general information about VCCV BFD, limitations, and configuring, see the VLL Services chapter.

VCCV BFD is supported on the following VPLS services:

- T-LDP spoke-SDP termination on VPLS (including I-VPLS, B-VPLS, and R-VPLS)
- H-VPLS spoke-SDP
- BGP VPLS
- VPLS with BGP autodiscovery

To configure VCCV BFD for H-VPLS (where the pseudowire template does not apply), configure the BFD template using the **configure service vpls spoke-sdp bfd-template name** command, then enable it using the **configure service vpls spoke-sdp bfd-enable** command.

For BGP VPLS, a BFD template is referenced from the pseudowire template binding context. To configure VCCV BFD for BGP VPLS, use the **configure service vpls bgp pw-template-binding bfd-template name** command and enable it using the **configure service vpls bgp pw-template-binding bfd-enable** command.

For BGP-AD VPLS, a BFD template is referenced from the pseudowire template context. To configure VCCV BFD for BGP-AD, use the **configure service vpls bgp-ad pw-template-binding bfd-template name** command, and enable it using the **configure service vpls bgp-ad pw-template-binding bfd-enable** command.

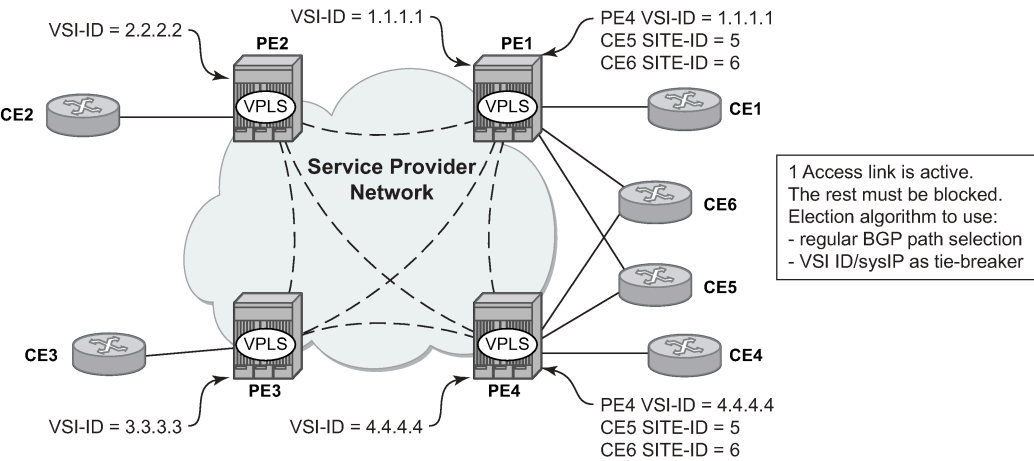
### 3.2.19 BGP multihoming for VPLS

This section describes BGP-based procedures for electing a designated forwarder among the set of PEs that are multihomed to a customer site. Only the local PEs are actively participating in the selection algorithm. The PEs remote from the dual-homed CE are not required to participate in the designated forwarding election for a remote dual-homed CE.

The main components of the BGP-based multihoming solution for VPLS are:

- Provisioning model
- MP-BGP procedures
- Designated Forwarder Election
- Blackhole avoidance, indicating the designated forwarder change toward the core PEs and access PEs or CEs
- The interaction with pseudowire signaling (BGP/LDP)

Figure 44: BGP multihoming for VPLS



OSSG489

Figure 44: BGP multihoming for VPLS shows the VPLS using BGP multihoming for the case of multihomed CEs. Although the figure shows the case of a pseudowire infrastructure signaled with LDP for an LDP VPLS using BGP-AD for discovery, the procedures are identical for BGP VPLS or for a mix of BGP- and LDP-signaled pseudowires.

3.2.19.1 Information model and required extensions to L2VPN NLRI

VPLS Multihoming using BGP-MP expands on the BGP AD and BGP VPLS provisioning model. The addressing for the multihomed site is still independent from the addressing for the base VSI (VSI-ID or, respectively, VE-ID). Every multihomed CE is represented in the VPLS context through a site-ID, which is the same on the local PEs. The site-ID is unique within the scope of a VPLS. It serves to differentiate between the multihomed CEs connected to the same VPLS Instance (VSI). For example, in [Figure 45: BGP MH-NLRI for VPLS multihoming](#), CE5 is assigned the same site-ID on both PE1 and PE4. For the same VPLS instance, different site-IDs are assigned for multihomed CE5 and CE6; for example, site-ID 5 is assigned for CE5 and site-ID 6 is assigned for CE6. The single-homed CEs (CE1, 2, 3, and 4) do not require allocation of a multihomed site-ID. They are associated with the addressing for the base VSI, either VSI-ID or VE-ID.

The new information model required changes to the BGP usage of the NLRI for VPLS. The extended MH NLRI for Multi-Homed VPLS is compared with the BGP AD and BGP VPLS NRIs in [Figure 45: BGP MH-NLRI for VPLS multihoming](#).

Figure 45: BGP MH-NLRI for VPLS multihoming

| BGP VPLS (RFC 4761)            | LDP VPLS (RFC 4762)            | BGP and/or LDP VPLS            |
|--------------------------------|--------------------------------|--------------------------------|
| <b>BGP BPLS NLRI</b>           | <b>BGP AD NLRI</b>             | <b>BGP MH NLRI</b>             |
| (2) Length=17                  | (2) Length=12                  | (2) Length=17                  |
| Route Distinguisher (8 Octets) | Route Distinguisher (8 Octets) | Route Distinguisher (8 Octets) |
| VE ID (2)                      | VSI-ID (System IP) (4)         | Site-ID (2)                    |
| VE BI Size (2)                 | VBS Not Used (2)               | ZEROs (2)                      |
| Label Base (3)                 | LB Not Used (3)                | ZEROs (3)                      |

OSSG490

The BGP VPLS NLRI described in RFC 4761 is used to carry a 2-byte site-ID that identifies the MH site. The last seven bytes of the BGP VPLS NLRI used to instantiate the pseudowire are not used for BGP-MH and are zeroed out. This NLRI format translates into the following processing path in the receiving VPLS PE:

- BGP VPLS PE: no label information means there is no need to set up a BGP pseudowire.
- BGP AD for LDP VPLS: length =17 indicates a BGP VPLS NLRI that does not require any pseudowire LDP signaling.

The processing procedures described in this section start from the above identification of the BGP update as not destined for pseudowire signaling.

The RD ensures that the NLRIs associated with a specific site-ID on different PEs are seen as different by any of the intermediate BGP nodes (RRs) on the path between the multihomed PEs. That is, different RDs must be used on the MH PEs every time an RR or an ASBR is involved to guarantee the MH NLRIs reach the PEs involved in VPLS MH.

The L2-Info extended community from RFC 4761 is used in the BGP update for MH NLRI to initiate a MAC flush for blackhole avoidance, to indicate the operational and admin status for the MH site or the DF election status.

After the pseudowire infrastructure between VSIs is built using either RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*, or RFC 4761 procedures, or a mix of pseudowire signaling procedures, on activation of a multihomed site, an election algorithm must be run on the local and remote PEs to determine which site is the designated forwarder (DF). The end result is that all the related MH sites in a VPLS are placed in standby except for the site selected as DF. Nokia BGP-based multihoming solution uses the DF election procedure described in the IETF working group document *draft-ietf-bess-vpls-multihoming-01*. The implementation allows the use of BGP local preference and the received VPLS preference but does not support setting the VPLS preference to a non-zero value.

### 3.2.19.2 Supported services and multihoming objects

This feature is supported for the following services:

- LDP VPLS with or without BGP-AD
- BGP VPLS (BGP multihoming for inter-AS BGP-VPLS services is not supported)
- mix of the above

The following access objects can be associated with MH Site:

- SAPs
- SDP bindings (pseudowire object), both mesh-SDP and spoke-SDP
- Split Horizon Group

Under the SHG we can associate either one or multiple of the following objects: SAPs, pseudowires (BGP VPLS, BGP-AD, provisioned and LDP-signaled spoke-SDP and mesh-SDP)

### 3.2.19.3 Blackhole avoidance

Blackholing refers to the forwarding of frames to a PE that is no longer carrying the designated forwarder. This could happen for traffic from:

- Core PE participating in the main VPLS



- Customer Edge devices (CEs)
- Access PEs (pseudowires between them and the MH PEs are associated with MH sites)

Changes in DF election results or MH site status must be detected by all of the above network elements to provide for Blackhole Avoidance.

### 3.2.19.3.1 MAC flush to the core PEs

Assuming that there is a transition of the existing DF to non-DF status, the PE that owns the MH site experiencing this transition generates a MAC flush-all-from-me (negative MAC flush) toward the related core PEs. Upon reception, the remote PEs flush all the MACs learned from the MH PE.

MAC flush-all-from-me indication message is sent using the following core mechanisms:

- For LDP VPLS running between core PEs, existing LDP MAC flush is used.
- For pseudowire signaled with BGP VPLS, MAC flush is provided implicitly using the L2-Info Extended community to indicate a transition of the active MH site; for example, the attached objects going down or more generically, the entire site going from Designated Forwarder (DF) to non-DF.
- Double flushing does not happen as it is expected that between any pair of PEs, there exists only one type of pseudowires, either BGP or LDP pseudowire, but not both.

### 3.2.19.3.2 Indicating non-DF status toward the access PE or CE

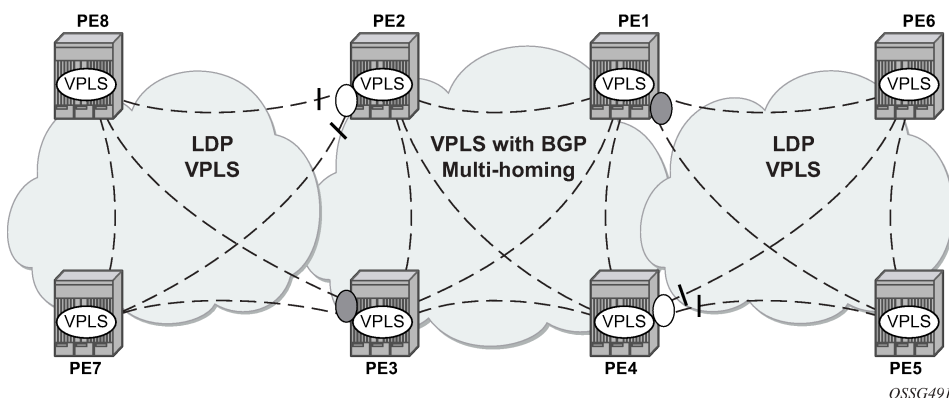
For the CEs or access PEs, support is provided for indicating the blocking of the MH site using the following procedures:

- For MH Access PE running LDP pseudowires, the LDP standby-status is sent to all LDP pseudowires.
- For MH CEs, site deactivation is linked to a CCM failure on a SAP that has a down MEP configured.

### 3.2.19.4 BGP multihoming for VPLS inter-domain resiliency

BGP MH for VPLS can be used to provide resiliency between different VPLS domains. An example of a multihoming topology is shown in [Figure 46: BGP MH used in an HVPLS topology](#).

Figure 46: BGP MH used in an HVPLS topology



LDP VPLS domains are interconnected using a core VPLS domain, either BGP VPLS or LDP VPLS. The gateway PEs, for example PE2 and PE3, are running BGP multihoming where one MH site is assigned to each of the pseudowires connecting the access PE, PE7, and PE8 in this example.

Alternatively, the MH site can be associated with multiple access pseudowires using an access SHG. The **configure service vpls site failed-threshold** command can be used to indicate the number of pseudowire failures that are required for the MH site to be declared down.

### 3.2.20 Multicast-aware VPLS

Because it is a Layer 2 service, multicast and broadcast frames are typically flooded in a VPLS. While broadcast frames are targeted to all receivers in the VPLS, IP multicast for a multicast group is typically targeted to selected receivers in the VPLS. Flooding to all sites can cause wasted network bandwidth and unnecessary replication on the ingress PE router.

To avoid this condition, VPLS is IP multicast-aware. It forwards IP multicast traffic to the object on which the IP multicast traffic is requested. This is achieved by enabling the following related multicast protocol snooping:

- IGMP snooping
- MLD snooping
- PIM snooping

#### 3.2.20.1 IGMP snooping for VPLS

When IGMP snooping is enabled in a VPLS service, IGMP messages received on SAPs and SDPs are snooped to determine the scope of flooding for a specified stream or (S,G). IGMP snooping operates in a proxy mode, where the system summarizes upstream IGMP reports and responds to downstream queries.

Streams are sent to all SAPs and SDPs on which there is a multicast router (either discovered dynamically from received query messages or configured statically using the **mrouter-port** command) and on which an active join for that stream has been received. The mrouter port configuration adds a (\*,\*) entry into the MFIB, which causes all groups (and IGMP messages) to be sent out of the respective object and causes IGMP messages received on that object to be discarded.

Directly-connected multicast sources are supported when IGMP snooping is enabled.

IGMP snooping is enabled at the service level.

IGMP is not supported on the following:

- B-VPLS, routed I-VPLS, PBB-VPLS services
- a router configured with **enable-inter-as-vpn** or **enable-rr-vpn-forwarding**
- the following forms of default SAP:
  - \*
  - \*.null
  - \*.\*.
- a VPLS service configured with a connection profile VLAN SAP

### 3.2.20.2 MLD snooping for VPLS

MLD snooping is an IPv6 version of IGMP snooping, and the guidelines and procedures are similar; see [IGMP snooping for VPLS](#) for more information. However, MLD snooping uses MAC-based forwarding. See [MAC-based IPv6 multicast forwarding](#) for more information. Directly connected multicast sources are supported when MLD snooping is enabled.

MLD snooping is enabled at the service level with the following restrictions.

- MLD snooping is not supported in the following services:
  - B-VPLS
  - Routed I-VPLS
  - EVPN-MPLS services
  - PBB-EVPN services
- MLD snooping is not supported under the following forms of default SAP:
  - \*
  - \*.null
  - \*.\*.
- MLD snooping is not supported in a VPLS service configured with a connection profile VLAN SAP.

### 3.2.20.3 PIM snooping for VPLS

PIM snooping for VPLS allows a VPLS PE router to build multicast states by snooping PIM protocol packets that are sent over the VPLS. The VPLS PE then forwards multicast traffic based on the multicast states. When all receivers in a VPLS are IP multicast routers running PIM, multicast forwarding in the VPLS is efficient when PIM snooping for VPLS is enabled.

Because of PIM join/prune suppression, to make PIM snooping operate over VPLS pseudowires, two options are available: plain PIM snooping and PIM proxy. PIM proxy is the default behavior when PIM snooping is enabled for a VPLS.

PIM snooping is supported for both IPv4 and IPv6 multicast by default and can be configured to use SG-based forwarding (see [IPv6 multicast forwarding](#) for more information).

Directly connected multicast sources are supported when PIM snooping is enabled.

The following restrictions apply to PIM snooping:

- PIM snooping for IPv4 and IPv6 is not supported:
  - in the following services:
    - PBB B-VPLS
    - R-VPLS (including I-VPLS and BGP EVPN)
    - PBB-EVPN B-VPLS
    - EVPN-VXLAN R-VPLS
  - on a router configured with **enable-inter-as-vpn** or **enable-rr-vpn-forwarding**
  - under the following forms of default SAP:
    - \*

- \*.null
- \*\*
- in a VPLS service configured with a connection profile VLAN SAP
- with connected SR OSs configured with **improved-assert**
- with subscriber management in the VPLS service
- as a mechanism to drive MCAC
- PIM snooping for IPv6 is not supported:
  - in the following services:
    - PBB I-VPLS
    - BGP-VPLS
    - BGP EVPN (including PBB-EVPN)
    - VPLS E-Tree
    - Management VPLS
  - with the configuration of MLD snooping

### 3.2.20.3.1 Plain PIM snooping

In a plain PIM snooping configuration, VPLS PE routers only snoop; PIM messages are generated on their own. Join/prune suppression must be disabled on CE routers.

When plain PIM snooping is configured, if a VPLS PE router detects a condition where join/prune suppression is not disabled on one or more CE routers, the PE router puts PIM snooping into the PIM proxy state. A trap is generated that reports the condition to the operator and is logged to the syslog. If the condition changes, for example, join/prune suppression is disabled on CE routers, the PE reverts to the plain PIM snooping state. A trap is generated and is logged to the syslog.

### 3.2.20.3.2 PIM proxy

For PIM proxy configurations, VPLS PE routers perform the following:

- snoop hellos and flood hellos in the fast datapath
- consume join/prune messages from CE routers
- generate join/prune messages upstream using the IP address of one of the downstream CE routers
- run an upstream PIM state machine to determine whether a join/prune message should be sent upstream

Join/prune suppression is not required to be disabled on CE routers, but it requires all PEs in the VPLS to have PIM proxy enabled. Otherwise, CEs behind the PEs that do not have PIM proxy enabled may not be able to get multicast traffic that they are interested in if they have join/prune suppression enabled.

When PIM proxy is enabled, if a VPLS PE router detects a condition where join/prune suppression is disabled on all CE routers, the PE router puts PIM proxy into a plain PIM snooping state to improve efficiency. A trap is generated to report the scenario to the operator and is logged to the syslog. If the condition changes, for example, join/prune suppression is enabled on a CE router, PIM proxy is placed

back into the operational state. Again, a trap is generated to report the condition to the operator and is logged to the syslog.

### 3.2.20.4 IPv6 multicast forwarding

When MLD snooping or PIM snooping for IPv6 is enabled, the forwarding of IPv6 multicast traffic is MAC-based; see [MAC-based IPv6 multicast forwarding](#) for more information.

The operation with PIM snooping for IPv6 can be changed to SG-based forwarding; see [SG-based IPv6 multicast forwarding](#) for more information.

The following command configures the IPv6 multicast forwarding mode with the default being **mac-based**:

**configure service vpls mcast-ipv6-snooping-scope {sg-based | mac-based}**

The forwarding mode can only be changed when PIM snooping for IPv6 is disabled.

#### 3.2.20.4.1 MAC-based IPv6 multicast forwarding

This section describes IPv6 multicast address to MAC address mapping and IPv6 multicast forwarding entries.

For IPv6 multicast address to MAC address mapping, Ethernet MAC addresses in the range of 33-33-00-00-00-00 to 33-33-FF-FF-FF-FF are reserved for IPv6 multicast. To map an IPv6 multicast address to a MAC-layer multicast address, the low-order 32 bits of the IPv6 multicast address are mapped directly to the low-order 32 bits in the MAC-layer multicast address.

For IPv6 multicast forwarding entries, IPv6 multicast snooping forwarding entries are based on MAC addresses, while native IPv6 multicast forwarding entries are based on IPv6 addresses. When both MLD snooping or PIM snooping for IPv6 and native IPv6 multicast are enabled on the same device, both types of forwarding entries are supported on the same forward plane, although they are used for different services.

The following output shows a service with PIM snooping for IPv6 that has received joins for two multicast groups from different sources. As the forwarding mode is MAC-based, there is a single MFIB entry created to forward these two groups.

```
*A:PE# show service id 1 pim-snooping group ipv6
=====
PIM Snooping Groups ipv6
=====
Group Address          Source Address          Type      Incoming
Intf                  Num
Oifs
-----
ff0e:db8:1000::1      2001:db8:1000::1      (S,G)    SAP:1/1/1      2
ff0e:db8:1001::1      2001:db8:1001::1      (S,G)    SAP:1/1/1      2
-----
Groups : 2
=====
*A:PE#

*A:PE# show service id 1 all | match "Mcast IPv6 scope"
Mcast IPv6 scope : mac-based
*A:PE#

*A:PE# show service id 1 mfib
=====
Multicast FIB, Service 1
```

```

=====
Source Address  Group Address          Port Id                      Svc Id  Fwd
Blk
-----
*              33:33:00:00:00:01  sap:1/1/1                  Local   Fwd
                                sap:1/1/2                  Local   Fwd
-----
Number of entries: 1
=====
*A:PE#

```

### 3.2.20.4.2 SG-based IPv6 multicast forwarding

When PIM snooping for IPv6 is configured, SG-based forwarding can be enabled, which causes the IPv6 multicast forwarding to be based on both the source (if specified) and destination IPv6 address in the received join.

Enabling SG-based forwarding increases the MFIB usage if the source IPv6 address or higher 96 bits of the destination IPv6 address varies in the received joins compared to using MAC-based forwarding.

The following output shows a service with PIM snooping for IPv6 that has received joins for two multicast groups from different sources. As the forwarding mode is SG-based, there are two MFIB entries, one for each of the two groups.

```

*A:PE# show service id 1 pim-snooping group ipv6
=====
PIM Snooping Groups ipv6
=====
Group Address          Source Address          Type      Incoming
Intf                  Num
Oifs
-----
ff0e:db8:1000::1      2001:db8:1000::1      (S,G)    SAP:1/1/1      2
ff0e:db8:1001::1      2001:db8:1001::1      (S,G)    SAP:1/1/1      2
-----
Groups : 2
=====
*A:PE#

*A:PE# show service id 1 all | match "Mcast IPv6 scope"
Mcast IPv6 scope : sg-based
*A:PE#

*A:PE# show service id 1 mfib
=====
Multicast FIB, Service 1
=====
Source Address  Group Address          Port Id                      Svc Id  Fwd
Blk
-----
2001:db8:1000:* ff0e:db8:1000::1      sap:1/1/1                  Local   Fwd
                                sap:1/1/2                  Local   Fwd
2001:db8:1001:* ff0e:db8:1001::1      sap:1/1/1                  Local   Fwd
                                sap:1/1/2                  Local   Fwd
-----
Number of entries: 2
=====
*A:PE#

```

SG-based IPv6 multicast forwarding is supported when both plain PIM snooping and PIM proxy are supported.

SG-based forwarding is only supported on FP3- or higher-based line cards. It is supported in all services in which PIM snooping for IPv6 is supported, with the same restrictions.

It is not supported in the following services:

- PBB B-VPLS
- PBB I-VPLS
- Routed-VPLS (including with I-VPLS and BGP-EVPN)
- BGP-EVPN-MPLS (including PBB-EVPN)
- VPLS E-Tree
- Management VPLS

In any specific service, SG-based forwarding and MLD snooping are mutually exclusive. Consequently, MLD snooping uses MAC-based forwarding.

It is not supported in services with:

- subscriber management
- multicast VLAN Registration
- video interface

It is not supported on connected SR OS routers configured with **improved-assert**.

It is not supported with the following forms of default SAP:

- \*
- \*.null
- \*.\*.

### 3.2.20.5 PIM and IGMP/MLD snooping interaction

When both PIM snooping for IPv4 and IGMP snooping are enabled in the same VPLS service, multicast traffic is forwarded based on the combined multicast forwarding table. When PIM snooping is enabled, IGMP queries are forwarded but not snooped, consequently the IGMP querier needs to be seen either as a PIM neighbor in the VPLS service or the SAP toward it configured as an IGMP Mrouter port.

There is no interaction between PIM snooping for IPv6 and PIM snooping for IPv4/IGMP snooping when all are enabled within the same VPLS service. The configurations of PIM snooping for IPv6 and MLD snooping are mutually exclusive.

When PIM snooping is enabled within a VPLS service, all IP multicast traffic and flooded PIM messages (these include all PIM snooped messages when not in PIM proxy mode and PIM hellos when in PIM proxy mode) are sent to any SAP or SDP binding configured with an IGMP-snooping Mrouter port. This occurs even without IGMP-snooping enabled but is not supported in a BGP-VPLS or M-VPLS service.

### 3.2.20.6 Multichassis synchronization for Layer 2 snooping states

To achieve a faster failover in scenarios with redundant active/standby routers performing Layer 2 multicast snooping, it is possible to synchronize the snooping state from the active router to the standby router, so that if a failure occurs the standby router has the Layer 2 multicast snooped states and is able to forward the multicast traffic immediately. Without this capability, there would be a longer delay in re-establishing the multicast traffic path because it would wait for the Layer 2 states to be snooped.

Multichassis synchronization (MCS) is enabled per peer router and uses a **sync-tag**, which is configured on the objects requiring synchronization on both of the routers. This allows MCS to map the state of a set of objects on one router to a set of objects on the other router. Specifically, objects relating to a **sync-tag** on one router are backed up by, or are backing up, the objects using the same **sync-tag** on the other router (the state is synchronized from the active object on one router to its backup objects on the standby router).

The object type must be the same on both routers; otherwise, a mismatch error is reported. The same **sync-tag** value can be reused for multiple peer/object combinations, where each combination represents a different set of synchronized objects; however, a **sync-tag** cannot be configured on the same object to more than one peer.

The **sync-tag** is configured per port and can relate to a specific set of dot1q or QinQ VLANs on that port, as follows.

CLI syntax:

```
configure
  redundancy
    multi-chassis
      peer ip-address [create]
      sync
        port port-id [sync-tag sync-tag] [create]
        range encap-range sync-tag sync-tag
```

For IGMP snooping and PIM snooping for IPv4 to work correctly with MCS on QinQ ports using x.\* SAPs, one of the following must be true:

- MCS is configured with a **sync-tag** for the entire port.
- The IGMP snooping SAP and the MCS **sync-tag** must be provisioned with the same Q-tag values when using the range parameter.

### 3.2.20.6.1 IGMP snooping synchronization

MCS for IGMP snooping synchronizes the join/prune state information from IGMP messages received on the related port/VLANs corresponding to their associated **sync-tag**. It is enabled as follows.

CLI syntax:

```
configure
  redundancy
    multi-chassis
      peer ip-address [create]
      sync
        igmp-snooping
```

IGMP snooping synchronization is supported wherever IGMP snooping is supported (except in EVPN for VXLAN). See [IGMP snooping for VPLS](#) for more information. IGMP snooping synchronization is also only supported for the following active/standby redundancy mechanisms:

- MC-LAG
- MC-Ring
- Single-Active Multihoming (EVPN-MPLS and PBB-EVPN I-VPLS)
- Single-Active Multihoming (EVPN-MPLS VPRN and IES routed VPLS)



Configuring an Mrouter port under an object that has the synchronization of IGMP snooping states enabled is not recommended. The Mrouter port configuration adds a (\*,\*) entry into the MFIB, which causes all groups (and IGMP messages) to be sent out of the respective object. In addition, the **mrouter-port** command causes all IGMP messages on that object to be discarded. However, the (\*,\*) entry is not synchronized by MCS. Consequently, the Mrouter port could cause the two MCS peers to be forwarding different sets of multicast streams out of the related object when each is active.

### 3.2.20.6.2 MLD snooping synchronization

MCS for MLD snooping is not supported. The command is not blocked for backward-compatibility reasons but has no effect on the system if configured.

### 3.2.20.6.3 PIM snooping for IPv4 synchronization

MCS for PIM snooping for IPv4 synchronizes the neighbor information from PIM hellos and join/prune state information from PIM for IPv4 messages received on the related SAPs and spoke-SDPs corresponding to the **sync-tag** associated with the related ports and SDPs, respectively. Use the following CLI syntax to enable MCS for PIM snooping for IPv4 synchronization.

CLI syntax:

```
configure
  redundancy
    multi-chassis
      peer ip-address [create]
      sync
        pim-snooping [saps] [spoke-sdps]
```

Any PIM hello state information received over the MCS connection from the peer router takes precedence over locally snooped hello information. This ensures that any PIM hello messages received on the active router that are then flooded, for example through the network backbone, and received over a local SAP or SDP on the standby router are not inadvertently used in the standby router's VPLS service.

The synchronization of PIM snooping state is only supported for manually configured spoke-SDPs. It is not supported for spoke-SDPs configured within an endpoint.

When synchronizing the PIM state between two spoke-SDPs, if both spoke-SDPs go down, the PIM state is maintained on both until one becomes active to ensure that the PIM state is preserved when a spoke-SDP recovers.

Appropriate actions based on the expiration of PIM-related timers on the standby router are only taken after it has become the active peer for the related object (after a failover).

PIM snooping for IPv4 synchronization is supported wherever PIM snooping for IPv4 is supported, excluding the following services:

- BGP-VPLS
- VPLS E-Tree
- management VPLS

See [PIM snooping for VPLS](#) for more details.

PIM snooping for IPv4 synchronization is also only supported for the following active/standby redundancy mechanisms on dual-homed systems:

- MC-LAG
- BGP multihoming
- active/standby pseudowires
- single-active multihoming (EVPN-MPLS and PBB-EVPN I-VPLS)

Configuring an Mrouter port under an object that has the synchronization of PIM snooping for IPv4 states enabled is not recommended. The Mrouter port configuration adds a (\*,\*) entry into the MFIB, which causes all groups (and PIM messages) to be sent out of the respective object. In addition, the **mrouter-port** command causes all PIM messages on that object to be discarded. However, the (\*,\*) entry is not synchronized by MCS. Consequently, the Mrouter port could cause the two MCS peers to be forwarding different sets of multicast streams out of the related object when each is active.

### 3.2.20.7 VPLS multicast-aware high availability features

The following features are High Availability capable:

- Configuration redundancy (all the VPLS multicast-aware configurations can be synchronized to the standby CPM)
- Local snooping states as well as states distributed by LDP can be synchronized to the standby CPM.
- Operational states can also be synchronized; for example, the operational state of PIM proxy.

### 3.2.21 RSVP and LDP P2MP LSP for forwarding VPLS/B-VPLS BUM and IP multicast packets

This feature enables the use of a P2MP LSP as the default tree for forwarding Broadcast, Unicast unknown, and Multicast (BUM) packets of a VPLS or B-VPLS instance. The P2MP LSP is referred to in this case as the Inclusive Provider Multicast Service Interface (I-PMSI).

When enabled, this feature relies on BGP autodiscovery (BGP-AD) or BGP-VPLS to discover the PE nodes participating in a specified VPLS/B-VPLS instance. The BGP route contains the information required to signal both the point-to-point (P2P) PWs used for forwarding unicast known Ethernet frames and the RSVP P2MP LSP used to forward the BUM frames. The root node signals the P2MP LSP based on an LSP template associated with the I-PMSI at configuration time. The leaf node automatically joins the P2MP LSP that matches the I-PMSI tunnel information discovered via BGP.

If IGMP or PIM snooping are configured on the VPLS instance, multicast packets matching an L2 multicast Forwarding Information Base (FIB) record are also forwarded over the P2MP LSP.

The user enables the use of an RSVP P2MP LSP as the I-PMSI for forwarding Ethernet BUM and IP multicast packets in a VPLS/B-VPLS instance using the following context:

```
config>service>vpls [b-vpls]>provider-tunnel>inclusive>rsvp>lsp-template p2mp-lsp-template-name
```

The user enables the use of an LDP P2MP LSP as the I-PMSI for forwarding Ethernet BUM and IP multicast packets in a VPLS instance using the following context:

```
config>service>vpls [b-vpls]>provider-tunnel>inclusive>mldp
```

After the user performs a **no shutdown** under the context of the inclusive node and the expiration of a delay timer, BUM packets are forwarded over an automatically signaled mLDP P2MP LSP or over an automatically signaled instance of the RSVP P2MP LSP specified in the LSP template.

The user can specify that the node is both root and leaf in the VPLS instance:

```
config>service>vpls [b-vpls]>provider-tunnel>inclusive>root-and-leaf
```

The **root-and-leaf** command is required; otherwise, this node behaves as a leaf-only node by default. When the node is leaf only for the I-PMSI of type P2MP RSVP LSP, no PMSI Tunnel Attribute is included in BGP-AD route update messages and, therefore, no RSVP P2MP LSP is signaled, but the node can join an RSVP P2MP LSP rooted at other PE nodes participating in this VPLS/B-VPLS service. The user must still configure an LSP template even if the node is a leaf only. For the I-PMSI of type mLDP, the leaf-only node joins I-PMSI rooted at other nodes it discovered but does not include a PMSI Tunnel Attribute in BGP route update messages. This way, a leaf-only node forwards packets to other nodes in the VPLS/B-VPLS using the point-to-point spoke-SDPs.

BGP-AD (or BGP-VPLS) must have been enabled in this VPLS/B-VPLS instance or the execution of the **no shutdown** command under the context of the inclusive node is failed and the I-PMSI does not come up.

Any change to the parameters of the I-PMSI, such as disabling the P2MP LSP type or changing the LSP template, requires that the inclusive node be first shut down. The LSP template is configured in MPLS.

If the P2MP LSP instance goes down, VPLS/B-VPLS immediately reverts the forwarding of BUM packets to the P2P PWs. However, the user can restore at any time the forwarding of BUM packets over the P2P PWs by performing a **shutdown** under the context of the inclusive node.

This feature is supported with VPLS, H-VPLS, B-VPLS, and BGP-VPLS. It is not supported with I-VPLS and R-VPLS.

### 3.2.22 MPLS EL and hash label

The router supports the MPLS EL (RFC 6790) and the FAT label, known as the hash label (RFC 6391). These labels allow LSR nodes in a network to load-balance labeled packets in a more granular way than by hashing on the standard label stack. See the *7705 SAR Gen 2 MPLS Guide* for more information.

The EL is supported for LDP VPLS and BGP-AD VPLS, as well as Epipe spoke-SDP termination on VPLS services. To configure insertion of the EL on a spoke-SDP or mesh-SDP of a specific service, use the **entropy-label** command in the **spoke-sdp**, **mesh-sdp**, or **pw-template** contexts. The EL is only inserted if the far end of the MPLS tunnel is also entropy-label-capable.

The hash label is supported for LDP VPLS, BGP-AD, and BGP-VPLS VPLS, as well as Epipe services. Use the following commands to configure the hash label.

```
configure service epipe spoke-sdp hash-label
configure service pw-template hash-label
configure service vpls mesh-sdp hash-label
configure service vpls spoke-sdp hash-label
```

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

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

## 3.3 Routed VPLS and I-VPLS

This section provides information about Routed VPLS (R-VPLS) and I-VPLS.



**Note:** I-VPLS is not supported on the 7705 SAR Gen 2. I-VPLS information is included in this guide for reference only.

### 3.3.1 IES or VPRN IP interface binding

For the remainder of this section, R-VPLS and Routed I-VPLS both are described as a VPLS service, and differences are pointed out where applicable.



**Note:** I-VPLS is not supported on the 7705 SAR Gen 2. I-VPLS information is included in this guide for reference only.

A standard IP interface within an existing IES or VPRN service context may be bound to a service name. Subscriber and group IP interfaces are not allowed to bind to a VPLS or I-VPLS service context or I-VPLS. A VPLS service only supports binding for a single IP interface.

While an IP interface may only be bound to a single VPLS service, the routing context containing the IP interface (IES or VPRN) may have other IP interfaces bound to other VPLS service contexts of the same type (all VPLS or all I-VPLS). That is, R-VPLS allows the binding of IP interfaces in IES or VPRN services to be bound to VPLS services and Routed I-VPLS allows of IP interfaces in IES or VPRN services to be bound to I-VPLS services.

#### 3.3.1.1 Assigning a service name to a VPLS service

When a service name is applied to any service context, the name and service ID association is registered with the system. Assigning a service name to more than one service ID is not allowed.

There is special consideration for a service name that is assigned to a VPLS service on which the **config service vpls allow-ip-int-bind** command is enabled. In such cases the system scans the existing IES and VPRN services for an IP interface that is bound to the specified service name. If an IP interface is found, the IP interface is attached to the VPLS service associated with the name. Only one interface can be bound to the specified name.

If the **allow-ip-int-bind** command is not enabled on the VPLS service, the system does not attempt to resolve the VPLS service name to an IP interface. As soon as the **allow-ip-int-bind** flag is configured on the VPLS, the corresponding IP interface is bound and becomes operationally up. Toggling the **shutdown** and **no shutdown** command is not required.

If an IP interface is not currently bound to the service name used by the VPLS service, no action is taken at the time of the service name assignment.

#### 3.3.1.2 Service binding requirements

If the defined service ID is created on the system, the system checks to ensure that the service type is VPLS. If the service type is not VPLS or I-VPLS, service creation is not allowed and the service ID remains undefined within the system.

If the created service type is VPLS, the IP interface is eligible to enter the operationally up state.

### 3.3.1.3 Bound service name assignment

If a bound service name is assigned to a service within the system, the system first checks to ensure the service type is VPLS or I-VPLS. Secondly, the system ensures that the service is not already bound to another IP interface through the service ID. If the service type is not VPLS or I-VPLS or the service is already bound to another IP interface through the service ID, the service name assignment fails.

If a single VPLS service ID and service name is assigned to two separate IP interfaces, the VPLS service is not allowed to enter the operationally up state.

### 3.3.1.4 Binding a service name to an IP interface

An IP interface within an IES or VPRN service context may be bound to a service name at any time. Only one interface can be bound to a service.

When an IP interface is bound to a service name and the IP interface is administratively up, the system scans for a VPLS service context using the name and performs the following actions.

- If the name is not currently in use by a service, the IP interface is placed in an operationally down state: non-existent service name or inappropriate service type.
- If the name is currently in use by a non-VPLS service or the wrong type of VPLS service, the IP interface is placed in the operationally down state: non-existent service name or inappropriate service type.
- If the name is currently in use by a VPLS service without the **allow-ip-int-bind** command enabled, the IP interface is placed in the operationally down state: VPLS service **allow-ip-int-bind** flag not set. Toggling the **shutdown/no shutdown** command is not required.
- If the name is currently in use by a valid VPLS service and the **allow-ip-int-bind** command is enabled, the IP interface is eligible to be placed in the operationally up state, if other operational criteria is met.

### 3.3.1.5 Bound service deletion or service name removal

If a VPLS service is deleted while bound to an IP interface, the IP interface enters the "Down: Non-existent svc-ID" operational state. If the IP interface was bound to the VPLS service name, the IP interface enters the "Down: Non-existent svc-name" operational state. A console warning is not generated.

If the created service type is VPLS, the IP interface is eligible to enter the operationally up state.

### 3.3.1.6 IP interface attached VPLS service constraints

After a VPLS service has been bound to an IP interface through its service name, the assigned service name cannot be removed or changed, unless the IP interface is first unbound from the VPLS service name.

A VPLS service that is currently attached to an IP interface cannot be deleted from the system, unless the IP interface is unbound from the VPLS service name.

The **allow-ip-int-bind** flag in an IP interface-attached VPLS service cannot be reset, unless the IP interface is first unbound from the VPLS service name.

### 3.3.1.7 IP interface and VPLS operational state coordination

When the IP interface is successfully attached to a VPLS service, the operational state of the IP interface is dependent upon the operational state of the VPLS service.

The VPLS service remains down until at least one virtual port (SAP, spoke-SDP, or mesh SDP) is operational.

### 3.3.2 IP interface MTU and fragmentation

The VPLS service is affected by two MTU values: port MTUs and the VPLS service MTU. The MTU on each physical port defines the largest Layer 2 packet (including all DLC headers) that may be transmitted out of a port. The VPLS has a service level MTU that defines the largest packet supported by the service. This MTU does not include the local encapsulation overhead for each port (QinQ, dot1q, TopQ, or SDP service delineation fields and headers) but does include the remainder of the packet.

As virtual ports are created in the system, the virtual port cannot become operational unless the configured port MTU minus the virtual port service delineation overhead is greater than or equal to the configured VPLS service MTU. Therefore, an operational virtual port is ensured to support the largest packet traversing the VPLS service. The service delineation overhead on each Layer 2 packet is removed before forwarding into a VPLS service. VPLS services do not support fragmentation and must discard any Layer 2 packet larger than the service MTU after the service delineation overhead is removed.

When an IP interface is associated with a VPLS service, the IP-MTU is based on either the administrative value configured for the IP interface or an operational value derived from VPLS service MTU. The operational IP-MTU cannot be greater than the VPLS service MTU minus 14 bytes.

- If the configured (administrative) IP-MTU is configured for a value greater than the normalized IP-MTU, based on the VPLS service-MTU, the operational IP-MTU is reset to equal the normalized IP-MTU value (VPLS service MTU – 14 bytes).
- If the configured (administrative) IP-MTU is configured for a value less than or equal to the normalized IP-MTU, based on the VPLS service-MTU, the operational IP-MTU is set to equal the configured (administrative) IP-MTU value.

#### 3.3.2.1 Unicast IP routing into a VPLS service

The VPLS service MTU and the IP interface MTU parameters may be changed at any time.

### 3.3.3 ARP and VPLS FDB interactions

Two address-oriented table entries are used when routing into a VPLS service.

The first address table entry that affects VPLS routed packets is an ARP entry on the routing side, which determines the destination MAC address used by an IP next-hop. In the case where the destination IP address in the routed packet is a host on the local subnet represented by the VPLS instance, the destination IP address is used as the next-hop IP address in the ARP cache lookup. If the destination IP address is in a remote subnet that is reached by another router attached to the VPLS service, the routing lookup returns the local IP address on the VPLS service of the remote router. If the next-hop is not currently in the ARP cache, the system generates an ARP request to determine the destination MAC address associated with the next-hop IP address.

IP routing to all destination hosts associated with the next-hop IP address stops until the ARP cache is populated with an entry for the next-hop. The ARP cache may be populated with a static ARP entry for the next-hop IP address. While dynamically populated ARP entries age out according to the ARP aging timer, static ARP entries never age out.

The second address table entry that affects VPLS routed packets is the MAC destination lookup in the VPLS service context. The MAC associated with the ARP table entry for the IP next-hop may or may not currently be populated in the VPLS Layer 2 FDB table. While the destination MAC is unknown (not populated in the VPLS FDB), the system floods all packets destined for that MAC (routed or bridged) to all virtual ports within the VPLS service context. When the MAC is known (populated in the VPLS FDB), all packets destined for the MAC (routed or bridged) are targeted to the specific virtual port where the MAC has been learned.

As with ARP entries, static MAC entries may be created in the VPLS FDB. Dynamically learned MAC addresses are allowed to age out or be flushed from the VPLS FDB while static MAC entries always remain associated with a specific virtual port. Dynamic MACs may also be relearned on another VPLS virtual port than the current virtual port in the FDB. In this case, the system automatically moves the MAC FDB entry to the new VPLS virtual port.

The MAC address associated with the R-VPLS IP interface is protected within its VPLS service such that frames received with this MAC address as the source address are discarded. VRRP MAC addresses are not protected in this way.

### 3.3.3.1 R-VPLS specific ARP cache behavior

In typical routing behavior, the system uses the IP route table to select the egress interface, and then at the egress forwarding engine, an ARP entry is used to forward the packet to the appropriate Ethernet MAC. With R-VPLS, the egress IP interface may be represented by a multiple egress forwarding engine (wherever the VPLS service virtual ports exist).

To optimize routing performance, the ingress forwarding engine processing is augmented to perform an ingress ARP lookup to resolve which VPLS MAC address the IP frame must be routed toward. This MAC address may be currently known or unknown within the VPLS FDB. If the MAC is unknown, the ingress forwarding engine floods the packet to all egress forwarding engines where the VPLS service exists. If the MAC is known on a virtual port, the ingress forwarding engine forwards the packet to the correct egress forwarding engine. The following table describes how the ARP cache and MAC FDB entry states interact at ingress.

*Table 9: Ingress routed to VPLS next-hop behavior*

| Next-hop ARP cache entry  | Next-hop MAC FDB entry | Ingress behavior  |
|---------------------------|------------------------|---|
| ARP Cache Miss (No Entry) | Known or Unknown       | Flood to all egress forwarding engines associated with the VPLS or I-VPLS context.                                  |
|                           | Unknown                | Flood to all egress forwarding engines associated with the VPLS or I-VPLS context.                                  |
|                           | Unknown                | Flood to all egress forwarding engines associated with the VPLS for forwarding to all VPLS or I-VPLS virtual ports. |



The following table describes the corresponding egress behavior.

Table 10: Egress R-VPLS next-hop behavior

| Next-hop ARP Cache entry  | Next-hop MAC FDB entry | Egress behavior   |
|---------------------------|------------------------|---|
| ARP Cache Miss (No Entry) | Known                  | No ARP entry. The MAC address is unknown and the ARP request is flooded out of all virtual ports of the VPLS or I-VPLS instance.  |
|                           | Unknown                | Request control engine processing the ARP request to transmit out of all virtual ports associated with the VPLS or I-VPLS service. Only the first egress forwarding engine ARP processing request triggers an egress ARP request. |
| ARP Cache Hit             | Known                  | Forward out of specific egress VPLS or I-VPLS virtual ports where MAC has been learned.   |
|                           | Unknown                | Flood to all egress VPLS or I-VPLS virtual ports on forwarding engine.  |

### 3.3.4 The allow-ip-int-bind VPLS flag

The **allow-ip-int-bind** flag on a VPLS service context is used to inform the system that the VPLS service is enabled for routing support. The system uses the setting of the flag as a key to determine the types of ports and forwarding planes the VPLS service may span.

The system also uses the flag state to define VPLS features that are configurable on the VPLS service and to prevent enabling a feature that is not supported when routing support is enabled.

#### 3.3.4.1 R-VPLS SAPs only supported on standard Ethernet ports

The **allow-ip-int-bind** flag is set (routing support enabled) on a VPLS/I-VPLS service. SAPs within the service can be created on standard Ethernet, and CCAG ports. POS is not supported.

#### 3.3.4.2 LAG port membership constraints

If a LAG has a non-supported port type as a member, a SAP for the routing-enabled VPLS service cannot be created on the LAG. When one or more routing enabled VPLS SAPs are associated with a LAG, a non-supported Ethernet port type cannot be added to the LAG membership.

#### 3.3.4.3 R-VPLS feature restrictions

When the **allow-ip-int-bind** flag is set on a VPLS service, the following restrictions apply. The flag also cannot be enabled while any of these features are applied to the VPLS service:

- SDPs used in spoke or mesh SDP bindings cannot be configured as GRE.



- The VPLS service type cannot be B-VPLS or M-VPLS.
- MVR from R-VPLS and to another SAP is not supported.
- Enhanced and Basic Subscriber Management (BSM) features cannot be enabled.
- Network domain on SDP bindings cannot be enabled.
- Per-service hashing is not supported.
- BGP-VPLS is not supported.
- Ingress queuing for split horizon groups is not supported.
- Multiple virtual routers are not supported.

### 3.3.5 IPv4 and IPv6 multicast routing support

IPv4 and IPv6 multicast routing is supported in a R-VPLS service through its IP interface when the source of the multicast stream is on one side of its IP interface and the receivers are on either side of the IP interface. For example, the source for multicast stream G1 could be on the IP side sending to receivers on both other regular IP interfaces and the VPLS of the R-VPLS service, while the source for group G2 could be on the VPLS side sending to receivers on both the VPLS and IP side of the R-VPLS service.

IPv4 and IPv6 multicast routing is not supported with Multicast VLAN Registration functions or the configuration of a video interface within the associated VPLS service. It is also not supported in a routed I-VPLS service, or for IPv6 multicast in BGP EVPN-MPLS routed VPLS services. Forwarding IPv4 or IPv6 multicast traffic from the R-VPLS IP interface into its VPLS service on a P2MP LSP is not supported.

The IP interface of a R-VPLS supports the configuration of both PIM and IGMP for IPv4 multicast and for both PIM and MLD for IPv6 multicast.

To forward IPv4/IPv6 multicast traffic from the VPLS side of the R-VPLS service to the IP side, the **forward-ipv4-multicast-to-ip-int** and/or **forward-ipv6-multicast-to-ip-int** commands must be configured as follows:

```
configure
  service
    vpls <service-id>
      allow-ip-int-bind
        forward-ipv4-multicast-to-ip-int
        forward-ipv6-multicast-to-ip-int
      exit
    exit
  exit
exit
```

Enabling IGMP snooping or MLD snooping in the VPLS service is optional, where supported. If IGMP/MLD snooping is enabled, IGMP/MLD must be enabled on the R-VPLS IP interface in order for multicast traffic to be sent into, or received from, the VPLS service. IPv6 multicast uses MAC-based forwarding; see [MAC-based IPv6 multicast forwarding](#) for more information.

If both IGMP/MLD and PIM for IPv4/IPv6 are configured on the R-VPLS IP interface in a redundant PE topology, the associated IP interface on one of the PEs must be configured as both the PIM designated router and the IGMP/MLD querier. This ensures that the multicast traffic is sent into the VPLS service, as IGMP/MLD joins are only propagated to the IP interface if it is the IGMP/MLD querier. An alternative to this is to configure the R-VPLS IP interface in the VPLS service as an Mrouter port, as follows:

```
configure
```

```

service
  vpls <service-id>
    allow-ip-int-bind
    igmp-snooping
      mrouter-port
    mld-snooping
      mrouter-port
  exit
exit
exit
exit

```

This configuration achieves a faster failover in scenarios with redundant routers where multicast traffic is sent to systems on the VPLS side of their R-VPLS services and IGMP/MLD snooping is enabled in the VPLS service. If the active router fails, the remaining router does not have to wait until it sends an IGMP/MLD query into the VPLS service before it starts receiving IGMP/MLD joins and starts sending the multicast traffic into the VPLS service. When the Mrouter port is configured as above, all IGMP/MLD joins (and multicast traffic) are sent to the VPLS service IP interface.

IGMP/MLD snooping should only be enabled when systems, as opposed to PIM routers, are connected to the VPLS service. If IGMP/MLD snooping is enabled when the VPLS service is used for transit traffic for connected PIM routers, the IGMP/MLD snooping would prevent multicast traffic being forwarded between the PIM routers (as PIM snooping is not supported). A workaround would be to configure the VPLS SAPs and spoke-SDPs (and the R-VPLS IP interface) to which the PIM routers are connected as Mrouter ports.

If IMPM is enabled on an FP on which there is a R-VPLS service with **forward-ipv4-multicast-to-ip-int** or **forward-ipv6-multicast-to-ip-int** configured, the IPv4/IPv6 multicast traffic received in the VPLS service that is forwarded through the IP interface is IMPM-managed even without IGMP/MLD snooping being enabled. This does not apply to traffic that is only flooded within the VPLS service.

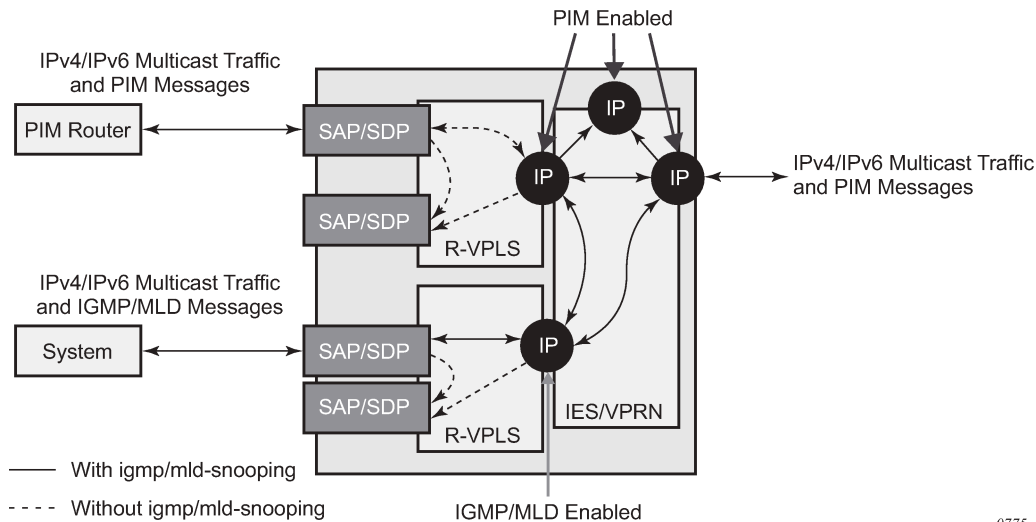
When IPv4/IPv6 multicast traffic is forwarded from a VPLS SAP through the R-VPLS IP interface, the packet count is doubled in the following statistics to represent both the VPLS and IP replication (this reflects the capacity used for this traffic on the ingress queues, which is subject to any configured rates and IMPM capacity management):

- Offered queue statistics
- IMPM managed statistics
- IMPM unmanaged statistics for policed traffic

IPv4 or IPv6 multicast traffic entering the IP side of the R-VPLS service and exiting over a multiport LAG on the VPLS side of the service is sent on a single link of that egress LAG, specifically the link used for all broadcast, unknown, and multicast traffic.

An example of IPv4/IPv6 multicast in a R-VPLS service is shown in [Figure 47: IPv4/IPv6 multicast with a router VPLS service](#). There are two R-VPLS IP interfaces connected to an IES service with the upper interface connected to a VPLS service in which there is a PIM router and the lower interface connected to a VPLS service in which there is a system using IGMP/MLD.

Figure 47: IPv4/IPv6 multicast with a router VPLS service



0775

The IPv4/IPv6 multicast traffic entering the IES/VRPN service through the regular IP interface is replicated to both the other regular IP interface and the two R-VPLS interfaces if PIM/IGMP/MLD joins have been received on the respective IP interfaces. This traffic is flooded into both VPLS services unless IGMP/MLD snooping is enabled in the lower VPLS service, in which case it is only sent to the system originating the IGMP/MLD join.

The IPv4/IPv6 multicast traffic entering the upper VPLS service from the connected PIM router is flooded in that VPLS service and, if related joins have been received, forwarded to the regular IP interfaces in the IES/VRPN. It is also be forwarded to the lower VPLS service if an IGMP/MLD join is received on its IP interface, and is flooded in that VPLS service unless IGMP/MLD snooping is enabled.

The IPv4/IPv6 multicast traffic entering the lower VPLS service from the connected system is flooded in that VPLS service, unless IGMP/MLD snooping is enabled, in which case it is only forwarded to SAPs, spoke-SDPs, or the R-VPLS IP interface if joins have been received on them. It is forwarded to the regular IP interfaces in the IES/VRPN service if related joins have been received on those interfaces, and it is also forwarded to the upper VPLS service if a PIM IPv4/IPv6 join is received on its IP interface, this being flooded in that VPLS service.

### 3.3.6 BGP-AD for R-VPLS support

BGP autodiscovery (BGP-AD) for R-VPLS is supported. BGP-AD for LDP VPLS is an already supported framework for automatically discovering the endpoints of a Layer 2 VPN offering an operational model similar to that of an IP VPN.

### 3.3.7 R-VPLS restrictions

This section describes restrictions that apply to R-VPLS.

### 3.3.7.1 VPLS SAP ingress IP filter override

When an IP Interface is attached to a VPLS or an I-VPLS service context, the VPLS SAP provisioned IP or IPv6 filter for ingress routed packets may be optionally overridden to provide special ingress filtering for routed packets. The filter override is defined on the IP interface bound to the VPLS service name. A separate override filter may be specified for IPv4 and IPv6 packet types. The filter override applies to unicast routed packets destined to the routed VPLS interface MAC address.

If a filter for a specified packet type (IPv4 or IPv6) is not overridden, the SAP specified filter is applied to the packet (if defined).

### 3.3.7.2 IP interface defined egress QoS reclassification

The SAP egress QoS policy defined forwarding class and profile reclassification rules are not applied to egress routed packets. To allow for egress reclassification, a SAP egress QoS policy ID may be optionally defined on the IP interface that is applied to routed packets that egress the SAPs on the VPLS or I-VPLS service associated with the IP interface. Both unicast directed and MAC unknown flooded traffic apply to this rule. Only the reclassification portion of the QoS policy is applied, which includes IP precedence or DSCP classification rules and any defined IP match criteria and their associated actions.

The policers and queues defined within the QoS policy applied to the IP interface are not created on the egress SAPs of the VPLS service. Instead, the QoS policy applied to the egress SAPs defines the egress policers and queues used by both routed and non-routed egress packets. The forwarding class mappings defined in the egress SAP's QoS policy also defines which policer or queue handles each forwarding class for both routed and non-routed packets.

### 3.3.7.3 Interface statistics collection

The 7705 SAR Gen 2 does not support statistics collection for R-VPLS interfaces.

### 3.3.7.4 Remarking for VPLS and routed packets

The remarking of packets to and from an IP interface in an R-VPLS service corresponds to that supported on IP interface, even though the packets ingress or egress a SAP in the VPLS service bound to the IP service. Specifically, this results in the ability to remark the DSCP/prec for these packets.

Packets that ingress and egress SAPs in the VPLS service (not routed through the IP interface) support the regular VPLS QoS and, therefore, the DSCP/prec cannot be remarked.

### 3.3.7.5 IPv4 multicast routing

When using IPv4 multicast routing, the following are not supported:

- The multicast VLAN registration functions within the associated VPLS service.
- The configuration of a video ISA within the associated VPLS service.
- The configuration of MFIB-allowed MDA destinations under spoke/mesh SDPs within the associated VPLS service.
- The IPv4 multicast routing is not supported in Routed I-VPLS.

- The RFC 6037 multicast tunnel termination (including when the system is a bud node) is not supported on the R-VPLS IP interface for multicast traffic received in the VPLS service.
- Forwarding of multicast traffic from the VPLS side of the service to the IP interface side of the service is not supported for R-VPLS services that have egress VXLAN VTEPs configured.

### 3.3.7.6 R-VPLS supported routing-related protocols

The following protocols are supported on IP interfaces bound to a VPLS service:

- BGP
- OSPF
- ISIS
- PIM
- IGMP
- BFD
- VRRP
- ARP
- DHCP Relay

### 3.3.7.7 Spanning Tree and split horizon

A R-VPLS context supports all spanning tree and split horizon capabilities that a non-R-VPLS service supports.

## 3.4 VPLS service considerations

This section describes the service features and any special capabilities or considerations as they relate to VPLS services.

### 3.4.1 SAP encapsulations

VPLS services are designed to carry Ethernet frame payloads, so the services can provide connectivity between any SAPs and SDPs that pass Ethernet frames. The following SAP encapsulations are supported on VPLS services:

- Ethernet null
- Ethernet dot1q
- Ethernet QinQ

### 3.4.2 VLAN processing

The following SAP encapsulation definitions on Ethernet ingress ports define which VLAN tags are used to determine the service to which the packet belongs:

- **null encapsulation defined on ingress**

Any VLAN tags are ignored and the packet goes to a default service for the SAP.

- **dot1q encapsulation defined on ingress**

Only the first VLAN tag is considered.

- **QinQ encapsulation defined on ingress**

Both labels are considered. The SAP can be defined with a wildcard for the inner label (for example, "100:100.\*"). In this situation, all packets with an outer label of 100 are treated as belonging to the SAP. If, on the same physical link, there is also a SAP defined with a QinQ encapsulation of 100:100.1, traffic with 100:1 goes to that SAP and all other traffic with 100 as the first label goes to the SAP with the 100:100.\* definition.

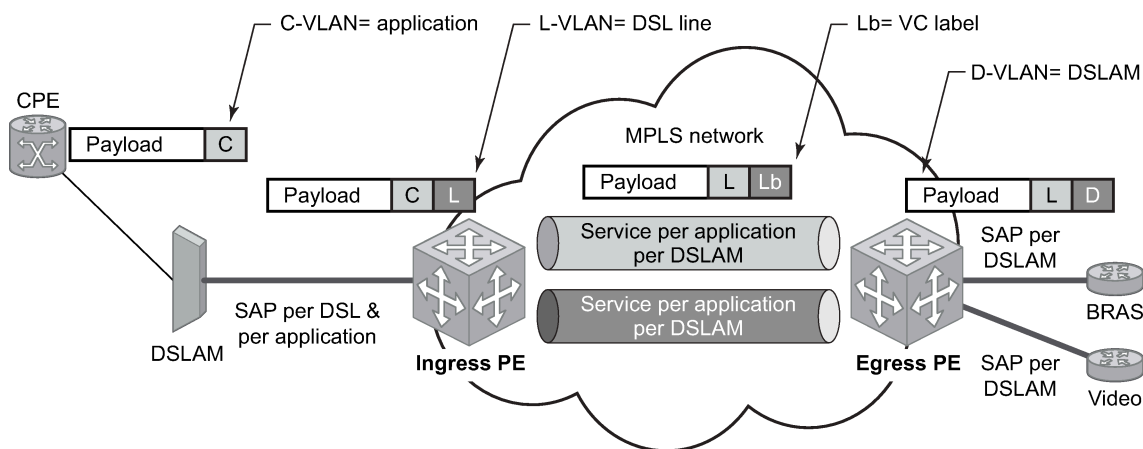
In the last two situations above, traffic encapsulated with tags for which there is no definition are discarded.

### 3.4.3 Ingress VLAN swapping

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

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

Figure 48: Ingress VLAN swapping



Fig\_36

Figure 48: Ingress VLAN swapping describes the network where, at user access side (DSLAM facing SAPs), every subscriber is represented by several QinQ SAPs with inner-tag encoding service and outer-tag encoding subscriber (DSL line). The aggregation side (BRAS or PE-facing SAPs) is represented by a

DSL line number (inner VLAN tag) and DSLAM (outer VLAN tag). The effective operation on the VLAN tag is to drop the inner tag at the access side and push another tag at the aggregation side.

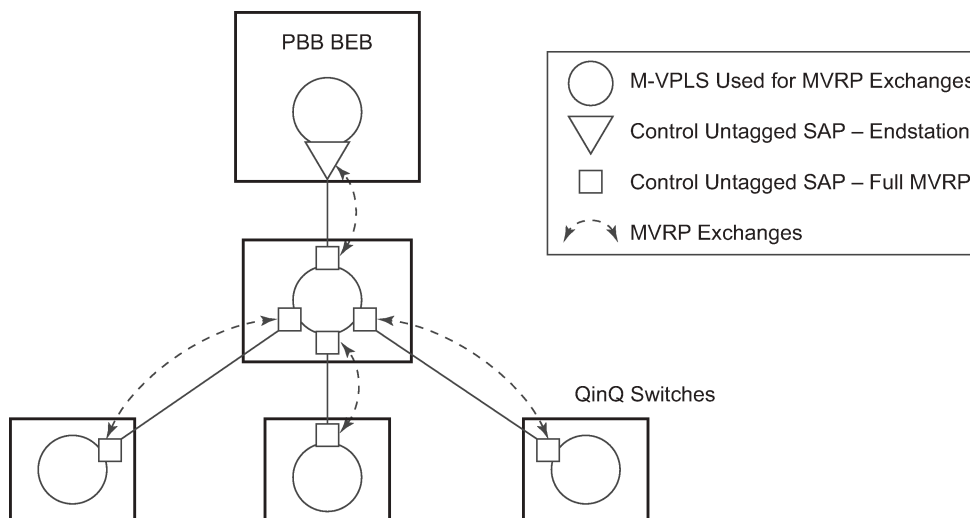
### 3.4.4 Service auto-discovery using MVRP

IEEE 802.1ak Multiple VLAN Registration Protocol (MVRP) is used to advertise throughout a native Ethernet switching domain one or multiple VLAN IDs to automatically build native Ethernet connectivity for multiple services. These VLAN IDs can be either Customer VLAN IDs (CVID) in an enterprise switching environment, Stacked VLAN IDs (SVID) in a Provider Bridging, QinQ Domain (see the *IEEE 802.1ad*), or Backbone VLAN IDs (BVID) in a Provider Backbone Bridging (PBB) domain (see the *IEEE 802.1ah*).

The initial focus of Nokia MVRP implementation is a Service Provider QinQ domain with or without a PBB core. The QinQ access into a PBB core example is used throughout this section to describe the MVRP implementation. With the exception of end-station components, a similar solution can be used to address a QinQ only or enterprise environments.

The components involved in the MVRP control plane are shown in [Figure 49: Infrastructure for MVRP exchanges](#).

Figure 49: Infrastructure for MVRP exchanges



OSSG492

All the devices involved are QinQ switches with the exception of the PBB BEB which delimits the QinQ domain and ensures the transition to the PBB core. The circles represent Management VPLS instances interconnected by SAPs to build a native Ethernet switching domain used for MVRP control plane exchanges.

The following high-level steps are involved in auto-discovery of VLAN connectivity in a native Ethernet domain using MVRP:

1. Configure the MVRP infrastructure

- This requires the configuration of a Management VPLS (M-VPLS) context.
- MSTP may be used in M-VPLS to provide the loop-free topology over which the MVRP exchanges take place.

2. Instantiate related VLAN FDB, trunks in the MVRP, M-VPLS scope

- The VLAN FDBs (VPLS instances) and associated trunks (SAPs) are instantiated in the same Ethernet switches and on the same “trunk ports” as the M-VPLS.
- There is no need to instantiate data VPLS instances in the BEB. I-VPLS instances and related downward-facing SAPs are provisioned manually because the ISID-to-VLAN association must be configured.

### 3. MVRP activation of service connectivity

When the first two customer UNI or PBB end-station SAPs, or both, are configured on different Ethernet switches in a specific service context, the MVRP exchanges activate service connectivity.

#### 3.4.4.1 Configure the MVRP infrastructure using an M-VPLS context

The following provisioning steps apply.

1. Configure the M-VPLS instances in the switches that participate in MVRP control plane.
2. Configure under the M-VPLS the untagged SAPs to be used for MVRP exchanges; only dot1q or QinQ ports are accepted for MVRP enabled M-VPLS.
3. Configure the MVRP parameters at M-VPLS instance or SAP level.

#### 3.4.4.2 Instantiate related VLAN FDBs and trunks in MVRP scope

This requires the configuration in the M-VPLS, under `vpls-group`, of the following attributes: VLAN ranges, `vpls-template` and `vpls-sap-template` bindings. As soon as the VPLS group is enabled, the configured attributes are used to auto-instantiate, on a per-VLAN basis, a VPLS FDB and related SAPs in the switches and on the “trunk ports” specified in the M-VPLS context. The trunk ports are ports associated with an M-VPLS SAP not configured as an end-station.

The following procedure is used:

- The `vpls-template` binding is used to instantiate the VPLS instance where the service ID is derived from the VLAN value as per service-range configuration.
- The `vpls-sap-template` binding is used to create dot1q SAPs by deriving from the VLAN value the service delimiter as per service-range configuration.

The above procedure may be used outside of the MVRP context to pre-provision a large number of VPLS contexts that share the same infrastructure and attributes.

The MVRP control of the auto-instantiated services can be enabled using the **`mvrp-control`** command under the **`vpls-group`**.

- If **`mvrp-control`** is disabled, the auto-created VPLS instances and related SAPs are ready to forward.
- If **`mvrp-control`** is enabled, the auto-created VPLS instances are instantiated initially with an empty flooding domain. According to the operator configuration the MVRP exchanges gradually enable service connectivity – between configured SAPs in the data VPLS context.

This also provides protection against operational mistakes that may generate flooding throughout the auto-instantiated VLAN FDBs.

From an MVRP perspective, these SAPs can be either “full MVRP” or “end-station” interfaces.

A full MVRP interface is a full participant in the local M-VPLS scope as described below.



- VLAN attributes received in an MVRP registration on this MVRP interface are declared on all the other full MVRP SAPs in the control VPLS.
- VLAN attributes received in an MVRP registration on other full MVRP interfaces in the local M-VPLS context are declared on this MVRP interface.

In an MVRP end-station interface, the attributes registered on that interface have local significance, as described below.

- VLAN attributes received in an MVRP registration on this interface are not declared on any other MVRP SAPs in the control VPLS. The attributes are registered only on the local port.
- Only locally active VLAN attributes are declared on the end-station interface; VLAN attributes registered on any other MVRP interfaces are not declared on end-station interfaces.
- Also defining an M-VPLS SAP as an end-station does not instantiate any objects on the local switch; the command is used just to define which SAP needs to be monitored by MVRP to declare the related VLAN value.

The following example describes the M-VPLS configuration required to auto-instantiate the VLAN FDBs and related trunks in non-PBB switches.

```
mrp
  - no shutdown
  - mvrp
    - shutdown
  - mvrp
    - no shutdown
sap 1/1/1:0
  - mvrp mvrp
    - no shutdown
sap 2/1/2:0
  - mvrp mvrp
    - no shutdown
sap 3/1/10:0
  - mvrp mvrp
    - no shutdown
vpls-group 1
  - service-range 100-2000
  - vpls-template-binding Autovpls1
  - sap-template-binding Autosap1
    - mvrp-control
  - no shutdown
```

A similar M-VPLS configuration may be used to auto-instantiate the VLAN FDBs and related trunks in PBB switches. The vpls-group command is replaced by the end-station command under the downward-facing SAPs as in the following example.

```
config>service>vpls control-mvrp m-vpls create customer 1
  - [...]
  - sap 1/1/1:0
    - mvrp mvrp
      - endstation-vid-group 1 vlan-id 100-2000
      - no shutdown
```

### 3.4.4.3 MVRP activation of service connectivity

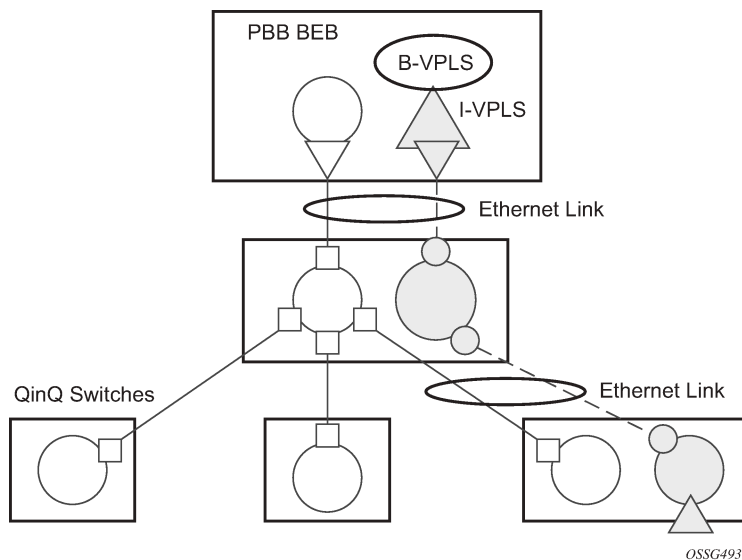
As new Ethernet services are activated, UNI SAPs need to be configured and associated with the VLAN IDs (VPLS instances) auto-created using the procedures described in the previous sections. These UNI

SAPs may be located in the same VLAN domain or over a PBB backbone. When UNI SAPs are located in different VLAN domains, an intermediate service translation point must be used at the PBB BEB, which maps the local VLAN ID through an I-VPLS SAP to a PBB ISID. This BEB SAP is playing the role of an end-station from an MVRP perspective for the local VLAN domain.

This section discusses how MVRP is used to activate service connectivity between a BEB SAP and a UNI SAP located on one of the switches in the local domain. A similar procedure is used in the case of UNI SAPs configured on two switches located in the same access domain. No end-station configuration is required on the PBB BEB if all the UNI SAPs in a service are located in the same VLAN domain.

The service connectivity instantiation through MVRP is shown in [Figure 50: Service instantiation with MVRP - QinQ to PBB example](#).

*Figure 50: Service instantiation with MVRP - QinQ to PBB example*



In this example, the UNI and service translation SAPs are configured in the data VPLS represented by the gray circles. This instance and associated trunk SAPs were instantiated using the procedures described in the previous sections. The following are configuration rules:

- on the BEB, an I-VPLS SAP must be configured toward the local switching domain (see yellow triangle facing downward in [Figure 50: Service instantiation with MVRP - QinQ to PBB example](#)).
- on the UNI facing the customer, a "customer" SAP is configured on the lower left switch (see yellow triangle facing upward in [Figure 50: Service instantiation with MVRP - QinQ to PBB example](#)).

As soon as the first UNI SAP becomes active in the data VPLS on the ES, the associated VLAN value is advertised by MVRP throughout the related M-VPLS context. As soon as the second UNI SAP becomes available on a different switch, or in our example on the PBB BEB, the MVRP proceeds to advertise the associated VLAN value throughout the same M-VPLS. The trunks that experience MVRP declaration and registration in both directions become active, instantiating service connectivity as represented by the big and small yellow circles shown in the figure.

A hold-time parameter (**config>service>vpls>mrp>mvrp>hold-time**) is provided in the M-VPLS configuration to control when the end-station or last UNI SAP is considered active from an MVRP perspective. The hold-time controls the amount of MVRP advertisements generated on fast transitions of the end-station or UNI SAPs.

If the **no hold-time** setting is used, the following rules apply:

- MVRP stops declaring the VLAN only when the last provisioned UNI SAP associated locally with the service is deleted.
- MVRP starts declaring the VLAN as soon as the first provisioned SAP is created in the associated VPLS instance, regardless of the operational state of the SAP.

If a non-zero "hold-time" setting is used, the following rules apply:

- When a SAP in down state is added, MVRP does not declare the associated VLAN attribute. The attribute is declared immediately when the SAP comes up.
- When the SAP goes down, the MVRP waits until "hold-time" expiry before withdrawing the declaration.

For QinQ end-station SAPs, only **no hold-time** setting is allowed.

Only the following PBB Epipe and I-VPLS SAP types are eligible to activate MVRP declarations:

- dot1q: for example, 1/1/2:100
- qinq or qinq default: for example, 1/1/1:100.1 and respectively 1/1/1:100.\*, respectively; the outer VLAN 100 is used as MVRP attribute as long as it belongs to the MVRP range configured for the port
- null port and dot1q default cannot be used

Examples of steps required to activate service connectivity for VLAN 100 using MVRP follows.

In the data VPLS instance (VLAN 100) controlled by MVRP, on the QinQ switch, example:

```
config>service>vpls 100
  - sap 9/1/1:10 //UNI sap using CVID 10 as service delimiter
  - no shutdown
```

In I-VPLS on PBB BEB, example:

```
config>service>vpls 1000 i-vpls
  - sap 8/1/2:100 //sap (using MVRP VLAN 100 on endstation port in M-VPLS)
  - no shutdown
```

#### 3.4.4.4 MVRP control plane

MVRP is based on the IEEE 802.1ak MRP specification where STP is the supported method to be used for loop avoidance in a native Ethernet environment. M-VPLS and the associated MSTP (or P-MSTP) control plane provides the loop avoidance component in the Nokia implementation. Nokia MVRP may also be used in a non-MSTP, loop-free topology.

#### 3.4.4.5 STP-MVRP interaction

[Table 11: MSTP and MVRP interaction table](#) shows the expected interaction between STP (MSTP or P-MSTP) and MVRP.

Table 11: MSTP and MVRP interaction table

| Item | M-VPLS service xSTP | M-VPLS SAP STP      | Register/declare data VPLS VLAN on M-VPLS SAP | DSFS (Data SAP Forwarding State) controlled by | Datapath forwarding with MVRP enabled controlled by |
|------|---------------------|---------------------|---|--|---|
| 1    | (p)MSTP             | Enabled             | Based on M-VPLS SAP's MSTP forwarding state   | MSTP only                                      | DSFS and MVRP                                       |
| 2    | (p)MSTP             | Disabled            | Based on M-VPLS SAP's operating state         | —  | MVRP  |
| 3    | Disabled            | Enabled or Disabled | Based on M-VPLS SAP's operating state         | —  | MVRP  |

**Note:**

- Running STP in data VPLS instances controlled by MVRP is not allowed.
- Running STP on MVRP-controlled end-station SAPs is not allowed.

### 3.4.4.5.1 Interaction between MVRP and instantiated SAP status

This section describes how MVRP reacts to changes in the instantiated SAP status.

There are a number of mechanisms that may generate operational or admin down status for the SAPs and VPLS instances controlled by MVRP:

1. Port down
2. MAC move
3. Port MTU too small
4. Service MTU too small

The shutdown of the whole instantiated VPLS or instantiated SAPs is disabled in both VPLS and VPLS SAP templates. The **no shutdown** option is automatically configured.

In the **port down** case, the MVRP is also operationally down on the port so no VLAN declaration occurs.

When MAC move is enabled in a data VPLS controlled by MVRP, in case a MAC move happens, one of the instantiated SAPs controlled by MVRP may be blocked. The SAP blocking by MAC move is not reported though to the MVRP control plane. As a result, MVRP keeps declaring and registering the related VLAN value on the control SAPs, including the one that shares the same port with the instantiate SAP blocked by MAC move, as long as MVRP conditions are met. For MVRP, an active control SAP is one that has MVRP enabled and MSTP is not blocking it for the VLAN value on the port. Also in the related data VPLS, one of the two conditions must be met for the declaration of the VLAN value: there must be either a local user SAP or at least one MVRP registration received on one of the control SAPs for that VLAN.

In the last two cases, VLAN attributes get declared or registered even when the instantiated SAP is operationally down, also with the MAC move case.

### 3.4.4.5.2 Using temporary flooding to optimize failover times

MVRP advertisements use the active topology, which may be controlled through loop avoidance mechanisms like MSTP. When the active topology changes as a result of network failures, the time it takes for MVRP to bring up the optimal service connectivity may be added on top of the regular MSTP convergence time. Full connectivity also depends on the time it takes for the system to complete flushing of bad MAC entries.

To minimize the effects of MAC flushing and MVRP convergence, a temporary flooding behavior is implemented. When enabled, the temporary flooding eliminates the time it takes to flush the MAC tables. In the initial implementation, the temporary flooding is initiated only on reception of an STP TCN.

While temporary flooding is active, all the frames received in the extended data VPLS context are flooded while the MAC flush and MVRP convergence take place. The extended data VPLS context comprises all instantiated trunk SAPs regardless of the MVRP activation status. A timer option is also available to configure a fixed period of time, in seconds, during which all traffic is flooded (BUM or known unicast). When the flood-time expires, traffic is delivered according to the regular FDB content. The timer value should be configured to allow auxiliary processes like MAC flush and MVRP to converge. The temporary flooding behavior applies to all VPLS types. MAC learning continues during temporary flooding. Temporary flooding behavior is enabled using the **temp-flooding** command under **config>service>vpls** or **config>service>template>vpls-template** contexts and is supported in VPLS regardless of whether MVRP is enabled.

For temporary flooding in VPLS, the following rules apply:

- If discard-unknown is enabled, there is no temporary flooding.
- Temporary flooding while active applies also to static MAC entries; after the MAC FDB is flushed it reverts back to the static MAC entries.
- If MAC learning is disabled, fast or temporary flooding is still enabled.
- Temporary flooding is not supported in B-VPLS context when MMRP is enabled. The use of a flood-time procedure provides a better procedure for this kind of environment.

## 3.5 Configuring a VPLS service using CLI

This section provides information to configure VPLS services using the CLI.

### 3.5.1 Basic configuration

The following fields require specific input (there are no defaults) to configure a basic VPLS service:

- Customer ID (for more information see the *7705 SAR Gen 2 Services Overview Guide*)
- For a local service, configure two SAPs, specifying local access ports and encapsulation values.
- For a distributed service, configure a SAP and an SDP for each far-end node.

The following example shows a configuration of a local VPLS service on ALA-1.

```
*A:ALA-1>config>service>vpls# info
-----
...
vpls 9001 customer 6 create
```

```

        description "Local VPLS"
        stp
            shutdown
        exit
        sap 1/2/2:0 create
            description "SAP for local service"
        exit
        sap 1/1/5:0 create
            description "SAP for local service"
        exit
        no shutdown
    -----
    *A:ALA-1>config>service>vpls#

```

The following example shows a configuration of a distributed VPLS service between ALA-1, ALA-2, and ALA-3.

```

    *A:ALA-1>config>service# info
    -----
    ...
        vpls 9000 customer 6 create
            shutdown
            description "This is a distributed VPLS."
        exit
    ...
    -----
    *A:ALA-1>config>service#

    *A:ALA-2>config>service# info
    -----
    ...
        vpls 9000 customer 6 create
            description "This is a distributed VPLS."
            stp
                shutdown
            exit
            sap 1/1/5:16 create
                description "VPLS SAP"
            exit
            spoke-sdp 2:22 create
            exit
            mesh-sdp 8:750 create
            exit
            no shutdown
        exit
    ...
    -----
    *A:ALA-2>config>service#

    *A:ALA-3>config>service# info
    -----
    ...
        vpls 9000 customer 6 create
            description "This is a distributed VPLS."
            stp
                shutdown
            exit
            sap 1/1/3:33 create
                description "VPLS SAP"
            exit
            spoke-sdp 2:22 create
            exit
            mesh-sdp 8:750 create

```

```

        exit
        no shutdown
    exit
    ...
-----
*A:ALA-3>config>service#

```

## 3.5.2 Common configuration tasks

### About this task

Perform the following steps to configure both local and distributed VPLS services.

### Procedure

- Step 1.** Associate the VPLS service with a customer ID.
- Step 2.** Define SAPs by performing the following steps.
  - a. Select nodes and ports.
  - b. Optional: Select QoS policies other than the default, configured in **config>qos** context.
  - c. Optional: Select filter policies, configured in **config>filter** context.
  - d. Optional: Select an accounting policy, configured in **config>log** context.
- Step 3.** Associate SDPs for distributed services.
- Step 4.** **Optional:** Modify STP default parameters (see [VPLS and STP](#) for more information).
- Step 5.** Enable the service.

## 3.5.3 Configuring VPLS components

This section describes the CLI syntax to configure VPLS components and provides configuration examples.

### 3.5.3.1 Creating a VPLS service

Use the following CLI syntax to create a VPLS service.

CLI syntax:

```

config>service# vpls service-id [customer customer-id] [vpn vpn-id] [m-vpls] [b-vpls | i-vpls]
[create]
    description description-string
    no shutdown

```

The following example shows a VPLS configuration:

```

*A:ALA-1>config>service>vpls# info
-----
...
    vpls 9000 customer 6 create
        description "This is a distributed VPLS."
        stp
        shutdown

```

```

        exit
    exit
...
-----
*A:ALA-1>config>service>vpls#

```

### 3.5.3.2 Enabling MAC move

The **mac-move** feature is useful to protect against undetected loops in the VPLS topology as well as the presence of duplicate MACs in a VPLS service. For example, if two clients in the VPLS have the same MAC address, the VPLS experiences a high re-learn rate for the MAC and shuts down the SAP or spoke SDP when the threshold is exceeded.

Use the following CLI syntax to configure **mac-move** parameters.

```

config>service# vpls service-id [customer customer-id] [vpn vpn-id] [m-vpls]
- mac-move
  - primary-ports
    - spoke-sdp
    - cumulative-factor
  - exit
  - secondary-ports
    - spoke-sdp
    - sap
  - exit
  - move-frequency frequency
  - retry-timeout timeout
  - no shutdown

```

#### Output example: mac-move information

```

*A:ALA-1# show service id 500 mac-move
....
*A:ALA-1#
=====
Service Mac Move Information
=====
Service Id       : 500                Mac Move       : Enabled
Primary Factor   : 4                  Secondary Factor : 2
Mac Move Rate    : 2                  Mac Move Timeout : 10
Mac Move Retries : 3
-----
SAP Mac Move Information 1/1/3:501
-----
Admin State      : Up                  Oper State      : Down
Flags            : RelearnLimitExceeded
Time to come up  : 1 seconds           Retries Left    : 1
Mac Move         : Blockable           Blockable Level : Tertiary
-----
SAP Mac Move Information 1/1/3:502
-----
Admin State      : Up                  Oper State      : Up
Flags            : None
Time to RetryReset: 267 seconds         Retries Left    : None
Mac Move         : Blockable           Blockable Level : Tertiary
-----
SDP Mac Move Information 21:501
-----
Admin State      : Up                  Oper State      : Up

```



```

Flags           : None
Time to RetryReset: Never      Retries Left      : 3
Mac Move        : Blockable    Blockable Level : Secondary

```

```

-----
SDP Mac Move Information 21:502
-----

```

```

Admin State      : Up           Oper State      : Down
Flags           : RelearnLimitExceeded
Time to RetryReset: Never      Retries Left      : None
Mac Move        : Blockable    Blockable Level : Tertiary

```

```

=====
...
**A:*A:ALA-1>config>service>vpls>mac-move#

```

### 3.5.3.3 Configuring STP bridge parameters in a VPLS

Modifying some of the STP parameters allows the operator to balance STP between resiliency and speed of convergence extremes. Modifying particular parameters, as following, must be done in the constraints of the following two formulas:

$$2 \times (\text{Bridge\_Forward\_Delay} - 1.0 \text{ seconds}) \geq \text{Bridge\_Max\_Age}$$

$$\text{Bridge\_Max\_Age} \geq 2 \times (\text{Bridge\_Hello0\_Time} + 1.0 \text{ seconds})$$

The following STP parameters can be modified at the VPLS level:

- [Bridge STP admin state](#)
- [Mode](#)
- [Bridge priority](#)
- [Max age](#)
- [Forward delay](#)
- [Hello time](#)
- [MST instances](#)
- [MST max hops](#)
- [MST name](#)
- [MST revision](#)

STP always uses the locally configured values for the first three parameters (Admin State, Mode, and Priority).

For the parameters Max Age, Forward Delay, Hello Time, and Hold Count, the locally configured values are only used when this bridge has been elected root bridge in the STP domain; otherwise, the values received from the root bridge are used. The exception to this rule is: when STP is running in RSTP mode, the Hello Time is always taken from the locally configured parameter. The other parameters are only used when running mode MSTP.

#### 3.5.3.3.1 Bridge STP admin state

The administrative state of STP at the VPLS level is controlled by the **shutdown** command.

When STP on the VPLS is administratively disabled, any BPDUs are forwarded transparently through the 7705 SAR Gen 2. When STP on the VPLS is administratively enabled, but the administrative state of a SAP or spoke-SDP is down, BPDUs received on such a SAP or spoke-SDP are discarded.

```
config>service>vpls service-id# stp
no shutdown
```

### 3.5.3.3.2 Mode

To be compatible with the different iterations of the IEEE 802.1D standard, the 7705 SAR Gen 2 supports several variants of the Spanning Tree protocol:

- **rstp**  
Rapid Spanning Tree Protocol (RSTP) compliant with IEEE 802.1D-2004 - default mode.
- **dot1w**  
Compliant with IEEE 802.1w.
- **comp-dot1w**  
Operation as in RSTP but backwards compatible with IEEE 802.1w (this mode was introduced for interoperability with some MTU types).
- **mstp**  
Compliant with the Multiple Spanning Tree Protocol specified in IEEE 802.1Q REV/D5.0-09/2005. This mode of operation is only supported in an MVPLS.
- **pmstp**  
Compliant with the Multiple Spanning Tree Protocol specified in IEEE 802.1Q REV/D3.0-04/2005 but with some changes to make it backwards compatible to 802.1Q 2003 edition and IEEE 802.1w.

See [Spanning Tree operating modes](#) for more information about these modes.

```
config>service>vpls service-id# stp
mode {rstp | comp-dot1w | dot1w | mstp | pmstp}
```

**Default:** rstp

### 3.5.3.3.3 Bridge priority

The **bridge-priority** command is used to populate the priority portion of the bridge ID field within outbound BPDUs (the most significant 4 bits of the bridge ID). It is also used as part of the decision process when determining the best BPDU between messages received and sent.

When running MSTP, this is the bridge priority used for the CIST.

All values are truncated to multiples of 4096, conforming with IEEE 802.1t and 802.1D-2004.

```
config>service>vpls service-id# stp
priority bridge-priority
```

- **Range:** 1 to 65535
- **Default:** 32768

- **Restore Default:** no priority

#### 3.5.3.3.4 Max age

The **max-age** command indicates how many hops a BPDU can traverse the network starting from the root bridge. The message age field in a BPDU transmitted by the root bridge is initialized to 0. Each other bridge takes the message\_age value from BPDUs received on their root port and increment this value by 1. Therefore, the message\_age reflects the distance from the root bridge. BPDUs with a message age exceeding max-age are ignored.

STP uses the max-age value configured in the root bridge. This value is propagated to the other bridges by the BPDUs.

The default value of **max-age** is 20. This parameter can be modified within a range of 6 to 40, limited by the standard STP parameter interaction formulas.

```
config>service>vpls service-id# stp
max-age max-info-age
```

- **Range:** 6 to 40 seconds
- **Default:** 20 seconds
- **Restore Default:** no max-age

#### 3.5.3.3.5 Forward delay

RSTP, as defined in the IEEE 802.1D-2004 standards, normally transitions to the forwarding state by a handshaking mechanism (rapid transition), without any waiting times. If handshaking fails (for example, on shared links, as follows), the system falls back to the timer-based mechanism defined in the original STP (802.1D-1998) standard.

A shared link is a link with more than two Ethernet bridges (for example, a shared 10/100BaseT segment). The **port-type** command is used to configure a link as point-to-point or shared (see [SAP link type](#) for more information).

For timer-based transitions, the 802.1D-2004 standard defines an internal variable forward-delay, which is used in calculating the default number of seconds that a SAP or spoke-SDP spends in the discarding and learning states when transitioning to the forwarding state. The value of the forward-delay variable depends on the STP operating mode of the VPLS instance:

- In RSTP mode, but only when the SAP or spoke-SDP has not fallen back to legacy STP operation, the value configured by the **hello-time** command is used.
- In all other situations, the value configured by the **forward-delay** command is used.

```
config>service>vpls service-id# stp
forward-delay seconds
```

- **Range:** 4 to 30 seconds
- **Default:** 15 seconds
- **Restore Default:** no forward-delay

### 3.5.3.3.6 Hello time

The **hello-time** command configures the STP hello time for the VPLS STP instance.

The *seconds* parameter defines the default timer value that controls the sending interval between BPDU configuration messages by this bridge, on ports where this bridge assumes the designated role.

The active hello time for the spanning tree is determined by the root bridge (except when the STP is running in RSTP mode, then the hello time is always taken from the locally configured parameter).

The configured hello-time value can also be used to calculate the bridge forward delay; see [Forward delay](#).

```
config>service>vpls service-id# stp
hello-time hello-time
```

- **Range:** 1 to 10 seconds
- **Default:** 2 seconds
- **Restore Default:** no hello-time

### 3.5.3.3.7 Hold count

The **hold-count** command configures the peak number of BPDUs that can be transmitted in a period of one second.

```
config>service>vpls service-id# stp
hold-countcount-value
```

- **Range:** 1 to 10
- **Default:** 6
- **Restore Default:** no hold-count

### 3.5.3.3.8 MST instances

You can create up to 15 MST-instances. They can range from 1 to 4094. By changing path cost and priorities, you can ensure that each instance forms a unique tree within the region, ensuring that different VLANs follow different paths.

You can assign non-overlapping VLAN ranges to each instance. VLANs that are not assigned to an instance are implicitly assumed to be in instance 0, which is also called the CIST. This CIST cannot be deleted or created.

The following parameters can be defined per instance:

- **mst-priority**

The bridge-priority for this specific mst-instance. It follows the same rules as bridge-priority. For the CIST, the bridge-priority is used.

- **vlan-range**

The VLANs are mapped to this specific mst-instance. If no VLAN-ranges are defined in any mst-instances, all VLANs are mapped to the CIST.

### 3.5.3.3.9 MST max hops

The **mst-max-hops** command defines the maximum number of hops the BPDU can traverse inside the region. Outside the region, max-age is used.

### 3.5.3.3.10 MST name

The MST name defines the name that the operator gives to a region. Together with MST revision and the VLAN to MST-instance mapping, it forms the MST configuration identifier. Two bridges that have the same MST configuration identifier form a region if they exchange BPDUs.

### 3.5.3.3.11 MST revision

The MST revision together with MST name and VLAN-to-MST-instance mapping define the MST configuration identifier. Two bridges that have the same MST configuration identifier form a region if they exchange BPDUs.

## 3.5.3.4 Configuring GSMP parameters

The following parameters must be configured in order for GSMP to function:

- One or more GSMP sessions
- One or more ANCP policies
- For basic subscriber management only, ANCP static maps
- For enhanced subscriber management only, associate subscriber profiles with ANCP policies

Use the following CLI syntax to configure GSMP parameters.

CLI syntax:

```
config>service>vpls# gsmp
  - group name [create]
    - ancp
      - dynamic-topology-discover
      - oam
    - description description-string
    - hold-multiplier multiplier
    - keepalive seconds
    - neighbor ip-address [create]
      - description v
      - local-address ip-address
      - priority-marking dscp dscp-name
      - priority-marking prec ip-prec-value
      - [no] shutdown
    - [no] shutdown
  - [no] shutdown
```

This example shows a GSMP group configuration.

```
A:ALA-48>config>service>vpls>gsmp# info
-----
      group "group1" create
```

```

        description "test group config"
        neighbor 10.10.10.104 create
            description "neighbor1 config"
            local-address 10.10.10.103
            no shutdown
        exit
        no shutdown
    exit
    no shutdown
-----
A:ALA-48>config>service>vpls>gsmp#

```

### 3.5.3.5 Configuring a VPLS SAP

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

See the following sections to configure local and distributed VPLS SAPs.

- [Local VPLS SAPs](#)
- [Distributed VPLS SAPs](#)

#### 3.5.3.5.1 Local VPLS SAPs

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

The following example shows a local VPLS configuration:

#### Example

```

*A:ALA-1>config>service# info
-----
...
    vpls 90001 customer 6 create
        description "Local VPLS"
        stp
            shutdown
        exit
        sap 1/2/2:0 create
            description "SAP for local service"
        exit
        sap 1/1/5:0 create
            description "SAP for local service"
        exit
        no shutdown
    exit
-----
*A:ALA-1>config>service#
*A:ALA-1>config>service# info
-----
    vpls 1150 customer 1 create
        fdb-table-size 1000
        fdb-table-low-wmark 5
        fdb-table-high-wmark 80
        local-age 60
        stp

```

```

        shutdown
    exit
    sap 1/1/1:1155 create
    exit
    sap 1/1/2:1150 create
    exit
    no shutdown
exit
-----
*A:ALA-1>config>service#

```

### 3.5.3.5.2 Distributed VPLS SAPs

To configure a distributed VPLS service, you must configure service entities on originating and far-end nodes. You must use the same service ID on all ends (for example, create a VPLS service ID 9000 on ALA-1, ALA-2, and ALA-3). A distributed VPLS consists of a SAP on each participating node and an SDP bound to each participating node.

For SDP configuration information, see the *7705 SAR Gen 2 Services Overview Guide*. For SDP binding information, see [Configuring SDP bindings](#).

The following example shows a configuration of VPLS SAPs configured for ALA-1, ALA-2, and ALA-3.

#### Example

```

*A:ALA-1>config>service# info
-----
...
    vpls 9000 customer 6 vpn 750 create
        description "Distributed VPLS services."
        stp
            shutdown
        exit
        sap 1/2/5:0 create
            description "VPLS SAP"
            multi-service-site "West"
        exit
    exit
...
-----
*A:ALA-1>config>service#

*A:ALA-2>config>service# info
-----
...
    vpls 9000 customer 6 vpn 750 create
        description "Distributed VPLS services."
        stp
            shutdown
        exit
        sap 1/1/2:22 create
            description "VPLS SAP"
            multi-service-site "West"
        exit
    exit
...
-----
*A:ALA-2>config>service#

```

```
*A:ALA-3>config>service# info
-----
...
    vpls 9000 customer 6 vpn 750 create
        description "Distributed VPLS services."
        stp
            shutdown
        exit
        sap 1/1/3:33 create
            description "VPLS SAP"
            multi-service-site "West"
        exit
    exit
...
-----
*A:ALA-3>config>service#
```

### 3.5.3.5.3 Configuring SAP-specific STP parameters

When a VPLS has STP enabled, each SAP within the VPLS has STP enabled by default. The operation of STP on each SAP is governed by the following:

- [SAP STP administrative state](#)
- [SAP virtual port number](#)
- [SAP priority](#)
- [SAP path cost](#)
- [SAP edge port](#)
- [SAP auto edge](#)
- [SAP link type](#)

#### 3.5.3.5.3.1 SAP STP administrative state

The administrative state of STP within a SAP controls how BPDUs are transmitted and handled when received. The allowable states are as follows:

- **SAP Admin Up**

The default administrative state is up for STP on a SAP. BPDUs are handled in the normal STP manner on a SAP that is administratively up.

- **SAP Admin Down**

An administratively down state allows a service provider to prevent a SAP from becoming operationally blocked. BPDUs do not originate out of the SAP toward the customer.

If STP is enabled on the VPLS level, but disabled on the SAP, received BPDUs are discarded. Discarding the incoming BPDUs allows STP to continue to operate normally within the VPLS service while ignoring the down SAP. The specified SAP is always in an operationally forwarding state.



**Note:** The administratively down state allows a loop to form within the VPLS.

```
config>service>vpls>sap>stp#
```



```
[no] shutdown
```

- **Range:** shutdown or no shutdown
- **Default:** no shutdown (SAP admin up)

### 3.5.3.5.3.2 SAP virtual port number

The virtual port number uniquely identifies a SAP within configuration BPDUs. The internal representation of a SAP is unique to a system and has a reference space much bigger than the 12 bits definable in a configuration BPDU. STP takes the internal representation value of a SAP and identifies it with a virtual port number that is unique to every other SAP defined on the VPLS. The virtual port number is assigned when the SAP is added to the VPLS.

Because the order in which SAPs are added to the VPLS is not preserved between reboots of the system, the virtual port number may change between restarts of the STP instance. To achieve consistency after a reboot, the virtual port number can be specified explicitly.

```
config>service>vpls>sap# stp
port-num number
```

- **Range:** 1 to 2047
- **Default:** (automatically generated)
- **Restore Default:** no port-num

### 3.5.3.5.3.3 SAP priority

SAP priority allows a configurable tie-breaking parameter to be associated with a SAP. When configuration BPDUs are being received, the configured SAP priority is used in some circumstances to determine whether a SAP is designated or blocked. These are the values used for CIST when running MSTP.

In traditional STP implementations (802.1D-1998), this field is called the port priority and has a value of 0 to 255. This field is coupled with the port number (0 to 255 also) to create a 16-bit value. In the latest STP standard (802.1D-2004), only the upper 4 bits of the port priority field are used to encode the SAP priority. The remaining 4 bits are used to extend the port ID field into a 12-bit virtual port number field. The virtual port number uniquely references a SAP within the STP instance. See [SAP virtual port number](#) for information about the virtual port number.

STP computes the actual SAP priority by taking the configured priority value and masking out the lower four bits. The result is the value that is stored in the SAP priority parameter. For example, if a value of 0 was entered, masking out the lower 4 bits would result in a parameter value of 0. If a value of 255 was entered, the result would be 240.

The default value for SAP priority is 128. This parameter can be modified within a range of 0 to 255, 0 being the highest priority. Masking causes the values actually stored and displayed to be 0 to 240, in increments of 16.

```
config>service>vpls>sap>stp#
priority stp-priority
```

- **Range:** 0 to 255 (240 largest value, in increments of 16)
- **Default:** 128

- **Restore Default:** no priority

#### 3.5.3.5.3.4 SAP path cost

The SAP path cost is used by STP to calculate the path cost to the root bridge. The path cost in BPDUs received on the root port is incremented with the configured path cost for that SAP. When BPDUs are sent out of other egress SAPs, the newly calculated root path cost is used. These are the values used for CIST when running MSTP.

STP suggests that the path cost is defined as a function of the link bandwidth. Because SAPs are controlled by complex queuing dynamics, the STP path cost is a purely static configuration.

The default value for SAP path cost is 10. This parameter can be modified within a range of 1 to 65535, 1 being the lowest cost.

```
config>service>vpls>sap>stp#  
path-cost sap-path-cost
```

- **Range:** 1 to 200000000
- **Default:** 10
- **Restore Default:** no path-cost

#### 3.5.3.5.3.5 SAP edge port

The SAP **edge-port** command is used to reduce the time it takes for a SAP to reach the forwarding state when the SAP is on the edge of the network, and therefore has no further STP bridge to handshake with.

The **edge-port** command is used to initialize the internal OPER\_EDGE variable. When OPER\_EDGE is false on a SAP, the normal mechanisms are used to transition to the forwarding state (see [Forward delay](#)). When OPER\_EDGE is true, STP assumes that the remote end agrees to transition to the forwarding state without actually receiving a BPDU with an agreement flag set.

The OPER\_EDGE variable is dynamically set to false if the SAP receives BPDUs (the configured **edge-port** value does not change). The OPER\_EDGE variable is dynamically set to true if **auto-edge** is enabled and STP concludes there is no bridge behind the SAP.

When STP on the SAP is administratively disabled and re-enabled, the OPER\_EDGE is reinitialized to the value configured for **edge-port**.

Valid values for SAP **edge-port** are **enabled** and **disabled** (default value).

To configure SAP **edge-port**, use the following command.

```
configure service vpls sap stp edge-port
```

#### 3.5.3.5.3.6 SAP auto edge

The SAP **edge-port** command is used to instruct the STP to dynamically decide whether the SAP is connected to another bridge.

If **auto-edge** is enabled, and STP concludes there is no bridge behind the SAP, the OPER\_EDGE variable is dynamically set to true. If **auto-edge** is enabled, and a BPDU is received, the OPER\_EDGE variable is dynamically set to false (see [SAP edge port](#)).

Valid values for SAP **auto-edge** are **enabled** (default value) and **disabled**.

To configure SAP **auto-edge**, use the following command.

```
configure service vpls sap stp auto-edge
```

### 3.5.3.5.3.7 SAP link type

The SAP **link-type** parameter instructs STP on the maximum number of bridges behind this SAP. If there is only a single bridge, transitioning to forwarding state is based on handshaking (fast transitions). If more than two bridges are connected by a shared media, their SAPs should all be configured as shared, and timer-based transitions are used.

Valid values for SAP link-type are shared and pt-pt, with pt-pt being the default.

```
config>service>vpls>sap>stp#  
    link-type {pt-pt | shared}
```

- **Default:** link-type pt-pt
- **Restore Default:** no link-type

### 3.5.3.5.4 STP SAP operational states

The operational state of STP within a SAP controls how BPDUs are transmitted and handled when received. The following STP SAP states are defined:

- [Operationally disabled](#)
- [Operationally discarding](#)
- [Operationally learning](#)
- [Operationally forwarding](#)
- [SAP BPDU encapsulation state](#)

#### 3.5.3.5.4.1 Operationally disabled

Operationally disabled is the normal operational state for STP on a SAP in a VPLS that has any of the following conditions:

- VPLS state administratively down
- SAP state administratively down
- SAP state operationally down

If the SAP enters the operationally up state with the STP administratively up and the SAP STP state is up, the SAP transitions to the STP SAP discarding state.

When, during normal operation, the router detects a downstream loop behind a SAP or spoke-SDP, BPDUs can be received at a very high rate. To recover from this situation, STP transitions the SAP to disabled state for the configured forward-delay duration.

#### 3.5.3.5.4.2 Operationally discarding

A SAP in the discarding state only receives and sends BPDUs, building the local correct STP state for each SAP while not forwarding actual user traffic. The duration of the discarding state is described in section [Forward delay](#).



**Note:** In previous versions of the STP standard, the discarding state was called a blocked state.

#### 3.5.3.5.4.3 Operationally learning

The learning state allows population of the MAC forwarding table before entering the forwarding state. In this state, no user traffic is forwarded.

#### 3.5.3.5.4.4 Operationally forwarding

Configuration BPDUs are sent out of a SAP in the forwarding state. Layer 2 frames received on the SAP are source learned and destination forwarded according to the FDB. Layer 2 frames received on other forwarding interfaces and destined for the SAP are also forwarded.

#### 3.5.3.5.4.5 SAP BPDUs encapsulation state

IEEE 802.1d (referred as dot1d) and Cisco's per VLAN Spanning Tree (PVST) BPDUs encapsulations are supported on a per-SAP basis. STP is associated with a VPLS service like PVST is associated per VLAN. The main difference resides in the Ethernet and LLC framing and a type-length-value (TLV) field trailing the BPDUs.

[Table 12: Spoke SDP BPDUs encapsulation states](#) shows differences between dot1d and PVST Ethernet BPDUs encapsulations based on the interface encap-type field.

Each SAP has a Read-Only operational state that shows which BPDUs encapsulation is currently active on the SAP. The states are as follows:

- **dot1d**

This state specifies that the switch is currently sending IEEE 802.1d standard BPDUs. The BPDUs are tagged or non-tagged based on the encapsulation type of the egress interface and the encapsulation value defined in the SAP. A SAP defined on an interface with encapsulation type dot1q continues in the dot1d BPDUs encapsulation state until a PVST encapsulated BPDUs is received, in which case, the SAP converts to the PVST encapsulation state. Each received BPDUs must be properly IEEE 802.1Q-tagged if the interface encapsulation type is defined as dot1q. PVST BPDUs are silently discarded if received when the SAP is on an interface defined with encapsulation type null.

- **PVST**

This state specifies that the switch is currently sending proprietary encapsulated BPDUs. PVST BPDUs are only supported on Ethernet interfaces with the encapsulation type set to dot1q. The SAP continues in the PVST BPDUs encapsulation state until a dot1d encapsulated BPDUs is received, in which case,

the SAP reverts to the dot1d encapsulation state. Each received BPDU must be properly IEEE 802.1Q-tagged with the encapsulation value defined for the SAP. PVST BPDUs are silently discarded if received when the SAP is on an interface defined with a null encapsulation type.

Dot1d is the initial and only SAP BPDU encapsulation state for SAPs defined on an Ethernet interface with encapsulation type set to null.

Each transition between encapsulation types optionally generates an alarm that can be logged and optionally transmitted as an SNMP trap.

### 3.5.3.5.5 Configuring VPLS SAPs with split-horizon

To configure a VPLS service with a split-horizon group, add the **split-horizon-group** parameter when creating the SAP. Traffic arriving on a SAP within a split-horizon group is not copied to other SAPs in the same split-horizon group.

#### Example: VPLS configuration output with split-horizon enabled

```
*A:ALA-1>config>service# info
-----
...
  vpls 800 customer 6001 vpn 700 create
    description "VPLS with split horizon for DSL"
    stp
      shutdown
    exit
    sap 1/1/3:100 split-horizon-group DSL-group1 create
      description "SAP for residential bridging"
    exit
    sap 1/1/3:200 split-horizon-group DSL-group1 create
      description "SAP for residential bridging"
    exit
    split-horizon-group DSL-group1
      description "Split horizon group for DSL"
    exit
    no shutdown
  exit
...
-----
*A:ALA-1>config>service#
```

### 3.5.3.5.6 Configuring MAC learning protection

To configure MAC learning protection, configure split horizon, MAC protection, and SAP parameters.

The following example shows a VPLS configuration with split horizon enabled:

```
A:ALA-48>config>service>vpls# info
-----
  description "local VPLS"
  split-horizon-group "DSL-group1" create
    restrict-protected-src
    restrict-unprotected-dst
  exit
  mac-protect
    mac ff:ff:ff:ff:ff:ff
  exit
  sap 1/1/9:0 create
```

```

        ingress
            scheduler-policy "SLA1"
            qos 100 shared-queuing
        exit
        egress
            scheduler-policy "SLA1"
            filter ip 10
        exit
        restrict-protected-src
        arp-reply-agent
        host-connectivity-verify source-ip 10.144.145.1
    exit
...
-----
A:ALA-48>config>service>vpls#

```

### 3.5.3.6 Configuring SAP subscriber management parameters

Use the following CLI syntax to configure subscriber management parameters on a VPLS service SAP. The policies and profiles that are referenced in the **def-sla-profile**, **def-sub-profile**, **non-sub-traffic**, and **sub-ident-policy** commands must already be configured in the **config>subscr-mgmt** context.

CLI syntax:

```

config>service>vpls service-id
  - sap sap-id [split-horizon-group group-name]
    - sub-sla-mgmt
      - def-sla-profile default-sla-profile-name
      - def-sub-profile default-subscriber-profile-name
      - mac-da-hashing
      - multi-sub-sap [number-of-sub]
      - no shutdown
      - single-sub-parameters
        - non-sub-traffic sub-profile sub-profile-name sla-profile sla-profile-name
    [subscriber sub-ident-string]
      - profiled-traffic-only
      - sub-ident-policy sub-ident-policy-name

```

The following example shows a subscriber management configuration:

```

A:ALA-48>config>service>vpls#
-----
        description "Local VPLS"
        stp
            shutdown
        exit
        sap 1/2/2:0 create
            description "SAP for local service"
            sub-sla-mgmt
                def-sla-profile "sla-profile1"
                sub-ident-policy "SubIdent1"
            exit
        exit
        sap 1/1/5:0 create
            description "SAP for local service"
        exit
        no shutdown
-----
A:ALA-48>config>service>vpls#

```

### 3.5.3.7 MSTP control over Ethernet tunnels

When MSTP is used to control VLANs, a range of VLAN IDs is normally used to specify the VLANs to be controlled.

If an Ethernet tunnel SAP is to be controlled by MSTP, the Ethernet tunnel SAP ID needs to be within the VLAN range specified under the mst-instance.

```
vpls 400 customer 1 m-vpls create
    stp
        mode mstp
        mst-instance 111 create
            vlan-range 1-100
        exit
        mst-name "abc"
        mst-revision 1
        no shutdown
    exit
    sap 1/1/1:0 create // untagged
    exit
    sap eth-tunnel-1 create
    exit
    no shutdown
exit
vpls 401 customer 1 create
    stp
        shutdown
    exit
    sap 1/1/1:12 create
    exit
    sap eth-tunnel-1:12 create
        // Ethernet tunnel SAP ID 12 falls within the VLAN
        // range for mst-instance 111
        eth-tunnel
            path 1 tag 1000
            path 8 tag 2000
        exit
    exit
    no shutdown
exit
```

### 3.5.3.8 Configuring SDP bindings

VPLS provides scaling and operational advantages. A hierarchical configuration eliminates the need for a full mesh of VCs between participating devices. Hierarchy is achieved by enhancing the base VPLS core mesh of VCs with access VCs (spoke) to form two tiers. Spoke SDPs are generally created between Layer 2 switches and placed at the MTU. The PE routers are placed at the service provider's Point of Presence (POP). Signaling and replication overhead on all devices is considerably reduced.

A spoke SDP is treated like the equivalent of a traditional bridge port where flooded traffic received on the spoke SDP is replicated on all other "ports" (other spoke and mesh SDPs or SAPs) and not transmitted on the port it was received (unless a split horizon group was defined on the spoke SDP; see section [Configuring VPLS spoke SDPs with split horizon](#)).

A spoke SDP connects a VPLS service between two sites and, in its simplest form, could be a single tunnel LSP. A set of ingress and egress VC labels are exchanged for each VPLS service instance to be transported over this LSP. The PE routers at each end treat this as a virtual spoke connection for the VPLS

service in the same way as the PE-MTU connections. This architecture minimizes the signaling overhead and avoids a full mesh of VCs and LSPs between the two metro networks.

A mesh SDP bound to a service is logically treated like a single bridge “port” for flooded traffic where flooded traffic received on any mesh SDP on the service is replicated to other “ports” (spoke SDPs and SAPs) and not transmitted on any mesh SDPs.

A VC-ID can be specified with the SDP-ID. The VC-ID is used instead of a label to identify a virtual circuit. The VC-ID is significant between peer SRs on the same hierarchical level. The value of a VC-ID is conceptually independent from the value of the label or any other datalink-specific information of the VC.

**Figure 51: SDPs — unidirectional tunnels** shows an example of a distributed VPLS service configuration of spoke and mesh SDPs (unidirectional tunnels) between routers and MTUs.

### 3.5.3.9 Configuring overrides on service SAPs

The following output shows a service SAP queue override configuration example:

```
*A:ALA-48>config>service>vpls>sap# info
-----
...
exit
ingress
  scheduler-policy "SLA1"
  scheduler-override
    scheduler "sched1" create
    parent weight 3 cir-weight 3
  exit
exit
  policer-control-policy "SLA1-p"
  policer-control-override create
  max-rate 50000
exit
  qos 100 multipoint-shared
  queue-override
    queue 1 create
    rate 1500000 cir 2000
  exit
exit
  policer-override
  policer 1 create
  rate 10000
  exit
exit
egress
  scheduler-policy "SLA1"
  policer-control-policy "SLA1-p"
  policer-control-override create
  max-rate 60000
exit
  qos 100
  queue-override
    queue 1 create
    adaptation-rule pir max cir max
  exit
exit
  policer-override
  policer 1 create
  mbs 2000 kilobytes
exit
```

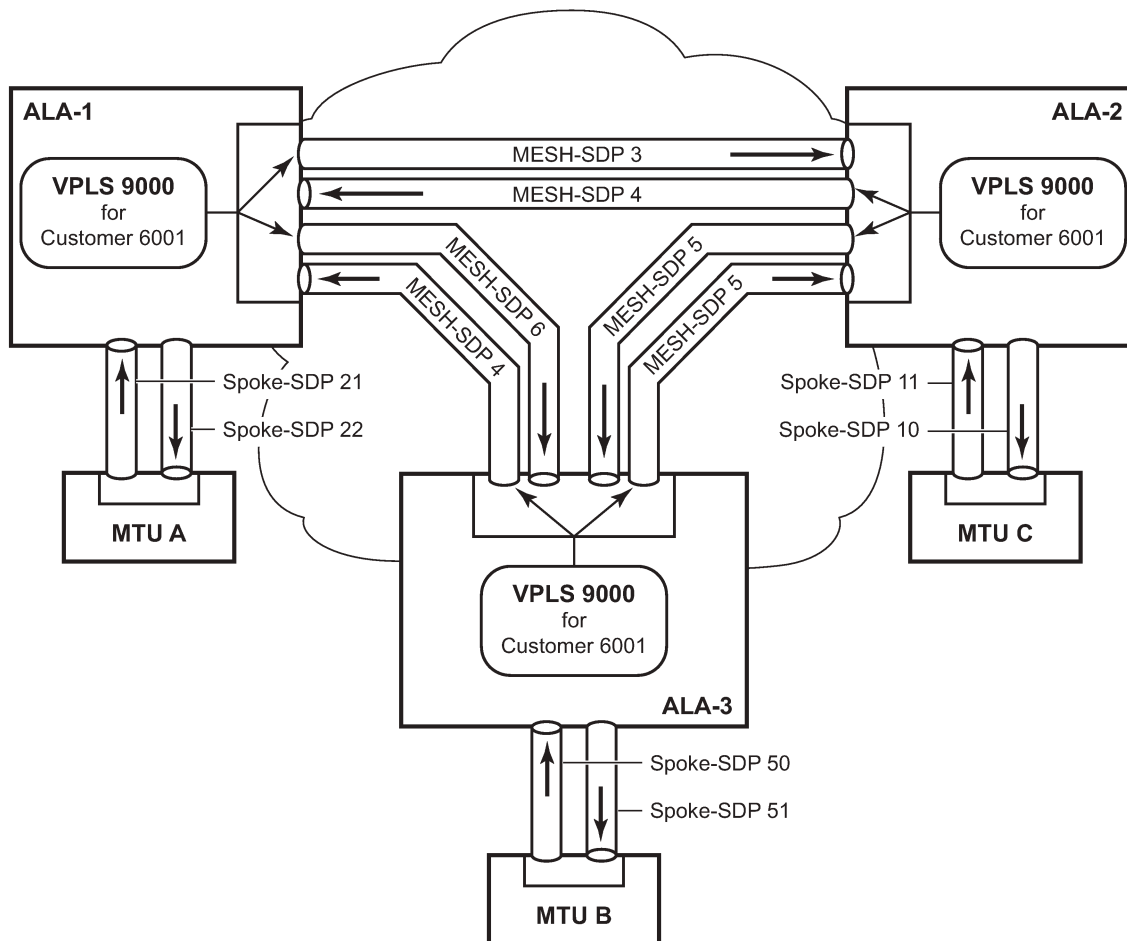


```

exit
filter ip 10
exit
-----
*A:ALA-48>config>service>vpls>sap#

```

Figure 51: SDPs — unidirectional tunnels



OSSG032

Use the following CLI syntax to create mesh or spoke-SDP bindings with a distributed VPLS service. SDPs must be configured before binding. For information about creating SDPs, see the *7705 SAR Gen 2 Services Overview Guide*.

Use the following CLI syntax to configure mesh SDP bindings.

CLI syntax:

```

config>service# vpls service-id
- mesh-sdp sdp-id[:vc-id] [vc-type {ether | vlan}]
- egress
  - filter {ip ip-filter-id|mac mac-filter-id}
  - mfib-allowed-mds-destinations
    - mda mda-id
  - vc-label egress-vc-label

```

```

- ingress
  - filter {ip ip-filter-id|mac mac-filter-id}
  - vc-label ingress-vc-label
- no shutdown
- static-mac ieee-address
- vlan-vc-tag 0..4094

```

Use the following CLI syntax to configure spoke-SDP bindings.

CLI syntax:

```

config>service# vpls service-id
- spoke-sdp sdp-id:vc-id [vc-type {ether | vlan}] [split-horizon-group group-name]
- egress
  - filter {ip ip-filter-id|mac mac-filter-id}
  - vc-label egress-vc-label
- ingress
  - filter {ip ip-filter-id|mac mac-filter-id}
  - vc-label ingress-vc-label
- limit-mac-move[non-blockable]
- vlan-vc-tag 0..4094
- no shutdown
- static-mac ieee-address
- stp
  - path-cost stp-path-cost
  - priority stp-priority
  - no shutdown
- vlan-vc-tag [0..4094]

```

The following examples show SDP binding configurations for ALA-1, ALA-2, and ALA-3 for VPLS service ID 9000 for customer 6:

```

*A:ALA-1>config>service# info
-----
...
    vpls 9000 customer 6 create
      description "This is a distributed VPLS."
      stp
        shutdown
      exit
      sap 1/2/5:0 create
      exit
      spoke-sdp 2:22 create
      exit
      mesh-sdp 5:750 create
      exit
      mesh-sdp 7:750 create
      exit
      no shutdown
    exit
-----
*A:ALA-1>config>service#

*A:ALA-2>config>service# info
-----
...
    vpls 9000 customer 6 create
      description "This is a distributed VPLS."
      stp
        shutdown
      exit

```

```

        sap 1/1/2:22 create
        exit
        spoke-sdp 2:22 create
        exit
        mesh-sdp 5:750 create
        exit
        mesh-sdp 7:750 create
        exit
        no shutdown
    exit
-----

*A:ALA-3>config>service# info
-----
...
    vpls 9000 customer 6 create
        description "This is a distributed VPLS."
        stp
            shutdown
        exit
        sap 1/1/3:33 create
        exit
        spoke-sdp 2:22 create
        exit
        mesh-sdp 5:750 create
        exit
        mesh-sdp 7:750 create
        exit
        no shutdown
    exit
-----
*A:ALA-3>config>service#

```

### 3.5.3.9.1 Configuring spoke-SDP specific STP parameters

When a VPLS has STP enabled, each spoke-SDP within the VPLS has STP enabled by default. Subsequent sections describe spoke-SDP specific STP parameters in detail.

#### 3.5.3.9.1.1 Spoke SDP STP administrative state

The administrative state of STP within a spoke SDP controls how BPDUs are transmitted and handled when received. The allowable states are:

- **spoke-sdp admin up**

The default administrative state is up for STP on a spoke SDP. BPDUs are handled in the normal STP manner on a spoke SDP that is administratively up.

- **spoke-sdp admin down**

An administratively down state allows a service provider to prevent a spoke SDP from becoming operationally blocked. BPDUs do not originate out the spoke SDP toward the customer.

If STP is enabled on VPLS level, but disabled on the spoke SDP, received BPDUs are discarded. Discarding the incoming BPDUs allows STP to continue to operate normally within the VPLS service while ignoring the down spoke SDP. The specified spoke SDP is always in an operationally forwarding state.



**Note:** The administratively down state allows a loop to form within the VPLS.

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#  
[no] shutdown
```

|         |                                  |
|---------|----------------------------------|
| Range   | shutdown or no shutdown          |
| Default | no shutdown (spoke-SDP admin up) |

3.5.3.9.1.2 Spoke SDP virtual port number

The virtual port number uniquely identifies a spoke SDP within configuration BPDUs. The internal representation of a spoke SDP is unique to a system and has a reference space much bigger than the 12 bits definable in a configuration BPDU. STP takes the internal representation value of a spoke SDP and identifies it with its own virtual port number that is unique to every other spoke-SDP defined on the VPLS. The virtual port number is assigned at the time that the spoke SDP is added to the VPLS.

Because the order in which spoke SDPs are added to the VPLS is not preserved between reboots of the system, the virtual port number may change between restarts of the STP instance. To achieve consistency after a reboot, the virtual port number can be specified explicitly.

CLI syntax:

```
config>service>vpls>spoke-sdp# stp  
port-num number
```

|                 |                         |
|-----------------|-------------------------|
| Range           | 1 to 2047               |
| Default         | automatically generated |
| Restore Default | no port-num             |

3.5.3.9.1.3 Spoke SDP priority

Spoke SDP priority allows a configurable tiebreaking parameter to be associated with a spoke SDP. When configuration BPDUs are being received, the configured spoke-SDP priority is used in some circumstances to determine whether a spoke SDP is designated or blocked.

In traditional STP implementations (802.1D-1998), this field is called the port priority and has a value of 0 to 255. This field is coupled with the port number (0 to 255 also) to create a 16-bit value. In the latest STP standard (802.1D-2004), only the upper 4 bits of the port priority field are used to encode the spoke SDP priority. The remaining 4 bits are used to extend the port ID field into a 12-bit virtual port number field. The virtual port number uniquely references a spoke SDP within the STP instance. See [Spoke SDP virtual port number](#) for more information about the virtual port number.

STP computes the actual spoke SDP priority by taking the configured priority value and masking out the lower four bits. The result is the value that is stored in the spoke SDP priority parameter. For instance, if a value of 0 was entered, masking out the lower 4 bits would result in a parameter value of 0. If a value of 255 was entered, the result would be 240.

The default value for spoke SDP priority is 128. This parameter can be modified within a range of 0 to 255; 0 being the highest priority. Masking causes the values actually stored and displayed to be 0 to 240, in increments of 16.

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#
priority stp-priority
```

|                        |   |
|------------------------|---|
| <b>Range</b>           | 0 to 255 (240 largest value, in increments of 16) |
| <b>Default</b>         | 128   |
| <b>Restore Default</b> | no priority                                       |

#### 3.5.3.9.1.4 Spoke SDP path cost

The spoke SDP path cost is used by STP to calculate the path cost to the root bridge. The path cost in BPDUs received on the root port is incremented with the configured path cost for that spoke-SDP. When BPDUs are sent out of other egress spoke SDPs, the newly calculated root path cost is used.

STP suggests that the path cost is defined as a function of the link bandwidth. Because spoke SDPs are controlled by complex queuing dynamics, the STP path cost is a purely static configuration.

The default value for spoke SDP path cost is 10. This parameter can be modified within a range of 1 to 200000000 (1 is the lowest cost).

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#
path-cost stp-path-cost
```

|                        |                |
|------------------------|----------------|
| <b>Range</b>           | 1 to 200000000 |
| <b>Default</b>         | 10             |
| <b>Restore Default</b> | no path-cost   |

#### 3.5.3.9.1.5 Spoke SDP edge port

The spoke SDP **edge-port** command is used to reduce the time it takes a spoke SDP to reach the forwarding state when the spoke SDP is on the edge of the network, and therefore has no further STP bridge to handshake with.

The **edge-port** command is used to initialize the internal OPER\_EDGE variable. When OPER\_EDGE is false on a spoke SDP, the normal mechanisms are used to transition to the forwarding state (see [Forward delay](#)). When OPER\_EDGE is true, STP assumes that the remote end agrees to transition to the forwarding state without actually receiving a BPDU with an agreement flag set.

The OPER\_EDGE variable is dynamically set to false if the spoke SDP receives BPDUs (the configured **edge-port** value does not change). The OPER\_EDGE variable is dynamically set to true if **auto-edge** is enabled and STP concludes there is no bridge behind the spoke SDP.

When STP on the spoke SDP is administratively disabled and re-enabled, the OPER\_EDGE is re-initialized to the spoke SDP configured for edge port.

Valid values for spoke-SDP **edge-port** are **enabled** and **disabled** (default value).

To configure spoke-SDP **edge-port**, use the following command.

```
configure service vpls spoke-sdp stp edge-port
```

### 3.5.3.9.1.6 Spoke SDP auto edge

The spoke SDP **edge-port** command is used to instruct STP to dynamically decide whether the spoke SDP is connected to another bridge.

If **auto-edge** is enabled, and STP concludes there is no bridge behind the spoke SDP, the OPER\_EDGE variable is dynamically set to true. If **auto-edge** is enabled, and a BPDU is received, the OPER\_EDGE variable is dynamically set to false (see [Spoke SDP edge port](#)).

Valid values for spoke SDP **auto-edge** are **enabled** (default value) and **disabled**.

To configure spoke SDP **auto-edge**, use the following command.

```
configure service vpls spoke-sdp stp auto-edge
```

### 3.5.3.9.1.7 Spoke SDP link type

The spoke SDP link-type command instructs STP on the maximum number of bridges behind this spoke SDP. If there is only a single bridge, transitioning to forwarding state is based on handshaking (fast transitions). If more than two bridges are connected by a shared media, their spoke SDPs should all be configured as shared, and timer-based transitions are used.

Valid values for spoke SDP link-type are shared and pt-pt, with pt-pt being the default.

CLI syntax:

```
config>service>vpls>spoke-sdp>stp#  
link-type {pt-pt|shared}
```

|                        |                 |
|------------------------|-----------------|
| <b>Default</b>         | link-type pt-pt |
| <b>Restore Default</b> | no link-type    |

### 3.5.3.9.2 Spoke SDP STP operational states

The operational state of STP within a spoke SDP controls how BPDUs are transmitted and handled when received. Subsequent sections describe spoke SDP operational states.

#### 3.5.3.9.2.1 Operationally disabled

Operationally disabled is the normal operational state for STP on a spoke SDP in a VPLS that has any of the following conditions:

- VPLS state administratively down
- Spoke SDP state administratively down

- Spoke SDP state operationally down

If the spoke SDP enters the operationally up state with the STP administratively up and the spoke SDP STP state is up, the spoke-SDP transitions to the STP spoke SDP discarding state.

When, during normal operation, the router detects a downstream loop behind a spoke SDP, BPDUs can be received at a very high rate. To recover from this situation, STP transitions the spoke SDP to a disabled state for the configured forward-delay duration.

### 3.5.3.9.2.2 Operationally discarding

A spoke-SDP in the discarding state only receives and sends BPDUs, building the local correct STP state for each spoke-SDP while not forwarding actual user traffic. The duration of the discarding state is described in section [Forward delay](#).



**Note:** In previous versions of the STP standard, the discarding state was called a blocked state.

### 3.5.3.9.2.3 Operationally learning

The learning state allows population of the MAC forwarding table before entering the forwarding state. In this state, no user traffic is forwarded.

### 3.5.3.9.2.4 Operationally forwarding

Configuration BPDUs are sent out of a SAP in the forwarding state. Layer 2 frames received on the SAP are source learned and destination forwarded according to the FDB. Layer 2 frames received on other forwarding interfaces and destined for the SAP are also forwarded.

### 3.5.3.9.2.5 Spoke SDP BPDU encapsulation states

IEEE 802.1D (referred as dot1d) and Cisco's per VLAN Spanning Tree (PVST) BPDU encapsulations are supported on a per spoke SDP basis. STP is associated with a VPLS service like PVST is per VLAN. The main difference resides in the Ethernet and LLC framing and a type-length-value (TLV) field trailing the BPDU.

[Table 12: Spoke SDP BPDU encapsulation states](#) shows differences between dot1D and PVST Ethernet BPDU encapsulations based on the interface encap-type field.

Table 12: Spoke SDP BPDU encapsulation states

| Field           | dot1d<br>encap-type null | dot1d<br>encap-type dot1q | PVST<br>encap-type<br>null | PVST<br>encap-type dot1q |
|-----------------|--------------------------|---------------------------|----------------------------|--------------------------|
| Destination MAC | 01:80:c2:00:00:00        | 01:80:c2:00:00:00         | N/A                        | 01:00:0c:cc:cc:cd        |
| Source MAC      | Sending Port MAC         | Sending Port MAC          | N/A                        | Sending Port MAC         |
| EtherType       | N/A                      | 0x81 00                   | N/A                        | 0x81 00                  |

| Field               | dot1d<br>encap-type null | dot1d<br>encap-type dot1q | PVST<br>encap-type<br>null | PVST<br>encap-type dot1q   |
|---------------------|--------------------------|---------------------------|----------------------------|----------------------------|
| Dot1p and DEI       | N/A                      | 0xe                       | N/A                        | 0xe                        |
| Dot1q               | N/A                      | VPLS spoke-SDP ID         | N/A                        | VPLS spoke-SDP encap value |
| Length              | LLC Length               | LLC Length                | N/A                        | LLC Length                 |
| LLC DSAP SSAP       | 0x4242                   | 0x4242                    | N/A                        | 0xaaaa (SNAP)              |
| LLC CNTL            | 0x03                     | 0x03                      | N/A                        | 0x03                       |
| SNAP OUI            | N/A                      | N/A                       | N/A                        | 00 00 0c (Cisco OUI)       |
| SNAP PID            | N/A                      | N/A                       | N/A                        | 01 0b                      |
| CONFIG or TCN BPDUs | Standard 802.1d          | Standard 802.1d           | N/A                        | Standard 802.1d            |
| TLV: Type and Len   | N/A                      | N/A                       | N/A                        | 58 00 00 00 02             |
| TLV: VLAN           | N/A                      | N/A                       | N/A                        | VPLS spoke-SDP encap value |
| Padding             | As Required              | As Required               | N/A                        | As Required                |

Each spoke SDP has a Read Only operational state that shows which BPDUs encapsulation is currently active on the spoke SDP. The following states apply:

- **dot1d**

Specifies that the switch is currently sending IEEE 802.1D standard BPDUs. The BPDUs are tagged or non-tagged based on the encapsulation type of the egress interface and the encapsulation value defined in the spoke-SDP. A spoke SDP defined on an interface with encapsulation type dot1q continues in the dot1d BPDUs encapsulation state until a PVST encapsulated BPDUs is received, after which the spoke-SDP converts to the PVST encapsulation state. Each received BPDUs must be properly IEEE 802.1q tagged if the interface encapsulation type is defined as dot1q.

- **PVST**

Specifies that the switch is currently sending proprietary encapsulated BPDUs. PVST BPDUs are only supported on Ethernet interfaces with the encapsulation type set to dot1q. The spoke SDP continues in the PVST BPDUs encapsulation state until a dot1d encapsulated BPDUs is received, in which case the spoke SDP reverts to the dot1d encapsulation state. Each received BPDUs must be properly IEEE 802.1q tagged with the encapsulation value defined for the spoke SDP.

Dot1d is the initial and only spoke-SDP BPDUs encapsulation state for spoke SDPs defined on an Ethernet interface with encapsulation type set to null.

Each transition between encapsulation types optionally generates an alarm that can be logged and optionally transmitted as an SNMP trap.



### 3.5.3.9.3 Configuring VPLS spoke SDPs with split horizon

To configure spoke SDPs with a split horizon group, add the **split-horizon-group** parameter when creating the spoke SDP. Traffic arriving on an SAP or spoke-SDP within a split horizon group is not copied to other SAPs or spoke SDPs in the same split horizon group.

The following example shows a VPLS configuration with split horizon enabled:

```

-----
*A:ALA-1>config>service# *A:ALA-1>config>service# info
-----
...
vpls 800 customer 6001 vpn 700 create
    description "VPLS with split horizon for DSL"
    stp
        shutdown
    exit
    spoke-sdp 51:15 split-horizon-group DSL-group1 create
    exit
    split-horizon-group DSL-group1
        description "Split horizon group for DSL"
    exit
    no shutdown
exit
...
-----
*A:ALA-1>config>service#

```

## 3.5.4 Configuring VPLS redundancy

This section discusses VPLS redundancy service management tasks.

### 3.5.4.1 Creating a management VPLS for SAP protection

This section provides a brief overview of the tasks that must be performed to configure a management VPLS for SAP protection and provides the CLI commands; see [Figure 52: Example configuration for protected VPLS SAP](#). The following tasks should be performed on both nodes providing the protected VPLS service.

Before configuring a management VPLS, read [VPLS redundancy](#) for an introduction to the concept of management VPLS and SAP redundancy.

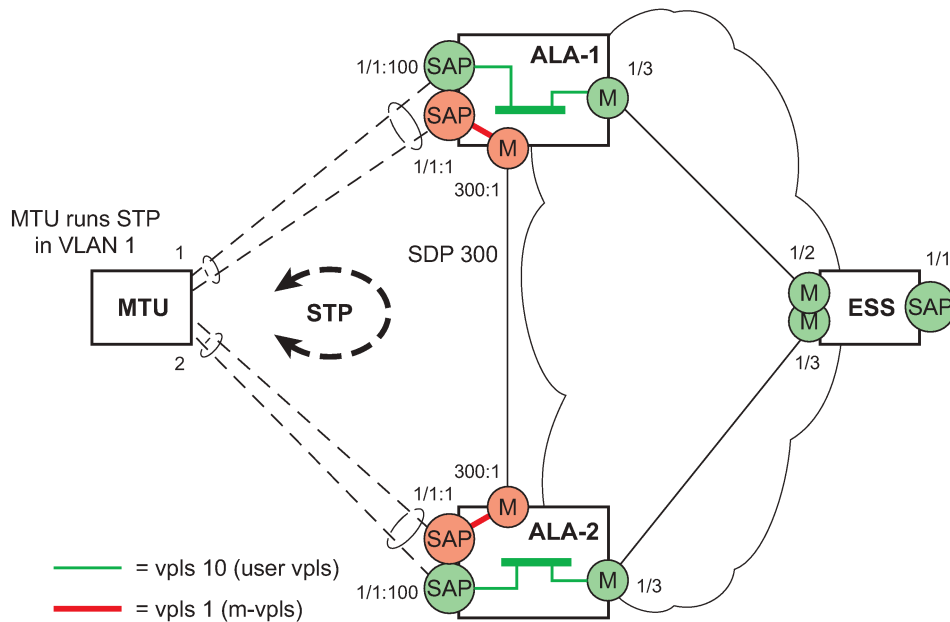
1. Create an SDP to the peer node.
2. Create a management VPLS.
3. Define a SAP in the M-VPLS on the port toward the MTU. The port must be dot1q or QinQ tagged. The SAP corresponds to the (stacked) VLAN on the MTU in which STP is active.
4. Optionally, modify STP parameters for load balancing.
5. Create a mesh SDP in the M-VPLS using the SDP defined in step 1. Ensure that this mesh SDP runs over a protected LSP.
6. Enable the management VPLS service and verify that it is operationally up.
7. Create a list of VLANs on the port that are to be managed by this management VPLS.

8. Create one or more user VPLS services with SAPs on VLANs in the range defined in step 6.



**Note:** The mesh SDP should be protected by a backup LSP or Fast Reroute. If the mesh SDP went down, STP on both nodes would go to forwarding state and a loop would occur.

Figure 52: Example configuration for protected VPLS SAP



OSSG047

Use the following CLI syntax to create a management VPLS.

```
config>service# sdp sdp-id mpls create
- far-end ip-address
- lsp lsp-name
- no shutdown
```

```
vpls service-id customer customer-id [m-vpls] create
- description description-string
- sap sap-id create
  - managed-vlan-list
    - range vlan-range
- mesh-sdp sdp-id:vc-id create
- stp
- no shutdown
```

### Example: VPLS configuration output

```
*A:ALA-1>config>service# info
-----
...
sdp 300 mpls create
  far-end 10.0.0.20
  lsp "toALA-A2"
  no shutdown
exit
```

```

vpls 1 customer 1 m-vpls create
  sap 1/1/1:1 create
    managed-vlan-list
      range 100-1000
    exit
  exit
  mesh-sdp 300:1 create
  exit
  stp
  exit
  no shutdown
exit
...
-----
*A:ALA-1>config>service#

```

### 3.5.4.2 Creating a management VPLS for spoke SDP protection

#### About this task

This section provides a brief overview of the tasks that must be performed to configure a management VPLS for spoke-SDP protection and provides the CLI commands; see [Figure 53: Example configuration for protected VPLS spoke SDP](#). The following tasks should be performed on all four nodes providing the protected VPLS service.

Before configuring a management VPLS, see [Configuring a VPLS SAP](#) for an introduction to the concept of management VPLS and spoke-SDP redundancy.

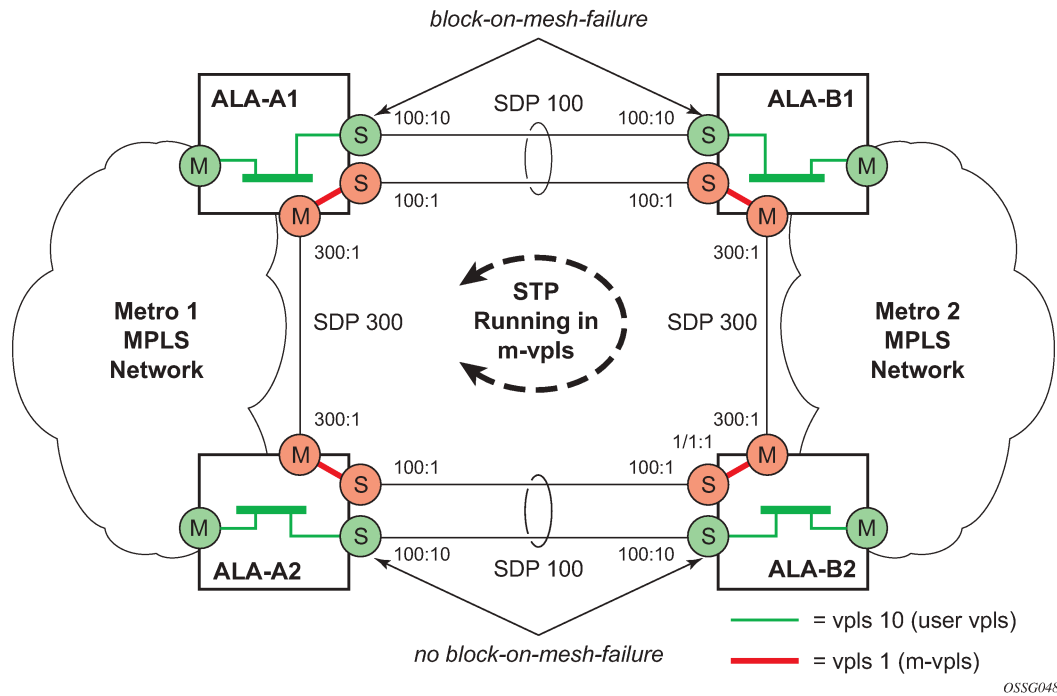
#### Procedure

- Step 1.** Create an SDP to the local peer node (node ALA-A2 in the following example).
- Step 2.** Create an SDP to the remote peer node (node ALA-B1 in the following example).
- Step 3.** Create a management VPLS.
- Step 4.** Create a spoke SDP in the M-VPLS using the SDP defined in step 1. Ensure that this mesh/spoke SDP runs over a protected LSP.
- Step 5.** Enable the management VPLS service and verify that it is operationally up.
- Step 6.** Create a spoke SDP in the M-VPLS using the SDP defined in Step 2. Optionally, modify STP parameters for load balancing (see [Configuring load balancing with management VPLS](#)).
- Step 7.** Create one or more user VPLS services with spoke SDPs on the tunnel SDP defined by step 2. As long as the user spoke SDPs created in step 7 are in this same tunnel SDP with the management spoke SDP created in step 6, the management VPLS protects them.



**Note:** The SDP should be protected by, for example, a backup LSP or Fast Reroute. If the SDP went down, STP on both nodes would go to forwarding state and a loop would occur.

Figure 53: Example configuration for protected VPLS spoke SDP



Use the following CLI syntax to create a management VPLS for spoke-SDP protection.

```
config>service# sdp sdp-id mpls create
- far-end ip-address
- lsp lsp-name
- no shutdown
```

```
vpls service-id customer customer-id [m-vpls] create
- description description-string
- mesh-sdp sdp-id:vc-id create
- spoke-sdp sdp-id:vc-id create
- stp
- no shutdown
```

### Example

#### VPLS configuration output

```
*A:ALA-A1>config>service# info
-----
...
sdp 100 mpls create
  far-end 10.0.0.30
  lsp "toALA-B1"
  no shutdown
exit
sdp 300 mpls create
  far-end 10.0.0.20
  lsp "toALA-A2"
  no shutdown
exit
```

```

vpls 101 customer 1 m-vpls create
spoke-sdp 100:1 create
exit
meshspoke-sdp 300:1 create
exit
stp
exit
no shutdown
exit
...
-----
*A:ALA-A1>config>service#

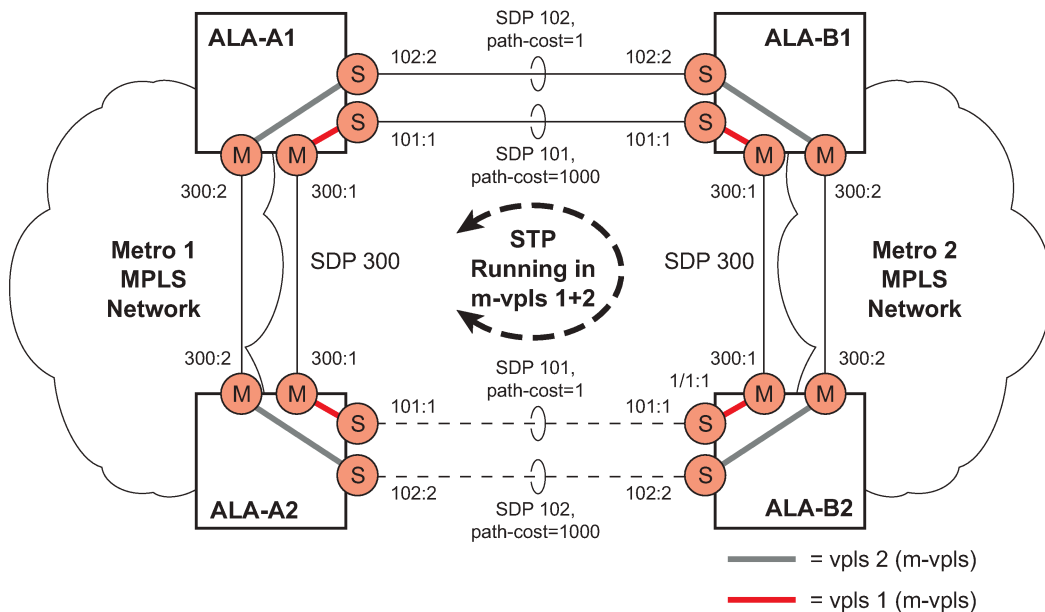
```

### 3.5.4.3 Configuring load balancing with management VPLS

With the concept of management VPLS, it is possible to load balance the user VPLS services across the two protecting nodes. This is done by creating two management VPLS instances, where both instances have different active QinQ spokes (by changing the STP path-cost). When user VPLS services are associated with either of the two management VPLS services, the traffic is split across the two QinQ spokes. Load balancing can be achieved in both the SAP protection and spoke-SDP protection scenarios.

[Figure 54: Example configuration for load balancing across two protected VPLS spoke-SDPs](#) shows an example configuration for load balancing across two protected VPLS spoke-SDPs.

*Figure 54: Example configuration for load balancing across two protected VPLS spoke-SDPs*



Use the following CLI syntax to create load balancing across two management VPLS instances.

CLI syntax:

```

config>service# sdp sdp-id mpls create
far-end ip-address
lsp lsp-name

```

```
no shutdown
```

CLI syntax:

```
vpls service-id customer customer-id [m-vpls] create
- description description-string
- mesh-sdp sdp-id:vc-id create
- spoke-sdp sdp-id:vc-id create
  - stp
    - path-cost
- stp
- no shutdown
```



**Note:** The STP path costs in each peer node should be reversed.

The following example shows the VPLS configuration on ALA-A1 (upper left, IP address 10.0.0.10):

```
*A:ALA-A1>config>service# info
-----
...
    sdp 101 mpls create
        far-end 10.0.0.30
        lsp "1toALA-B1"
        no shutdown
    exit
    sdp 102 mpls create
        far-end 10.0.0.30
        lsp "2toALA-B1"
        no shutdown
    exit
...
    vpls 101 customer 1 m-vpls create
        spoke-sdp 101:1 create
            stp
                path-cost 1
            exit
        exit
        mesh-sdp 300:1 create
        exit
        stp
        exit
        no shutdown
    exit
    vpls 102 customer 1 m-vpls create
        spoke-sdp 102:2 create
            stp
                path-cost 1000
            exit
        exit
        mesh-sdp 300:2 create
        exit
        stp
        exit
        no shutdown
    exit
...
-----
*A:ALA-A1>config>service#
```

The following example shows the VPLS configuration on ALA-A2 (lower left, IP address 10.0.0.20):

```
*A:ALA-A2>config>service# info
-----
...
    sdp 101 mpls create
        far-end 10.0.0.40
        lsp "1toALA-B2"
        no shutdown
    exit
    sdp 102 mpls create
        far-end 10.0.0.40
        lsp "2toALA-B2"
        no shutdown
    exit
...
    vpls 101 customer 1 m-vpls create
        spoke-sdp 101:1 create
            stp
            path-cost 1000
        exit
        exit
        mesh-sdp 300:1 create
        exit
        stp
        exit
    no shutdown
exit
vpls 102 customer 1 m-vpls create
    spoke-sdp 102:2 create
        stp
        path-cost 1
    exit
    exit
    mesh-sdp 300:2 create
    exit
    stp
    exit
no shutdown
exit
...
-----
*A:ALA-A2>config>service#
```

The following example shows the VPLS configuration on ALA-A3 (upper right, IP address 10.0.0.30):

```
*A:ALA-A1>config>service# info
-----
...
    sdp 101 mpls create
        far-end 10.0.0.10
        lsp "1toALA-A1"
        no shutdown
    exit
    sdp 102 mpls create
        far-end 10.0.0.10
        lsp "2toALA-A1"
        no shutdown
    exit
...
    vpls 101 customer 1 m-vpls create
        spoke-sdp 101:1 create
            stp
```

```

        path-cost 1
        exit
    exit
    mesh-sdp 300:1 create
    exit
    stp
    exit
    no shutdown
exit
vpls 102 customer 1 m-vpls create
    spoke-sdp 102:2 create
        stp
        path-cost 1000
    exit
    exit
    mesh-sdp 300:2 create
    exit
    stp
    exit
    no shutdown
exit
...
-----
*A:ALA-A1>config>service#

```

The following example shows the VPLS configuration on ALA-A4 (lower right, IP address 10.0.0.40):

```

*A:ALA-A2>config>service# info
-----
...
    sdp 101 mpls create
        far-end 10.0.0.20
        lsp "1toALA-B2"
        no shutdown
    exit
    sdp 102 mpls create
        far-end 10.0.0.20
        lsp "2toALA-B2"
        no shutdown
    exit
...
vpls 101 customer 1 m-vpls create
    spoke-sdp 101:1 create
        stp
        path-cost 1000
    exit
    exit
    mesh-sdp 300:1 create
    exit
    stp
    exit
    no shutdown
exit
vpls 102 customer 1 m-vpls create
    spoke-sdp 102:2 create
        stp
        path-cost 1
    exit
    exit
    mesh-sdp 300:2 create
    exit
    stp
    exit

```



```

        no shutdown
    exit
    ...
-----
*A:ALA-A2>config>service#

```

#### 3.5.4.4 Configuring selective MAC flush

Use the following CLI syntax to enable selective MAC flush in a VPLS.

```

config>service# vpls service-id
- send-flush-on-failure

```

Use the following CLI syntax to disable selective MAC flush in a VPLS.

```

config>service# vpls service-id
- no send-flush-on-failure

```

#### 3.5.4.5 Configuring multichassis endpoints

The following output shows configuration examples of multichassis redundancy and the VPLS configuration. The configurations in the graphics depicted in [Inter-domain VPLS resiliency using multichassis endpoints](#) are represented in this output.

Node mapping to the following examples in this section:

- PE3 = Dut-B
- PE3' = Dut-C
- PE1 = Dut-D
- PE2 = Dut-E

PE3 Dut-B

```

*A:Dut-B>config>redundancy>multi-chassis# info
-----
    peer 10.1.1.3 create
        peer-name "Dut-C"
        description "mcep-basic-tests"
        source-address 10.1.1.2
        mc-endpoint
            no shutdown
            bfd-enable
            system-priority 50
        exit
    no shutdown
exit
-----
*A:Dut-B>config>redundancy>multi-chassis#

*A:Dut-B>config>service>vpls# info
-----
    fdb-table-size 20000
    send-flush-on-failure
    stp

```

```

        shutdown
    exit
    endpoint "mcep-t1" create
        no suppress-standby-signaling
        block-on-mesh-failure
        mc-endpoint 1
        mc-ep-peer Dut-C
    exit
exit
mesh-sdp 201:1 vc-type vlan create
exit
mesh-sdp 211:1 vc-type vlan create
exit
spoke-sdp 221:1 vc-type vlan endpoint "mcep-t1" create
    stp
        shutdown
        exit
        block-on-mesh-failure
        precedence 1
    exit
spoke-sdp 231:1 vc-type vlan endpoint "mcep-t1" create
    stp
        shutdown
        exit
        block-on-mesh-failure
        precedence 2
    exit
exit
no shutdown

```

-----  
 \*A:Dut-B>config>service>vpls#

### PE3' Dut-C

```

:Dut-C>config>redundancy>multi-chassis# info
-----
    peer 10.1.1.2 create
        peer-name "Dut-B"
        description "mcep-basic-tests"
        source-address 10.1.1.3
        mc-endpoint
            no shutdown
            bfd-enable
            system-priority 21
        exit
        no shutdown
    exit

```

-----  
 \*A:Dut-C>config>redundancy>multi-chassis#

\*A:Dut-C>config>service>vpls# info

```

-----
    fdb-table-size 20000
    send-flush-on-failure
    stp
        shutdown
    exit
    endpoint "mcep-t1" create
        no suppress-standby-signaling
        block-on-mesh-failure
        mc-endpoint 1
        mc-ep-peer Dut-B
    exit
exit

```

```

mesh-sdp 301:1 vc-type vlan create
exit
mesh-sdp 311:1 vc-type vlan create
exit
spoke-sdp 321:1 vc-type vlan endpoint "mcep-t1" create
    stp
        shutdown
    exit
    block-on-mesh-failure
    precedence 3
exit
spoke-sdp 331:1 vc-type vlan endpoint "mcep-t1" create
    stp
        shutdown
    exit
    block-on-mesh-failure
exit
no shutdown
-----
*A:Dut-C>config>service>vpls#

```

### PE1 Dut-D

```

*A:Dut-D>config>redundancy>multi-chassis# info
-----
peer 10.1.1.5 create
    peer-name "Dut-E"
    description "mcep-basic-tests"
    source-address 10.1.1.4
    mc-endpoint
        no shutdown
        bfd-enable
        system-priority 50
        passive-mode
    exit
    no shutdown
exit
-----
*A:Dut-D>config>redundancy>multi-chassis#

*A:Dut-D>config>service>vpls# info
-----
fdb-table-size 20000
propagate-mac-flush
stp
    shutdown
exit
endpoint "mcep-t1" create
    block-on-mesh-failure
    mc-endpoint 1
    mc-ep-peer Dut-E
exit
exit
mesh-sdp 401:1 vc-type vlan create
exit
spoke-sdp 411:1 vc-type vlan endpoint "mcep-t1" create
    stp
        shutdown
    exit
    block-on-mesh-failure
    precedence 2
exit
spoke-sdp 421:1 vc-type vlan endpoint "mcep-t1" create

```

```

        stp
        shutdown
    exit
    block-on-mesh-failure
    precedence 1
exit
mesh-sdp 431:1 vc-type vlan create
exit
no shutdown
-----
*A:Dut-D>config>service>vpls#

```

## PE2 Dut-E

```

*A:Dut-E>config>redundancy>multi-chassis# info
-----
peer 10.1.1.4 create
peer-name "Dut-D"
description "mcep-basic-tests"
source-address 10.1.1.5
mc-endpoint
    no shutdown
    bfd-enable
    system-priority 22
    passive-mode
exit
no shutdown
exit
-----
*A:Dut-E>config>redundancy>multi-chassis#

*A:Dut-E>config>service>vpls# info
-----
fdb-table-size 20000
propagate-mac-flush
stp
    shutdown
exit
endpoint "mcep-t1" create
    block-on-mesh-failure
    mc-endpoint 1
        mc-ep-peer Dut-D
    exit
exit
spoke-sdp 501:1 vc-type vlan endpoint "mcep-t1" create
    stp
        shutdown
    exit
    block-on-mesh-failure
    precedence 3
exit
spoke-sdp 511:1 vc-type vlan endpoint "mcep-t1" create
    stp
        shutdown
    exit
    block-on-mesh-failure
exit
mesh-sdp 521:1 vc-type vlan create
exit
mesh-sdp 531:1 vc-type vlan create
exit
no shutdown

```

```
-----
*A:Dut-E>config>service>vpls#
```

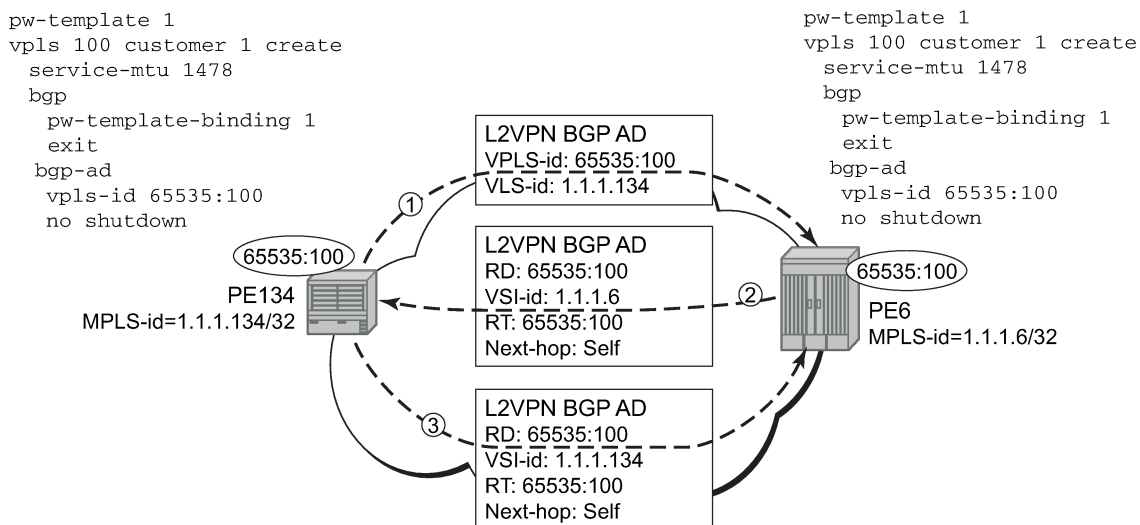
### 3.5.5 Configuring BGP AD

This section describes the different configuration options used to populate the required BGP AD and generate the LDP generalized pseudowire-ID FEC fields. Although several configuration options are available, not all options are required to start using BGP AD. As described in this section, a simple configuration can automatically generate the required values used by BGP and LDP. In most cases, deployments provide full mesh connectivity between all nodes across a VPLS instance. However, capabilities are available to influence the topology, and build hierarchies or hub and spoke models.

#### 3.5.5.1 Configuration steps

Using [Figure 55: BGP AD configuration example](#), assume PE6 was previously configured with VPLS 100 as indicated by the configurations code in the upper right. The BGP AD process commences after PE134 is configured with the VPLS 100 instance, as shown in the upper left. This shows a basic BGP AD configuration. The minimum requirement for enabling BGP AD on a VPLS instance is configuring the VPLS-ID and pointing to a pseudowire template.

*Figure 55: BGP AD configuration example*



OSSG244

In many cases, VPLS connectivity is based on a pseudowire mesh. To reduce the configuration requirement, the BGP values can be automatically generated using the VPLS-ID and the MPLS router-ID. By default, the lower six bytes of the VPLS-ID are used to generate the RD and the RT values. The VSI-ID value is generated from the MPLS router-ID. All of these parameters are configurable and can be coded to suit requirements and build different topologies.

```

PE134>config>service>vpls>bgp-ad#
[no] shutdown - Administratively enable/disable BGP auto-discovery
vpls-id - Configure VPLS-ID

```

## vsi-id + Configure VSI-id

The following command displays the service information, the BGP parameters, and the SDP bindings in use. When the discovery process is completed successfully, each endpoint has an entry for the service.

```
show service l2-route-table
```

```
PE134># show service l2-route-table
=====
Services: L2 Route Information - Summary Service
=====
Svc Id      L2-Routes (RD-Prefix)      Next Hop      Origin
          Sdp Bind Id
-----
100         65535:100-1.1.1.6          1.1.1.6       BGP-L2
          17406:4294967295
-----
No. of L2 Route Entries: 1
=====
PERs6>#

PERs6># show service l2-route-table
=====
Services: L2 Route Information - Summary Service
=====
Svc Id      L2-Routes (RD-Prefix)      Next Hop      Origin
          Sdp Bind Id
-----
100         65535:100-1.1.1.134       1.1.1.134     BGP-L2
          17406:4294967295
-----
No. of L2 Route Entries: 1
=====
PERs6>#
```

When only one of the endpoints has an entry for the service in the l2-routing-table, it is most likely a problem with the RT values used for import and export. This would most likely happen when different import and export RT values are configured using a router policy or the route-target command.

Service-specific commands continue to be available to display service-specific information, including status:

```
show service sdp-using
```

```
PERs6# show service sdp-using
=====
SDP Using
=====
SvcId      SdpId      Type      Far End      Opr S* I.Label E.Label
-----
100         17406:4294967295  BgpAd  10.1.1.134   Up    131063  131067
-----
Number of SDPs : 1
=====
* indicates that the corresponding row element may have been truncated.
```

BGP AD advertises the VPLS-ID in the extended community attribute, VSI-ID in the NLRI, and the local PE ID in the BGP next hop. At the receiving PE, the VPLS-ID is compared against locally provisioned

information to determine whether the two PEs share a common VPLS. If they do, the BGP information is used in the signaling phase (see [Configuring BGP VPLS](#)).

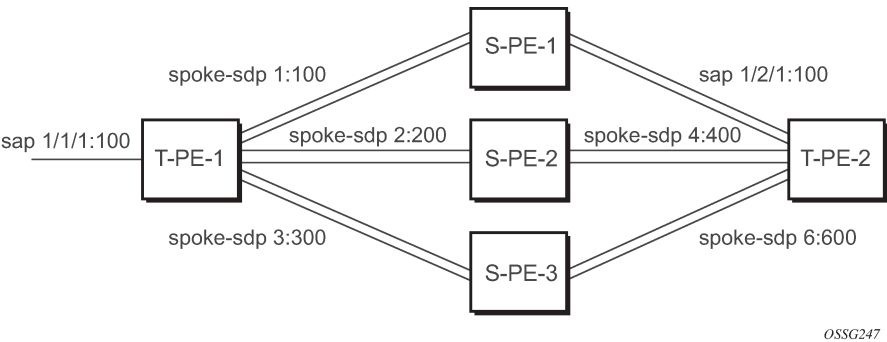
3.5.5.2 LDP signaling

T-LDP is triggered when the VPN endpoints have been discovered using BGP. The T-LDP session between the PEs is established when a session does not exist. The far-end IP address required for the T-LDP identification is learned from the BGP AD next hop information. The pw-template and pw-template-binding configuration statements are used to establish the automatic SDP or to map to the appropriate SDP. The FEC129 content is built using the following values:

- AGI from the locally configured VPLS-ID
- SAll from the locally configured VSI-ID
- TAll from the VSI-ID contained in the last 4 bytes of the received BGP NLRI

[Figure 56: BGP AD triggering LDP functions](#) shows the different detailed phases of the LDP signaling path, post BGP AD completion. It also indicates how some fields can be auto-generated when they are not specified in the configuration.

Figure 56: BGP AD triggering LDP functions



The following command shows the LDP peering relationships that have been established (see [Figure 57: Show router LDP session output](#)). The type of adjacency is displayed in the "Adj Type" column. In this case, the type is "Both" meaning link and targeted sessions have been successfully established.

Figure 57: Show router LDP session output

```
PERs6# show router ldp session
LDP Sessions
```

| Peer LDP Id | Adj Type | State       | Msg Sent | Msg Recv | Up Time     |
|-------------|----------|-------------|----------|----------|-------------|
| 1.1.1.134:0 | Both     | Established | 21482    | 21482    | 0d 15:38:44 |

No. of Sessions: 1

0988

The following command shows the specific LDP service label information broken up per FEC element type: 128 or 129, basis (see [Figure 58: Show router LDP bindings FEC-type services](#)). The information for FEC element 129 includes the AGI, SAll, and the TAll.

**Figure 58: Show router LDP bindings FEC-type services**

```

PERs6# show router ldp bindings fec-type services
LDP LSR ID: 1.1.1.6
Legend: U - Label In Use, N - Label Not In Use, W - Label Withdrawn
        S - Status Signaled Up, D - Status Signaled Down
        E - Epipe Service, V - VPLS Service, M - Mirror Service
        A - Apipe Service, F - Fpipe Service, I - IES Service, R - VPRN service
        P - Ipipe Service, C - Cpipe Service
        TLV - (Type, Length: Value)
LDP Service FEC 128 Bindings

```

| Type                      | VCId | SvcId | SDPId | Peer | IngLbl | EgrLbl | LMTU | RMTU |
|---------------------------|------|-------|-------|------|--------|--------|------|------|
| No Matching Entries Found |      |       |       |      |        |        |      |      |

```

LDP Service FEC 129 Bindings

```

| AGI                |       |       | SAII      |         |           | TAII |      |  |
|--------------------|-------|-------|-----------|---------|-----------|------|------|--|
| Type               | SvcId | SDPId | Peer      | IngLbl  | EgrLbl    | LMTU | RMTU |  |
| 65535:100          |       |       | 1.1.1.6   |         | 1.1.1.134 |      |      |  |
| V-Eth              | 100   | 17406 | 1.1.1.134 | 131063U | 131067S   | 1464 | 1464 |  |
| No. of FEC 129s: 1 |       |       |           |         |           |      |      |  |

0989

### 3.5.5.3 Pseudowire template

The pseudowire template is defined under the top-level **service** command (**config>service>pw-template**) and specifies whether to use an automatically generated SDP or manually configured SDP. It also provides the set of parameters required for establishing the pseudowire (SDP binding) as follows:

```

PERs6>config>service# pw-template 1 create
-[no] pw-template <policy-id> [use-provisioned-sdp | prefer-provisioned-sdp]
<policy-id> : [1..2147483647]
<use-provisioned-s*> : keyword
<prefer-provisioned*> : keyword

[no] accounting-pol*      - Configure accounting-policy to be used
[no] auto-learn-mac*      - Enable/Disable automatic update of MAC protect list
[no] block-on-peer-*      - Enable/Disable block traffic on peer fault
[no] collect-stats        - Enable/disable statistics collection
[no] control word         - Enable/Disable the use of Control Word
[no] disable-aging        - Enable/disable aging of MAC addresses
[no] disable-learn*       - Enable/disable learning of new MAC addresses
[no] discard-unknown*     - Enable/disable discarding of frames with unknown source
                           MAC address
    egress                 + Spoke SDP binding egress configuration
[no] force-qinq-vc-*      - Forces qinq-vc-type forwarding in the data-path
[no] force-vlan-vc-*      - Forces vlan-vc-type forwarding in the data-path
[no] hash-label           - Enable/disable use of hash-label
    igmp-snooping          + Configure IGMP snooping parameters
    ingress               + Spoke SDP binding ingress configuration
[no] l2pt-terminati*      - Configure L2PT termination on this spoke SDP
[no] limit-mac-move       - Configure mac move
[no] mac-pinning          - Enable/disable MAC address pinning on this spoke SDP
[no] max-nbr-mac-ad*      - Configure the maximum number of MAC entries in the FDB
                           from this SDP
[no] restrict-protect*    - Enable/disable protected src MAC restriction
[no] sdp-exclude          - Configure excluded SDP group

```



|                      |                                   |
|----------------------|-----------------------------------|
| [no] sdp-include     | - Configure included SDP group    |
| [no] split-horizon-* | + Configure a split horizon group |
| stp                  | + Configure STP parameters        |
| vc-type              | - Configure VC type               |
| [no] vlan-vc-tag     | - Configure VLAN VC tag           |

A **pw-template-binding** command configured within the VPLS service under the **bgp-ad** sub-command is a pointer to the pw-template that should be used. If a VPLS service does not specify an import-rt list, then that binding applies to all route targets accepted by that VPLS. The **pw-template-bind** command can select a different template on a per import-rt basis. It is also possible to specify specific pw-templates for some route targets with a VPLS service and use the single **pw-template-binding** command to address all unspecified but accepted imported targets.

Figure 59: PW-template-binding CLI syntax

```

PERs6>config>service>vpls>bgp-ad# pw-template-binding
- pw-template-binding <policy-id> [split-horizon-group <group-name>] [import-rt
{ext-community, ...(upto 5 max)}]
- no pw-template-binding <policy-id>

<policy-id>          : [1..2147483647]
<group-name>         : [32 chars max]
<ext-community>      : target:{<ip-addr:comm-val>|<as-number:ext-comm-val>}
ip-addr              - a.b.c.d
comm-val             - [0..65535]
as-number            - [1..65535]
ext-comm-val         - [0..4294967295]

```

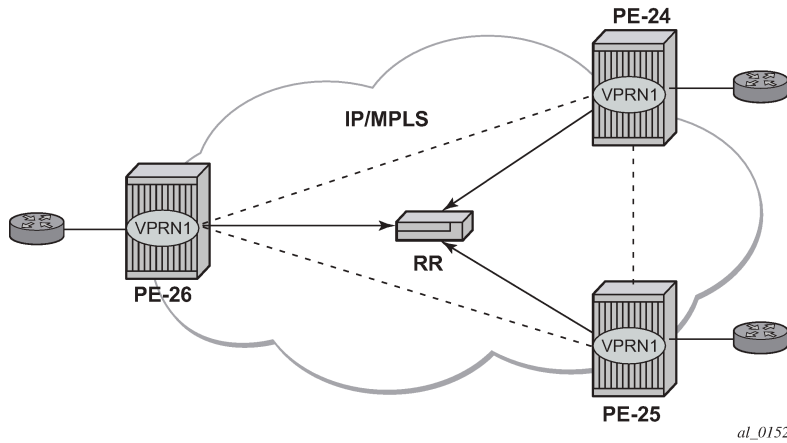
0990

It is important to understand the significance of the split horizon group used by the pw-template. Traditionally, when a VPLS instance was manually created using mesh-SDP bindings, these were automatically placed in a common split horizon group to prevent forwarding between the pseudowire in the VPLS instances. This prevents loops that would have otherwise occurred in the Layer 2 service. When automatically discovering VPLS service using BGP AD, the service provider has the option of associating the auto-discovered pseudowire with a split horizon group to control the forwarding between pseudowires.

### 3.5.6 Configuring BGP VPLS

This section provides a configuration example required to bring up BGP VPLS in the VPLS PEs depicted in [Figure 60: BGP VPLS example](#).

Figure 60: BGP VPLS example



The red BGP VPLS is configured in the PE24, PE25, and PE26 using the commands shown in the following CLI examples:

```
*A:PE24>config>service>vpls# info
-----
    bgp
      route-distinguisher 65024:600
      route-target export target:65019:600 import target:65019:600
      pw-template-binding 1
    exit
    bgp-vpls
      max-ve-id 100
      ve-name 24
      ve-id 24
    exit
    no shutdown
  exit
  sap 1/1/20:600.* create
  exit
  no shutdown
-----

*A:PE24>config>service>vpls#

*A:PE25>config>service>vpls# info
-----
    bgp
      route-distinguisher 65025:600
      route-target export target:65019:600 import target:65019:600
      pw-template-binding 1
    exit
    bgp-vpls
      max-ve-id 100
      ve-name 25
      ve-id 25
    exit
    no shutdown
  exit
  sap 1/1/19:600.* create
  exit
  no shutdown
-----

*A:PE25>config>service>vpls#
```

```
*A:PE26>config>service>vpls# info
-----
      bgp
        route-distinguisher 65026:600
        route-target export target:65019:600 import target:65019:600
        pw-template-binding 1
      exit
    bgp-vpls
      max-ve-id 100
      ve-name 26
      ve-id 26
    exit
    no shutdown
  exit
  sap 5/2/20:600.* create
  exit
  no shutdown
-----
*A:PE26>config>service>vpls#
```

### 3.5.6.1 Configuring a VPLS management interface

Use the following CLI syntax to create a VPLS management interface:

CLI syntax:

```
config>service>vpls# interface ip-int-name
address ip-address[/mask] [netmask]
arp-timeout seconds
description description-string
mac ieee-address
no shutdown
static-arp ip-address ieee-address
```

The following example shows the configuration.

```
A:ALA-49>config>service>vpls>interface# info detail
-----
      no description
      mac 14:31:ff:00:00:00
      address 10.231.10.10/24
      no arp-timeout
      no shutdown
-----
A:ALA-49>config>service>vpls>interface#
```

### 3.5.7 Configuring policy-based forwarding for DPI in VPLS

The purpose of policy-based forwarding is to capture traffic from a customer and perform a deep packet inspection (DPI) and forward traffic, if allowed, by the DPI.

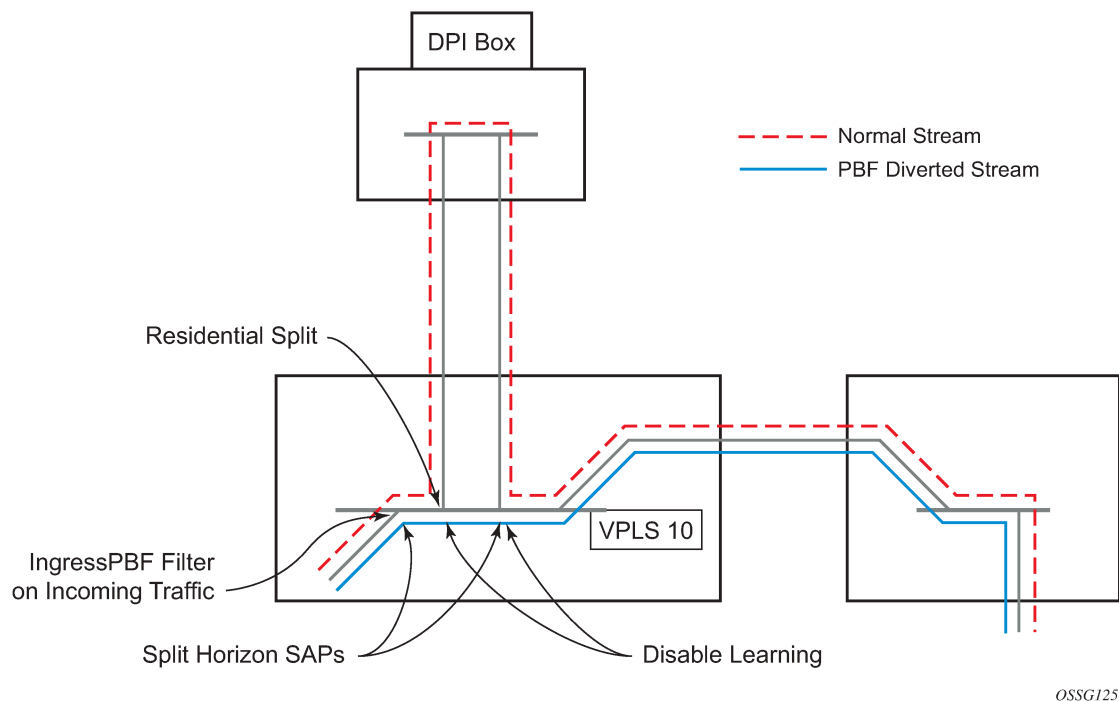
In the following example, the split horizon groups are used to prevent flooding of traffic. Traffic from customers enter at SAP 1/1/5:5. Because of the mac-filter 100 that is applied on ingress, all traffic with dot1p 07 marking is forwarded to SAP 1/1/22:1, which is the DPI.

DPI performs packet inspection/modification and either drops the traffic or forwards the traffic back into the box through SAP 1/1/21:1. Traffic is then sent to spoke-SDP 3:5.

SAP 1/1/23:5 is configured to determine whether the VPLS service is flooding all the traffic. If flooding is performed by the router, traffic would also be sent to SAP 1/1/23:5 (which it should not).

[Figure 61: Policy-based forwarding for deep packet inspection](#) shows an example to configure policy-based forwarding for deep packet inspection on a VPLS service. For information about configuring filter policies, see the *7705 SAR Gen 2 Router Configuration Guide*.

*Figure 61: Policy-based forwarding for deep packet inspection*



The following example shows the service configuration:

```
*A:ALA-48>config>service# info
-----
...
vpls 10 customer 1 create
  service-mtu 1400
  split-horizon-group "dpi" residential-group create
  exit
  split-horizon-group "split" create
  exit
  stp
    shutdown
  exit
  igmp-host-tracking
    expiry-time 65535
    no shutdown
  exit
  sap 1/1/21:1 split-horizon-group "split" create
    disable-learning
    static-mac 00:00:00:31:11:01 create
  exit
```

```

        sap 1/1/22:1 split-horizon-group "dpi" create
        disable-learning
        static-mac 00:00:00:31:12:01 create
    exit
    sap 1/1/23:5 create
        static-mac 00:00:00:31:13:05 create
    exit
    no shutdown
exit
...
-----
*A:ALA-48>config>service#

```

The following example shows the MAC filter configuration:

```

*A:ALA-48>config>filter# info
-----
...
    mac-filter 100 create
    default-action forward
    entry 10 create
        match
            dot1p 7 7
        exit
        log 101
        action forward sap 1/1/22:1
    exit
exit
...
-----
*A:ALA-48>config>filter#

```

The following example shows the service configuration with a MAC filter:

```

*A:ALA-48>config>service# info
-----
...
    vpls 10 customer 1 create
    service-mtu 1400
    split-horizon-group "dpi" residential-group create
    exit
    split-horizon-group "split" create
    exit
    stp
        shutdown
    exit
    igmp-host-tracking
        expiry-time 65535
        no shutdown
    exit
    sap 1/1/5:5 split-horizon-group "split" create
        ingress
            filter mac 100
        exit
        static-mac 00:00:00:31:15:05 create
    exit
    sap 1/1/21:1 split-horizon-group "split" create
        disable-learning
        static-mac 00:00:00:31:11:01 create
    exit
    sap 1/1/22:1 split-horizon-group "dpi" create
        disable-learning
        static-mac 00:00:00:31:12:01 create

```

```

        exit
        sap 1/1/23:5 create
            static-mac 00:00:00:31:13:05 create
        exit
        spoke-sdp 3:5 create
        exit
        no shutdown
    exit
....
-----
*A:ALA-48>config>service#

```

## 3.6 Service management tasks

This section describes the management tasks for VPLS services.

### 3.6.1 Modifying VPLS service parameters

You can change existing service parameters. The changes are applied immediately. To display a list of services, use the **show service service-using vpls** command. Enter the parameter such as description, SAP, SDP, or service-MTU command syntax, then enter the new information.

The following shows a modified VPLS configuration:

```

*A:ALA-1>config>service>vpls# info
-----
    description "This is a different description."
    disable-learning
    disable-aging
    discard-unknown
    local-age 500
    remote-age 1000
    stp
        shutdown
    exit
    sap 1/1/5:22 create
        description "VPLS SAP"
    exit
    spoke-sdp 2:22 create
    exit
    no shutdown
-----
*A:ALA-1>config>service>vpls#

```

### 3.6.2 Modifying management VPLS parameters

To modify the range of VLANs on an access port that are to be managed by an existing management VPLS, the new range should be entered, then the old range removed. If the old range is removed before a new range is defined, all customer VPLS services in the old range become unprotected and may be disabled.

```

config>service# vpls service-id
- sap sap-id
- managed-vlan-list

```

```
– [no] range vlan-range
```

### 3.6.3 Deleting a management VPLS

As with normal VPLS service, a management VPLS cannot be deleted until SAPs and SDPs are unbound (deleted), interfaces are shutdown, and the service is shutdown on the service level.

Use the following CLI syntax to delete a management VPLS service.

```
config>service
[no] vpls service-id
shutdown
[no] spoke-sdp sdp-id
[no] mesh-sdp sdp-id
shutdown
[no] sap sap-id
shutdown
```

### 3.6.4 Disabling a management VPLS

Use the following syntax to shut down a management VPLS without deleting the service parameters.

When a management VPLS is disabled, all associated user VPLS services are also disabled (to prevent loops). If this is not needed, unmanage the user VPLS services by removing them from the managed-vlan-list or moving the spoke-SDPs to another tunnel SDP.

```
config>service
vpls service-id
– shutdown
```

#### Example

```
config>service# vpls 1
config>service>vpls# shutdown
config>service>vpls# exit
```

### 3.6.5 Deleting a VPLS service

A VPLS service cannot be deleted until SAPs and SDPs are unbound (deleted), interfaces are shutdown, and the service is shut down on the service level.

Use the following CLI syntax to delete a VPLS service.

```
config>service
[no] vpls service-id
shutdown
[no] mesh-sdp sdp-id
shutdown
sap sap-id [split-horizon-group group-name]
no sap sap-id
shutdown
```

### 3.6.6 Disabling a VPLS service

Use the following syntax to shut down a VPLS service without deleting the service parameters.

```
config>service> vpls service-id  
[no] shutdown
```

#### Example

```
config>service# vpls 1  
config>service>vpls# shutdown  
config>service>vpls# exit
```

### 3.6.7 Re-enabling a VPLS service

Use the following syntax to re-enable a VPLS service that was shut down.

```
config>service> vpls service-id  
[no] shutdown
```

#### Example

```
config>service# vpls 1  
config>service>vpls# no shutdown  
config>service>vpls# exit
```



## 4 Layer 2 control protocols

SR OS has awareness of multiple Layer 2 Control Protocols (L2CP). In some configurations, these L2CP frames are extracted and processed by the receiving router. However, it is useful in some deployments to have the router transparently forward the L2CP frames through a Layer 2 VLL or VPLS service. SR OS support for transparent tunneling is configured on a protocol basis, as described in this section.

L2CP frames are processed as follows:

- tunneled - pass the frames through any associated service
- peered - extract and process the frame
- discarded - extract the frame and then discard it

The following L2CP frames are always tunneled:

- **STP/RSTP/MSTP**

This applies only if Spanning Tree is not enabled if the ingress SAP is attached to a VPLS service. These frames are identified as frames with a destination MAC address of 01:80:C2:00:00:00.

- **LAMP**

These frames are identified as frames with a destination MAC address of 01:80:C2:00:00:02, Ethertype (0x8809), and slow-protocol subtype 0x02.

- **MAC specific control protocols**

These frames are identified as frames with a destination MAC address of 01:80:C2:00:00:04.

- **provider bridge group address 01:80:C2:00:00:08**

These frames are identified as frames with a destination MAC address of 01:80:C2:00:00:08

- **provider bridge MVRP address 01:80:C2:00:00:0D**

These frames are identified as frames with a destination MAC address of 01:80:C2:00:00:0D

Other frame types are processed according to the CLI configuration, as follows:

- **PAUSE frames**

PAUSE frames are transmitted to request backward direction flow control. By default, SR OS peers pause frames on reception and pause the transmit side of the port. On some ports, this behavior can be changed to discard PAUSE frames. For applicable ports, use the following command to discard PAUSE frames:

- **MD-CLI**

```
configure port ethernet discard-rx-pause-frames true
```

- **classic CLI**

```
configure port ethernet discard-rx-pause-frames
```

PAUSE frames are never tunneled.

PAUSE frames are identified as frames with a destination MAC address of 01:80:C2:00:00:01, Ethertype (0x8808), and subtype 0x0001.

- **LACP frames**

If the port is part of a LAG, the LACP frames are peered. If the port is not part of a LAG, the LACP frames are discarded or tunneled based on the configuration of the following command.

```
configure port ethernet lacp-tunnel
```

LACP frames are identified as frames with a destination MAC address of 01:80:C2:00:00:02, Ethertype (0x8809), and the slow-protocol subtype (0x01).

- **EFM-OAM frames**

If the port has EFM-OAM processing enabled, the EFM-OAM frames are peered. If the port does not have EFM-OAM processing enabled, the EFM-OAM frames are discarded or tunneled based on the configuration of the following command.

```
configure port ethernet efm-oam tunneling
```

EFM-OAM frames are identified as frames with a destination MAC address of 01:80:C2:00:00:02, Ethertype (0x8809), and the slow-protocol sub-type (0x03).

- **ESMC frames**

If the port is an input reference to the central frequency clock, the ESMC frames are peered. If the port is not an input reference to the central frequency clock, the ESMC frames are discarded. Use the following command to override the preceding scenarios and tunnel the ESMC frames:

- **MD-CLI**

```
configure port ethernet ssm esmc-tunnel true
```

- **classic CLI**

```
configure port ethernet ssm esmc-tunnel
```

ESMC frames are identified as frames with a destination MAC address of 01:80:C2:00:00:02, Ethertype (0x8809), and the slow-protocol sub-type (0x0A).

- **802.1x frames**

By default, the router extracts 802.1x frames when they are received. For extracted frames, if a RADIUS server is configured, the frames are peered; otherwise, they are discarded. Use the following command to override the extraction and tunnel the 802.1x frames:

- **MD-CLI**

```
configure port ethernet dot1x tunneling true
```

- **classic CLI**

```
configure port ethernet dot1x tunneling
```

802.1x frames are identified as frames with a destination MAC address of 01:80:C2:00:00:03, and Ethertype (0x888E).

- **E-LMI frames**

If Ethernet Local Management Interface (E-LMI) processing is enabled on the port, E-LMI frames are peered. Otherwise, the E-LMI frames are dropped from VPLS and tunneled for Epipe.

E-LMI frames from VPLS are tunneled when:

- the following command is included in the service

```
configure service vpls tunnel-elmi
```

- the following command is not enabled on the port

```
configure port ethernet elmi
```

- the encapsulation of the E-LMI frame matches the VPLS service tagging

E-LMI frames are identified as frames with a destination MAC address of 01:80:C2:00:00:07, and Ethertype (0x88EE).

- **LLDP frames**

If LLDP processing is enabled on the port, LLDP frames are peered. Otherwise, LLDP frames are discarded or tunneled based on the configuration of the following commands.

```
configure port ethernet lldp dest-mac tunnel-nearest-bridge
configure port ethernet lldp dest-mac tunnel-nearest-customer
configure port ethernet lldp dest-mac tunnel-nearest-non-tpmr
```

LLDP frames are identified as frames with a destination MAC address depending on the tunnel configuration:.

- nearest bridge - 01:80:C2:00:00:0E and Ethertype (0x88CC)
- nearest customer - 01:80:C2:00:00:00 and Ethertype (0x88CC)
- nearest non-TPMR - 01:80:C2:00:00:03 and Ethertype (0x88CC)

- **PTP peer delay frames**

If the port is configured in the router as an active port within the PTP process, the frames are peered; otherwise, the frames are tunneled.

PTP message frames are identified as frames with destination MAC address 01:80:C2:00:00:0E, and Ethertype (0x88F7).



**Note:** E-LMI must be disabled and the port must not be configured as a PTP port.

Configure the following commands on the port to achieve the maximum transparency of L2CP frames:

- **MD-CLI**

```
configure port ethernet discard-rx-pause-frames true
configure port ethernet lacp-tunnel true
configure port ethernet efm-oam tunneling true
configure port ethernet ssm esmc-tunnel true
configure port ethernet dot1x tunneling true
configure port ethernet lldp dest-mac tunnel-nearest-bridge true
```

- **classic CLI**

```
configure port ethernet discard-rx-pause-frames
configure port ethernet lacp-tunnel
configure port ethernet efm-oam tunneling
configure port ethernet ssm esmc-tunnel
```

---

```
configure port ethernet dot1x tunneling
configure port ethernet lldp dest-mac tunnel-nearest-bridge
```

## 5 Ethernet Virtual Private Networks

This chapter provides information about Ethernet Virtual Private Networks (EVPN).

### 5.1 Overview of EVPN applications

EVPN is an IETF technology as defined in RFC 7432, *BGP MPLS-Based Ethernet VPN*, that uses a specific BGP address family and allows VPLS services to be operated as IP-VPNs, where the MAC addresses and the information to set up the flooding trees are distributed by BGP.

EVPN is defined to fill the gaps of other L2VPN technologies such as VPLS. The main objective of the EVPN is to build E-LAN services in a similar way to RFC 4364 IP-VPNs, while supporting MAC learning within the control plane (distributed by MP-BGP), efficient multidestination traffic delivery, and active/active multihoming.

EVPN can be used as the control plane for different data plane encapsulations. The Nokia implementation supports the following data planes:

- **EVPN for VXLAN overlay tunnels (EVPN-VXLAN)**

EVPN for VXLAN overlay tunnels (EVPN-VXLAN), being the Data Center Gateway (DGW) function, is the main application for this feature. In this application, VXLAN is expected within the Data Center and VPLS SDP bindings or SAPs are expected for WAN connectivity. R-VPLS and VPRN connectivity to the WAN is also supported.

The EVPN-VXLAN functionality is standardized in RFC 8365.

- **EVPN for MPLS tunnels (EVPN-MPLS)**

EVPN-MPLS is supported where PEs are connected by any type of MPLS tunnel. EVPN-MPLS is generally used as an evolution for VPLS services in the WAN, and Data Center Interconnect is one of the main applications.

The EVPN-MPLS functionality is standardized in RFC 7432.

- **EVPN for PBB over MPLS tunnels (PBB-EVPN)**

PEs are connected by PBB over MPLS tunnels in this data plane. It is usually used for large scale E-LAN and E-Line services in the WAN.

The PBB-EVPN functionality is standardized in RFC 7623.

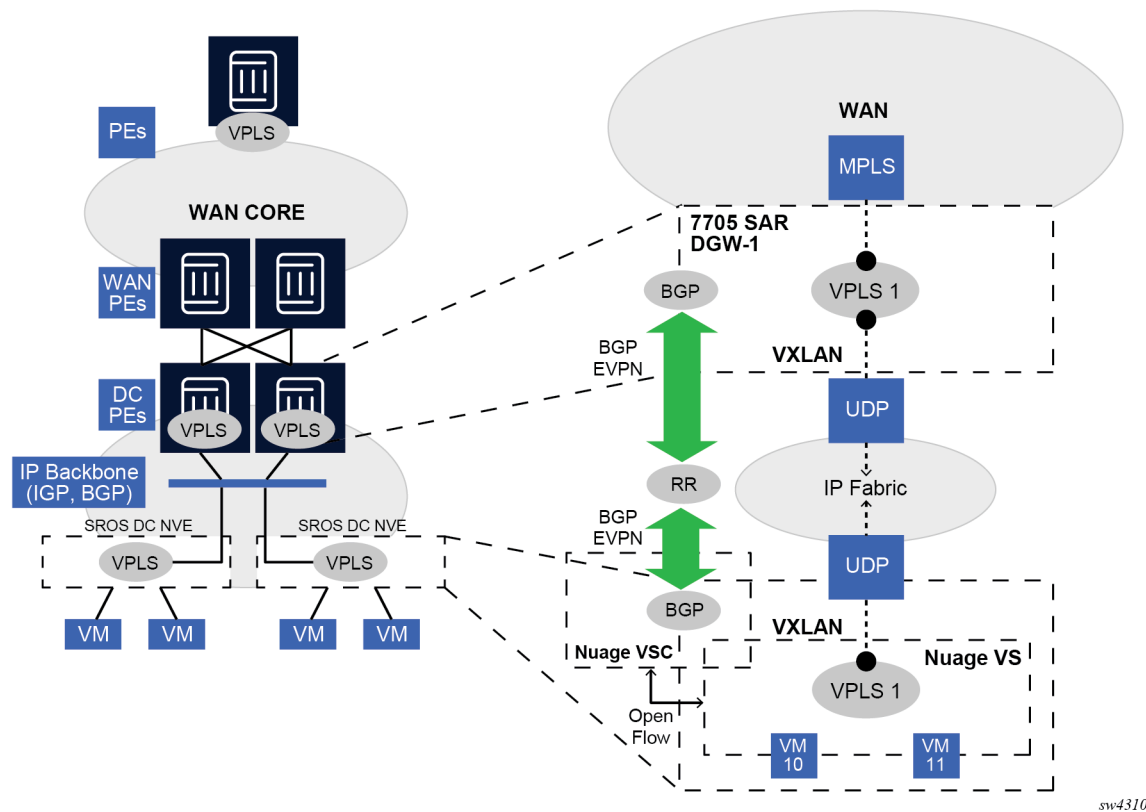
The 7705 SAR Gen 2 EVPN VXLAN implementation is integrated in the Nuage Data Center architecture, where the router serves as the DGW.

For more information about the Nuage Networks architecture and products, see the *Nuage Networks Virtualized Service Platform Guide*. The following sections describe the applications supported by EVPN.

#### 5.1.1 EVPN for VXLAN tunnels in a Layer 2 DGW (EVPN-VXLAN)

The following figure shows the use of EVPN for VXLAN overlay tunnels when it is used as a Layer 2 DGW.

Figure 62: Layer 2 DC PE with VPLS to the WAN



Data Center (DC) providers require a DGW solution that can extend tenant subnets to the WAN. Customers can deploy the NVO3-based solutions in the DC, where EVPN is the standard control plane and VXLAN is a predominant data plane encapsulation. The Nokia DC architecture uses EVPN and VXLAN as the control and data plane solutions for Layer 2 connectivity within the DC and so does the SR OS.

While EVPN VXLAN is used within the DC, some service providers use VPLS and H-VPLS as the solution to extend Layer 2 VPN connectivity. [Figure 62: Layer 2 DC PE with VPLS to the WAN](#) shows the Layer 2 DGW function on the 7705 SAR Gen 2 routers, providing VXLAN connectivity to the DC and regular VPLS connectivity to the WAN.

The WAN connectivity is based on VPLS where SAPs (null, dot1q, and qinq), spoke SDPs (FEC type 128 and 129), and mesh-SDPs are supported.

The DC GWs can provide multihoming resiliency through the use of BGP multihoming.

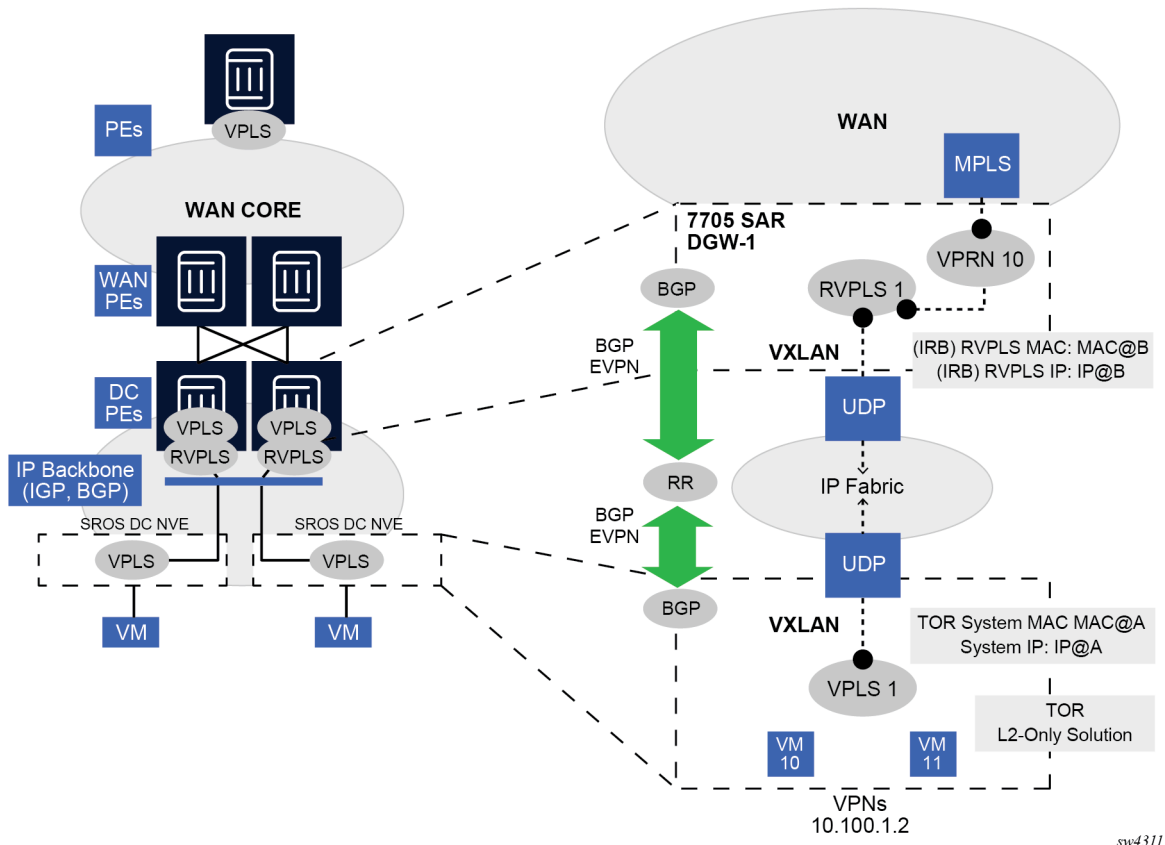
EVPN-MPLS can also be used in the WAN. In this case, the Layer 2 DGW function provides translation between EVPN-VXLAN and EVPN-MPLS. EVPN multihoming can be used to provide DGW redundancy.

If point-to-point services are needed in the DC, SR OS supports the use of EVPN-VPWS for VXLAN tunnels, including multihoming, in accordance with RFC 8214.

### 5.1.2 EVPN for VXLAN tunnels in a Layer 2 DC with integrated routing bridging connectivity on the DGW

Figure 63: Gateway IRB on the DC PE for an L2 EVPN/VXLAN DC shows the use of EVPN for VXLAN overlay tunnels when the DC provides Layer 2 connectivity and the DGW can route the traffic to the WAN through an R-VPLS and linked VPRN.

Figure 63: Gateway IRB on the DC PE for an L2 EVPN/VXLAN DC

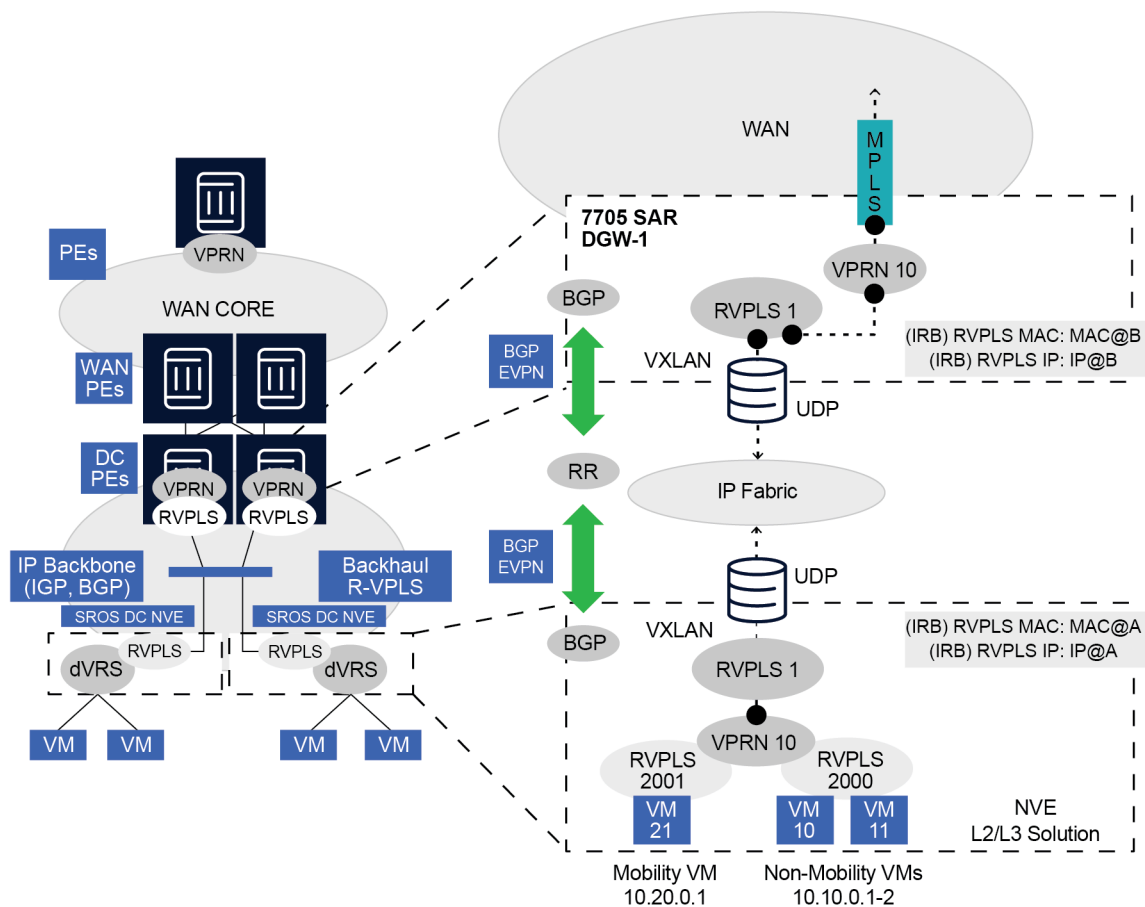


In some cases, the DGW must provide a Layer 3 default gateway function to all the hosts in a specified tenant subnet. In this case, the VXLAN data plane is terminated in an R-VPLS on the DGW, and connectivity to the WAN is accomplished through regular VPRN connectivity. The 7705 SAR Gen 2 supports IPv4 and IPv6 interfaces as default gateways in this scenario.

### 5.1.3 EVPN for VXLAN tunnels in a Layer 3 DC with integrated routing bridging connectivity among VPRNs

The following figure shows the use of EVPN for VXLAN tunnels when the DC provides distributed Layer 3 connectivity to the DC tenants.

Figure 64: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC



sw4312

Each tenant has several subnets for which each DC Network Virtualization Edge (NVE) provides intra-subnet forwarding. An NVE may be a Nokia VSG (Virtual Switch Gateway), VSC (Virtual Switch Controller)/VRS (Virtual Routing and Switching), or any other NVE in the market supporting the same constructs, and each subnet typically corresponds to an R-VPLS. For example, in the preceding figure, 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 also provides inter-subnet forwarding by connecting all the local subnets to a VPRN instance. When the tenant requires Layer 3 connectivity to the IP-VPN in the WAN, a VPRN is defined in the DC GWs, which connects the tenant to the WAN. That VPRN instance is connected to the VPRNs in the NVEs by means of an Integrated Routing and Bridging (IRB) backhaul R-VPLS. This IRB backhaul R-VPLS provides a scalable solution because it allows Layer 3 connectivity to the WAN without the need for defining all of the subnets in the DC gateway.

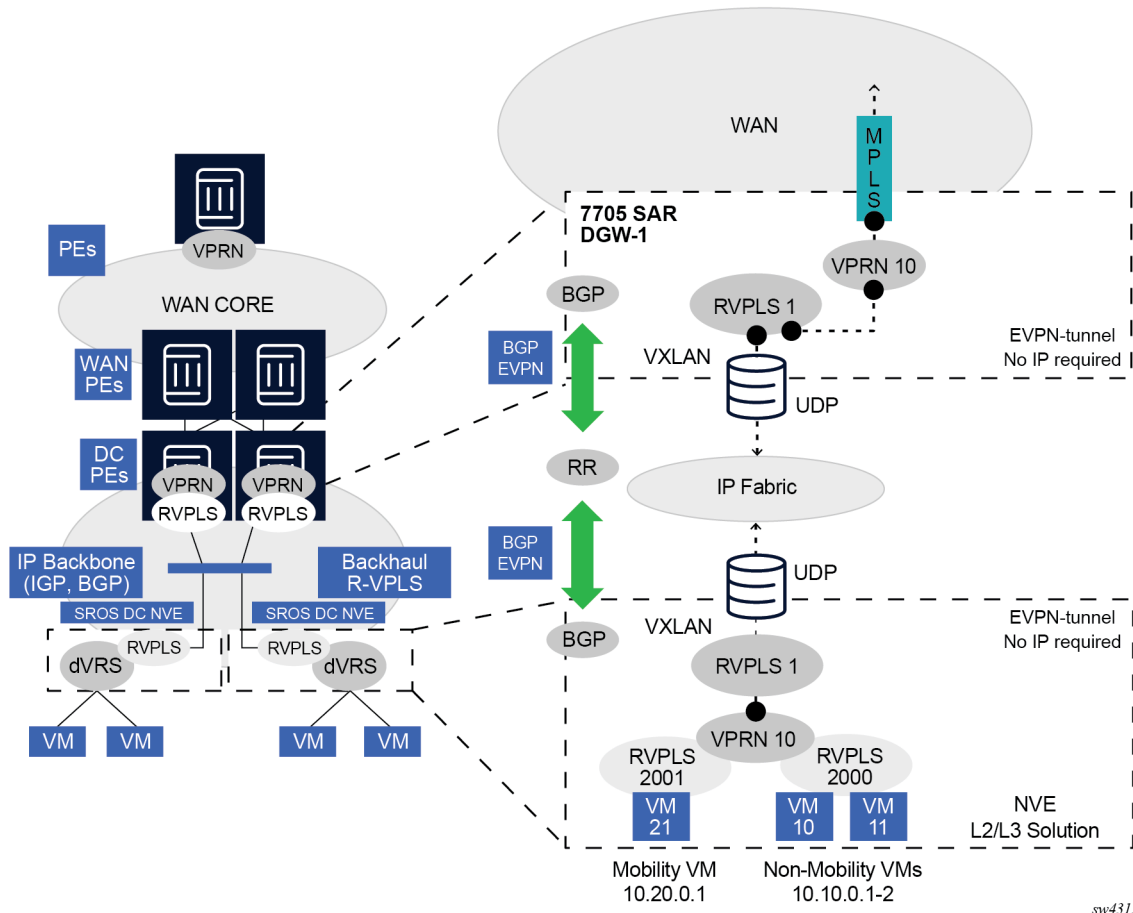
The 7705 SAR Gen 2 DGW supports the IRB backhaul R-VPLS model, where the R-VPLS runs EVPN-VXLAN and the VPRN instances exchange IP prefixes (IPv4 and IPv6) through the use of EVPN. Interoperability between the EVPN and IP-VPN for IP prefixes is also fully supported.



### 5.1.4 EVPN for VXLAN tunnels in a Layer 3 DC with EVPN-tunnel connectivity among VPRNs

The following figure shows the use of EVPN for VXLAN tunnels when the DC provides distributed Layer 3 connectivity to the DC tenants and the VPRN instances are connected through EVPN tunnels.

Figure 65: EVPN-tunnel gateway IRB on the DC PE for a Layer 3 EVPN/VXLAN DC



The solution described in section [EVPN for VXLAN tunnels in a Layer 3 DC with integrated routing bridging connectivity among VPRNs](#) provides a scalable IRB backhaul R-VPLS service where IRB interfaces can be used to connect all the VPRN instances for a specified tenant. 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. Provisioning is 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**

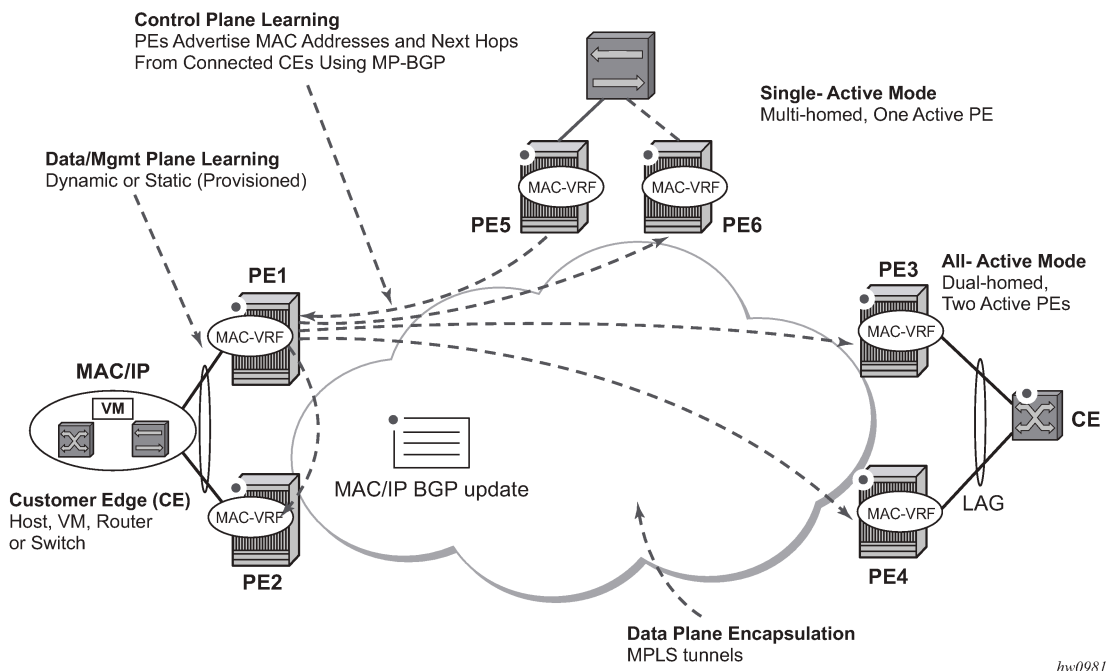
If EVPN tunnels are enabled, multicast traffic is suppressed in the EVPN-VXLAN IRB backhaul R-VPLS service (it is not required). As a result, the number of VXLAN binds in IRB backhaul R-VPLS services with EVPN-tunnels can be much higher.

This optimization is fully supported by the 7705 SAR Gen 2.

### 5.1.5 EVPN for MPLS tunnels in E-LAN services

The following figure shows the use of EVPN for MPLS tunnels. In this case, EVPN is used as the control plane for E-LAN services in the WAN.

Figure 66: EVPN for MPLS in VPLS services



As defined in RFC 7432, EVPN-MPLS is an L2VPN technology that can fill the gaps in VPLS for E-LAN services. Service providers that offer E-LAN services request EVPN for its multihoming capabilities and to leverage the optimization EVPN provides.

EVPN supports both all-active multihoming (per flow load-balancing multihoming) as well as single-active multihoming (per-service load-balancing multihoming). Although VPLS already supports single-active multihoming, EVPN single-active multihoming is deemed the superior technology because of its mass-withdrawal capabilities to speed up convergence in scaled environments.

EVPN technology provides significant benefits, including:

- superior multihoming capabilities

- IP-VPN-like operation and control for E-LAN services
- reduction and (in some cases) suppression of the broadcast, unknown unicast, and multicast (BUM) traffic in the network
- simple provision and management
- new set of tools to control the distribution of MAC addresses and ARP entries in the network

The SR OS EVPN-MPLS implementation is compliant with RFC 7432.

EVPN-MPLS can also be enabled in R-VPLS services with the same feature-set that is described for VXLAN tunnels in sections and [EVPN for VXLAN tunnels in a Layer 3 DC with integrated routing bridging connectivity among VPRNs](#) and [EVPN for VXLAN tunnels in a Layer 3 DC with EVPN-tunnel connectivity among VPRNs](#).

### 5.1.6 EVPN for MPLS tunnels in E-Line services

The MPLS network can be shared between EVPN for E-LAN and E-Line services using EVPN in the control plane. EVPN for E-Line services (EVPN-VPWS) is a simplification of the RFC 7432 procedures, and it is supported in compliance with RFC 8214.

### 5.1.7 EVPN for MPLS tunnels in E-Tree services

The MPLS network used by E-LAN and E-Line services can also be shared by Ethernet-Tree (E-Tree) services using the EVPN control plane. EVPN E-Tree services use the EVPN control plane extensions described in IETF RFC 8317 and are supported on the 7705 SAR Gen 2.

## 5.2 EVPN for VXLAN tunnels and cloud technologies

This section provides information about EVPN for VXLAN tunnels and cloud technologies.

### 5.2.1 VXLAN

The SR OS, SR Linux and Nuage solution for DC supports VXLAN (Virtual eXtensible Local Area Network) overlay tunnels as per RFC 7348.

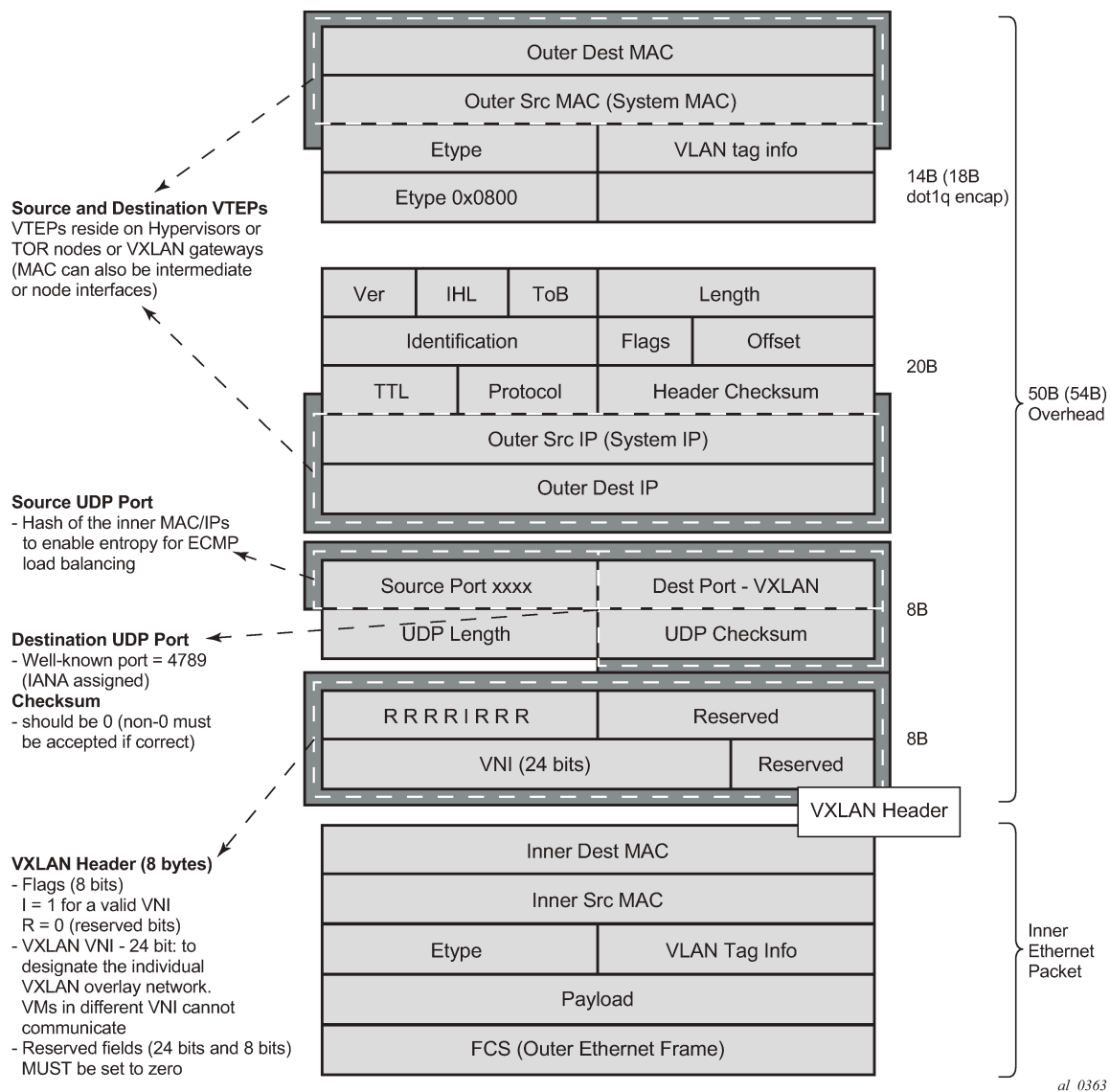
VXLAN addresses the data plane needs for overlay networks within virtualized DCs accommodating multiple tenants. The main attributes of the VXLAN encapsulation are the following:

- VXLAN is an overlay network encapsulation used to carry MAC traffic between VMs over a logical Layer 3 tunnel.
- VXLAN encapsulation 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).
- It supports multipath 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.

- It 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.

The following figure shows an example of the VXLAN encapsulation supported by the Nokia implementation.

Figure 67: VXLAN frame format



VXLAN encapsulates the inner Ethernet frames into VXLAN + UDP/IP packets. The main pieces of information encoded in this encapsulation are the following:

- 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 remaining flags ("Reserved" bits) are set to 0.

- Includes the VNI field (24-bit value) or VXLAN network identifier that identifies an isolated Layer 2 domain within the DC network.
- Remaining fields are reserved for future use.
- **UDP header (8 bytes)**
  - 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 7705 SAR Gen 2 does at ingress. This creates an “entropy” value that can be used by the core DC nodes for load balancing on ECMP paths.
  - The checksum is set to zero.
- **Outer IP and Ethernet headers (34 or 38 bytes)**
  - The source IP and source MAC addresses identify the source VTEP. That is, these fields are populated with the PE system IP and chassis MAC address.



**Note:** The source MAC address is changed on all the IP hops along the path, as is typical in regular IP routing.

- The destination IP identifies the remote VTEP (remote system IP) and is the result of the destination MAC lookup in the service Forwarding Database (FDB).



**Note:** All remote MACs are learned by the EVPN BGP and associated with a remote VTEP address and VNI.

Some considerations related to the support of VXLAN are:

- VXLAN is only supported on network or hybrid ports with null or dot1q encapsulation.
- VXLAN is supported on Ethernet/LAG and POS/APS.
- IPv4 and IPv6 unicast addresses are supported as VTEPs.
- By default, system IP addresses are supported, as VTEPs, for originating and terminating VXLAN tunnels. Non-system IPv4 and IPv6 addresses are supported by using a Forwarding Path Extension (FPE).

### 5.2.1.1 VXLAN ECMP and LAG

The DGW supports ECMP load balancing to reach the destination VTEP. Also, any intermediate core node in the DC should be able to provide further load balancing across ECMP paths, because the source UDP port of each tunneled packet is derived from a hash of the customer inner packet. The following must be considered:

- ECMP for VXLAN is supported on VPLS services but not for BUM traffic. Unicast spraying is based on the packet contents.
- ECMP for VXLAN on R-VPLS services is supported for VXLAN IPv6 tunnels.
- ECMP for VXLAN IPv4 tunnels on R-VPLS is only supported if the **configure service vpls allow-ip-int-bind vxlan-ipv4-tep-ecmp** command is enabled on the R-VPLS (as well as **configure router ecmp**).
- ECMP for Layer 3 multicast traffic on R-VPLS services with EVPN-VXLAN destinations is only supported if the **vpls allow-ip-int-bind ip-multicast-ecmp** command is enabled (as well as **configure router ecmp**).

- In the cases where ECMP is not supported (BUM traffic in VPLS and ECMP on R-VPLS if not enabled), each VXLAN binding is tied to a single (different) ECMP path, so that in a normal deployment with a reasonable number of remote VTEPs, there should be a fair distribution of the traffic across the paths. That is, only per-VTEP load-balancing is supported, instead of per-flow load-balancing.
- LAG spraying based on the packet hash is supported in all the cases (VPLS unicast, VPLS BUM, and R-VPLS).

### 5.2.1.2 VXLAN VPLS tag handling

The following describes the behavior with respect to VLAN tag handling for VXLAN VPLS services:

- Dot1q, QinQ, and null SAPs, as well as regular VLAN-handling procedures at the WAN side, are supported on VXLAN VPLS services.
- No "vc-type vlan" like VXLAN VNI bindings are supported. Therefore, at the egress of the VXLAN network port, the router does not add any inner VLAN tag on top of the VXLAN encapsulation, and at the ingress network port, the router ignores any VLAN tag received and handles it as part of the payload.

### 5.2.1.3 VXLAN MTU considerations

For VXLAN VPLS services, the network port MTU must be at least 50 Bytes (54 Bytes if dot1q) greater than the service MTU to allow enough room for the VXLAN encapsulation.

The service MTU is only enforced on SAPs (any SAP ingress packet with MTU greater than the service MTU is discarded) and not on VXLAN termination (any VXLAN ingress packet makes it to the egress SAP regardless of the configured service MTU).

If BGP-EVPN is enabled in a VXLAN VPLS service, the service MTU can be advertised in the Inclusive Multicast Ethernet Tag routes and enforce that all the routers attached to the same EVPN service have the same service MTU configured.



**Note:** The router never fragments or reassembles the VXLAN packets. In addition, the DF (Do not Fragment) flag is always set in the VXLAN outer IP header.

### 5.2.1.4 VXLAN QoS

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

#### 5.2.1.4.1 Ingress

The network ingress QoS policy can be applied either to the network interface over which the VXLAN traffic arrives or under **vlan/network/ingress** within the EVPN service.

Regardless of where the network QoS policy is applied, the ingress network QoS policy is used to classify the VXLAN packets based on the outer dot1p (if present), then the outer DSCP, to yield an FC/profile.

If the ingress network QoS policy is applied to the network interface over which the VXLAN traffic arrives then the VXLAN unicast traffic uses the network ingress queues configured on FP where the network interface resides. QoS control of BUM traffic received on the VXLAN tunnels is possible by separately

redirecting these traffic types to policers within an FP ingress network queue group. This QoS control uses the per forwarding class **fp-redirect-group** parameter together with **broadcast-policer**, **unknown-policer**, and **mcast-policer** within the ingress section of a network QoS policy. This QoS control applies to all BUM traffic received for that forwarding class on the network IP interface on which the network QoS policy is applied.

The ingress network QoS policy can also be applied within the EVPN service by referencing an FP queue group instance, as follows:

```
configure
  service
    vpls <service-id>
      vxlan vni <vni-id>
        network
          ingress
            qos <network-policy-id>
              fp-redirect-group <queue-group-name>
                instance <instance-id>
```

In this case, the redirection to a specific ingress FP queue group applies as a single entity (per forwarding class) to all VXLAN traffic received only by this service. This overrides the QoS applied to the related network interfaces for traffic arriving on VXLAN tunnels in that service but does not affect traffic received on a spoke SDP in the same service. It is possible to also redirect unicast traffic to a policer using the per forwarding class **fp-redirect-group policer** parameter, as well as the BUM traffic as above, within the ingress section of a network QoS policy. The use of **ler-use-dscp**, **ip-criteria** and **ipv6-criteria** statements are ignored if configured in the ingress section of the referenced network QoS policy. If the instance of the named queue group template referenced in the **qos** command is not configured on an FP receiving the VXLAN traffic, then the traffic uses the ingress network queues or queue group related to the network interface.

#### 5.2.1.4.2 Egress

On egress, there is no need to specify "remarking" in the policy to mark the DSCP. This is because the VXLAN adds a new IPv4 header, and the DSCP is always marked based on the egress network qos policy.

#### 5.2.1.5 VXLAN ping

A new VXLAN troubleshooting tool, VXLAN Ping, is available to verify VXLAN VTEP connectivity. The **VXLAN Ping** command is available from interactive CLI and SNMP.

This tool allows the user 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 user to specify a different **test-id** (equates to originator handle) for each active and outstanding test. The required local **service** identifier from which the test is launched determines the source IP (the system IP address) to use in the outer IP header of the packet. This IP address is encoded into the VXLAN header Source IP TLV. The service identifier also encodes the local VNI. The **outer-ip-destination** must equal the VTEP termination point on the remote node, and the **dest-vni** must be a valid VNI within the associated service on the remote node. The outer source IP address is automatically detected and inserted in the IP header of the packet. The outer source IP address uses the IPv4 system address by default.

If the VTEP is created using a non-system source IP address through the **vxlan-src-vtep** command, the outer source IP address uses the address specified by **vxlan-src-vtep**. The remainder of the variables are optional.



The VXLAN PDU is encapsulated in the appropriate transport header and forwarded within the overlay to the appropriate VTEP termination. The VXLAN router alert (RA) bit is set to prevent forwarding OAM PDU beyond the terminating VTEP. Because handling of the router alert bit was not defined in some early releases of VXLAN implementations, the VNI Informational bit (I-bit) is set to "0" for OAM packets. This indicates that the VNI is invalid, and the packet should not be forwarded. This safeguard can be overridden by including the **i-flag-on** option that sets the bit to "1", valid VNI. Ensure that OAM frames meant to be contained to the VTEP are not forwarded beyond its endpoints.

The supporting VXLAN OAM ping draft includes a requirement to encode a reserved IEEE MAC address as the inner destination value. However, at the time of implementation, that IEEE MAC address had not been assigned. The inner IEEE MAC address defaults to 00:00:00:00:00:00, but may be changed using the **inner-l2** option. Inner IEEE MAC addresses that are included with OAM packets are not learned in the local Layer 2 forwarding databases.

The echo responder terminates the VXLAN OAM frame, and takes the appropriate response action, and include relevant return codes. By default, the response is sent back using the IP network as an IPv4 UDP response. The user can choose to override this default by changing the **reply-mode** to **overlay**. The overlay return mode forces the responder to use the VTEP connection representing the source IP and source VTEP. If a return overlay is not available, the echo response is dropped by the responder.

Support is included for:

- IPv4 VTEP
- Optional specification of the outer UDP Source, which helps downstream network elements along the path with ECMP to hash to flow to the same path
- Optional configuration of the inner IP information, which helps the user test different equal paths where ECMP is deployed on the source. A test only validates a single path where ECMP functions are deployed. The inner IP information is processed by a hash function, and there is no guarantee that changing the IP information between tests selects different paths.
- Optional end system validation for a single L2 IEEE MAC address per test. This function checks the remote FDB for the configured IEEE MAC Address. Only one end system IEEE MAC Address can be configured per test.
- Reply mode UDP (default) or Overlay
- Optional additional padding can be added to each packet. There is an option that indicates how the responder should handle the pad TLV. By default, the padding is not reflected to the source. The user can change this behavior by including the **reflect-pad** option. The **reflect-pad** option is not supported when the reply mode is set to UDP.
- Configurable send counts, intervals, times outs, and forwarding class

The VXLAN OAM PDU includes two timestamps. These timestamps are used to report forward direction delay. Unidirectional delay metrics require accurate time of day clock synchronization. Negative unidirectional delay values are reported as "0.000". The round trip value includes the entire round trip time including the time that the remote peer takes to process that packet. These reported values may not be representative of network delay.

The following example commands and outputs show how the VXLAN Ping function can be used to validate connectivity. The echo output includes a new header to better describe the VXLAN ping packet headers and the various levels.

```
oam vxlan-ping test-id 1 service 1 dest-vni 2 outer-ip-destination 10.20.1.4
interval
0.1 send-count 10
```



```

TestID 1, Service 1, DestVNI 2, ReplyMode UDP, IFlag Off, PadSize 0, ReflectPad No,
SendCount 10, Interval 0.1, Timeout 5
Outer: SourceIP 10.20.1.3, SourcePort Dynamic, DestIP 10.20.1.4, TTL 10, FC be, Prof
ile
In
Inner: DestMAC 00:00:00:00:00:00, SourceIP 10.20.1.3, DestIP 127.0.0.1

! ! ! ! ! ! ! !
---- vxlan-id 2 ip-address 10.20.1.4 PING Statistics ----
10 packets transmitted, 10 packets received, 0.00% packet loss
    10 non-errored responses(!), 0 out-of-order(*), 0 malformed echo responses(.)
    0 send errors(.), 0 time outs(.)
    0 overlay segment not found, 0 overlay segment not operational
forward-delay min = 1.097ms, avg = 2.195ms, max = 2.870ms, stddev = 0.735ms
round-trip-delay min = 1.468ms, avg = 1.693ms, max = 2.268ms, stddev = 0.210ms

oam vxlan-ping test-id 2 service 1 dest-vni 2 outer-ip-destination 10.20.1.4 outer-
ip-source-udp 65000 outer-ip-ttl 64 inner-l2 d0:0d:1e:00:00:01 inner-ip-source
192.168.1.2 inner-ip-destination 127.0.0.8 reply-mode overlay send-count 20
interval
1 timeout 3 padding 1000 reflect-pad fc nc profile out

TestID 2, Service 1, DestVNI 2, ReplyMode overlay, IFlag Off, PadSize 1000, ReflectP
ad
Yes, SendCount 20, Interval 1, Timeout 3
Outer: SourceIP 10.20.1.3, SourcePort 65000, DestIP 10.20.1.4, TTL 64, FC nc, Profil
e
out
Inner: DestMAC d0:0d:1e:00:00:01, SourceIP 192.168.1.2, DestIP 127.0.0.8

=====
rc=1 Malformed Echo Request Received, rc=2 Overlay Segment Not Present, rc=3 Overlay
Segment Not Operational, rc=4 Ok
=====

1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=1 ttl=255 rtt-time=1.733ms fwd
-time=0.302ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=2 ttl=255 rtt-time=1.549ms fwd
-time=1.386ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=3 ttl=255 rtt-time=3.243ms fwd
-time=0.643ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=4 ttl=255 rtt-time=1.551ms fwd
-time=2.350ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=5 ttl=255 rtt-time=1.644ms fwd
-time=1.080ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=6 ttl=255 rtt-time=1.670ms fwd
-time=1.307ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=7 ttl=255 rtt-time=1.636ms fwd
-time=0.490ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=8 ttl=255 rtt-time=1.649ms fwd
-time=0.005ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=9 ttl=255 rtt-time=1.401ms fwd
-time=0.685ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=10 ttl=255 rtt-time=1.634ms fwd
-time=0.373ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=11 ttl=255 rtt-time=1.559ms fwd
-time=0.679ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=12 ttl=255 rtt-time=1.666ms fwd
-time=0.880ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=13 ttl=255 rtt-time=1.629ms fwd
-time=0.669ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=14 ttl=255 rtt-time=1.280ms fwd
-time=1.029ms. rc=4

```

```

1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=15 ttl=255 rtt-time=1.458ms fwd
-time=0.268ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=16 ttl=255 rtt-time=1.659ms fwd
-time=0.786ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=17 ttl=255 rtt-time=1.636ms fwd
-time=1.071ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=18 ttl=255 rtt-time=1.568ms fwd
-time=2.129ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=19 ttl=255 rtt-time=1.657ms fwd
-time=1.326ms. rc=4
1132 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=20 ttl=255 rtt-time=1.762ms fwd
-time=1.335ms. rc=4

---- vxlan-id 2 ip-address 10.20.1.4 PING Statistics ----
20 packets transmitted, 20 packets received, 0.00% packet loss
  20 valid responses, 0 out-of-order, 0 malformed echo responses
  0 send errors, 0 time outs
  0 overlay segment not found, 0 overlay segment not operational
forward-delay min = 0.005ms, avg = 0.939ms, max = 2.350ms, stddev = 0.577ms
round-trip-delay min = 1.280ms, avg = 1.679ms, max = 3.243ms, stddev = 0.375ms

oam vxlan-ping test-id 1 service 1 dest-vni 2 outer-ip-destination 10.20.1.4 send
-count 10 end-system 00:00:00:00:00:01 interval 0.1
TestID 1, Service 1, DestVNI 2, ReplyMode UDP, IFlag Off, PadSize 0, ReflectPad No,
EndSystemMAC 00:00:00:00:00:01, SendCount 10, Interval 0.1, Timeout 5
Outer: SourceIP 10.20.1.3, SourcePort Dynamic, DestIP 10.20.1.4, TTL 10, FC be, Prof
ile
In
Inner: DestMAC 00:00:00:00:00:00, SourceIP 10.20.1.3, DestIP 127.0.0.1

2 2 2 2 2 2 2 2 2 2
---- vxlan-id 2 ip-address 10.20.1.4 PING Statistics ----
10 packets transmitted, 10 packets received, 0.00% packet loss
  10 non-errored responses(!), 0 out-of-order(*), 0 malformed echo responses(.)
  0 send errors(.), 0 time outs(.)
  0 overlay segment not found, 0 overlay segment not operational
  0 end-system present(1), 10 end-system not present(2)
forward-delay min = 0.467ms, avg = 0.979ms, max = 1.622ms, stddev = 0.504ms
round-trip-delay min = 1.501ms, avg = 1.597ms, max = 1.781ms, stddev = 0.088ms

oam vxlan-ping test-id 1 service 1 dest-vni 2 outer-ip-destination 10.20.1.4 send
-count 10 end-system 00:00:00:00:00:01
TestID 1, Service 1, DestVNI 2, ReplyMode UDP, IFlag Off, PadSize 0, ReflectPad No,
EndSystemMAC 00:00:00:00:00:01, SendCount 10, Interval 1, Timeout 5
Outer: SourceIP 10.20.1.3, SourcePort Dynamic, DestIP 10.20.1.4, TTL 10, FC be, Prof
ile
In
Inner: DestMAC 00:00:00:00:00:00, SourceIP 10.20.1.3, DestIP 127.0.0.1

=====
rc=1 Malformed Echo Request Received, rc=2 Overlay Segment Not Present, rc=3 Overlay
Segment Not Operational, rc=4 Ok
mac=1 End System Present, mac=2 End System Not Present
=====

92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=1 ttl=255 rtt-time=2.883ms fwd
-time=4.196ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=2 ttl=255 rtt-time=1.596ms fwd
-time=1.536ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=3 ttl=255 rtt-time=1.698ms fwd

```

```

-time=0.000ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=4 ttl=255 rtt-time=1.687ms fwd
-time=1.766ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=5 ttl=255 rtt-time=1.679ms fwd
-time=0.799ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=6 ttl=255 rtt-time=1.678ms fwd
-time=0.000ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=7 ttl=255 rtt-time=1.709ms fwd
-time=0.031ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=8 ttl=255 rtt-time=1.757ms fwd
-time=1.441ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=9 ttl=255 rtt-time=1.613ms fwd
-time=2.570ms. rc=4 mac=2
92 bytes from vxlan-id 2 10.20.1.4: vxlan_seq=10 ttl=255 rtt-time=1.631ms fwd
-time=2.130ms. rc=4 mac=2

---- vxlan-id 2 ip-address 10.20.1.4 PING Statistics ----
10 packets transmitted, 10 packets received, 0.00% packet loss
  10 valid responses, 0 out-of-order, 0 malformed echo responses
  0 send errors, 0 time outs
  0 overlay segment not found, 0 overlay segment not operational
  0 end-system present, 10 end-system not present
forward-delay min = 0.000ms, avg = 1.396ms, max = 4.196ms, stddev = 1.328ms
round-trip-delay min = 1.596ms, avg = 1.793ms, max = 2.883ms, stddev = 0.366ms

```

### 5.2.1.6 EVPN-VXLAN routed VPLS multicast routing support

IPv4 and IPv6 multicast routing is supported in an EVPN-VXLAN VPRN and IES routed VPLS service through its IP interface when the source of the multicast stream is on one side of its IP interface and the receivers are on either side of the IP interface. For example, the source for multicast stream G1 could be on the IP side, sending to receivers on both other regular IP interfaces and the VPLS of the routed VPLS service, while the source for group G2 could be on the VPLS side sending to receivers on both the VPLS and IP side of the routed VPLS service. See [IPv4 and IPv6 multicast routing support](#) for more details.

### 5.2.1.7 IGMP and MLD snooping on VXLAN

The delivery of IP multicast in VXLAN services can be optimized with IGMP and MLD snooping. IGMP and MLD snooping are supported in EVPN-VXLAN VPLS services and in EVPN-VXLAN VPRN/IES R-VPLS services. When enabled, IGMP and MLD reports are snooped on SAPs or SDP bindings, but also on VXLAN bindings, to create or modify entries in the MFIB for the VPLS service.

When configuring IGMP and MLD snooping in EVPN-VXLAN VPLS services, consider the following:

- To enable IGMP snooping in the VPLS service on VXLAN, use the **configure service vpls igmp-snooping no shutdown** command.
- To enable MLD snooping in the VPLS service on VXLAN, use the **configure service vpls mld-snooping no shutdown** command.
- The VXLAN bindings only support basic IGMP/MLD snooping functionality. Features configurable under SAPs or SDP bindings are not available for VXLAN (VXLAN bindings are configured with the default values used for SAPs and SDP bindings). By default, a specified VXLAN binding only becomes a dynamic Mrouter when it receives IGMP or MLD queries and adds a specified multicast group to the MFIB when it receives an IGMP or MLD report for that group.

Alternatively, it is possible to configure all VXLAN bindings for a particular VXLAN instance to be Mrouter ports using the **configure service vpls vxlan igmp-snooping mrouter-port** and **configure service vpls vxlan mld-snooping mrouter-port** commands.

- The **show service id igmp-snooping**, **clear service id igmp-snooping**, **show service id mld-snooping**, and **clear service id mld-snooping** commands are also available for VXLAN bindings.



**Note:** MLD snooping uses MAC-based forwarding. See [MAC-based IPv6 multicast forwarding](#) for more details.

The following CLI commands show how the system displays IGMP snooping information and statistics on VXLAN bindings (the equivalent MLD output is similar).

```
*A:PE1# show service id 1 igmp-snooping port-db vxlan vtep 192.0.2.72 vni 1 detail
=====
IGMP Snooping VXLAN 192.0.2.72/1 Port-DB for service 1
=====
-----
IGMP Group 239.0.0.1
-----
Mode           : exclude           Type           : dynamic
Up Time        : 0d 19:07:05       Expires        : 137s
Compat Mode    : IGMP Version 3
V1 Host Expires : 0s              V2 Host Expires : 0s
-----
Source Address  Up Time      Expires      Type      Fwd/Blk
-----
No sources.
-----
IGMP Group 239.0.0.2
-----
Mode           : include           Type           : dynamic
Up Time        : 0d 19:06:39       Expires        : 0s
Compat Mode    : IGMP Version 3
V1 Host Expires : 0s              V2 Host Expires : 0s
-----
Source Address  Up Time      Expires      Type      Fwd/Blk
-----
10.0.0.232     0d 19:06:39  137s        dynamic   Fwd
-----
Number of groups: 2
=====

*A:PE1# show service id 1 igmp-snooping
statistics vxlan vtep 192.0.2.72 vni 1
=====
IGMP Snooping Statistics for VXLAN 192.0.2.72/1 (service 1)
=====
Message Type      Received      Transmitted    Forwarded
-----
General Queries   0             0              556
Group Queries     0             0              0
Group-Source Queries 0             0              0
V1 Reports        0             0              0
V2 Reports        0             0              0
V3 Reports        553           0              0
V2 Leaves         0             0              0
Unknown Type      0             N/A            0
-----
Drop Statistics
-----
```

```

Bad Length           : 0
Bad IP Checksum      : 0
Bad IGMP Checksum    : 0
Bad Encoding         : 0
No Router Alert      : 0
Zero Source IP       : 0
Wrong Version        : 0
Lcl-Scope Packets    : 0
Rsvd-Scope Packets   : 0

Send Query Cfg Drops : 0
Import Policy Drops  : 0
Exceeded Max Num Groups : 0
Exceeded Max Num Sources : 0
Exceeded Max Num Grp Srcs: 0
MCAC Policy Drops    : 0
=====
*A:PE1# show service id 1 mfib
=====
Multicast FIB, Service 1
=====
Source Address  Group Address      SAP or SDP Id          Svc Id  Fwd/Blk
-----
*              *              sap:1/1/1:1           Local   Fwd
*              239.0.0.1        sap:1/1/1:1           Local   Fwd
                  vxlan:192.0.2.72/1      Local   Fwd
10.0.0.232      239.0.0.2        sap:1/1/1:1           Local   Fwd
                  vxlan:192.0.2.72/1      Local   Fwd
-----
Number of entries: 3
=====

```

### 5.2.1.8 PIM snooping on VXLAN

PIM snooping for IPv4 and IPv6 are supported in an EVPN-EVPN-VXLAN VPLS or R-VPLS service (with the R-VPLS attached to a VPRN or IES service). The snooping operation is similar to that within a VPLS service (see [PIM snooping for VPLS](#)) and supports both PIM snooping and PIM proxy modes.

PIM snooping for IPv4 is enabled using the **configure service vpls pim-snooping** command.

PIM snooping for IPv6 is enabled using the **configure service vpls pim-snooping no ipv6-multicast-disable** command.

When using PIM snooping for IPv6, the default forwarding is MAC-based with optional support for SG-based (see [IPv6 multicast forwarding](#)). SG-based forwarding requires FP3- or higher-based hardware.

It is not possible to configure **max-num-groups** for VXLAN bindings.

### 5.2.1.9 Static VXLAN termination in Epipe services

By default, the system IP address is used to terminate and generate VXLAN traffic. The following configuration example shows an Epipe service that supports static VXLAN termination:

```

config service epipe 1 name "epipe1" customer 1 create
  sap 1/1/1:1 create
  exit
  vxlan vni 100 create
  egr-vtep 192.0.2.1

```

```

    oper-group op-grp-1
    exit
no shutdown

```

Where:

- **vxlan vni vni create** specifies the ingress VNI the router uses to identify packets for the service. The following considerations apply:
  - In services that use EVPN, the configured VNI is only used as the ingress VNI to identify packets that belong to the service. Egress VNIs are learned from the BGP EVPN. In the case of Static VXLAN, the configured VNI is also used as egress VNI (because there is no BGP EVPN control plane).
  - The configured VNI is unique in the system, and as a result, it can only be configured in one service (VPLS or Epipe).
- **egr-vtep ip-address** specifies the remote VTEP the router uses when encapsulating frames into VXLAN packets. The following consideration apply:
  - When the PE receives VXLAN packets, the source VTEP is not checked against the configured egress VTEP.
  - The **ip-address** must be present in the global routing table so that the VXLAN destination is operationally up.
- The **oper-group** may be added under **egr-vtep**. The expected behavior for the operational group and service status is as follows:
  - If the **egr-vtep** entry is not present in the routing table, the VXLAN destination (in the **show service id vxlan** command) and the provisioned operational group under **egr-vtep** enters into the operationally down state.
  - If the Epipe SAP goes down, the service goes down, but it is not affected if the VXLAN destination goes down.
  - If the service is **admin shutdown**, then in addition to the SAP, the VXLAN destination and the **oper-group** also enters the operationally down state.



**Note:** The operational group configured under **egr-vtep** cannot be monitored on the SAP of the Epipe where it is configured.

The following features are not supported by Epipe services with VXLAN destinations:

- per-service hashing
- SDP-binds
- PBB context
- BGP-VPWS
- spoke SDP-FEC
- PW-port

### 5.2.1.10 Static VXLAN termination in VPLS/R-VPLS services

VXLAN instances in VPLS and R-VPLS can be configured with egress VTEPs. This is referred as static vxlan-instances. The following configuration example shows a VPLS service that supports a static vxlan-instance:

```
config service vpls 1 name "vpls-1" customer 1 create
  sap 1/1/1:1 create
  exit
  vxlan instance 1 vni 100 create
    source-vtep-security
    no disable-aging /* default: disable-aging
    no disable-learning /* default: disable-learning
    no discard-unknown-source
    no max-nbr-mac-addr <table-size>
    restrict-protected-src discard-frame
    egr-vtep 192.0.2.1 create
    exit
    egr-vtep 192.0.2.2 create
    exit
  vxlan instance 2 vni 101 create
    egr-vtep 192.0.2.3 create
    exit

  vxlan instance 2 vni 101 create
    egr-vtep 192.0.2.3 create
    exit
  no shutdown
```

Specifically the following can be stated:

- Each VPLS service can have up to two static VXLAN instances. Each instance is an implicit split-horizon-group, and up to 255 static VXLAN binds are supported in total, shared between the two VXLAN instances.
- Single VXLAN instance VPLS services with static VXLAN are supported along with SAPs and SDP bindings. Therefore:
  - VNIs configured in static VXLAN instances are "symmetric", that is, the same ingress and egress VNIs are used for VXLAN packets using that instance. Note that asymmetric VNIs are actually possible in EVPN VXLAN instances.
  - The addresses can be IPv4 or IPv6 (but not a mix within the same service).
  - A specified VXLAN instance can be configured with static egress VTEPs, or be associated with BGP EVPN, but the same instance cannot be configured to support both static and BGP-EVPN based VXLAN bindings.
- Up to two VXLAN instances are supported per VPLS (up to two).
  - When two VXLAN instances are configured in the same VPLS service, any combination of static and BGP-EVPN enabled instances are supported. That is, the two VXLAN instances can be static, or BGP-EVPN enabled, or one of each type.
  - When a service is configured with EVPN and there is a static BGP-EVPN instance in the same service, the user must configure **restrict-protected-src discard-frame** along with no **disable-learning** in the static BGP-EVPN instance, **service>vpls>vxlan**.

- MAC addresses are learned also on the VXLAN bindings of the static VXLAN instance. Therefore, they are shown in the FDB commands. Note that disable-learning and disable-aging are by default enabled in static vxlan-instance.
  - The learned MAC addresses are subject to the remote-age, and not the local-age (only MACs learned on SAPs use the local-age setting).
  - MAC addresses are learned on a VTEP as long as no disable-learning is configured, and the VXLAN VTEP is present in the base route table. When the VTEP disappears from the route table, the associated MACs are flushed.
- The **vpls vxlan source-vtep-security** command can be configured per VXLAN instance on VPLS services. When enabled, the router performs an IPv4 **source-vtep** lookup to discover if the VXLAN packet comes from a trusted VTEP. If not, the router discards the frame. If the lookup yields a trusted source VTEP, then the frame is accepted.
  - A trusted VTEP is an egress VTEP that has been statically configured, or dynamically learned (through EVPN) in any service, Epipe or VPLS
  - The command **show service vxlan** shows the list of trusted VTEPs in the router.
  - The command **source-vtep-security** works for static VXLAN instances or BGP-EVPN enabled VXLAN instances, but only for IPv4 VTEPs.
  - The command is **mutually exclusive** with assisted-replication (replicator or leaf) in the VNI instance. AR can still be configured in a different instance.

Static VXLAN instances can use non-system IPv4/IPv6 termination.

### 5.2.1.11 Non-system IPv4 and IPv6 VXLAN termination in VPLS, R-VPLS, and Epipe services

#### Prerequisites

By default, only VXLAN packets with the same IP destination address as the system IPv4 address of the router can be terminated and processed for a subsequent MAC lookup. A router can simultaneously terminate VXLAN tunnels destined for its system IP address and three additional non-system IPv4 or IPv6 addresses, which can be on the base router or VPRN instances. This section describes the configuration requirements for services to terminate VXLAN packets destined for a non-system loopback IPv4 or IPv6 address on the base router or VPRN.

#### About this task

Perform the following steps to configure a service with non-system IPv4 or IPv6 VXLAN termination:

#### Procedure

- Step 1.** Create the FPE (see FPE creation)
- Step 2.** Associate the FPE with VXLAN termination (see FPE association with VXLAN termination)
- Step 3.** Configure the router loopback interface (see VXLAN router loopback interface)
- Step 4.** Configure VXLAN termination (non-system) VTEP addresses (see VXLAN termination VTEP addresses)
- Step 5.** Add the service configuration (see VXLAN services)



## What to do next

The following actions must be considered when the aforementioned steps are completed.

- **FPE creation**

A Forwarding Path Extension (FPE) is required to terminate non-system IPv4 or IPv6 VXLAN tunnels.

In a non-system IPv4 VXLAN termination, the FPE function is used for additional processing required at ingress (VXLAN tunnel termination) only, and not at egress (VXLAN tunnel origination).

If the IPv6 VXLAN terminates on a VPLS or Epipe service, the FPE function is used at ingress only, and not at egress.

For R-VPLS services terminating IPv6 VXLAN tunnels and also for VPRN VTEPs, the FPE is used for the egress as well as the VXLAN termination function. In the case of R-VPLS, an internal static SDP is created to allow the required extra processing.

For information about FPE configuration and functions, see the *7705 SAR Gen 2 Interface Configuration Guide, "Forwarding Path Extension"*.

- **FPE association with VXLAN termination**

The FPE must be associated with the VXLAN termination application. The following example configuration shows two FPEs and their corresponding association. FPE 1 uses the base router and FPE 2 is configured for VXLAN termination on VPRN 10.

```
configure
  fwd-path-ext
    fpe 1 create
      path pxc pxc-1
      vxlan-termination
    fpe 2 create
      path pxc pxc-2
      vxlan-termination router 10
```

- **VXLAN router loopback interface**

Create the interface that terminates and originates the VXLAN packets. The interface is created as a router interface, which is added to the Interior Gateway Protocol (IGP) and used by the BGP as the EVPN NLRI next hop.

Because the system cannot terminate the VXLAN on a local interface address, a subnet must be assigned to the loopback interface and not a host IP address that is /32 or /128. In the following example, all the addresses in subnet 11.11.11.0/24 (except 11.11.11.1, which is the interface IP) and subnet 10.1.1.0/24 (except 10.1.1.1) can be used for tunnel termination. The subnet is advertised using the IGP and is configured on either the base router or a VPRN. In the example, two subnets are assigned, in the base router and VPRN 10 respectively.

```
configure
  router
    interface "lo1"
      loopback
      address 10.11.11.1/24
  isis
    interface "lo1"
      passive
      no shutdown
```

```
configure
  service
```

```

vprn 10 name "vprn10" customer 1 create
  interface "lo1"
    loopback
    address 10.1.1.1/24
  isis
    interface "lo1"
      passive
      no shutdown

```

A local interface address cannot be configured as a VXLAN tunnel-termination IP address in the CLI, as shown in the following example.

```

*A:PE-3# configure service system vxlan tunnel-termination 192.0.2.3 fpe 1 create
MINOR: SVCNMR #8353 VXLAN Tunnel termination IP address cannot be configured -
IP address in use by another application or matches a local interface IP address

```

The subnet can be up to 31 bits. For example, to use 10.11.11.1 as the VXLAN termination address, the subnet should be configured and advertised as shown in the following example configuration.

```

interface "lo1"
  address 10.11.11.0/31
  loopback
  no shutdown
exit
isis 0
  interface "lo1"
    passive
    no shutdown
  exit
  no shutdown
exit

```

It is not a requirement for the remote PEs and NVEs to have the specific /32 or /128 IP address in their RTM to resolve the BGP EVPN NLRI next hop or forward the VXLAN packets. An RTM with a subnet that contains the remote VTEP can also perform these tasks.



**Note:** The system does not check for a pre-existing local base router loopback interface with a subnet corresponding to the VXLAN tunnel termination address. If a tunnel termination address is configured and the FPE is operationally up, the system starts terminating VXLAN traffic and responding ICMP messages for that address. The following conditions are ignored in this scenario:

- the presence of a loopback interface in the base router
- the presence of an interface with the address contained in the configured subnet, and no loopback

The following example output includes an IPv6 address in the base router. It could also be configured in a VPRN instance.

```

configure
  router
    interface "lo1"
      loopback
      address 10.11.11.1/24
      ipv6
        address 2001:db8::/127
    exit
  isis

```

```
interface "lo1"
  passive
  no shutdown
```

- **VXLAN termination VTEP addresses**

The **service>system>vxlan>tunnel-termination** context allows the user to configure non-system IP addresses that can terminate the VXLAN and their corresponding FPEs.

As shown in the following example, an IP address may be associated with a new or existing FPE already terminating the VXLAN. The list of addresses that can terminate the VXLAN can include IPv4 and IPv6 addresses.

```
config service system vxlan#
  tunnel-termination 10.11.11.1 fpe 1 create
  tunnel-termination 2001:db8:1000::1 fpe 1 create

config service vprn 10 vxlan#
  tunnel-termination 10.1.1.2 fpe 2 create
```

The **tunnel-termination** command creates internal loopback interfaces that can respond to ICMP requests. In the following sample output, an internal loopback is created when the tunnel termination address is added (for 10.11.11.1 and 2001:db8:1000::1). The internal FPE router interfaces created by the VXLAN termination function are also shown in the output. Similar loopback and interfaces are created for tunnel termination addresses in a VPRN (not shown).

```
*A:PE1# show router interface
=====
Interface Table (Router: Base)
=====
```

| Interface-Name<br>IP-Address              | Adm   | Opr(v4/v6) | Mode    | Port/SapId<br>PfxState |
|---|-------|------------|---------|------------------------|
| -----                                     | ----- | -----      | -----   | -----                  |
| _tmnx_fpe_1.a<br>fe80::100/64             | Up    | Up/Up      | Network | pxc-2.a:1<br>PREFERRED |
| _tmnx_fpe_1.b<br>fe80::101/64             | Up    | Up/Up      | Network | pxc-2.b:1<br>PREFERRED |
| _tmnx_vli_vxlan_1_131075<br>10.11.11.1/32 | Up    | Up/Up      | Network | loopback<br>n/a        |
| 2001:db8:1000::1                          |       |            |         | PREFERRED              |
| fe80::6cfb:ffff:fe00:0/64                 |       |            |         | PREFERRED              |
| lo1<br>10.11.11.0/31                      | Up    | Up/Down    | Network | loopback<br>n/a        |
| system<br>1.1.1.1/32                      | Up    | Up/Down    | Network | system<br>n/a          |
| <snip>                                    |       |            |         |                        |

- **VXLAN services**

By default, the VXLAN services use the system IP address as the source VTEP of the VXLAN encapsulated frames. The **vxlan-src-vtep** command in the **config>service>vpls** or **config>service>epipe** context enables the system to use a non-system IPv4 or IPv6 address as the source VTEP for the VXLAN tunnels in that service.

A different **vxlan-src-vtep** can be used for different services, as shown in the following example where two different services use different non-system IP addresses as source VTEPs.

```
configure service vpls 1
  vxlan-src-vtep 10.11.11.1
```

```
configure service vpls 2
vxlan-src-vtep 2001:db8:1000::1
```

In addition, if a **vxlan-src-vtep** is configured and the service uses EVPN, the IP address is also used to set the BGP NLRI next hop in EVPN route advertisements for the service.



**Note:** The BGP EVPN next hop can be overridden by the use of export policies based on the following rules:

- A BGP peer policy can override a next hop pushed by the **vxlan-src-vtep** configuration.
- If the VPLS service is IPv6 (that is, the **vxlan-src-vtep** is IPv6) and a BGP peer export policy is configured with **next-hop-self**, the BGP next-hop is overridden with an IPv6 address auto-derived from the IP address of the system. The auto-derivation is based on RFC 4291. For example, ::ffff:10.20.1.3 is auto-derived from system IP 10.20.1.3.
- The policy checks the address type of the next hop provided by the **vxlan-src-vtep** command. If the command provides an IPv6 next hop, the policy is unable use an IPv4 address to override the IPv6 address provided by the **vxlan-src-vtep** command.

After the preceding steps are performed to configure a VXLAN termination, the VPLS, R-VPLS, or Epipe service can be used normally, except that the service terminates VXLAN tunnels with a non-system IPv4 or IPv6 destination address (in the base router or a VPRN instance) instead of the system IP address only.

The FPE **vxlan-termination** function creates internal router interfaces and loopbacks that are displayed by the **show** commands. When configuring IPv6 VXLAN termination on an R-VPLS service, as well as the internal router interfaces and loopbacks, the system creates internal SDP bindings for the required egress processing. The following output shows an example of an internal FPE-type SDP binding created for IPv6 R-VPLS egress processing.

```
*A:PE1# show service sdp-using
=====
SDP Using
=====
SvcId      SdpId          Type  Far End          Opr   I.Label E.Label
State
-----
2002       17407:2002     Fpe   fpe_1.b          Up    262138 262138
-----
Number of SDPs : 1
=====
```

When BGP EVPN is used, the BGP peer over which the EVPN-VXLAN updates are received can be an IPv4 or IPv6 peer, regardless of whether the next-hop is an IPv4 or IPv6 address.

The same VXLAN tunnel termination address cannot be configured on different router instances; that is, on two different VPRN instances or on a VPRN and the base router.

## 5.2.2 EVPN for overlay tunnels

This section describes the specifics of EVPN for non-MPLS Overlay tunnels.

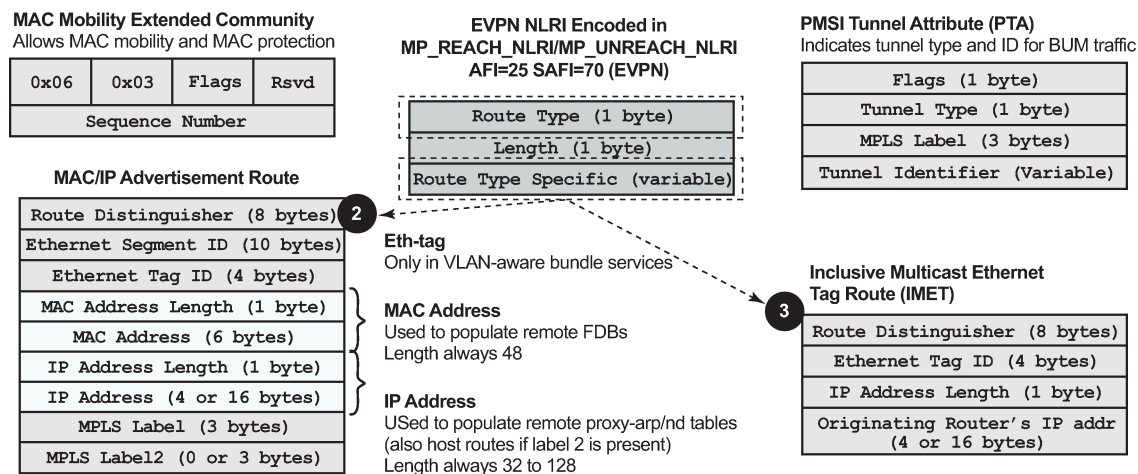
### 5.2.2.1 BGP-EVPN control plane for VXLAN overlay tunnels

RFC 8365 describes EVPN as the control plane for overlay-based networks. The 7705 SAR Gen 2 supports all routes and features described in RFC 7432 that are required for the DGW function. EVPN multihoming and BGP multihoming based on the L2VPN BGP address family are both supported if redundancy is needed.

The following figure shows the EVPN MP-BGP NLRI, required attributes and extended communities, and two route types supported for the DGW Layer 2 applications:

- route type 3** Inclusive Multicast Ethernet Tag (IMET) route
- route type 2** MAC/IP advertisement route

Figure 68: EVPN-VXLAN required routes and communities



#### EVPN route type 3 – IMET route

Route type 3 is used to set up the flooding tree (BUM flooding) for a specified VPLS service in the data center. The received inclusive multicast routes add entries to the VPLS flood list. The tunnel types supported in an EVPN route type 3 when BGP-EVPN MPLS is enabled are ingress replication, P2MP MLDP, and composite tunnels.

Ingress Replication (IR) and Assisted Replication (AR) are supported for VXLAN tunnels. See [Layer 2 multicast optimization for VXLAN \(Assisted-Replication\)](#) for more information about the AR.


If **ingress-repl-inc-mcast-advertisement** is enabled, a route type 3 is generated by the router per VPLS service as soon as the service is in an operationally up state. The following fields and values are used:

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context.



**Note:** The RD can be configured or derived from the **bgp-evpn evi** value.

- Ethernet Tag ID is 0.
- IP address length is always 32.
- Originating router's IP address carries an IPv4 or IPv6 address.

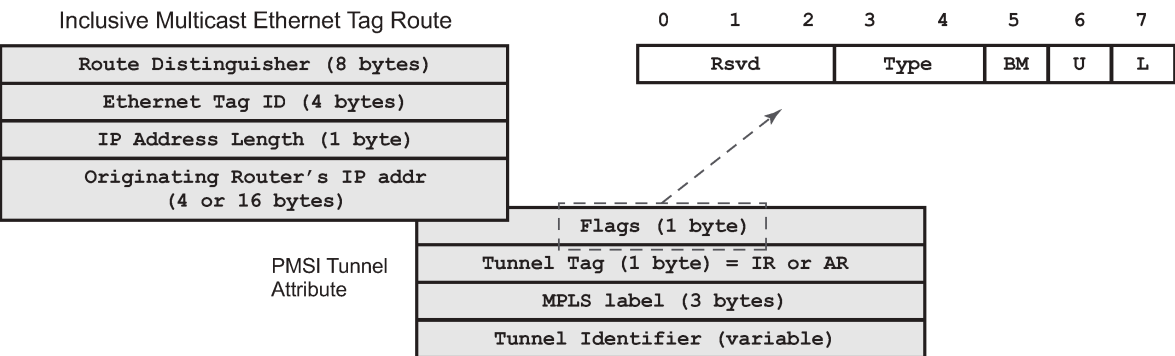


**Note:** By default, the IP address of the Originating router is derived from the system IP address. However, this can be overridden by the **configure service vpls bgp-evpn incl-mcast-orig-ip ip-address** command for the Ingress Replication (and mLDP if MPLS is used) tunnel type.

- For PMSI Tunnel Attribute (PTA), tunnel type = Ingress replication (6) or Assisted Replication (10). The following applies:
  - Leaf is not required for Flags.
  - MPLS label carries the VNI configured in the VPLS service. Only one VNI can be configured per VPLS service.
  - Tunnel endpoint is equal to the system IP address.

As shown in the following figure, additional flags are used in the PTA when the service is configured for AR.

Figure 69: PMSI attribute flags field for AR




1040

The Flags field is defined as a Type field (for AR) with two new flags that are defined as follows:

- T is the AR Type field (2 bits):
  - 00 (decimal 0) = RNVE (non-AR support)
  - 01 (decimal 1) = AR REPLICATOR
  - 10 (decimal 2) = AR LEAF
- The U and BM flags defined in IETF Draft *draft-ietf-bess-evpn-optimized-ir* are not used in the SR OS.

The following table describes the inclusive multicast route information sent per VPLS service when the router is configured as **assisted-replication replicator** (AR-R) or **assisted-replication leaf** (AR-L). A Regular Network Virtualization Edge device (RNVE) is defined as an EVPN-VXLAN router that does not support (or is not configured for) Assisted-Replication.



**Note:** For AR-R, two inclusive multicast routes may be advertised if **ingress-repl-inc-mcast-advertisement** is enabled: a route with tunnel-type IR, tunnel-id = IR IP (generally system-ip) and a route with tunnel-type AR, tunnel-id = AR IP (the address configured in the **assisted-replication-ip** command).

Table 13: AR-R and AR-L routes and usage

| AR role | Function               | Inclusive Mcast routes advertisement  |
|---------|------------------------|---|
| AR-R    | Assists AR-LEAFs       | <ul style="list-style-type: none"> <li>IR included in the Mcast route (uses IR IP) if <b>ingress-repl-inc-mcast-advertisement</b> is enabled</li> <li>AR included in the Mcast route (uses AR IP, tunnel type=AR, T=1)</li> </ul> |
| AR-LEAF | Sends BM only to AR-Rs | IR inclusive multicast route (IR IP, T=2) if <b>ingress-repl-inc-mcast-advertisement</b> is enabled   |
| RNVE    | Non-AR support         | IR inclusive multicast route (IR IP) if <b>ingress-repl-inc-mcast-advertisement</b> is enabled  |

### EVPN route type 2 – MAC/IP advertisement route

The 7705 SAR Gen 2 generates this route type for advertising MAC addresses. If mac-advertisement is enabled, the router generates MAC advertisement routes for the following:

- learned MACs on SAPs or SDP bindings
- conditional static MACs



**Note:** To address unknown MAC routes, if **unknown-mac-route** is enabled, there is no bgp-mh site in the service or there is a (single) DF site

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

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context.



**Note:** The RD can be configured or derived from the **bgp-evpn evi** value.

- Ethernet Segment Identifier (ESI) value = 0:0:0:0:0:0:0:0 or non-zero, depending on whether the MAC addresses are learned on an Ethernet Segment.
- Ethernet Tag ID is 0.
- MAC address length is always 48.
- MAC Address:
  - is 00:00:00:00:00:00 for the Unknown MAC route address.
  - is different from 00:...:00 for the rest of the advertised MACs.
- IP address and IP address length:
  - The length of the IP address associated with the MAC being advertised is either 32 for IPv4 or 128 for IPv6.
  - If the MAC address is the Unknown MAC route, the IP address length is zero and the IP omitted.
  - In general, any MAC route without IP has IPL=0 (IP length) and the IP is omitted.
  - When received, any IPL value not equal to zero, 32, or 128 discards the route.

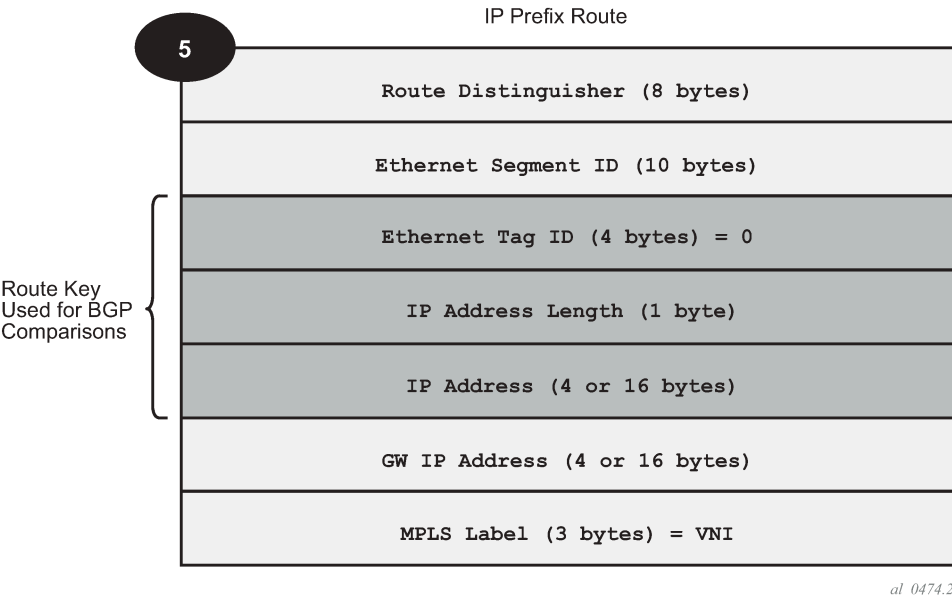
- MPLS Label 1 carries the VNI configured in the VPLS service. Only one VNI can be configured per VPLS.
- MPLS Label 2 is 0.
- MAC Mobility extended community is used to signal the sequence number in case of MAC moves and the sticky bit in case of advertising conditional static MACs. If a MAC route is received with a MAC mobility **ext-community**, the sequence number and the sticky bit are considered for route selection.

When EVPN-VXLAN multihoming is enabled, type 1 routes (Auto-Discovery per-ES and per-EVI routes) and type 4 routes (ES routes) are also generated and processed. See [BGP-EVPN control plane for MPLS tunnels](#) for more information about route types 1 and 4.

**EVPN route type 5 – IP prefix route**

The following figure shows the IP prefix route or route-type 5.

Figure 70: EVPN route-type 5



The router generates this route type to advertise IP prefixes in EVPN. The IP prefix advertisement routes are generated for existing IP prefixes in a VPRN linked to the IRB backhaul R-VPLS service.

The route-type 5 generated by a router uses the following fields and values:

- Route Distinguisher: taken from the RD configured in the IRB backhaul R-VPLS service within the BGP context
- Ethernet Segment Identifier (ESI): value = 0:0:0:0:0:0:0:0
- Ethernet Tag ID: 0
- IP address length: any value in the 0 to 128 range
- IP address: any valid IPv4 or IPv6 address
- Gateway IP address: can carry two different values:
  - if different from zero, the route-type 5 carries the primary IP interface address of the VPRN behind which the IP prefix is known. This is the case for the regular IRB backhaul R-VPLS model.



- if 0.0.0.0, the route-type 5 is sent with a MAC next-hop extended community that carries the VPRN interface MAC address. This is the case for the EVPN tunnel R-VPLS model.
- MPLS Label: carries the VNI configured in the VPLS service. Only one VNI can be configured per VPLS service.

All routes in EVPN-VXLAN are sent with the RFC 5512 tunnel encapsulation extended community, with the tunnel type value set to VXLAN.

### 5.2.2.2 EVPN for VXLAN in VPLS services

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

**Figure 62: Layer 2 DC PE with VPLS to the WAN** shows a DC with a Layer 2 service that carries the traffic for a tenant who wants to extend a subnet beyond the DC. The DC PE function is carried out by the 7705 SAR Gen 2 where a VPLS instance exists for that particular tenant. Within the DC, the tenant has VPLS instances in all the Network Virtualization Edge (NVE) devices where they require connectivity (such as VPLS instances can be instantiated in TORs, Nuage VRS, VSG, and so on). The VPLS instances in the redundant DGW and the DC NVEs are connected by VXLAN bindings. BGP-EVPN provides the required control plane for such VXLAN connectivity.

The DGW routers are configured with a VPLS per tenant that provides the VXLAN connectivity to the Nuage VPLS instances. On the router, each tenant VPLS instance is configured with:

- WAN-related parameters (SAPs, spoke SDPs, mesh-SDPs, BGP-AD, and so on).
- BGP-EVPN and VXLAN (VNI) parameters. The following CLI output is an example of an EVPN-VXLAN VPLS service.

#### Example: EVPN-VXLAN VPLS service

```
*A:DGW1>config>service>vpls# info
-----
description "vxlan-service"
vxlan instance 1 vni 1 create
exit
bgp
    route-distinguisher 65001:1
    route-target export target:65000:1 import target:65000:1
exit
bgp-evpn
    unknown-mac-route
    mac-advertisement
    vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
sap 1/1/1:1 create
exit
no shutdown
-----
```

The **bgp-evpn** context specifies the encapsulation type (only VXLAN is supported) used by EVPN, and other parameters like the **unknown-mac-route** and **mac-advertisement** commands. These commands are typically configured in three different ways:

- If the operator configures **no unknown-mac-route** and **mac-advertisement** (default option), the router advertises new learned MACs (on the SAPs or SDP bindings) or new conditional static MACs.

- If the operator configures **unknown-mac-route** and **no mac-advertisement**, the router only advertises an unknown MAC route as long as the service is operationally up (if no BGP-MH site is configured in the service) or the router is the DF (if BGP-MH is configured in the service).
- If the operator configures **unknown-mac-route** and **mac-advertisement**, the router advertises new learned MACs, conditional static MACs, and the **unknown-mac-route**. The **unknown-mac-route** is only advertised under the preceding conditions.

Other parameters related to EVPN or VXLAN are:

- MAC duplication parameters
- VXLAN VNI (defines the VNI that the router uses in the EVPN routes generated for the VPLS service)

After the VPLS is configured and operationally up, the router sends or receives inclusive multicast Ethernet Tag routes, and a full-mesh of VXLAN connections is automatically created. These VXLAN "auto-bindings" can be characterized as follows:

- The VXLAN auto-binding model is based on an IP-VPN-like design, where no SDPs or SDP binding objects are created by or visible to the user. The VXLAN auto-binds are composed of remote VTEPs and egress VNIs, and can be displayed with the following command:

```
show service id 112 vxlan destinations
```

#### Output example

```
=====
Egress VTEP, VNI (Instance 1)
=====
VTEP Address                               Egress VNI Oper Mcast Num
                                           State      MACs
-----
192.0.2.2                                112        Up   BUM   1
192.0.2.3                                112        Down BUM   0
-----
Number of Egress VTEP, VNI : 2
=====
```

```
show service id 112 vxlan destinations detail
```

#### Output example

```
=====
Egress VTEP, VNI (Instance 1)
=====
VTEP Address                               Egress VNI Oper Mcast Num
                                           State      MACs
-----
192.0.2.2                                112        Up   BUM   1
Oper Flags      : None
Type            : evpn
L2 PBR         : No
Sup BCast Domain : No
Last Update    : 02/03/2023 22:15:06
192.0.2.3                                112        Down BUM   0
Oper Flags      : MTU-Mismatch
Type            : evpn
L2 PBR         : No
Sup BCast Domain : No
Last Update    : 01/31/2023 21:28:39
-----
```

```
Number of Egress VTEP, VNI : 2
```

```
=====
```

- If the following command is configured on the PEs attached to the same service, the service MTU value is advertised in the EVPN Layer-2 Attributes extended community along with the IMET routes.

– **MD-CLI**

```
configure service vpls bgp-evpn routes incl-mcast advertise-l2-attributes
```

– **classic CLI**

```
configure service vpls bgp-evpn incl-mcast-l2-attributes-advertisement
```

Upon receiving the signaled MTU from an egress PE, the ingress PE compares the MTU with the local one and, in case of mismatch, the EVPN VXLAN destination is brought operationally down. An operational flag MTU-Mismatch shows the reason why the VXLAN destination is operationally down in this case. The following command makes the router ignore the MTU signaled by the remote PE and bring up the VXLAN destination if there are no other reasons to keep it down.

```
configure service vpls bgp-evpn ignore-mtu-mismatch
```

- The VXLAN bindings observe the VPLS split-horizon rule. This is performed automatically without the need for any split-horizon configuration.
- BGP Next-Hop Tracking for EVPN is fully supported. If the BGP next-hop for a specified received BGP EVPN route disappears from the routing table, the BGP route is not marked as "used" and the respective entry in **show service id vxlan destinations** is removed.

After the flooding domain is setup, the routers and DC NVEs start advertising MAC addresses, and the routers can learn MACs and install them in the FDB. Some considerations are the following:

- All the MAC addresses associated with remote VTEP/VNIs are always learned in the control plane by EVPN. Data plane learning on VXLAN auto-bindings is not supported.
- When **unknown-mac-route** is configured, it is generated when no (BGP-MH) site is configured, or a site is configured and the site is DF in the PE.



**Note:** The **unknown-mac-route** is not installed in the FDB (therefore, it does not display in the **show service id svc-id fdb detail** command).

- The router can be configured with only one VNI (and signals a single VNI per VPLS) and the same VNI value must be configured in all DC NVEs (remote PEs). The VTEPs and VNIs display in the FDB information associated with MAC addresses:

**Output example**

```
A:PE65# show service id 1000 fdb detail
```

```
=====
Forwarding Database, Service 1000
=====
```

| ServId | MAC               | Source-Identifier           | Type<br>Age | Last Change       |
|--------|-------------------|-----------------------------|-------------|-------------------|
| 1000   | 00:00:00:00:00:01 | vxlan-1:<br>192.0.2.63:1063 | Evpn        | 10/05/13 23:25:57 |
| 1000   | 00:00:00:00:00:65 | sap:1/1/1:1000              | L/30        | 10/05/13 23:25:57 |
| 1000   | 00:ca:ca:ca:ca:00 | vxlan-1:                    | EvpnS       | 10/04/13 17:35:43 |

```

192.0.2.63:1063
-----
No. of MAC Entries: 3
-----
Legend:  L=Learned  O=0am  P=Protected-MAC  C=Conditional  S=Static
=====

```

### 5.2.2.2.1 Resiliency and BGP multihoming

The DC overlay infrastructure relies on IP tunneling, that is, VXLAN; therefore, the underlay IP layer resolves failure in the DC core. The IGP should be optimized to get the fastest convergence.

From a service perspective, resilient connectivity to the WAN may be provided by BGP multihoming.

### 5.2.2.2.2 Use of BGP-EVPN, BGP-AD, and sites in the same VPLS service

All BGP-EVPN (control plane for a VXLAN DC), BGP-AD (control plane for MPLS-based spoke SDPs connected to the WAN), and one site for BGP multihoming (control plane for the multihomed connection to the WAN) can be configured in a single service in a specified system. In this 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. To use different import/export Route Targets (RTs), use the VSI import and export policies.
- The **pw-template-binding** command under BGP does not affect EVPN or BGP-MH. It is only used for the instantiation of the BGP-AD spoke SDPs.
- If the same import/export RTs are used in the two redundant DGWs, VXLAN binding as well as a fec129 spoke SDP binding is established between the two DGWs, creating a loop. To avoid creating a loop, the router allows the establishment of an EVPN VXLAN binding and an SDP binding to the same far end, but the SDP binding is kept operationally down. Only the VXLAN binding is operationally up.

### 5.2.2.2.3 Use of the unknown-mac-route

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

The use of EVPN, as the control plane of NVO networks in the DC, provides significant benefits, as described in IETF draft *draft-ietf-bess-evpn-overlay*.

However, there is a potential issue that must be addressed when a VPLS DCI is used for an NVO3-based DC: all the MAC addresses learned from the WAN side of the VPLS must be advertised by BGP EVPN updates. Even if optimized BGP techniques like RT-constraint are used, the number of MAC addresses to advertise or withdraw (in case of failure) from the DC GWs can be difficult to control and overwhelming for the DC network, especially when the NVEs reside in the hypervisors.

The solution to this issue is based on the use of an unknown-mac-route address that is advertised by the DC PEs. By using this unknown-mac-route advertisement, the DC tenant may decide to optionally turn off the advertisement of WAN MAC addresses in the DGW, therefore, reducing the control plane overhead and the size of the FDB tables in the NVEs.

The use of the **unknown-mac-route** is optional and helps to reduce the amount of unknown-unicast traffic within the data center. All the receiving NVEs supporting this concept send any unknown-unicast packet

to the owner of the **unknown-mac-route**, as opposed to flooding the unknown-unicast traffic to all other NVEs that are part of the same VPLS.



**Note:** Although the router can be configured to generate and advertise the **unknown-mac-route**, the router never honors the **unknown-mac-route** and floods to the TLS-flood list when an unknown-unicast packet arrives at an ingress SAP or SDP binding.

The use of the **unknown-mac-route** assumes the following:

- A fully virtualized DC where all the MACs are control-plane learned, and learned previous to any communication (no legacy TORs or VLAN connected servers).
- The only exception is MACs learned over the SAPs/SDP bindings that are part of the BGP-MH WAN site ID. Only one site ID is supported in this case.
- No other SAPs/SDP bindings out of the WAN site ID are supported, unless only static MACs are used on those SAPs/SDP bindings.

Therefore, when **unknown-mac-route** is configured, it is only generated when one of the following applies:

- No site is configured and the service is operationally up.
- A BGP-MH site is configured and the DGW is Designated Forwarder (DF) for the site. In case of BGP-MH failover, the **unknown-mac-route** is withdrawn by the former DF and advertised by the new DF.

### 5.2.2.3 EVPN for VXLAN in R-VPLS services

**Figure 63: Gateway IRB on the DC PE for an L2 EVPN/VXLAN DC** shows a DC with a Layer 2 service that carries the traffic for a tenant who extends a subnet within the DC, while the DGW is the default gateway for all the hosts in the subnet. The DGW function is carried out by the 7705 SAR Gen 2 where an R-VPLS instance exists for that particular tenant. Within the DC, the tenant has VPLS instances in all the NVE devices where they require connectivity (such VPLS instances can be instantiated in TORs, Nuage VRS, VSG, and so on). The WAN connectivity is based on existing IP-VPN features.

In this model, the DGW routers are configured with a R-VPLS (bound to the VPRN that provides the WAN connectivity) per tenant that provides the VXLAN connectivity to the Nuage VPLS instances. This model provides inter-subnet forwarding for L2-only TORs and other L2 DC NVEs.

On the router:

- The VPRN is configured with an interface bound to the backhaul R-VPLS. That interface is a regular IP interface (IP address configured or possibly a Link Local Address if IPv6 is added).
- The VPRN can support other numbered interfaces to the WAN or even to the DC.
- The R-VPLS is configured with the BGP, BGP-EVPN and VXLAN (VNI) parameters.

The Nuage VSGs and NVEs use a regular VPLS service model with BGP EVPN and VXLAN parameters.

Consider the following:

- Route-type 2 routes with MACs and IPs are advertised. Some considerations about MAC+IP and ARP/ND entries are:
  - The node advertises its IRB MAC+IP in a route type 2 route and possibly the VRRP vMAC+vIP if it runs VRRP and the node is the active router. In both cases, the MACs are advertised as static MACs, therefore, protected by the receiving PEs.
  - If the VPRN interface is configured with one or more additional secondary IP addresses, they are all advertised in routes type 2, as static MACs.

- The node processes route-type 2 routes as usual, populating the FDB with the received MACs and the VPRN ARP/ND table with the MAC and IPs, respectively.



**Note:** ND entries received from the EVPN are installed as Router entries. The ARP/ND entries coming from the EVPN are tagged as **evpn**.

- When a VPLS containing proxy-ARP/proxy-ND entries is bound to a VPRN (allow-ip-int-bind) all the proxy-ARP/proxy-ND entries are moved to the VPRN ARP/ND table. ARP/ND entries are also moved to proxy-ARP/proxy-ND entries if the VPLS is unbound.
- EVPN does not program EVPN-received ARP/ND entries if the receiving VPRN has no IP addresses for the same subnet. The entries are added when the IP address for the same subnet is added.
- Static ARP/ND entries have precedence over dynamic and EVPN ARP/ND entries.
- VPRN interface binding to VPLS service brings down the VPRN interface operational status, if the VPRN interface MAC or the VRRP MAC matches a static-mac or OAM MAC configured in the associated VPLS service. If that is the case, a trap is generated.
- Redundancy is handled by VRRP. The active node advertises vMAC and vIP, as discussed, including the MAC mobility extended community and the sticky bit.

EVPN-enabled R-VPLS services are also supported on IES interfaces.

### 5.2.2.3.1 EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes

**Figure 64: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC** shows a Layer 3 DC model, in which a VPRN is defined in the DGWs connects the tenant to the WAN. That VPRN instance is connected to the VPRNs in the NVEs by means of an IRB backhaul R-VPLS. Because the IRB backhaul R-VPLS provides connectivity only to all the IRB interfaces and the DGW VPRN is not directly connected to all the tenant subnets, the WAN IP prefixes in the VPRN routing table must be advertised in EVPN. In the same way, the NVEs send IP prefixes in EVPN that are received by the DGW and imported in the VPRN routing table.



**Note:** To generate or process IP prefixes sent or received in EVPN route type 5, support for IP route advertisement must be enabled in BGP-EVPN using the **bgp-evpn ip-route-advertisement** command. This command is disabled by default and must be explicitly enabled. The command is tied to the **allow-ip-int-bind** command required for R-VPLS, and it is not supported on an R-VPLS linked to IES services.

#### Example: Local router interface host addresses advertised in EVPN

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, as shown in the following 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
10.1.1.100/32                   Local  Host   00h00m11s  0
      if                          Y
=====
```

For the case displayed by the preceding output, 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

### Output example: VPRN with two IRB interfaces

The following example shows a VPRN (500) with two IRB interfaces connected to backhaul R-VPLS services 501 and 502 where EVPN-VXLAN runs.

```
vprn 500 customer 1 create
    ecmp 4
    route-distinguisher 65072:500
    vrf-target target:65000:500
    interface "evi-502" create
        address 10.20.20.72/24
        vpls "evpn-vxlan-502"
    exit
    exit
    interface "evi-501" create
        address 10.10.10.72/24
        vpls "evpn-vxlan-501"
    exit
    exit
    no shutdown
vpls 501 name "evpn-vxlan-501" customer 1 create
    allow-ip-int-bind
    vxlan instance 1 vni 501 create
    exit
    bgp
        route-distinguisher 65072:501
        route-target export target:65000:501 import target:65000:501
    exit
    bgp-evpn
        ip-route-advertisement incl-host
        vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
    exit
    no shutdown
    exit
vpls 502 name "evpn-vxlan-502" customer 1 create
    allow-ip-int-bind
    vxlan instance 1 vni 502 create
    exit
    bgp
        route-distinguisher 65072:502
        route-target export target:65000:502 import target:65000:502
    exit
    bgp-evpn
        ip-route-advertisement incl-host
        vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
    exit
    no shutdown
    exit
```

When the preceding commands are enabled, the router behaves as follows:



- Receive route-type 5 routes and import the IP prefixes and associated IP next-hops into the VPRN routing table.
  - If the route-type 5 is successfully imported by the router, the prefix included in the route-type 5 (for example, 10.0.0.0/24), is added to the VPRN routing table with a next hop equal to the gateway IP included in the route (for example, 192.0.0.1. that refers to the IRB IP address of the remote VPRN behind which the IP prefix sits).
  - When the router receives a packet from the WAN to the 10.0.0.0/24 subnet, the IP lookup on the VPRN routing table yields 192.0.0.1 as the next-hop. That next-hop is resolved to a MAC in the ARP table and the MAC resolved to a VXLAN tunnel in the FDB table



**Note:** IRB MAC and IP addresses are advertised in the IRB backhaul R-VPLS in routes type 2.

- Generate route-type 5 routes for the IP prefixes in the associated VPRN routing table.  
For example, if VPRN-1 is attached to EVPN R-VPLS 1 and EVPN R-VPLS 2, and R-VPLS 2 has **bgp-evpn ip-route-advertisement** configured, the node advertises the R-VPLS 1 interface subnet in one route-type 5.
- Routing policies can filter the imported and exported IP prefix routes accordingly.

### Output example: VPRN routing table receiving routes

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 in the following example.

```
*A:PE72# show router 500 route-table
=====
Route Table (Service: 500)
=====
```

| Dest Prefix[Flags]<br>Next Hop[Interface Name] | Type   | Proto    | Age<br>Metric  | Pref |
|--|--------|----------|----------------|------|
| 10.20.20.0/24<br>evi-502                       | Local  | Local    | 01d11h10m<br>0 | 0    |
| 10.20.20.71/32<br>10.10.10.71                  | Remote | BGP EVPN | 00h02m26s<br>0 | 169  |
| 10.10.10.0/24<br>10.10.10.71                   | Remote | Static   | 00h00m05s<br>1 | 5    |
| 10.16.0.1/32<br>10.10.10.71                    | Remote | BGP EVPN | 00h02m26s<br>0 | 169  |

```
-----
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. If the same route is received from the WAN over BGP-VPRN and from BGP-EVPN, the EVPN route is preferred.
- When the same route-type 5 prefix is received from different gateway IPs, ECMP is supported if configured in the VPRN.
- All routes in the VPRN routing table (as long as they do not point back to the EVPN R-VPLS interface) are advertised via EVPN.

Although the preceding description focuses on IPv4 interfaces and prefixes, it also applies to IPv6 interfaces. The following considerations are specific to the IPv6 VPRN R-VPLS interfaces:



- The IPv4 and IPv6 interfaces can be defined on R-VPLS IP interfaces at the same time (dual stack).
- The user may configure specific IPv6 global addresses on the VPRN R-VPLS interfaces. If a specific global IPv6 address is not configured on the interface, the link-local address interface MAC/IP is advertised in a route type 2 as soon as IPv6 is enabled on the VPRN R-VPLS interface.
- Routes type 5 for IPv6 prefixes are advertised using either the configured global address or the implicit link-local address (if no global address is configured).

If more than one global address is configured, typically the first IPv6 address is used as gateway IP. The is the first address on the list of IPv6 addresses displayed using SNMP or using the following command.

```
show router interface ipv6
```

The remaining addresses are advertised only in MAC-IP routes (Route Type 2) but not used as gateway IP for IPv6 prefix routes.

### 5.2.2.3.2 EVPN for VXLAN in EVPN tunnel R-VPLS services

[Figure 65: EVPN-tunnel gateway IRB on the DC PE for a Layer 3 EVPN/VXLAN DC](#) shows a Layer 3 connectivity model that optimizes the solution described in [EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes](#). Instead of regular IRB backhaul R-VPLS services for the connectivity of all the VPRN IRB interfaces, EVPN tunnels can be configured. The main advantage of using EVPN tunnels is that, unlike regular IRB R-VPLS interfaces, they do not need configuration of IP addresses.

In addition to the **ip-route-advertisement** command, this model requires the configuration of the following command

```
configure service vprn interface vpls evpn-tunnel
```




**Note:** Although EVPN tunnels can be enabled independently of the **ip-route-advertisement** command, no route-type 5 advertisements are sent or processed. Neither the **evpn-tunnel** nor the **ip-route-advertisement** command, is supported on R-VPLS services linked to IES interfaces.

#### Example: VPRN (500) with an EVPN-tunnel R-VPLS (504)

```
vprn 500 name "vprn500" customer 1 create
    ecmp 4
    route-distinguisher 65071:500
    vrf-target target:65000:500
    interface "evi-504" create
        vpls "evpn-vxlan-504"
        evpn-tunnel
    exit
exit
no shutdown
exit
vpls 504 name "evpn-vxlan-504" customer 1 create
    allow-ip-int-bind
    vxlan instance 1 vni 504 create
    exit
    bgp
        route-distinguisher 65071:504
        route-target export target:65000:504 import target:65000:504
    exit
    bgp-evpn
        ip-route-advertisement
```

```
        vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
exit
no shutdown
exit
```

A specified VPRN supports regular IRB backhaul R-VPLS services as well as EVPN tunnel R-VPLS services.



**Note:** EVPN tunnel R-VPLS services do not support SAPs or SDP-binds.

The process followed upon receiving a route type 5 on a regular IRB R-VPLS interface differs from the one for an EVPN-tunnel type:

- IRB backhaul R-VPLS VPRN interface:
  - When a route-type 2 that includes an IP prefix is received and it becomes active, the MAC/IP information is added to the FDB and ARP tables. This can be checked with the **show router arp** command and the **show service id fdb detail** command.
  - When route-type 5 is received and becomes active for the R-VPLS service, the IP prefix is added to the VPRN routing table, regardless of the existence of a route-type 2 that can resolve the gateway IP address. If a packet is received from the WAN side and the IP lookup hits an entry for which the gateway IP (IP next-hop) does not have an active ARP entry, the system uses ARP to get a MAC. If ARP is resolved but the MAC is unknown in the FDB table, the system floods into the TLS multicast list. Routes type 5 can be checked in the routing table with the **show router route-table** and **show router fib** commands.
- EVPN tunnel R-VPLS VPRN interface:
  - When route-type 2 is received and becomes active, the MAC address is added to the FDB only.
  - When a route-type 5 is received and active, the IP prefix is added to the VPRN routing table with next-hop equal to EVPN tunnel: GW-MAC.

For example, ET-d8:45:ff:00:01:35, where the GW-MAC is added from the GW-MAC extended community sent along with the route-type 5.

If a packet is received from the WAN side and the IP lookup hits an entry for which the next-hop is a EVPN tunnel: GW-MAC, the system looks up the GW-MAC in the FDB. Usually a route-type 2 with the GW-MAC is previously received so that the GW-MAC can be added to the FDB. If the GW-MAC is not present in the FDB, the packet is dropped.

- IP prefixes with GW-MACs as next hops are displayed by the **show router** command, as shown below:

Output example: IP prefixes with GW-MACs as next hops

```
*A:PE71# show router 500 route-table
=====
Route Table (Service: 500)
=====
Dest Prefix[Flags]                Type  Proto  Age      Pref
Next Hop[Interface Name]          Metric
-----
10.20.20.72/32                    Remote BGP  EVPN  00h23m50s 169
10.10.10.72                        0
10.30.30.0/24                     Remote BGP  EVPN  01d11h30m 169
evi-504 (ET-d8:45:ff:00:01:35)    0
```

```

10.10.10.0/24 Remote BGP VPN 00h20m52s 170
    192.0.2.69 (tunneled) 0
10.1.0.0/16 Remote BGP EVPN 00h22m33s 169
    evi-504 (ET-d8:45:ff:00:01:35) 0
-----
No. of Routes: 4

```

### Output example: GW-MAC and remaining IP prefix BGP attributes

```

*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
Aggregator AS : None
Atomic Aggr. : Not Atomic
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
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : No As-Path
EVPN type    : IP-PREFIX
ESI          : N/A
Gateway Address: 00:00:01:00:01:02
Prefix       : 3.0.1.6/32
MPLS Label   : 262140
Route Tag    : 0xb
Neighbor-AS  : N/A
Orig Validation: N/A
Source Class : 0

Interface Name : NotAvailable
Aggregator     : None
MED            : 0
Peer Router Id : 10.20.1.2
Route Dist.    : 10.20.1.2:1
Tag            : 1
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
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric  : None
Connector    : None
Community    : target:100:1 mac-nh:00:00:01:00:01:02
              bgp-tunnel-encap:VXLAN
Interface Name : NotAvailable
Aggregator     : None
MED            : 0

```

```

Cluster      : No Cluster Members
Originator Id : None                      Peer Router Id : 10.20.1.2
Flags        : Used Valid Best IGP
Route Source  : Internal
AS-Path       : 111
EVPN type     : IP-PREFIX
ESI           : N/A                      Tag           : 1
Gateway Address: 00:00:01:00:01:02
Prefix        : 3.0.1.6/32              Route Dist.     : 10.20.1.2:1
MPLS Label    : 262140
Route Tag     : 0xb
Neighbor-AS   : 111
Orig Validation: N/A
Source Class  : 0                      Dest Class      : 0

```

```

-----
Routes : 1
=====

```

EVPN tunneling is also supported on IPv6 VPRN interfaces. When sending IPv6 prefixes from IPv6 interfaces, the GW-MAC in the route type 5 (IP-prefix route) is always zero. If no specific Global Address is configured on the IPv6 interface, the routes type 5 for IPv6 prefixes are always sent using the Link Local Address as GW-IP.

### Output example: IPv6 prefix received via BGP EVPN

```

*A:PE71# show router 30 route-table ipv6

=====
IPv6 Route Table (Service: 30)
=====
Dest Prefix[Flags]                Type   Proto   Age      Pref
Next Hop[Interface Name]          Metric
-----
2001:db8:1000::/64                Local   Local   00h01m19s  0
int-PE-71-CE-1                    0
2001:db8:2000::1/128              Remote  BGP EVPN 00h01m20s 169
fe80::da45:ffff:fe00:6a-"int-evi-301" 0
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

*A:PE71# show router bgp routes evpn ipv6-prefix prefix 2001:db8:2000::1/128 hunt
=====
BGP Router ID:192.0.2.71          AS:64500          Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked
Origin codes  : i - IGP, e - EGP, ? - incomplete, > - best, b - backup
=====
BGP EVPN IP-Prefix Routes
=====
RIB In Entries
-----
Network      : N/A

```

```

Nexthop      : 192.0.2.69
From         : 192.0.2.69
Res. Nexthop : 192.168.19.2
Local Pref.  : 100
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric  : None
Connector    : None
Community    : target:64500:301 bgp-tunnel-encap:VXLAN
Cluster      : No Cluster Members
Originator Id : None
Flags        : Used Valid Best IGP
Route Source  : Internal
AS-Path       : No As-Path
EVPN type     : IP-PREFIX
ESI          : N/A
Gateway Address: fe80::da45:ffff:fe00:*
Prefix       : 2001:db8:2000::1/128
MPLS Label   : 0
Route Tag    : 0
Neighbor-AS   : N/A
Orig Validation: N/A
Source Class  : 0
Add Paths Send : Default
Last Modified : 00h41m17s

```

```

-----
RIB Out Entries
-----

```

```

-----
Routes : 1
=====

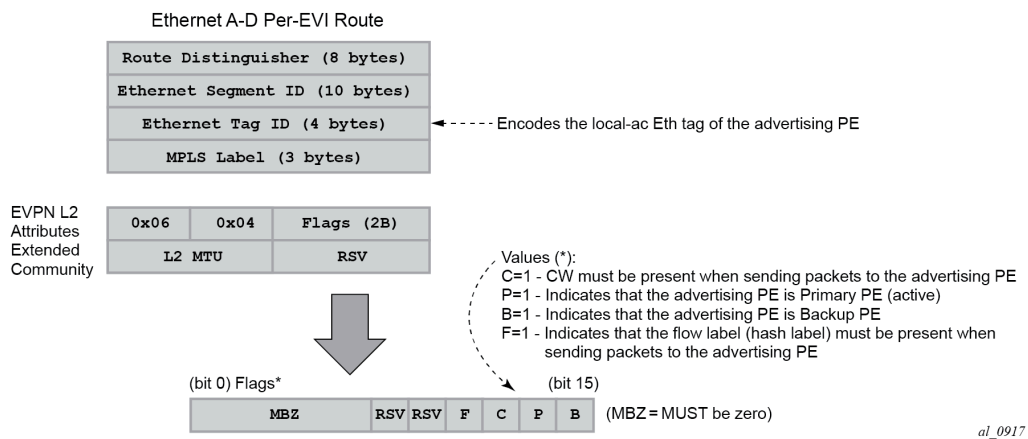
```

### 5.2.2.4 EVPN-VPWS for VXLAN tunnels

#### BGP-EVPN control plane for EVPN-VPWS

EVPN-VPWS uses route-type 1 and route-type 4; it does not use route-types 2, 3 or 5. [Figure 71: EVPN-VPWS BGP extensions](#) shows the encoding of the required extensions for the Ethernet A-D per-EVI routes. The encoding follows the guidelines described in RFC 8214.

Figure 71: EVPN-VPWS BGP extensions



If the advertising PE has an access SAP-SDP or spoke SDP that is not part of an Ethernet Segment (ES), the PE populates the fields of the AD per-EVI route with the following values:

- Ethernet Tag ID field is encoded with the value configured by the user in the **service bgp-evpn local-attachment-circuit eth-tag value** command.
- RD and MPLS label values are encoded as specified in RFC 7432. For VXLAN, the MPLS field encodes the VXLAN VNI.
- ESI is 0.
- The route is sent along an EVPN L2 attributes extended community, as specified in RFC 8214, where:
  - type and subtype are 0x06 and 0x04 as allocated by IANA
  - flag C is set if a control word is configured in the service; C is always zero for VXLAN tunnels
  - P and B flags are zero
  - L2 MTU is encoded with a service MTU configured in the Epipe service

If the advertising PE has an access SAP-SDP or spoke SDP that is part of an ES, the AD per-EVI route is sent with the information described above, with the following minor differences:

- The ESI encodes the corresponding non-zero value.
- The P and B flags are set in the following cases:
  - All-active multihoming
    - All PEs that are part of the ES always set the P flag.
    - The B flag is never set in the all-active multihoming ES case.
  - Single-active multihoming
    - Only the DF PE sets the P bit for an EVI and the remaining PEs send it as P=0.
    - Only the backup DF PE sets the B bit.

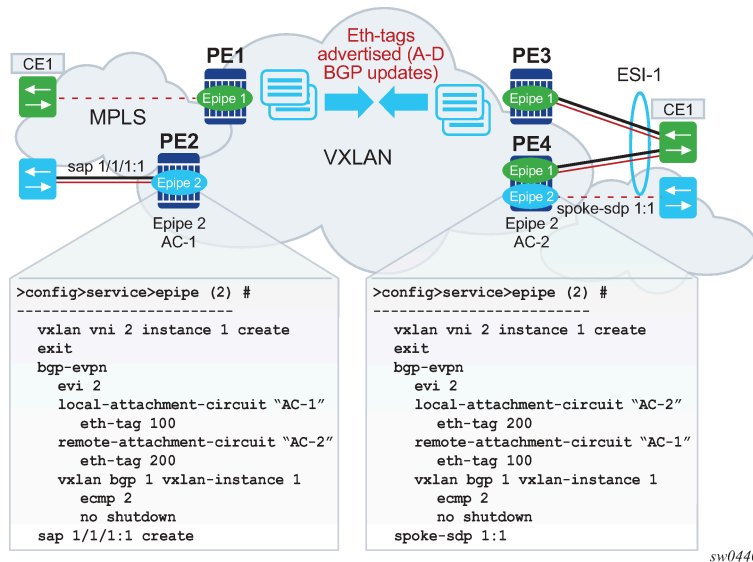
If more than two PEs are present in the same single-active ES, the backup PE is the winner of a second DF election (excluding the DF). The remaining non-DF PEs send B=0.

Also, ES and AD per-ES routes are advertised and processed for the Ethernet-Segment, as described in RFC 7432 ESs. The ESI label sent with the AD per-ES route is used by BUM traffic on VPLS services; it is not used for Epipe traffic.

### EVPN-VPWS for VXLAN tunnels in Epipe services

BGP-EVPN can be enabled in Epipe services with either SAPs or spoke SDPs at the access, as shown in [Figure 72: EVPN-MPLS VPWS](#).

Figure 72: EVPN-MPLS VPWS



EVPN-VPWS is supported in VXLAN networks that also run EVPN-VXLAN in VPLS services. From a control plane perspective, EVPN-VPWS is a simplified point-to-point version of RFC 7432 for E-Line services for the following reasons:

- EVPN-VPWS does not use inclusive multicast, MAC/IP routes or IP-prefix routes.
- AD Ethernet per-EVI routes are used to advertise the local attachment circuit identifiers at each side of the VPWS instance. The attachment circuit identifiers are configured as local and remote Ethernet tags. When an AD per-EVI route is imported and the Ethernet tag matches the configured remote Ethernet tag, an EVPN destination is created for the Epipe.

In the following configuration example, Epipe 2 is an EVPN-VPWS service between PE2 and PE4 (as shown in [Figure 72: EVPN-MPLS VPWS](#)).

```
PE2>config>service>epipe(2)#
-----
vxlan vni 2 instance 1 create
exit
bgp
exit
bgp-evpn
evi 2
local-attachment-circuit "AC-1"
eth-tag 100
remote-attachment-circuit "AC-2"
eth-tag 200
vxlan bgp 1 vxlan-instance 1
```

```

    ecmp 2
    no shutdown
    sap 1/1/1:1 create

PE4>config>service>epipe(2)#
-----
vxlan vni 2 instance 1 create
exit
bgp
exit
bgp-evpn
    evi 2
        local-attachment-circuit "AC-2"
            eth-tag 200
        remote-attachment-circuit "AC-1"
            eth-tag 100
    vxlan bgp 1 vxlan-instance 1
        ecmp 2
        no shutdown
    spoke-sdp 1:1

```

The following considerations apply to the preceding example configuration:

- When the EVI value is lower than 65535, the EVI is used to automatically derive the route-target or route-distinguisher of the service. For EVI values greater than 65535, the route-distinguisher is not automatically derived and the route-target is automatically derived, if **evi-three-byte-auto-rt** is configured. The EVI values must be unique in the system regardless of the type of service to which they are assigned (Epipe or VPLS).
- Support for the following BGP-EVPN commands in Epipe services is the same as in VPLS services:
  - **vxlan bgp 1 vxlan-instance 1**
  - **vxlan send-tunnel-encap**
  - **vxlan shutdown**
  - **vxlan ecmp**
- The following BGP-EVPN commands identify the local and remote attachment circuits, with the configured Ethernet tags encoded in the advertised and received AD Ethernet per-EVI routes:
  - **local-attachment-circuit name**
  - **local-attachment-circuit name eth-tag tag-value**; where **tag-value** is 1 to 16777215
  - **remote-attachment-circuit name**
  - **remote-attachment-circuit name eth-tag tag-value**; where **tag-value** is 1 to 16777215

Changes to remote Ethernet tags are allowed without shutting down BGP-EVPN VXLAN or the Epipe service. The local AC Ethernet tag value cannot be changed without BGP-EVPN VXLAN shutdown.

Both local and remote Ethernet tags are mandatory to bring up the Epipe service.

EVPN-VPWS Epipes can also be configured with the following characteristics:

- Access attachment circuits can be SAPs or spoke SDP. Only manually-configured spoke SDP is supported; BGP-VPWS and endpoints are not supported. The VC switching configuration is not supported on BGP-EVPN enabled pipes.
- EVPN-VPWS Epipes can advertise the Layer 2 (service) MTU and check its consistency as follows:



1. The advertised MTU value is taken from the configured service MTU in the Epipe service.
2. The received L2 MTU is compared to the local value. In case of a mismatch between the received MTU and the configured service MTU, the system does not set up the EVPN destination; as a result, the service does not come up.

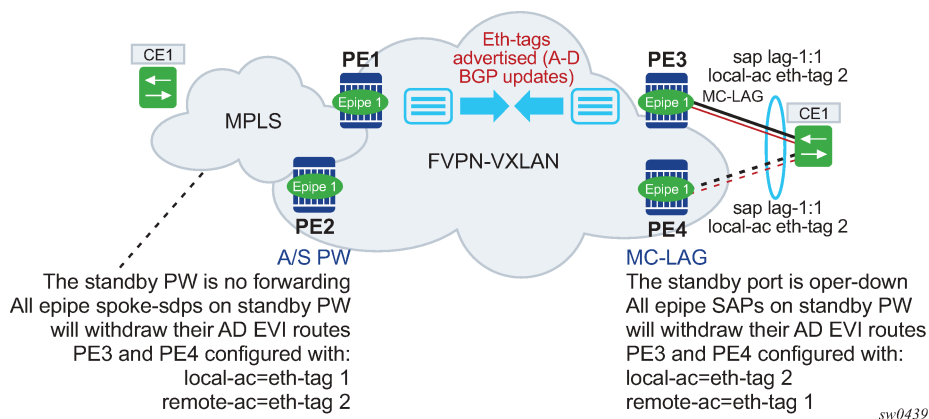
Consider the following:

- The system does not check the network port MTU value.
- If the received L2 MTU value is 0, the MTU is ignored.

### Using A/S PW and MC-LAG with EVPN-VPWS Epipes

The use of A/S PW (for access spoke SDP) and MC-LAG (for access SAPs) provides an alternative redundant solution for EVPN-VPWS that do not use the EVPN multi homing procedures described in RFC 8214. [Figure 73: A/S PW and MC-LAG support on EVPN-VPWS](#) shows the use of both mechanisms in a single Epipe.

Figure 73: A/S PW and MC-LAG support on EVPN-VPWS



In [Figure 73: A/S PW and MC-LAG support on EVPN-VPWS](#), an A/S PW connects the CE to PE1 and PE2 (left side of the diagram), and an MC-LAG connects the CE to PE3 and PE4 (right side of the diagram). As EVPN multi homing is not used, there are no AD per-ES routes or ES routes. The redundancy is handled as follows:

- PE1 and PE2 are configured with Epipe-1, where a spoke SDP connects the service in each PE to the access CE. The local AC Ethernet tag is 1 and the remote AC Ethernet tag is 2 (in PE1/PE2).
- PE3 and PE4 are configured with Epipe-1, where each PE has a lag SAP that belongs to a previously-configured MC-LAG construct. The local AC Ethernet tag is 2 and the remote AC Ethernet tag is 1.
- An endpoint and A/S PW is configured on the CE on the left side of the diagram. PE1/PE2 are able to advertise Ethernet tag 1 based on the operating status or the forwarding status of the spoke SDP.  
For example, if PE1 receives a standby PW status indication from the CE and the previous status was forward, it withdraws the AD EVI route for Ethernet tag 1. If PE2 receives a forward PW status indication and the previous status was standby or down, it advertises the AD EVI route for Ethernet tag 1.
- The user can configure MC-LAG for access SAPs using the example configuration of PE3 and PE4, as shown in [Figure 73: A/S PW and MC-LAG support on EVPN-VPWS](#). In this case, the MC-LAG determines which chassis is active and which is standby.

If PE4 becomes the standby chassis, the entire LAG port is brought down. As a result, the SAP goes operationally down and PE4 withdraws any previous AD EVI routes for Ethernet tag 2.

If PE3 becomes the active chassis, the LAG port becomes operationally up. As a result, the SAP and the PE3 advertise the AD per-EVI route for Ethernet tag 2.

## EVPN multihoming for EVPN-VPWS services

EVPN multihoming is supported for EVPN-VPWS Epipe services with the following considerations:

- Single-active and all-active multihoming is supported for SAPs and spoke SDP.
- ESs can be shared between the Epipe (MPLS and VXLAN) and VPLS (MPLS) services for LAGs, ports, and SDPs.
- No split-horizon function is required because no traffic exists between the Designated Forwarder (DF) and the non-DF for Epipe services. As a result, the ESI label is never used, and the following commands do not affect Epipe services. Additionally, configure the **single-active-no-esi-label** or **all-active-no-esi-label** modes to increase the scale of Ethernet Segments for EVPN VPWS services.

### – MD-CLI

```
configure service system bgp evpn ethernet-segment multi-homing-mode single-active-no-esi-label
configure service system bgp evpn ethernet-segment multi-homing-mode all-active-no-esi-label
configure service system bgp evpn ethernet-segment pbb source-bmac-lsb
```

### – classic CLI

```
configure service system bgp-evpn ethernet-segment multi-homing single-active no-esi-label
configure service system bgp-evpn ethernet-segment multi-homing all-active no-esi-label
configure service system bgp-evpn ethernet-segment source-bmac-lsb
```

- The local Ethernet tag values must match on all PEs that are part of the same ES, regardless of the multi homing mode. The PEs in the ES use the AD per-EVI routes from the peer PEs to validate the PEs as DF election candidates for a specific EVI.

The DF election for Epipes that is defined in an all-active multi homing ES is not relevant because all PEs in the ES behave in the same way as follows:

- All PEs send P=1 on the AD per-EVI routes.
- All PEs can send upstream and downstream traffic, regardless of whether the traffic is unicast, multicast, or broadcast (all traffic is treated as unicast in the Epipe services).

Therefore, the following tools command shows **N/A** when all-active multihoming is configured.

```
*A:PE-2# tools dump service system bgp-evpn ethernet-segment "ESI-12" evi 6000 df
[03/18/2016 20:31:35] All Active VPWS - DF N/A
```

Aliasing is supported for traffic sent to an ES destination. If ECMP is enabled on the ingress PE, per-flow load balancing is performed to all PEs that advertise P=1. The PEs that advertise P=0, are not considered as next hops for an ES destination.

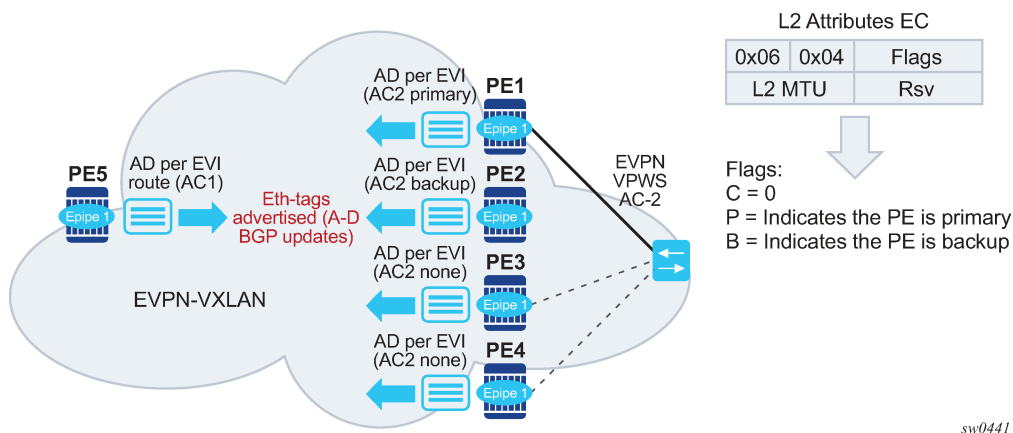


**Note:** The ingress PE load balances the traffic if shared queuing or ingress policing is enabled on the access SAPs.

Although DF election is not relevant for Epipes in an all-active multi homing ES, it is essential for the following forwarding and backup functions in a single-active multihoming ES:

- The PE elected as DF is the primary PE for the ES in the Epipe. The primary PE unblocks the SAP or spoke SDP for upstream and downstream traffic; the remaining PEs in the ES bring their ES SAPs or spoke SDPs operationally down.
- The DF candidate list is built from the PEs sending ES routes for the same ES and is pruned for a specific service, depending on the availability of the AD per-ES and per-EVI routes.
- When the SAP or spoke SDPs that are part of the ES come up, the AD per-EVI routes are sent with P=0 and B=0. The remote PEs do not start sending traffic until the DF election process is complete and the ES activation timer is expired, and the PEs advertise AD per-EVI routes with P and B bits other than zero.
- The backup PE function is supported as defined in RFC 8214. The primary PE, backup, or none status is signaled by the PEs (part of the same single-active MH ES) in the P or B flags of the EVPN L2 attributes extended community. [Figure 74: EVPN-VPWS single-active multihoming](#) shows the advertisement and use of the primary, backup, or none indication by the PEs in the ES.

Figure 74: EVPN-VPWS single-active multihoming



As specified in RFC 7432, the remote PEs in VPLS services have knowledge of the primary PE in the remote single-active ES, based on the advertisement of the MAC/IP routes because only the DF learns and advertises MAC/IP routes.

Because there are no MAC/IP routes in EVPN-VPWS, the remote PEs can forward the traffic based on the P/B bits. The process is described in the following list:

1. The DF PE for an EVI (PE1) sends P=1 and B=0.
  2. For each ES or EVI, a second DF election is run among the PEs in the backup candidate list to elect the backup PE. The backup PE sends P=0 and B=1 (PE2).
  3. All remaining multi homing PEs send P=0 and B=0 (PE3 and PE4).
  4. At the remote PEs (PE5), the P and B flags are used to identify the primary and backup PEs within the ES destination. The traffic is then sent to the primary PE, provided that it is active.
- When a remote PE receives the withdrawal of an Ethernet AD per-ES (or per-EVI) route from the primary PE, the remote PE immediately switches the traffic to the backup PE for the affected EVIs. The backup PE takes over immediately without waiting for the ES activation timer to bring up its SAP or spoke SDP.

- The BGP-EVPN MPLS ECMP setting also governs the forwarding in single-active multi homing, regardless of the single-active multi homing bit in the AD per-ES route received at the remote PE (PE5).
  - PE5 always sends the traffic to the primary remote PE (the owner of the P=1 bit). In case of multiple primary PEs and ECMP>1, PE5 load balances the traffic to all primary PEs, regardless of the multi homing mode.
  - If the last primary PE withdraws its AD per-EVI or per-ES route, PE5 sends the traffic to the backup PE or PEs. In case of multiple backup PEs and ECMP>1, PE1 load balances the traffic to the backup PEs.

### Non-system IPv4/IPv6 VXLAN termination for EVPN-VPWS services

EVPN-VPWS services support non-system IPv4/IPv6 VXLAN termination. For system configuration information, see [Non-system IPv4 and IPv6 VXLAN termination in VPLS, R-VPLS, and Epipe services](#).

EVPN multihoming is supported when the PEs use non-system IP termination, however additional configuration steps are needed in this case:

- The **configure service system bgp-evpn eth-seg es-orig-ip ip-address** command must be configured with the non-system IPv4/IPv6 address used for the EVPN-VPWS VXLAN service. As a result, this command modifies the originating-ip field in the ES routes advertised for the Ethernet Segment, and makes the system use this IP address when adding the local PE as DF candidate.
- The **configure service system bgp-evpn eth-seg route-next-hop ip-address** command must be configured with the non-system IP address, too. The command changes the next-hop of the ES and AD per-ES routes to the configured address.
- The non-system IP address (in each of the PEs in the ES) must match in these three commands for the local PE to be considered suitable for DF election:
  - **es-orig-ip ip-address**
  - **route-next-hop ip-address**
  - **vxlan-src-vtep ip-address**

#### 5.2.2.4.1 EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes

**Figure 64: Gateway IRB on the DC PE for an L3 EVPN/VXLAN DC** shows a Layer 3 DC model, where a VPRN is defined in the DGWs, connecting the tenant to the WAN. That VPRN instance is connected to the VPRNs in the NVEs by means of an IRB backhaul R-VPLS. Because the IRB backhaul R-VPLS provides connectivity only to all the IRB interfaces and the DGW VPRN is not directly connected to all the tenant subnets, the WAN ip-prefixes in the VPRN routing table must be advertised in EVPN. In the same way, the NVEs send IP prefixes in EVPN that is received by the DGW and imported in the VPRN routing table.



**Note:** To generate or process IP prefixes sent or received in EVPN route type 5, the support for IP route advertisement must be enabled in BGP-EVPN. This is performed through the **bgp-evpn ip-route-advertisement** command. This command is disabled by default and must be explicitly enabled. The command is tied to the **allow-ip-int-bind** command required for R-VPLS, and it is not supported on R-VPLS linked to IES services.

Local router interface host addresses are not advertised in EVPN by default. To advertise them, the **ip-route-advertisement incl-host** command must be enabled. For example:

```
=====
Route Table (Service: 2)
```

| Dest Prefix[Flags]<br>Next Hop[Interface Name] | Type  | Proto<br>Active | Age<br>Metric  | Pref |
|--|-------|-----------------|----------------|------|
| 10.1.1.0/24<br>if                              | Local | Local<br>Y      | 00h00m11s<br>0 | 0    |
| 10.1.1.100/32<br>if                            | Local | Host<br>Y       | 00h00m11s<br>0 | 0    |

For the case displayed by the output above, the behavior is the following:

- **ip-route-advertisement** only local subnet (default) - 10.1.1.0/24 is advertised
- **ip-route-advertisement incl-host** local subnet, host - 10.1.1.0/24 and 10.1.1.100/32 are advertised

Below is an example of VPRN (500) with two IRB interfaces connected to backhaul R-VPLS services 501 and 502 where EVPN-VXLAN runs:

```
vprn 500 customer 1 create
    ecmp 4
    route-distinguisher 65072:500
    vrf-target target:65000:500
    interface "evi-502" create
        address 10.20.20.72/24
        vpls "evpn-vxlan-502"
    exit
    interface "evi-501" create
        address 10.10.10.72/24
        vpls "evpn-vxlan-501"
    exit
    no shutdown
vpls 501 name "evpn-vxlan-501" customer 1 create
    allow-ip-int-bind
    vxlan instance 1 vni 501 create
    exit
    bgp
        route-distinguisher 65072:501
        route-target export target:65000:501 import target:65000:501
    exit
    bgp-evpn
        ip-route-advertisement incl-host
        vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
    no shutdown
    exit
vpls 502 name "evpn-vxlan-502" customer 1 create
    allow-ip-int-bind
    vxlan instance 1 vni 502 create
    exit
    bgp
        route-distinguisher 65072:502
        route-target export target:65000:502 import target:65000:502
    exit
    bgp-evpn
        ip-route-advertisement incl-host
        vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
    no shutdown
    exit
```

```
no shutdown
exit
```

When the above commands are enabled, the router behaves as follows:

- Receive route-type 5 routes and import the IP prefixes and associated IP next-hops into the VPRN routing table.
  - If the route-type 5 is successfully imported by the router, the prefix included in the route-type 5 (for example, 10.0.0.0/24), is added to the VPRN routing table with a next-hop equal to the gateway IP included in the route (for example, 192.0.0.1. that refers to the IRB IP address of the remote VPRN behind which the IP prefix sits).
  - When the router receives a packet from the WAN to the 10.0.0.0/24 subnet, the IP lookup on the VPRN routing table yields 192.0.0.1 as the next-hop. That next-hop is resolved to a MAC in the ARP table and the MAC resolved to a VXLAN tunnel in the FDB table



**Note:** IRB MAC and IP addresses are advertised in the IRB backhaul R-VPLS in routes type 2.

- Generate route-type 5 routes for the IP prefixes in the associated VPRN routing table.

For example, if VPRN-1 is attached to EVPN R-VPLS 1 and EVPN R-VPLS 2, and R-VPLS 2 has **bgp-evpn ip-route-advertisement** configured, the router advertises the R-VPLS 1 interface subnet in one route-type 5.

- Routing policies can filter the imported and exported IP prefix routes accordingly.

The VPRN routing table can receive routes from all the supported protocols (BGP-VPN, OSPF, IS-IS, RIP, static routing) as well as from IP prefixes from EVPN, as shown below:

```
*A:PE72# show router 500 route-table
=====
Route Table (Service: 500)
=====
```

| Dest Prefix[Flags]<br>Next Hop[Interface Name] | Type   | Proto    | Age<br>Metric  | Pref |
|--|--------|----------|----------------|------|
| 10.20.20.0/24<br>evi-502                       | Local  | Local    | 01d11h10m<br>0 | 0    |
| 10.20.20.71/32<br>10.10.10.71                  | Remote | BGP EVPN | 00h02m26s<br>0 | 169  |
| 10.10.10.0/24<br>10.10.10.71                   | Remote | Static   | 00h00m05s<br>1 | 5    |
| 10.16.0.1/32<br>10.10.10.71                    | Remote | BGP EVPN | 00h02m26s<br>0 | 169  |

```
-----
No. of Routes: 4
```

The following considerations apply:

- The route Preference for EVPN IP prefixes is 169.  
BGP IP-VPN routes have a preference of 170 by default, therefore, if the same route is received from the WAN over BGP-VPRN and from BGP-EVPN, then the EVPN route is preferred.
- When the same route-type 5 prefix is received from different gateway IPs, ECMP is supported if configured in the VPRN.

- All routes in the VPRN routing table (as long as they do not point back to the EVPN R-VPLS interface) are advertised via EVPN.

Although the description above is focused on IPv4 interfaces and prefixes, it applies to IPv6 interfaces too. The following considerations are specific to IPv6 VPRN R-VPLS interfaces:

- IPv4 and IPv6 interfaces can be defined on R-VPLS IP interfaces at the same time (dual-stack).
- The user may configure specific IPv6 Global Addresses on the VPRN R-VPLS interfaces. If a specific Global IPv6 Address is not configured on the interface, the Link Local Address interface MAC/IP is advertised in a route type 2 as soon as IPv6 is enabled on the VPRN R-VPLS interface.
- Routes type 5 for IPv6 prefixes are advertised using either the configured Global Address or the implicit Link Local Address (if no Global Address is configured).

If more than one Global Address is configured, normally the first IPv6 address is used as gateway IP. The "first IPv6 address" refers to the first one on the list of IPv6 addresses shown through the **show router <id> interface interface** IPv6 or through SNMP.

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

#### 5.2.2.4.2 EVPN for VXLAN in EVPN tunnel R-VPLS services

Figure 65: EVPN-tunnel gateway IRB on the DC PE for a Layer 3 EVPN/VXLAN DC shows an L3 connectivity model that optimizes the solution described in [EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes](#). Instead of regular IRB backhaul R-VPLS services for the connectivity of all the VPRN IRB interfaces, EVPN tunnels can be configured. The main advantage of using EVPN tunnels is that they do not need the configuration of IP addresses, as regular IRB R-VPLS interfaces do.

In addition to the **ip-route-advertisement** command, this model requires the configuration of the **config>service>vprn>if>vpls <name> evpn-tunnel**.



**Note:** The **evpn-tunnel** can be enabled independently of **ip-route-advertisement**, however, no route-type 5 advertisements are sent or processed in that case. Neither command, **evpn-tunnel** and **ip-route-advertisement**, is supported on R-VPLS services linked to IES interfaces.

The example below shows a VPRN (500) with an EVPN-tunnel R-VPLS (504):

```
vprn 500 customer 1 create
    ecmp 4
    route-distinguisher 65071:500
    vrf-target target:65000:500
    interface "evi-504" create
        vpls "evpn-vxlan-504"
            evpn-tunnel
        exit
    exit
    no shutdown
exit
vpls 504 name "evpn-vxlan-504" customer 1 create
    allow-ip-int-bind
    vxlan instance 1 vni 504 create
    exit
    bgp
        route-distinguisher 65071:504
        route-target export target:65000:504 import target:65000:504
    exit
    bgp-evpn
```



```

        ip-route-advertisement
        vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
exit
no shutdown
exit

```

A specified VPRN supports regular IRB backhaul R-VPLS services as well as EVPN tunnel R-VPLS services.



**Note:** EVPN tunnel R-VPLS services do not support SAPs or SDP-binds.

The process followed upon receiving a route-type 5 on a regular IRB R-VPLS interface differs from the one for an EVPN-tunnel type:

- IRB backhaul R-VPLS VPRN interface:
  - When a route-type 2 that includes an IP prefix is received and it becomes active, the MAC/IP information is added to the FDB and ARP tables. This can be checked with the **show router arp** command and the **show service id fdb detail** command.
  - When route-type 5 is received and becomes active for the R-VPLS service, the IP prefix is added to the VPRN routing table, regardless of the existence of a route-type 2 that can resolve the gateway IP address. If a packet is received from the WAN side and the IP lookup hits an entry for which the gateway IP (IP next-hop) does not have an active ARP entry, the system uses ARP to get a MAC. If ARP is resolved but the MAC is unknown in the FDB table, the system floods into the TLS multicast list. Routes type 5 can be checked in the routing table with the **show router route-table** and **show router fib** commands.
- EVPN tunnel R-VPLS VPRN interface:
  - When route-type 2 is received and becomes active, the MAC address is added to the FDB (only).
  - When a route-type 5 is received and active, the IP prefix is added to the VPRN routing table with next-hop equal to EVPN tunnel: GW-MAC.

For example, ET-d8:45:ff:00:01:35, where the GW-MAC is added from the GW-MAC extended community sent along with the route-type 5.

If a packet is received from the WAN side, and the IP lookup hits an entry for which the next-hop is a EVPN tunnel: GW-MAC, the system looks up the GW-MAC in the FDB. Usually a route-type 2 with the GW-MAC is previously received so that the GW-MAC can be added to the FDB. If the GW-MAC is not present in the FDB, the packet is dropped.

- IP prefixes with GW-MACs as next-hops are displayed by the show router command, as shown below:

```

*A:PE71# show router 500 route-table
=====
Route Table (Service: 500)
=====
Dest Prefix[Flags]                Type   Proto   Age      Pref
Next Hop[Interface Name]          Metric
-----
10.20.20.72/32                    Remote BGP EVPN 00h23m50s 169
10.10.10.72                        0
10.30.30.0/24                     Remote BGP EVPN 01d11h30m 169
evi-504 (ET-d8:45:ff:00:01:35)    0
10.10.10.0/24                     Remote BGP VPN 00h20m52s 170

```



```

192.0.2.69 (tunneled)                                0
10.1.0.0/16                                           Remote BGP EVPN 00h22m33s 169
evi-504 (ET-d8:45:ff:00:01:35)                       0
-----
No. of Routes: 4

```

The GW-MAC as well as the rest of the IP prefix BGP attributes are displayed by the **show router bgp routes evpn ip-prefix** command.

```

*A:Dut-A# show router bgp routes evpn ip-prefix prefix 3.0.1.6/32 detail
=====
BGP Router ID:10.20.1.1      AS:100      Local AS:100
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
Origin codes : i - IGP, e - EGP, ? - incomplete, > - best, b - backup

=====
BGP EVPN IP-Prefix Routes
=====
-----
Original Attributes

Network      : N/A
Nexthop      : 10.20.1.2
From         : 10.20.1.2
Res. Nexthop : 192.168.19.1
Local Pref.  : 100
Aggregator AS : None
Atomic Aggr. : Not Atomic
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
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : No As-Path
EVPN type    : IP-PREFIX
ESI          : N/A
Gateway Address: 00:00:01:00:01:02
Prefix       : 3.0.1.6/32
MPLS Label   : 262140
Route Tag    : 0xb
Neighbor-AS  : N/A
Orig Validation: N/A
Source Class : 0

Interface Name : NotAvailable
Aggregator     : None
MED            : 0
Tag            : 1
Route Dist.    : 10.20.1.2:1
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
Aggregator AS : None
Atomic Aggr. : Not Atomic
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
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : No As-Path
EVPN type    : IP-PREFIX
ESI          : N/A
Gateway Address: 00:00:01:00:01:02
Prefix       : 3.0.1.6/32
MPLS Label   : 262140
Route Tag    : 0xb
Neighbor-AS  : N/A
Orig Validation: N/A
Source Class : 0

Interface Name : NotAvailable
Aggregator     : None
MED            : 0
Tag            : 1
Route Dist.    : 10.20.1.2:1
Dest Class     : 0

```

```

Cluster      : No Cluster Members
Originator Id : None                      Peer Router Id : 10.20.1.2
Flags        : Used Valid Best IGP
Route Source  : Internal
AS-Path       : 111
EVPN type     : IP-PREFIX
ESI          : N/A                      Tag           : 1
Gateway Address: 00:00:01:00:01:02
Prefix       : 3.0.1.6/32                Route Dist.    : 10.20.1.2:1
MPLS Label    : 262140
Route Tag     : 0xb
Neighbor-AS   : 111
Orig Validation: N/A
Source Class  : 0                      Dest Class     : 0

```

```

-----
Routes : 1
=====

```

EVPN tunneling is also supported on IPv6 VPRN interfaces. When sending IPv6 prefixes from IPv6 interfaces, the GW-MAC in the route type 5 (IP-prefix route) is always zero. If no specific Global Address is configured on the IPv6 interface, the routes type 5 for IPv6 prefixes are always sent using the Link Local Address as GW-IP. The following example output shows an IPv6 prefix received through BGP EVPN.

```
*A:PE71# show router 30 route-table ipv6
```

```

=====
IPv6 Route Table (Service: 30)
=====
Dest Prefix[Flags]                Type   Proto   Age           Pref
  Next Hop[Interface Name]                Metric
-----
2001:db8:1000::/64                Local  Local   00h01m19s    0
      int-PE-71-CE-1                  0
2001:db8:2000::1/128              Remote BGP EVPN 00h01m20s   169
      fe80::da45:ffff:fe00:6a-"int-evi-301" 0
-----
No. of Routes: 2
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====

```

```
*A:PE71# show router bgp routes evpn ipv6-prefix prefix 2001:db8:2000::1/128 hunt
```

```

=====
BGP Router ID:192.0.2.71      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
              l - leaked
Origin codes  : i - IGP, e - EGP, ? - incomplete, > - best, b - backup
=====
BGP EVPN IP-Prefix Routes
=====
RIB In Entries
-----
Network      : N/A
Nexthop      : 192.0.2.69
From         : 192.0.2.69

```

```

Res. Nexthop      : 192.168.19.2
Local Pref.       : 100
Aggregator AS    : None
Atomic Aggr.     : Not Atomic
AIGP Metric      : None
Connector        : None
Community        : target:64500:301 bgp-tunnel-encap:VXLAN
Cluster          : No Cluster Members
Originator Id    : None
Flags            : Used Valid Best IGP
Route Source     : Internal
AS-Path          : No As-Path
EVPN type        : IP-PREFIX
ESI              : N/A
Gateway Address  : fe80::da45:ffff:fe00:*
Prefix           : 2001:db8:2000::1/128
MPLS Label       : 0
Route Tag        : 0
Neighbor-AS      : N/A
Orig Validation  : N/A
Source Class     : 0
Add Paths Send   : Default
Last Modified    : 00h41m17s

Interface Name   : int-71-69
Aggregator      : None
MED             : 0
Tag             : 301
Route Dist.     : 192.0.2.69:301
Dest Class      : 0

```

```

-----
RIB Out Entries
-----

```

```

-----
Routes : 1
=====

```

### 5.2.3 Layer 2 multicast optimization for VXLAN (Assisted-Replication)

The Assisted-Replication feature for IPv4 VXLAN tunnels (both Leaf and Replicator functions) is supported in compliance with the non-selective mode described in IETF Draft *draft-ietf-bess-evpn-optimized-ir*.

The Assisted-Replication feature is a Layer 2 multicast optimization feature that helps software-based PE and NVEs with low-performance replication capabilities to deliver broadcast and multicast Layer 2 traffic to remote VTEPs in the VPLS service.

The EVPN and proxy-ARP/ND capabilities can reduce the amount of broadcast and unknown unicast in the VPLS service; ingress replication is sufficient for most use cases in this scenario. However, when multicast applications require a significant amount of replication at the ingress node, software-based nodes struggle because of their limited replication performance. By enabling the Assisted-Replication Leaf function, all the broadcast and multicast packets are sent to a SR OS router configured as a Replicator, which replicates the traffic to all the VTEPs in the VPLS service on behalf of the Leaf. This guarantees that the broadcast or multicast traffic is delivered to all the VPLS participants without any packet loss caused by performance issues.

The Leaf or Replicator function is enabled per VPLS service by the **configure service vpls vxlan assisted-replication {replicator | leaf}** command. In addition, the Replicator requires the configuration of an Assisted-Replication IP (AR-IP) address. The AR-IP loopback address indicates whether the received VXLAN packets have to be replicated to the remote VTEPs. The AR-IP address is configured using the **configure service system vxlan assisted-replication-ip <ip-address>** command.

Based on the **assisted-replication {replicator | leaf}** configuration, the 7705 SAR Gen 2 can behave as a Replicator (AR-R), Leaf (AR-L), or Regular Network Virtualization Edge (RNVE) router. An RNVE router does not support the Assisted-Replication feature. Because it is configured with no assisted replication, the

RNVE router ignores the AR-R and AR-L information and replicates to its flooding list where VTEPs are added based on the regular ingress replication routes.

### 5.2.3.1 Replicator (AR-R) procedures

An AR-R configuration is shown in the following example.

```
*A:PE-2>config>service>system>vxlan# info
-----
    assisted-replication-ip 10.2.2.2
-----
*A:PE-2>config>service>vpls# info
-----
    vxlan instance 1 vni 4000 create
        assisted-replication replicator
    exit
    bgp
    exit
    bgp-evpn
        evi 4000
        vxlan bgp 1 vxlan-instance 1
            no shutdown
        exit
<snip>
        no shutdown
-----
```

In this example configuration, the BGP advertises a new inclusive multicast route with tunnel-type = AR, type (T) = AR-R, and tunnel-id = originating-ip = next-hop = assisted-replication-ip (IP address 10.2.2.2 in the preceding example). In addition to the AR route, the AR-R sends a regular IR route if **ingress-repl-inc-mcast-advertisement** is enabled.



**Note:** You should disable the **ingress-repl-inc-mcast-advertisement** command if the AR-R does not have any SAP or SDP bindings and is used solely for Assisted-Replication functions.

The AR-R builds a flooding list composed of ACs (SAPs and SDP bindings) and VXLAN tunnels to remote nodes in the VPLS. All objects in the flooding list are broadcast/multicast (BM) and unknown unicast (U) capable. The following example output of the **show service id vxlan** command shows that the VXLAN destinations in the flooding list are tagged as "BUM".

```
*A:PE-2# show service id 4000 vxlan
=====
Vxlan Src Vtep IP: N/A
=====
VPLS VXLAN, Ingress VXLAN Network Id: 4000
Creation Origin: manual
Assisted-Replication: replicator
RestProtSrcMacAct: none
=====
VPLS VXLAN service Network Specifics
=====
Ing Net QoS Policy : none                Vxlan VNI Id      : 4000
Ingress FP QGrp   : (none)              Ing FP QGrp Inst : (none)
=====
Egress VTEP, VNI
=====
VTEP Address                Egress VNI  Num. MACs   Mcast Oper L2
                               State PBR
```

```

-----
192.0.2.3          4000      0      BUM    Up    No
192.0.2.5          4000      0      BUM    Up    No
192.0.2.6          4000      0      BUM    Up    No
-----

```

```

-----
Number of Egress VTEP, VNI : 3
-----
=====

```

When the AR-R receives a BUM packet on an AC, the AR-R forwards the packet to its flooding list (including the local ACs and remote VTEPs).

When the AR-R receives a BM packet on a VXLAN tunnel, it checks the IP DA of the underlay IP header and performs the following BM packet processing.



**Note:** The AR-R function is only relevant to BM packets; it does not apply to unknown unicast packets. If the AR-R receives unknown unicast packets, it sends them to the flooding list, skipping the VXLAN tunnels.

- If the destination IP matches its AR-IP, the AR-R forwards the BM packet to its flooding list (ACs and VXLAN tunnels). The AR-R performs source suppression to ensure that the traffic is not sent back to the originating Leaf.
- If the destination IP matches its regular VXLAN termination IP (IR-IP), the AR-R skips all the VXLAN tunnels from the flooding list and only replicates to the local ACs. This is the default Ingress Replication (IR) behavior.

### 5.2.3.2 Leaf (AR-L) procedures

An AR-L is configured as shown in the following example.

```

A:PE-3>config>service>vpls# info
-----
vxlan instance 1 vni 4000 create
    assisted-replication leaf replicator-activation-time 30
bgp
exit
bgp-evpn
    evi 4000
    vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
    mpls
        shutdown
    exit
exit
stp
    shutdown
exit
sap 1/1/1:4000 create
    no shutdown
exit
no shutdown
-----

```

In this example configuration, the BGP advertises a new inclusive multicast route with a tunnel-type = IR, type (T) = AR-L and tunnel-id = originating-ip = next-hop = IR-IP (IP address terminating VXLAN normally, either system-ip or vxlan-src-vtep address).

The AR-L builds a single flooding list per service but controlled by the BM and U flags. These flags are displayed in the following **show service id vxlan** command example output.

```
A:PE-3# show service id 4000 vxlan
=====
Vxlan Src Vtep IP: N/A
=====
VPLS VXLAN, Ingress VXLAN Network Id: 4000
Creation Origin: manual
Assisted-Replication: leaf      Replicator-Activation-Time: 30
RestProtSrcMacAct: none
=====
VPLS VXLAN service Network Specifics
=====
Ing Net QoS Policy : none                Vxlan VNI Id      : 4000
Ingress FP QGrp    : (none)              Ing FP QGrp Inst : (none)
=====
Egress VTEP, VNI
=====
VTEP Address                Egress VNI  Num. MACs  Mcast Oper L2
                               State PBR
-----
10.2.2.2                    4000        0          BM    Up   No
10.4.4.4                    4000        0          -     Up   No
192.0.2.2                   4000        0          U     Up   No
192.0.2.5                   4000        0          U     Up   No
192.0.2.6                   4000        0          U     Up   No
-----
Number of Egress VTEP, VNI : 5
=====
```

The AR-L creates the following VXLAN destinations when it receives and selects a Replicator-AR route or the Regular-IR routes:

- A VXLAN destination to each remote PE that sent an IR route. These bindings have the U flag set.
- A VXLAN destination to the selected AR-R. These bindings have only the BM flag set; the U flag is not set.
- The non-selected AR-Rs create a binding with flag "-" (in the CPM) that is displayed by the **show service id vxlan** command. Although the VXLAN destinations to non-selected AR-Rs do not carry any traffic, the destinations count against the total limit and must be considered when accounting for consumed VXLAN destinations in the router.

The BM traffic is only sent to the selected AR-R, whereas the U (unknown unicast) traffic is sent to all the destinations with the U flag.

The AR-L performs per-service load-balancing of the BM traffic when two or more AR-Rs exist in the same service. The AR Leaf creates a list of candidate PEs for each AR-R (ordered by IP and VNI; candidate 0 being the lowest IP and VNI). The replicator is selected out of a modulo function of the service-id and the number of replicators, as shown in the following example output.

```
A:PE-3# show service id 4000 vxlan assisted-replication replicator
=====
Vxlan AR Replicator Candidates
=====
VTEP Address                Egress VNI  In Use  In Candidate List Pending Time
-----
10.2.2.2                    4000        yes    yes                0
10.4.4.4                    4000        no     yes                0
-----
```

```
Number of entries : 2
-----
=====
```

A change in the number of Replicator-AR routes (for example, if a route is withdrawn or a new route appears) affects the result of the hashing, which may cause a different AR-R to be selected.



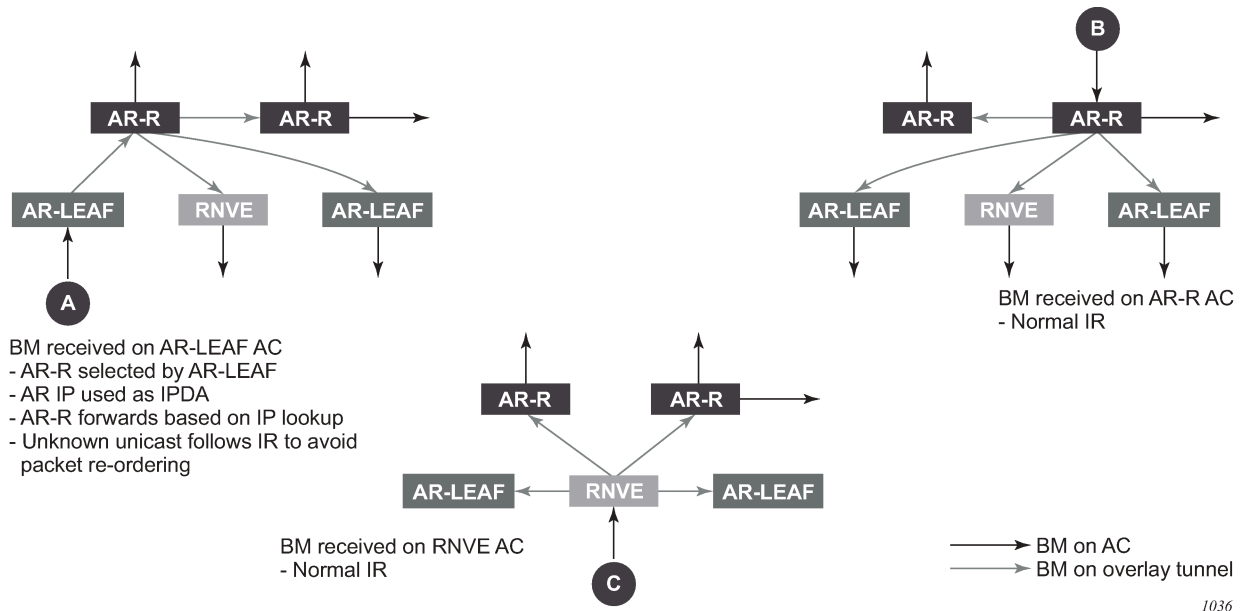
**Note:** An AR-L waits for the configured replicator-activation-time before sending the BM packets to the AR-R. In the interim, the AR-L uses regular ingress replication procedures. This activation time allows the AR-R to program the Leaf VTEP. If the timer is zero, the AR-R may receive packets from a not-yet-programmed source VTEP, in which case it discards the packets.

The following list summarizes other aspects of the AR-L behavior:

- When a Leaf receives a BM packet on an AC, it sends the packet to its flood list that includes access SAP or SDP bindings and VXLAN destinations with BM or BUM flags. If a single AR-R is selected, only a VXLAN destination includes the BM flags.
- Control plane-generated BM packets, such as ARP/ND (when proxy-ARP/ND is enabled) or Eth-CFM, follow the behavior of regular data plane BM packets.
- When a Leaf receives an unknown unicast packet on an AC, it sends the packet to the flood-list, skipping the AR destination because the U flag is set to 0. To avoid packet re-ordering, the unknown unicast packets do not go through the AR-R.
- When a Leaf receives a BUM packet on an overlay tunnel, it forwards the packet to the flood list, skipping the VXLAN tunnels (that is, the packet is sent to the local ACs and never to a VXLAN tunnel). This is the default IR behavior.
- When the last Replicator-AR route is withdrawn, the AR-L removes the AR destination from the flood list and falls back to ingress replication.

[Figure 75: AR BM replication behavior for a BM packet](#) shows the expected replication behavior for BM traffic when received at the access on an AR-R, AR-L, or RNVE router. Unknown unicast follows regular ingress replication behavior regardless of the role of the ingress node for the specific service.

Figure 75: AR BM replication behavior for a BM packet



### 5.2.3.3 Assisted-Replication interaction with other VPLS features

The Assisted-Replication feature has the following limitations:

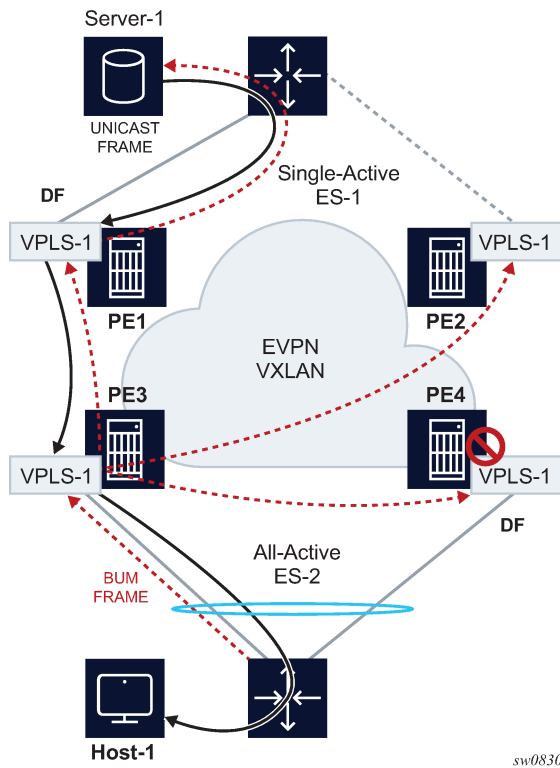
- The following features are not supported on the same service where the Assisted-Replication feature is enabled.
  - Aggregate QoS per VNI
  - VXLAN IPv6 transport
  - IGMP/MLD/PIM-snooping
- Assisted-Replication Leaf and Replicator functions are mutually exclusive within the same VPLS service.
- The Assisted-Replication feature is supported with IPv4 non-system-ip VXLAN termination. However, the configured assisted-replication-ip (AR-IP) must be different from the tunnel termination IP address.
- The AR-IP address must be a /32 loopback interface on the base router.
- The Assisted-Replication feature is only supported in EVPN-VXLAN services (VPLS with BGP-EVPN vxlan enabled). Although services with a combination of EVPN-MPLS and EVPN-VXLAN are supported, the Assisted-Replication configuration is only relevant to the VXLAN.

### 5.2.4 EVPN VXLAN multihoming

SR OS supports EVPN VXLAN multihoming as specified in RFC 8365. Similar to EVPN-MPLS, as described in [EVPN for MPLS tunnels](#), ESs and virtual ESs can be associated with VPLS and R-VPLS services where BGP-EVPN VXLAN is enabled. [Figure 76: EVPN multihoming for EVPN-VXLAN](#) illustrates the use of ESs in EVPN VXLAN networks.



Figure 76: EVPN multihoming for EVPN-VXLAN



The multihoming procedures consist of three components:

- Designated Forwarder (DF) election
- split-horizon
- aliasing

DF election is the mechanism by which the PEs attached to the same ES elect a single PE to forward all traffic (in case of single-active mode) or all BUM traffic (in case of all-active mode) to the multihomed CE. The same DF Election mechanisms described in [EVPN for MPLS tunnels](#) are supported for VXLAN services.

Split-horizon is the mechanism by which BUM traffic received from a peer ES PE is filtered so that it is not looped back to the CE that first transmitted the frame. It is applicable to all-active multihoming. This is illustrated in [Figure 76: EVPN multihoming for EVPN-VXLAN](#), where PE4 receives BUM traffic from PE3 but, in spite of being the DF for ES-2, PE4 filters the traffic and does not send it back to host-1. While split-horizon filtering uses ESI-labels in EVPN MPLS services, an alternative procedure called "Local Bias" is applied in VXLAN services, as described in RFC 8365. In MPLS services, split-horizon filtering may be used in single-active mode to avoid in-flight BUM packets from being looped back to the CE during transient times. In VXLAN services, split-horizon filtering is only used with all-active mode.

Aliasing is the procedure by which PEs that are not attached to the ES can process non-zero MAC/IP and AD routes and create ES destinations to which per-flow ecmp can be applied. Aliasing only applies to all-active mode.

As an example, the configuration of an ES that is used for VXLAN services follows. Note that this ES can be used for VXLAN services and MPLS services (in both cases VPLS and Epipes).

```
A:PE-3# configure service system bgp-evpn ethernet-segment "ES-2"
A:PE-3>config>service>system>bgp-evpn>eth-seg# info
-----
esi 01:02:00:00:00:00:00:00
service-carving
  mode manual
  manual
    preference non-revertive create
    value 10
  exit
exit
exit
multi-homing all-active
lag 1
no shutdown
-----
```

An example of configuration of a VXLAN service using the above ES follows:

```
A:PE-3# configure service vpls 1
A:PE-3>config>service>vpls# info
-----
vxlan instance 1 vni 1 create
exit
bgp
exit
bgp-evpn
  evi 1
  vxlan bgp 1 vxlan-instance 1
  ecmp 2
  auto-disc-route-advertisement
  mh-mode network
  no shutdown
  exit
exit
stp
  shutdown
exit
sap lag-1:30 create
  no shutdown
exit
no shutdown
-----
```

The **auto-disc-route-advertisement** and **mh-mode network** commands are required in all services that are attached to at least one ES, and they must be configured in both, the PEs attached to the ES locally and the remote PEs in the same service. The former enables the advertising of multihoming routes in the service, whereas the latter activates the multihoming procedures for the service, including the local bias mode for split-horizon.

In addition, the configuration of **vpls>bgp-evpn>vxlan>ecmp 2** (or greater) is required so that VXLAN ES destinations with two or more next hops can be used for per-flow load balancing. The following command shows how PE1, as shown in [Figure 76: EVPN multihoming for EVPN-VXLAN](#), creates an ES destination composed of two VXLAN next hops.

```
A:PE-1# show service id 1 vxlan destinations
=====
```

```

Egress VTEP, VNI
=====
Instance      VTEP Address      Egress VNI  Evpn/  Num.
Mcast         Oper State        L2 PBR      Static  MACs
-----
1             192.0.2.3         1           evpn    0
BUM           Up                No
1             192.0.2.4         1           evpn    0
BUM           Up                No
-----
Number of Egress VTEP, VNI : 2
=====

BGP EVPN-VXLAN Ethernet Segment Dest
=====
Instance  Eth SegId                Num. Macs    Last Change
-----
1          01:02:00:00:00:00:00:00  1            04/01/2019 08:54:54
-----
Number of entries: 1
=====

A:PE-1# show service id 1 vxlan esi 01:02:00:00:00:00:00:00
=====
BGP EVPN-VXLAN Ethernet Segment Dest
=====
Instance  Eth SegId                Num. Macs    Last Change
-----
1          01:02:00:00:00:00:00:00  1            04/01/2019 08:54:54
-----
Number of entries: 1
=====

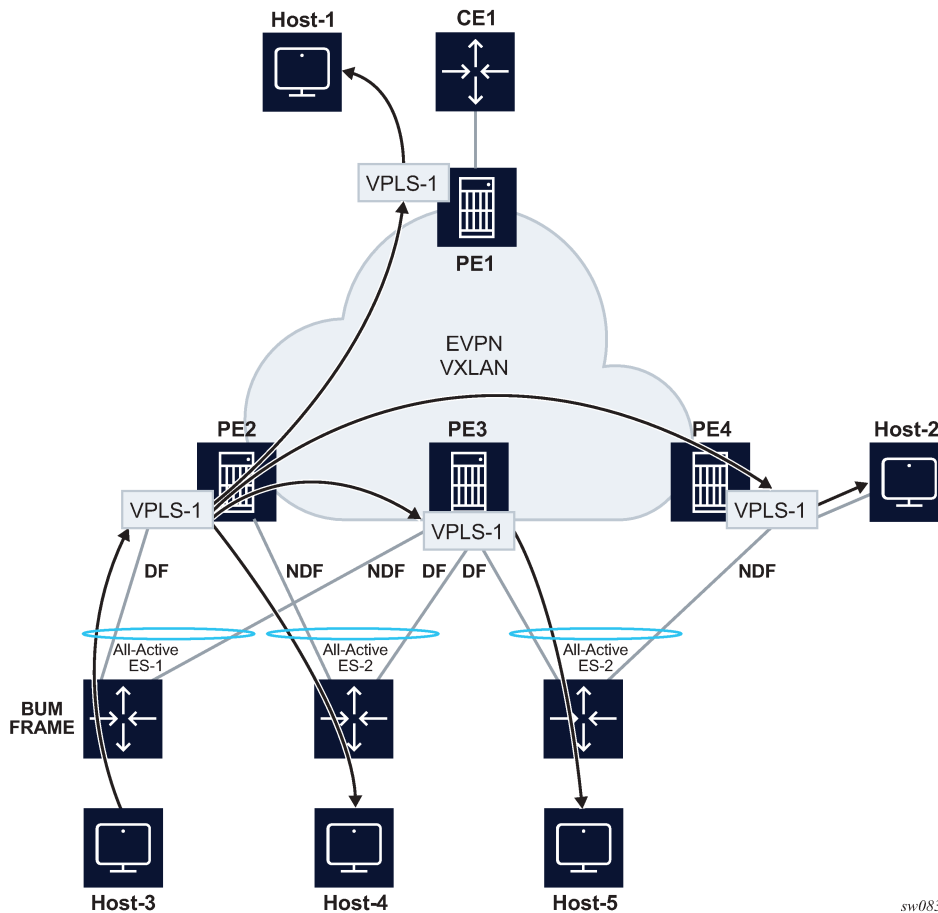
BGP EVPN-VXLAN Dest TEP Info
=====
Instance  TEP Address              Egr VNI      Last Change
-----
1          192.0.2.3                1            04/01/2019 08:54:54
1          192.0.2.4                1            04/01/2019 08:54:54
-----
Number of entries : 2
=====

```

### 5.2.4.1 Local bias for EVPN VXLAN multihoming

EVPN MPLS, as described in [EVPN for MPLS tunnels](#), uses ESI-labels to identify the BUM traffic sourced from a specified ES. The egress PE performs a label lookup to find the ESI label below the EVI label and to determine if a frame can be forwarded to a local ES. Because VXLAN does not support ESI-labels, or any MPLS label for that matter, the split-horizon filtering must be based on the tunnel source IP address. This also implies that the SAP-to-SAP forwarding rules must be changed when the SAPs belong to local ESs, irrespective of the DF state. This new forwarding is what RFC 8365 refers to as local bias. [Figure 77: EVPN-VXLAN multihoming with local bias](#) illustrates the local bias forwarding behavior.

Figure 77: EVPN-VXLAN multihoming with local bias



Local bias is based on the following principles:

- Every PE knows the IP addresses associated with the other PEs with which it has shared multihomed ESs.
- When the PE receives a BUM frame from a VXLAN bind, it looks up the source IP address in the tunnel header and filters out the frame on all local interfaces connected to ESs that are shared with the ingress PE.

With this approach, the ingress PE must perform replication locally to all directly-attached ESs (regardless of the DF Election state) for all flooded traffic coming from the access interfaces. BUM frames received on any SAP are flooded to:

- local non-ES SAPs and non-ES SDP-binds
- local all-active ES SAPs (DF and NDF)
- local single-active ES SDP-binds and SAPs (DF only)
- EVPN-VXLAN destinations

As an example, in [Figure 77: EVPN-VXLAN multihoming with local bias](#), PE2 receives BUM traffic from Host-3 and it forwards it to the remote PEs and the local ES SAP, even though the SAP is in NDF state.

The following rules apply to egress PE forwarding for EVPN-VXLAN services:

- The source VTEP is looked up for BUM frames received on EVPN-VXLAN.
- If the source VTEP matches one of the PEs with which the local PE shares both an ES and a VXLAN service:
  - the local PE is not forwarded to the shared ES local SAPs
  - the local PE forwards normally to ES SAPs unless they are in NDF state
- Because there is no multicast label or multicast B-MAC in VXLAN, the egress PE only identifies BUM traffic using the customer MAC DA; as a result, BM or unknown MAC DAs identify BUM traffic.

For example, in [Figure 77: EVPN-VXLAN multihoming with local bias](#), PE3 receives BUM traffic on VXLAN. PE3 identifies the source VTEP as a PE with which two ESs are shared, therefore it does not forward the BUM frames to the two shared ESs. It forwards to the non-shared ES (Host-5) because it is in DF state. PE4 receives BUM traffic and forwards it based on normal rules because it does not share any ESs with PE2.

The following command can be used to check whether the local PE has enabled the local bias procedures for a specific ES:

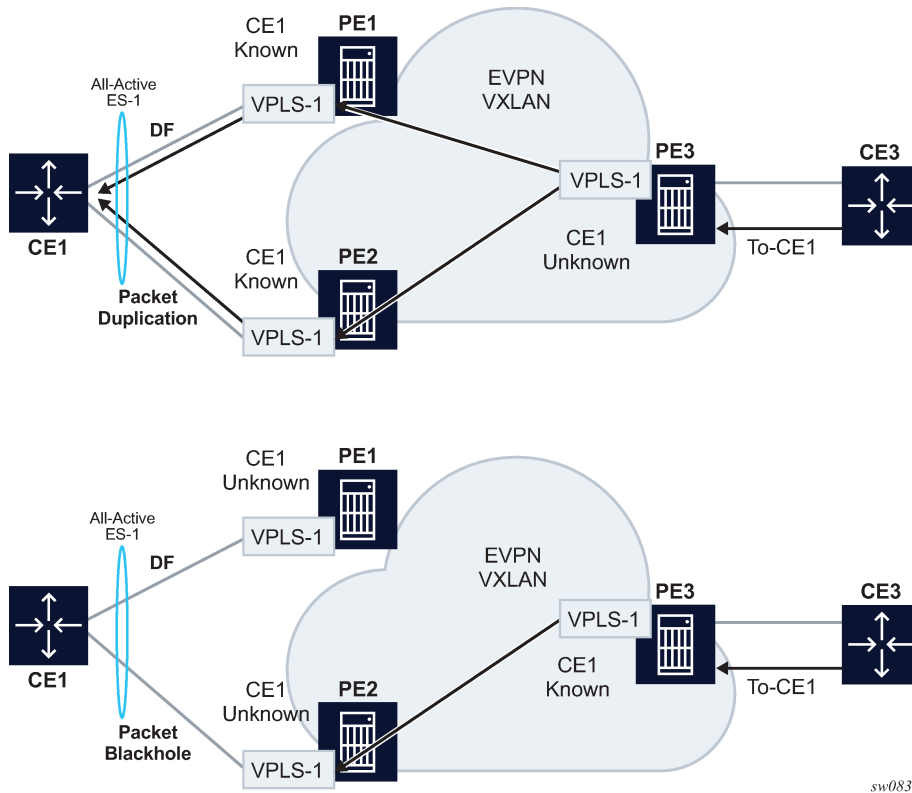
```
A:PE-2# tools dump service system bgp-evpn ethernet-segment "ES-1" local-bias
-----
[04/01/2019 08:45:08] Vxlan Local Bias Information
-----+-----
Peer                                         | Enabled
-----+-----
192.0.2.3                                   | Yes
-----+-----
```

5.2.4.2 Known limitations for local bias

In EVPN MPLS networks, an ingress PE that uses ingress replication to flood unknown unicast traffic pushes a BUM MPLS label that is different from a unicast label. The egress PEs use this BUM label to identify such BUM traffic to apply DF filtering for All-Active multihomed sites. In PBB-EVPN, in addition to the multicast label, the egress PE can also rely on the multicast B-MAC DA to identify customer BUM traffic.

In VXLAN there are no BUM labels or any tunnel indication that can assist the egress PE in identifying the BUM traffic. As such, the egress PE must solely rely on the C-MAC destination address, which may create some transient issues that are depicted in [Figure 78: EVPN-VXLAN multihoming and unknown unicast issues](#).

Figure 78: EVPN-VXLAN multihoming and unknown unicast issues



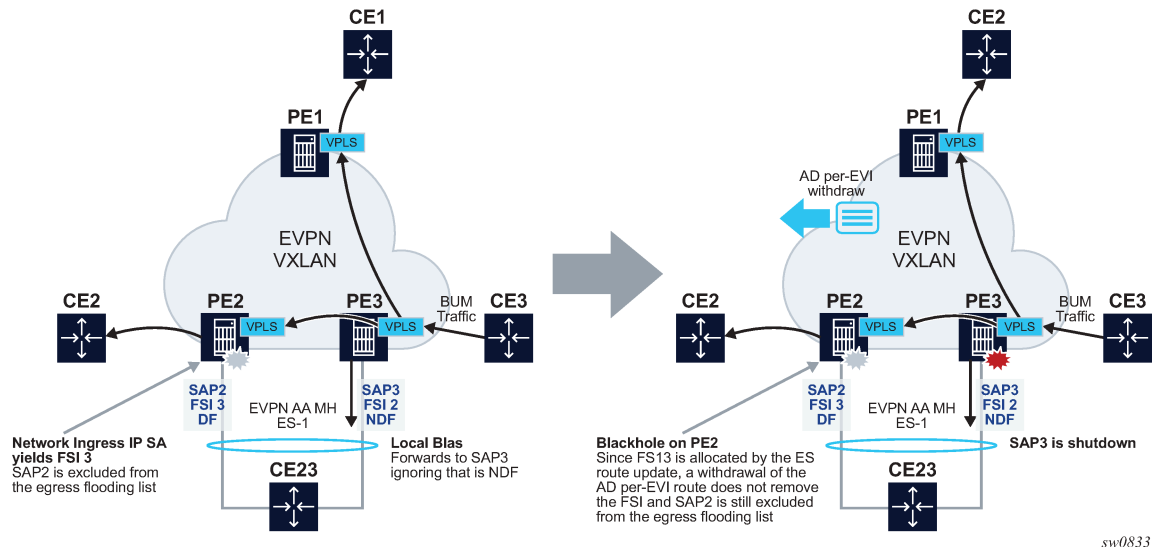
As shown in [Figure 78: EVPN-VXLAN multihoming and unknown unicast issues](#), top diagram, in absence of the mentioned unknown unicast traffic indication there can be transient duplicate traffic to All-Active multihomed sites under the following condition: CE1's MAC address is learned by the egress PEs (PE1 and PE2) and advertised to the ingress PE3; however, the MAC advertisement has not been received or processed by the ingress PE, resulting in the host MAC address to be unknown on the ingress PE3 but known on the egress PEs. Therefore, when a packet destined for CE1 address arrives on PE3, it floods it through ingress replication to PE1 or PE2 and, because CE1's MAC is known to PE1 and PE2, multiple copies are sent to CE1.

Another issue is shown at the bottom of [Figure 78: EVPN-VXLAN multihoming and unknown unicast issues](#). In this case, CE1's MAC address is known on the ingress PE3 but unknown on PE1 and PE2. If PE3's aliasing hashing picks up the path to the ES' NDF, a black-hole occurs.

The above two issues are solved in MPLS, as unicast known and unknown frames are identified with different labels.

Finally, another issue is described in [Figure 79: Blackhole created by a remote SAP shutdown](#). Under normal circumstances, when CE3 sends BUM traffic to PE3, the traffic is "local-biased" to PE3's SAP3 even though it is NDF for the ES. The flooded traffic to PE2 is forwarded to CE2, but not to SAP2 because the local bias split-horizon filtering takes place.

Figure 79: Blackhole created by a remote SAP shutdown



The right side of the diagram in [Figure 79: Blackhole created by a remote SAP shutdown](#) shows an issue when SAP3 is manually shutdown. In this case, PE3 withdraws the AD per-EVI route corresponding to SAP3; however, this does not change the local bias filtering for SAP2 in PE2. Therefore, when CE3 sends BUM traffic, it can neither be forwarded to CE23 via local SAP3 nor can it be forwarded by PE2.

### 5.2.4.3 Non-system IPv4 and IPv6 VXLAN termination for EVPN VXLAN multihoming

EVPN VXLAN multihoming is supported on VPLS and R-VPLS services when the PEs use non-system IPv4 or IPv6 termination, however, as with EVPN VPWS services, additional configuration steps are required.

- The **configure service system bgp-evpn eth-seg es-orig-ip ip-address** command must be configured with the non-system IPv4 or IPv6 address used for the EVPN-VXLAN service. This command modifies the originating-ip field in the ES routes advertised for the Ethernet Segment, and makes the system use this IP address when adding the local PE as DF candidate.
- The **configure service system bgp-evpn eth-seg route-next-hop ip-address** command must also be configured with the non-system IP address. This command changes the next-hop of the ES and AD per-ES routes to the configured address.
- Finally, the non-system IP address (in each of the PEs in the ES) must match in these three commands for the local PE to be considered suitable for DF election:
  - **es-orig-ip ip-address**
  - **route-next-hop ip-address**
  - **vlan-src-vtep ip-address**

## 5.3 EVPN for MPLS tunnels

This section provides information about EVPN for MPLS tunnels.

### 5.3.1 BGP-EVPN control plane for MPLS tunnels

[Table 14: EVPN routes and usage](#) lists all the EVPN routes and their usage in EVPN-VXLAN, EVPN-MPLS, and PBB-EVPN.



**Note:** Route type 1 is not required in PBB-EVPN as per RFC 7623.

*Table 14: EVPN routes and usage*

| EVPN route  | Usage  | EVPN-VXLAN | EVPN-MPLS | PBB-EVPN |
|---|--|------------|-----------|----------|
| Type 1 - Ethernet Auto-Discovery route (A-D)                                  | Mass-withdraw, ESI labels, Aliasing                          | Y          | Y         | —        |
| Type 2 - MAC/IP Advertisement route   | MAC/IP advertisement, IP advertisement for ARP resolution    | Y          | Y         | Y        |
| Type 3 - Inclusive Multicast Ethernet Tag route                               | Flooding tree setup (BUM flooding)                           | Y          | Y         | Y        |
| Type 4 - ES route   | ES discovery and DF election                                 | Y          | Y         | Y        |
| Type 5 - IP Prefix advertisement route  | IP Routing   | Y          | Y         | —        |
| Type 6 - Selective Multicast Ethernet Tag route                               | Signal interest on a multicast group                         | Y          | Y         | —        |
| Type 7 - Multicast Join Synch route   | Join a multicast group on a multihomed ES                    | Y          | Y         | —        |
| Type 8 - Multicast Leave Synch route  | Leave a multicast group on a multihomed ES                   | Y          | Y         | —        |
| Type 10 - Selective Provider Multicast Service Interface Auto-Discovery route | Signal and setup Selective Provider Tunnels for IP Multicast | -          | Y         | -        |

RFC 7432 describes the BGP-EVPN control plane for MPLS tunnels. If EVPN multihoming is not required, two route types are needed to set up a basic EVI (EVPN Instance): MAC/IP Advertisement and the Inclusive Multicast Ethernet Tag routes. If multihoming is required, the ES and the Auto-Discovery routes are also needed.

The route fields and extended communities for route types 2 and 3 are shown in [Figure 68: EVPN-VXLAN required routes and communities. BGP-EVPN control plane for VXLAN overlay tunnels](#). The changes compared to their use in EVPN-VXLAN are described below.



### EVPN route type 3 - inclusive multicast Ethernet tag route

As in EVPN-VXLAN, route type 3 is used for setting up the flooding tree (BUM flooding) for a specified VPLS service. The received inclusive multicast routes add entries to the VPLS flood list in the 7705 SAR Gen 2. Ingress replication, p2mp mLDP, and composite tunnels are supported as tunnel types in route type 3 when BGP-EVPN MPLS is enabled

The following route values are used for EVPN-MPLS services:

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context. The RD can be configured or derived from the **bgp-evpn evi** value.
- Ethernet Tag ID is 0.
- IP address length is always 32.
- Originating router's IP address carries an IPv4 or IPv6 address.
- The PMSI attribute can have different formats depending on the tunnel type enabled in the service.

#### – Tunnel type = Ingress replication (6)

The route is referred to as an Inclusive Multicast Ethernet Tag IR (IMET-IR) route and the PMSI Tunnel Attribute (PTA) fields are populated as follows:

- Leaf not required for Flags.
- MPLS label carries the MPLS label allocated for the service in the high-order 20 bits of the label field.

Unless **bgp-evpn mpls ingress-replication-bum-label** is configured in the service, the MPLS label used is the same as that used in the MAC/IP routes for the service.

- Tunnel endpoint is equal to the originating IP address.

#### – Tunnel type=p2mp mLDP (2)

The route is referred to as an IMET-P2MP route and its PTA fields are populated as follows:

- Leaf not required for Flags.
- MPLS label is 0.
- Tunnel endpoint includes the route node address and an opaque number. This is the tunnel identifier that the leaf-nodes use to join the mLDP P2MP tree.

#### – Tunnel type=Composite tunnel (130)

The route is referred to as an IMET-P2MP-IR route and its PTA fields are populated as follows:

- Leaf not required for Flags.
- MPLS label 1 is 0.
- Tunnel endpoint identifier includes the following:

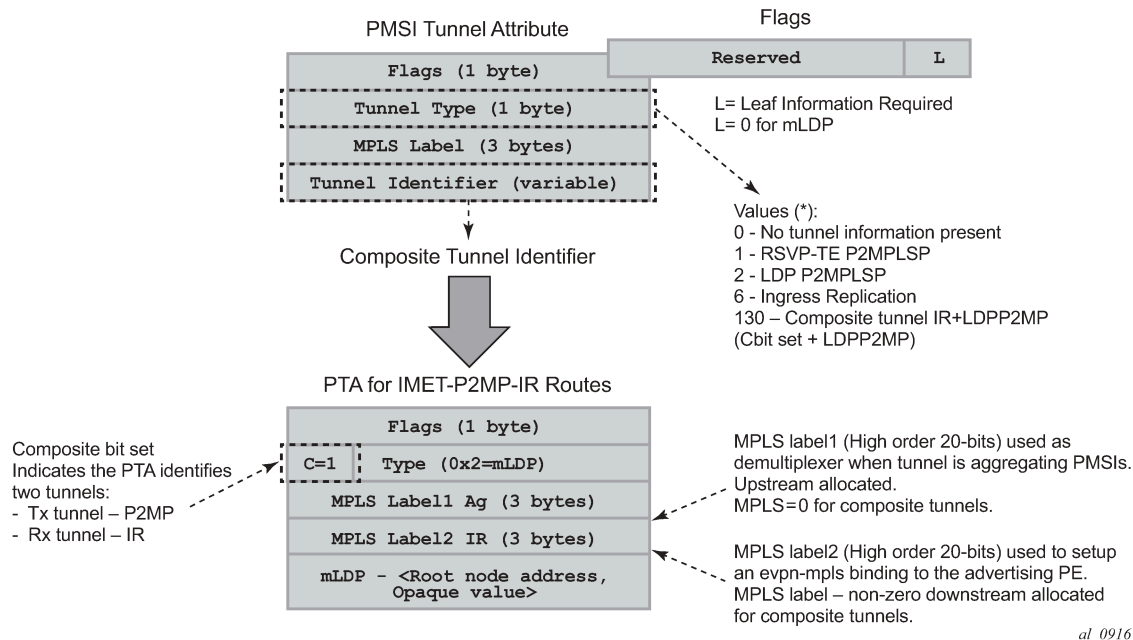
|                               |   |
|-------------------------------|---|
| <b>MPLS label2</b>            | non-zero, downstream allocated label (like any other IR label). The leaf-nodes use the label to set up an EVPN-MPLS destination to the root and add it to the default-multicast list. |
| <b>mLDP tunnel identifier</b> | the route node address and an opaque number. This is the tunnel identifier that the leaf-nodes use to join the mLDP P2MP tree.  |

IMET-P2MP-IR routes are used in EVIs with a few root nodes and a significant number of leaf-only PEs. In this scenario, a combination of P2MP and IR tunnels can be used in the network, such that the root nodes use P2MP tunnels to send broadcast, Unknown unicast, and Multicast traffic but the leaf-PE nodes use

IR to send traffic to the roots. This use case is documented in IETF RFC 8317 and the main advantage it offers is the significant savings in P2MP tunnels that the PE/P routers in the EVI need to handle (as opposed to a full mesh of P2MP tunnels among all the PEs in an EVI).

In this case, the root PEs signals a special tunnel type in the PTA, indicating that they intend to transmit BUM traffic using an mLDP P2MP tunnel but they can also receive traffic over an IR evpn-mpls binding. An IMET route with this special "composite" tunnel type in the PTA is called an IMET-P2MP-IR route and the encoding of its PTA is shown in [Figure 80: Composite p2mp mLDP and IR tunnels—PTA](#).

Figure 80: Composite p2mp mLDP and IR tunnels—PTA



## EVPN route type 2 - MAC/IP advertisement route

The 7705 SAR Gen 2 router generates this route type for advertising MAC addresses (and IP addresses if proxy-ARP/proxy-ND is enabled). If mac-advertisement is enabled, the router generates MAC advertisement routes for the following:

- learned MACs on SAPs or SDP bindings
- conditional static MACs



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

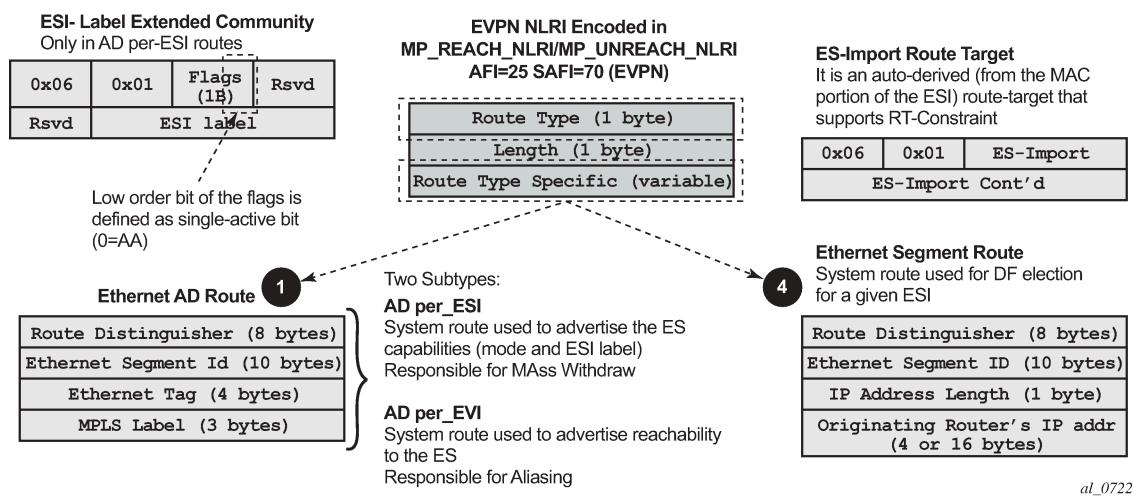
The route type 2 generated by a router uses the following fields and values:

- Route Distinguisher is taken from the RD of the VPLS service within the BGP context. The RD can be configured or derived from the **bgp-evpn evi** value.
- Ethernet Segment Identifier (ESI) is zero for MACs learned from single-homed CEs and different from zero for MACs learned from multihomed CEs.
- Ethernet Tag ID is 0.
- MAC address length is always 48.

- MAC address can be learned or statically configured.
- IP address and IP address length:
  - It is the IP address associated with the MAC being advertised with a length of 32 (or 128 for IPv6).
  - In general, any MAC route without IP has IPL=0 (IP length) and the IP is omitted.
  - When received, any IPL value not equal to zero, 32, or 128 discards the route.
  - MPLS Label 1 carries the MPLS label allocated by the system to the VPLS service. The label value is encoded in the high-order 20 bits of the field and is the same label used in the routes type 3 for the same service unless **bgp-evpn mpls ingress-replication-bum-label** is configured in the service.
- MPLS Label 2 is 0.
- The MAC mobility extended community is used for signaling the sequence number in case of MAC moves and the sticky bit in case of advertising conditional static MACs. If a MAC route is received with a MAC mobility **ext-community**, the sequence number and the 'sticky' bit are considered for the route selection.

When EVPN multihoming is enabled in the system, two more routes are required. [Figure 81: EVPN routes type 1 and 4](#) shows the fields in routes type 1 and 4 and their associated extended communities.

Figure 81: EVPN routes type 1 and 4



### EVPN route type 1 - Ethernet auto-discovery route (AD route)

The 7705 SAR Gen 2 router generates this route type for advertising for multihoming functions. The system can generate two types of AD routes:

- Ethernet AD route per-ESI (Ethernet Segment ID)
- Ethernet AD route per-EVI (EVPN Instance)

The Ethernet AD per-ESI route generated by a router uses the following fields and values:

- Route Distinguisher is taken from the system level RD or service level RD.
- Ethernet Segment Identifier (ESI) contains a 10-byte identifier as configured in the system for a specified **ethernet-segment**.
- Ethernet Tag ID is MAX-ET (0xFFFFFFFF). This value is reserved and used only for AD routes per ESI.

- MPLS label is 0.
- ESI Label Extended community includes the single-active bit (0 for all-active and 1 for single-active) and ESI label for all-active multihoming split-horizon.
- Route target extended community is taken from the service level RT or an RT-set for the services defined on the Ethernet segment.

The system can either send a separate Ethernet AD per-ESI route per service, or a few Ethernet AD per-ESI routes aggregating the route-targets for multiple services. While both alternatives inter-operate, RFC 7432 states that the EVPN Auto-Discovery per-ES route must be sent with a set of route-targets corresponding to all the EVIs defined on the Ethernet Segment (ES). Either option can be enabled using the command: **config>service>system>bgp-evpn#ad-per-es-route-target <[evi-rt ] | [evi-rt-set]> route-distinguisher ip-address [extended-evi-range]**

The default option **ad-per-es-route-target evi-rt** configures the system to send a separate AD per-ES route per service. When enabled, the **evi-rt-set** option supports route aggregation: a single AD per-ES route with the associated RD (**ip-address:1**) and a set of EVI route targets are advertised (up to a maximum of 128). When the number of EVIs defined in the Ethernet Segment is significant (therefore the number of route-targets), the system sends more than one route. For example:

- AD per-ES route for **evi-rt-set 1** is sent with RD **ip-address:1**
- AD per-ES route for **evi-rt-set 2** is sent with RD **ip-address:2**
- up to an AD per-ES route is sent with RD **ip-address:512**

The **extended-evi-range** option is needed for the use of **evi-rt-set** with a **comm-val** extended range of 1 through 65535. This option is recommended when EVIs greater than 65535 are configured in some services. In this case, there are more EVIs for which the route-targets must be packed in the AD per-ES routes. This command option extends the maximum number of AD per-ES routes that can be sent (since the RD now supports up to ip-address:65535) and allows many more route-targets to be included in each set.



**Note:** When **evi-rt-set** is configured, no vsi-export policies are possible on the services defined on the Ethernet Segment. If vsi-export policies are configured for a service, the system sends an individual AD per-ES route for that service. The maximum standard BGP update size is 4KB, with a maximum of 2KB for the route-target extended community attribute.

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

- Route Distinguisher is taken from the service level RD.
- Ethernet Segment Identifier (ESI) contains a 10-byte identifier as configured in the system for a specified Ethernet Segment.
- Ethernet Tag ID is 0.
- MPLS label encodes the unicast label allocated for the service (high-order 20 bits).
- Route-target extended community is taken from the service level RT.



**Note:** The AD per-EVI route is not sent with the ESI label Extended Community.

## EVPN route type 4 - ES route

The router generates this route type for multihoming ES discovery and DF (Designated Forwarder) election.

- Route Distinguisher is taken from the service level RD.

- Ethernet Segment Identifier (ESI) contains a 10-byte identifier as configured in the system for a specified **ethernet-segment**.
- The value of ES-import route-target community is automatically derived from the MAC address portion of the ESI. This extended community is treated as a route-target and is supported by RT-constraint (route-target BGP family).

### EVPN route type 5 - IP prefix route

IP Prefix Routes are also supported for MPLS tunnels. The route fields for route type 5 are shown in [Figure 70: EVPN route-type 5](#). The 7705 SAR Gen 2 router generates this route type for advertising IP prefixes in EVPN using the same fields that are described in section [BGP-EVPN control plane for VXLAN overlay tunnels](#), with the following exceptions:

- MPLS label carries the MPLS label allocated for the service.
- This route is sent with the RFC 5512 tunnel encapsulation extended community with the tunnel type value set to MPLS

### RFC 5512 - BGP tunnel encapsulation extended community

The following routes are sent with the RFC 5512 BGP Encapsulation Extended Community: MAC/IP, Inclusive Multicast Ethernet Tag, and AD per-EVI routes. ES and AD per-ESI routes are not sent with this Extended Community.

The router processes the following BGP Tunnel Encapsulation tunnel values registered by IANA for RFC 5512:

- VXLAN encapsulation is 8.
- MPLS encapsulation is 10.

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

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

If the encapsulation extended community (as defined in RFC 5512) is not present in a received route, BGP treats the route as an MPLS or VXLAN-based configuration of the **config>router>bgp>neighbor# def-recv-evpn-encap [mpls | vxlan]** command. The command is also available at the **bgp** and **group** levels.

## 5.3.2 EVPN for MPLS tunnels in VPLS services

EVPN can be used in MPLS networks where PEs are interconnected through any type of tunnel, including RSVP-TE, Segment-Routing TE, LDP, BGP, Segment Routing IS-IS, Segment Routing OSPF, RIB-API, MPLS-forwarding-policy, SR-Policy, or MPLSoUDP. As with VPRN services, tunnel selection for a VPLS service (with BGP-EVPN MPLS enabled) is based on the **auto-bind-tunnel** command. The BGP EVPN routes next-hops can be IPv4 or IPv6 addresses and can be resolved to a tunnel in the IPv4 tunnel-table or IPv6 tunnel-table.

EVPN-MPLS is modeled similarly to EVPN-VXLAN and uses a VPLS service where EVPN-MPLS "bindings" can coexist with SAPs and SDP-bindings.

### Example: VPLS service with EVPN-MPLS

```
*A:node-2config>service>vpls# info
```

```

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

```

First configure a **bgp-evpn** context where VXLAN must be disabled and MPLS enabled. In addition to enabling MPLS the command, the minimum set of commands to be configured to set up the EVPN-MPLS instance are the **evi** and the **auto-bind-tunnel resolution** commands. The relevant configuration options are the following.

**evi {1..16777215}** — This EVPN identifier is unique in the system and is used for the service-carving algorithm used for multihoming (if configured), and for auto-deriving the route target and route distinguishers (if lower than 65535) in the service. It can be used for EVPN-MPLS and EVPN-VXLAN services.

The following options are supported:

- If this EVPN identifier is not specified, the value is zero and no route distinguisher or route target is automatically derived from it.
- If the specified EVPN identifier is lower than 65535 and no other route distinguisher or route target is configured in the service, the following applies:
  - The route distinguisher is derived from <system\_ip>:evi.
  - The route target is derived from <autonomous-system>:evi.
- If the specified EVPN identifier is higher than 65535 and no other route distinguisher or route target is configured in the service, the following applies:
  - The route distinguisher cannot be automatically derived. An error is generated if enabling EVPN is attempted without a route distinguisher. A manual or an **auto-rd** route distinguisher must be configured.
  - The route target can only be automatically derived if the **evi-three-byte-auto-rt** command is configured. If configured, the route target is automatically derived in accordance with the following rules described in RFC8365.
    - The route target is composed of ASN(2-octets):A/type/D-ID/EVI.
    - The ASN is a 2-octet value configured in the system. For AS numbers exceeding the 2-byte limit, the low order 16-bit value is used.
    - The A=0 value is used for auto-derivation.
    - The type=4 (EVI-based) is used.
    - The BGP instance is encoded using D-ID= [1..2]. This allows the automatic derivation of different RTs in multi-instance services. The value is inherited from the corresponding BGP instance.
    - EVI indicates the configured EVI in the service

## Example

Consider a service with the following characteristics:

- ASN=64500

- VPLS with BGP instance **bgp 1** for EVPN-MPLS
- EVI=100000

The automatically derived route targets for this service are:

- bgp 1 — 64500:1090619040 (ASN:0x410186A0)
- bgp 2 — 64500:1107396256 (ASN:0x420186A0)

If this EVPN identifier is not specified, the value is zero and no route distinguisher or route targets is automatically derived from it. If specified and no other route distinguisher/route target are configured in the service:, then the following applies:

- the route distinguisher is derived from: **<system\_ip>:evi**
- the route target is derived from: **<autonomous-system>:evi**



**Note:** When the vsi-import/export policies are configured, the route target must be configured in the policies and those values take preference over the automatically derived route targets. The operational route target for a service is displayed by the **show service id svc-id bgp** command. If the **bgp-ad vpls-id** is configured in the service, the **vpls-id** derived route target takes precedence over the evi-derived route target.

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

The following options are specific to EVPN-MPLS and are configured in the **configure service vpls bgp-evpn mpls** context:

- **control word**

Enable or disable control word capability to guarantee interoperability to other vendors. When enabled along with the following command, the control word capability is signaled in the C flag of the EVPN Layer 2 attributes extended community, as defined in *draft-ietf-bess-rfc7432bis*;

- **MD-CLI**

```
configure service vpls bgp-evpn routes incl-mcast advertise-l2-attributes
```

- **classic CLI**

```
configure service vpls bgp-evpn incl-mcast-l2-attributes-advertisement
```

On reception, the router compares the C flag with the local setting for **control-word**. In case of a mismatch, the EVPN destination goes operationally down with the corresponding operational flag indicating the reason.



**Note:** The **control-word** is required, as described in RFC 7432, to avoid frame disordering.

- **auto bind tunnel**

This command selects the type of MPLS transport tunnel to use for a specific instance; this command is used in the same way as in VPRN services.

For BGP-EVPN MPLS, you must explicitly add BGP to the resolution filter in EVPN (BGP is implicit in VPRNs).



- **force VLAN VC forwarding**

This option allows the system to preserve the VLAN ID and P-bits of the service-delimiting qtag in a new tag added in the customer frame before sending it to the EVPN core.



**Note:** You can use this option in conjunction with the **sap ingress vlan-translation** command. If so, the configured translated VLAN ID is sent to the EVPN binds as opposed to the service-delimiting tag VLAN ID. If the ingress SAP/binding is null-encapsulated, the output VLAN ID and pbits are zero.

- **force QinQ VC forwarding with c-tag-c-tag or s-tag-c-tag**

This command allows the system to preserve the VLAN ID and pbits of the service-delimiting Q-tags (up to two tags) in customer frames before sending them to the EVPN core.



**Note:** You can use this option in conjunction with the **sap ingress qinq-vlan-translation s-tag.c-tag** command. If so, the configured translated S-tag and C-tag VLAN IDs are the VLAN IDs sent to the EVPN binds as opposed to the service-delimiting tags VLAN IDs. If the ingress SAP or binding is null-encapsulated, the output VLAN ID and pbits are zero.

- **split horizon group**

This command allows the association of a user-created split horizon group to all the EVPN-MPLS destinations. See [EVPN and VPLS integration](#) for more information.

- **ecmp**

Set this option to a value greater than 1 to activate aliasing to the remote PEs that are defined in the same all-active multihoming ES.

- **ingress replication bum label**

You can use this option when you want the PE to advertise a label for BUM traffic (Inclusive Multicast routes) that is different from the label advertised for unicast traffic (with the MAC/IP routes). This is useful to avoid potential transient packet duplication in all-active multihoming.

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

- **mac-advertisement**

- **mac-duplication** and settings

- **incl-mcast advertise-l2-attributes** (MD-CLI)

**incl-mcast-l2-attributes-advertisement** (classic CLI)

This function enables the advertisement and processing of the EVPN Layer 2 attributes extended community. The control word, hash-label configuration, and the service-MTU value are advertised in the extended community. On reception, the received MTU, hash-label, and control-word flags are compared with the local MTU and hash-label or control-word configuration. In case of a mismatch in any of the three settings, the EVPN destination goes operationally down, and the corresponding operational flag describes the mismatch. The absence of an IMET route from an egress PE or the absence of the EVPN Layer 2 attributes extended community on a received IMET route from the PE, causes the route to bring down the EVPN destinations to that PE.

- **ignore-mtu-mismatch**

This command makes the router ignore the received Layer 2 MTU in the EVPN Layer 2 attributes extended community of the IMET route for a peer. If disabled, the local service MTU is compared



against the received Layer 2 MTU. If there is a mismatch, the EVPN destinations to the peer stay oper-state down.

When EVPN-MPLS is established among some PEs in the network, EVPN unicast and multicast “bindings” to the remote EVPN destinations are created on each PE. A specified ingress PE creates the following:

- 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**.

These bindings, as well as the MACs learned on them, can be checked using the following commands.

### Output example: EVPN-MPLS destination bindings

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**. As a result, the device has a single EVPN-MPLS destination binding to PE(192.0.2.69) and two bindings (unicast and multicast) to PE(192.0.2.70).

```
show service id 1 evpn-mpls
```

```
=====
BGP EVPN-MPLS Dest
=====
```

| TEP Address | Egr Label<br>Transport | Num. MACs | Mcast | Last Change         |
|-------------|------------------------|-----------|-------|---------------------|
| 192.0.2.69  | 262118<br>ldp          | 1         | Yes   | 06/11/2015 19:59:03 |
| 192.0.2.70  | 262139<br>ldp          | 0         | Yes   | 06/11/2015 19:59:03 |
| 192.0.2.70  | 262140<br>ldp          | 1         | No    | 06/11/2015 19:59:03 |
| 192.0.2.72  | 262140<br>ldp          | 0         | Yes   | 06/11/2015 19:59:03 |
| 192.0.2.72  | 262141<br>ldp          | 1         | No    | 06/11/2015 19:59:03 |
| 192.0.2.73  | 262139<br>ldp          | 0         | Yes   | 06/11/2015 19:59:03 |
| 192.0.2.254 | 262142<br>bgp          | 0         | Yes   | 06/11/2015 19:59:03 |

```
-----
Number of entries : 7
-----
=====
```

```
show service id 1 fdb detail
```

```
=====
Forwarding Database, Service 1
=====
```

| ServId | MAC               | Source-Identifier           | Type<br>Age | Last Change       |
|--------|-------------------|-----------------------------|-------------|-------------------|
| 1      | 00:ca:fe:ca:fe:69 | eMpls:<br>192.0.2.69:262118 | EvpnS       | 06/11/15 21:53:48 |
| 1      | 00:ca:fe:ca:fe:70 | eMpls:                      | EvpnS       | 06/11/15 19:59:57 |

```

1          00:ca:fe:ca:fe:72 192.0.2.70:262140      EvpnS      06/11/15 19:59:57
          eMpls:
          192.0.2.72:262141
-----
No. of MAC Entries: 3
-----
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
=====

```

### 5.3.2.1 EVPN and VPLS integration

The 7705 SAR Gen 2 EVPN implementation supports RFC 8560 so that EVPN-MPLS and VPLS can be integrated into the same network and within the same service. Because EVPN is not deployed in green-field networks, this feature is useful for the integration between both technologies and even for the migration of VPLS services to EVPN-MPLS.

The following behavior enables the integration of EVPN and SDP-bindings in the same VPLS network.

#### Systems with EVPN endpoints and SDP-bindings to the same far-end bring down the SDP-bindings.

- The router allows the establishment of an EVPN endpoint and an SDP-binding to the same far-end, but the SDP-binding is kept operationally down. Only the EVPN endpoint is operationally up. This applies to spoke-SDPs (manual, BGP-AD, and BGP-VPLS) and mesh-SDPs. It is also possible between VXLAN and SDP bindings.
- If there is an existing EVPN endpoint to a specified far-end and a spoke-SDP establishment is attempted, the spoke-SDP is set up 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 or used EVPN route arrives, the EVPN endpoint is set up and the spoke-SDP is brought down with an operational flag indicating that there is an EVPN route to the same far-end.
- In the case of an SDP-binding and EVPN endpoint to different far-end IPs on the same remote PE, both links are up. This may occur if the SDP-binding is terminated in an IPv6 or IPv4 address that is different from the system address where the EVPN endpoint is terminated.

#### The user can add spoke-SDPs and all the EVPN-MPLS endpoints in the same SHG.

- A CLI command is added under the **bgp-evpn>mpls** context so that the EVPN-MPLS endpoints can be added to an SHG:

```
bgp-evpn mpls [no] split-horizon-group group-name
```

- The **bgp-evpn mpls split-horizon-group** command must reference a user-configured SHG. User-configured SHGs 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 **bgp-evpn mpls split-horizon-group** command is not used, the default SHG (that contains all the EVPN endpoints) is still used but cannot be referred to on SAPs/spoke-SDPs.
- SAPs and SDP-bindings that share the same SHG of the EVPN-MPLS provider-tunnel are brought operationally down if the point-to-multipoint tunnel is operationally up.

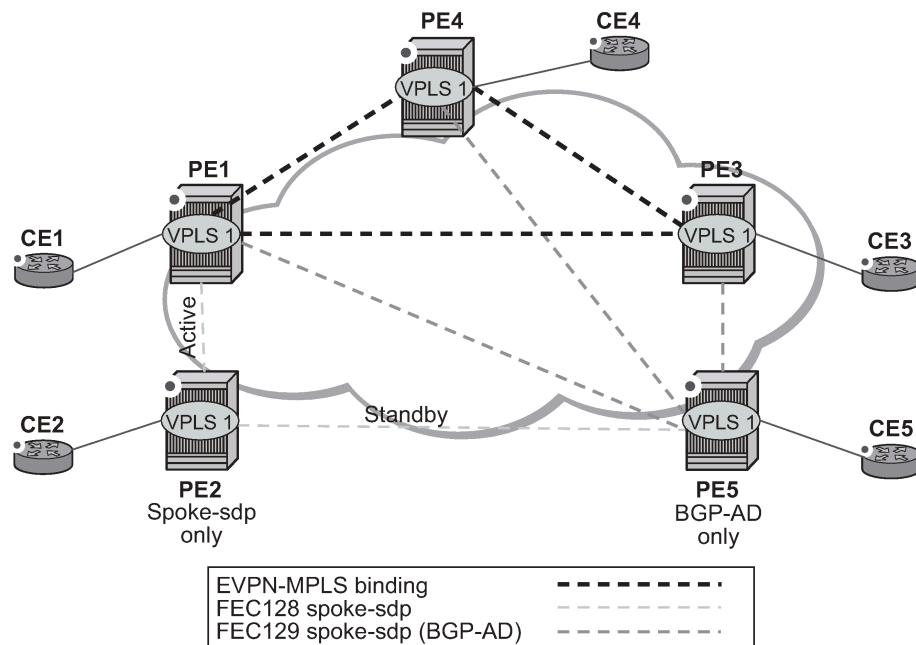
#### The system disables the advertisement of MACs learned on spoke-SDPs or SAPs that are part of an EVPN SHG.

- When the SAPs and spoke-SDPs (manual or BGP-AD/VPLS-discovered) are configured within the same SHG as the EVPN endpoints, MAC addresses are still learned on them but they are not advertised in EVPN.

- The preceding statement is also true if proxy-ARP/ND is enabled and an IP→MAC pair is learned on a SAP or SDP-binding that belongs to the EVPN SHG.
- The SAPs and spoke-SDPs added to an EVPN SHG should not be part of any EVPN multihomed ES. If that occurs, the PE would still advertise the AD per-EVI route for the SAP or spoke-SDP, attracting EVPN traffic that could not possibly be forwarded to that SAP or SDP-binding.
- Similar to the preceding statement, an SHG 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 forwarding of traffic from the EVPN to the BGP-MH site, regardless of the DF/NDF state.

The following figure shows an example of EVPN-VPLS integration.

Figure 82: EVPN-VPLS integration



al\_0723

The following is an example configuration output of PE1, PE5, and PE2.

```
*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 bgp 1
  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

```

The following applies to the configuration described in the preceding example.

- 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. The following applies to this configuration.
  - PE1, PE3, and PE4 attempt to establish BGP-AD spoke-SDPs, but they are kept operationally down as long as there are EVPN endpoints active among them.
  - BGP-AD spoke-SDPs and EVPN endpoints are instantiated within the same split horizon group, for example, SHG-1.
  - Manual spoke-SDPs from PE1 and PE5 to PE2 are not part of SHG-1.
- EVPN MAC advertisements
  - MACs learned on FEC128 spoke-SDPs are advertised normally in EVPN.
  - MACs learned on FEC129 spoke-SDPs are not advertised in EVPN (because they are part of SHG-1, which is the split horizon group used for **bgp-evpn mpls**). This prevents any data plane MACs learned on the SHG from being advertised in EVPN.
- BUM operation on PE1
  - When CE1 sends BUM, PE1 floods to all the active bindings.
  - When CE2 sends BUM, PE2 sends it to PE1 (active spoke-SDP) and PE1 floods to all the bindings and SAPs.
  - When CE5 sends BUM, PE5 floods to the three EVPN PEs. PE1 floods to the active spoke-SDP and SAPs, never to the EVPN PEs because they are part of the same SHG.

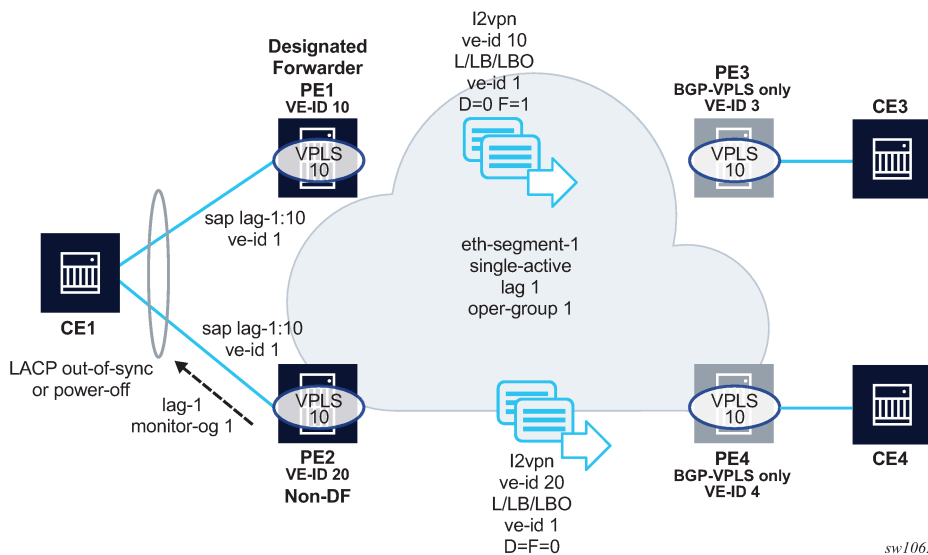
The operation in services with BGP-VPLS and BGP-EVPN is equivalent to the operation for BGP-AD and BGP EVPN described in [EVPN and VPLS integration](#).

### 5.3.2.2 EVPN single-active multihoming and BGP-VPLS integration

In a VPLS service to which multiple EVPN PE and BGP-VPLS PE are attached, single-active multihoming is supported on two or more of the EVPN PE with no special considerations. However, all-active multihoming is not supported because traffic from the all-active multihomed CE could cause a MAC flip-flopping effect on remote BGP-VPLS PEs, asymmetric flows, or other issues.

The following figure shows a scenario with a single-active ES used in a service where EVPN PEs and BGP-VPLS are integrated.

Figure 83: BGP-VPLS to EVPN integration and single-active MH



Although other single-active examples are supported, in the preceding figure, CE1 is connected to the EVPN PEs via a single LAG (lag-1). The LAG is associated with eth-segment-1 on PE1 and PE2, which is configured as single-active and with oper-group 1. PE1 and PE2 use **lag>monitor-oper-group 1** so that the non-DF PE can signal the non-DF state to CE1 (in the form of Link Aggregation Control Protocol (LACP) out-of-sync or power-off).

In addition to the BGP-VPLS routes sent for the service virtual-edge (VE) ID, the multihoming PEs, in this case, must generate additional BGP-VPLS routes per ES (per VPLS service) to generate a MAC flush on the remote BGP-VPLS PEs in case of failure.

The **sap>bgp-vpls-mh-veid number** command should be configured on the SAPs that are part of an EVPN single-active ES. This command allows the advertisement of L2VPN routes that indicate the state of the multihomed SAPs to the remote BGP-VPLS PEs. After a DF switchover, the F and D bits of the generated L2VPN routes for the SAP VE ID are updated so that the remote BGP-VPLS PEs can perform a MAC flush operation on the service and avoid blackholes.

For example, in case of a failure on the ES SAP on PE1, PE1 must indicate to PE3 and PE4 the need to flush MAC addresses learned from PE1 (flush-all-from-me message). Otherwise, PE3 continues to send traffic with MAC DA = CE1 to PE1, and PE1 blackholes the traffic.

The following applies to the example in [Figure 83: BGP-VPLS to EVPN integration and single-active MH](#).

- Both ES peers (PE1 and PE2) should be configured with the same VE ID for the ES SAP. However, this is not mandatory.

- In addition to the regular service ve-id L2VPN route, based on the **sap>bgp-vpls-mh-ve-id** configuration and on BGP VPLS being enabled, the PE advertises an L2VPN route with the following fields:
  - VE ID that contains the value configured using the **sap bgp-vpls-mh-ve-id** command
  - RD, RT, next hop, and other attributes that are the same as the service BGP VPLS route
  - L2VPN information extended community with the following flags:
    - **D=0**  
This value applies If the SAP is oper-up or oper-down with a flag MHStandby (for example, the PE is non-DF in single-active MH).  
This value also applies if there is an ES oper-group and the port is down as a result of the oper-group.
    - **D=1**  
This value applies if the SAP is oper-down with a different flag (for example, port-down or admin-down).
    - **F (DF bit) =1**  
This value applies if the SAP is oper-up. Otherwise, the value is F=0.
- After a failure on the access SAP, only MAC flush messages are triggered if the **bgp-vpls-mh-ve-id** command is configured in the failing SAP. If the SAP is configured with VE ID 1, the following applies.
  - If the non-DF PE has a failure on the access SAP, PE2 sends an update with VE ID=1/D=1/F=0. This is an indication for PE3 and PE4 that the PE2 SAP is oper-down but triggering a MAC flush on PE3 and PE4 is not required.
  - If the DF PE has a failure on the SAP, PE1 advertises VE ID=1/D=1/F=0. After receiving this update, PE3 and PE4 flush all their MACs associated with the PE1 spoke-SDP. The failure on PE1 triggers an EVPN DF election on PE2, which becomes DF and advertises VE ID=1/D=0/F=1. This message does not trigger any MAC flush procedures.

The following considerations apply for EVPN single-active multihoming and BGP-VPLS integration.

- PE3 and PE4 can be Nokia nodes running SR OS or any third-party PEs that support the procedures defined in *draft-ietf-bess-vpls-multihoming*, such that BGP-VPLS MAC flush signaling is understood.
- PE1 and PE2 are expected to run an SR OS version that supports the configuration of the **sap bgp-vpls-mh-veid number** command on multihomed SAPs. Otherwise, the MAC flush behavior does not work as expected.
- The procedures described in this section are also supported if the EVPN PEs use MC-LAG instead of an ES for CE1 redundancy. In this case, the SAP VE ID route for the standby PE is sent as VE ID=1/D=1/F=0, whereas the active chassis advertises VE ID=1/D=0/F=1. A switchover triggers a MAC flush on the remote PEs, as described in [EVPN single-active multihoming and BGP-VPLS integration](#).
- L2VPN routes generated for ESs or SAPs with the **sap bgp-vpls-mh-veid number** command are decoded in the remote nodes as BGP-MH routes (because they do not have label information) in the **show router bgp routes l2vpn** command and debug.

### 5.3.2.3 Auto-derived route-distinguisher in services with multiple BGP families

Multiple BGP families and protocols can be enabled at the same time in a VPLS. When **bgp-evpn** is enabled, **bgp-ad** and **bgp-mh** are also supported. A single RD is used per service and not per BGP family or protocol.

The following rules apply.

- The VPLS RD is selected based on the following precedence.
  - Manual RD or automatic RD always take precedence when configured.
  - If manual or automatic RD is not configured, the RD is derived from the **bgp-ad>vpls-id**.
  - If manual RD, automatic RD, or VPLS ID are not configured, the RD is derived from the **bgp-evpn>evi**, except for **bgp-mh**, which does not support EVI-derived RD, and except when the EVI is greater than 65535. In these two cases, no EVI-derived RD is possible.
  - If manual RD, automatic RD, VPLS ID, or EVI is not configured, there is no RD and the service fails.
- The selected RD (see the preceding selection criteria) is displayed by the Oper Route Dist field of the **show service id bgp** command.
- The service supports dynamic RD changes. For example, the CLI allows the dynamic update of the VPLS ID, even if it is used to automatically derive the service RD for **bgp-ad**, **bgp-vpls**, or **bgp-mh**.



**Note:** When the RD changes, the active routes for that VPLS are withdrawn and readvertised with the new RD.

- If one of the mechanisms to derive the RD for a specified service is removed from the configuration, the system selects a new RD based on the preceding rules. For example, if the VPLS ID is removed from the configuration, the routes are withdrawn, the new RD selected from the EVI, and the routes readvertised with the new RD.



**Note:** The reconfiguration fails if the new RD already exists in a different VPLS or Epipe service.

- Because the **vpls-id** takes precedence over the **evi** when the RD is auto-derived, the existing RD is not affected when **evpn** is added to an existing **bgp-ad** service. This is important to support **bgp-ad** to **evpn** migration.

### 5.3.3 P2MP mLDP tunnels for BUM traffic in EVPN-MPLS services

P2MP mLDP tunnels for BUM traffic in EVPN-MPLS services are supported and enabled through the use of the provider-tunnel context. If EVPN-MPLS takes ownership over the provider-tunnel, **bgp-ad** is still supported in the service but it does not generate BGP updates, including the PMSI Tunnel Attribute. The following CLI example shows an EVPN-MPLS service that uses P2MP mLDP LSPs for BUM traffic.

```
*A:PE-1>config>service>vpls(vpls or b-vpls)# info
-----
description "evpn-mpls-service with p2mp mLDP"
bgp-evpn
  evi 10
  no ingress-repl-inc-mcast-advertisement
  mpls bgp 1
```

```

    no shutdown
    auto-bind-tunnel resolution any
  exit
  provider-tunnel
    inclusive
    owner bgp-evpn-mpls
    root-and-leaf
    mldp
    no shutdown
  exit
  exit
  sap 1/1/1:1 create
  exit
  spoke-sdp 1:1 create
  exit

```

When **provider-tunnel inclusive** is used in EVPN-MPLS services, the following commands can be used in the same way as for BGP-AD or BGP-VPLS services:

- **data-delay-interval**
- **root-and-leaf**
- **mldp**
- **shutdown**

The following commands are used by **provider-tunnel** in BGP-EVPN MPLS services:

- **[no] ingress-repl-inc-mcast-advertisement**

This command allows you to control the advertisement of IMET-IR and IMET-P2MP-IR routes for the service. See [BGP-EVPN control plane for MPLS tunnels](#) for a description of the IMET routes. The following considerations apply:

- If configured as **no ingress-repl-inc-mcast-advertisement**, the system does not send the IMET-IR or IMET-P2MP-IR routes, regardless of the service being enabled for BGP-EVPN MPLS or BGP-EVPN VXLAN.
- If configured as **ingress-repl-inc-mcast-advertisement** and the PE is **root-and-leaf**, the system sends an IMET-P2MP-IR route.
- If configured as **ingress-repl-inc-mcast-advertisement** and the PE is **no root-and-leaf**, the system sends an IMET-IR route.
- Default value is **ingress-repl-inc-mcast-advertisement**.

- **[no] owner {bgp-ad | bgp-vpls | bgp-evpn-mpls}**

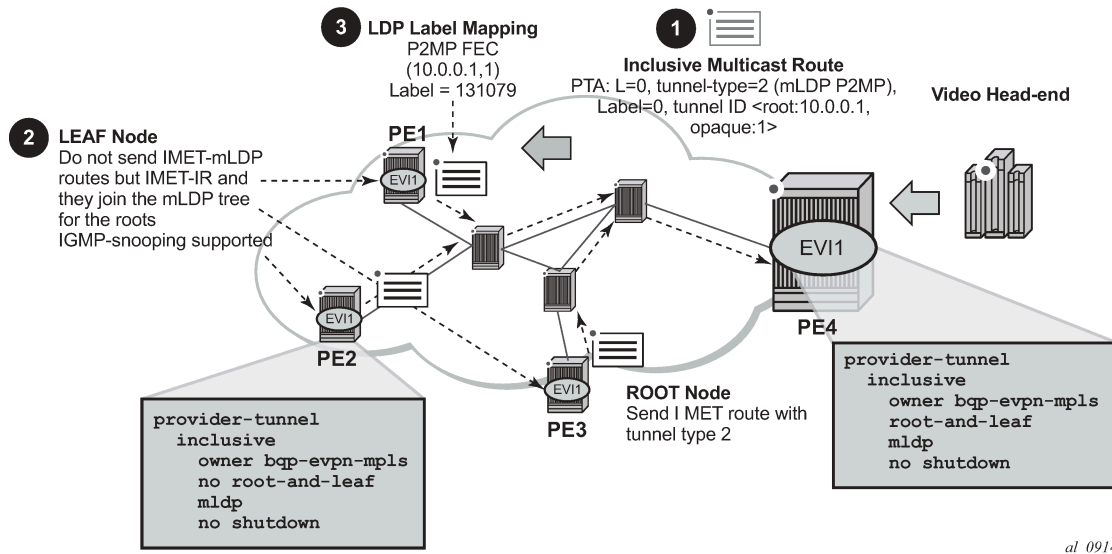
The owner of the provider tunnel must be configured. The default value is **no owner**. The following considerations apply:

- Only one of the protocols supports a provider tunnel in the service and it must be explicitly configured.
- **bgp-vpls** and **bgp-evpn** are mutually exclusive.
- While **bgp-ad** and **bgp-evpn** can coexist in the same service, only **bgp-evpn** can be the provider-tunnel owner in such cases.

[Figure 84: EVPN services with p2mp mLDp—control plane](#) shows the use of P2MP mLDp tunnels in an EVI with a root node and a few leaf-only nodes.



Figure 84: EVPN services with p2mp mLDP—control plane



Consider the use case of a root-and-leaf PE4 where the other nodes are configured as leaf-only nodes (**no root-and-leaf**). This scenario is handled as follows:

1. If **ingress-repl-inc-mcast-advertisement** is configured, then as soon as the **bgp-evpn mpls** option is enabled, the PE4 sends an IMET-P2MP route (tunnel type mLDP), or optionally, an IMET-P2MP-IR route (tunnel type composite). IMET-P2MP-IR routes allow leaf-only nodes to create EVPN-MPLS multicast destinations and send BUM traffic to the root.
2. If **ingress-repl-inc-mcast-advertisement** is configured, PE1/2/3 do not send IMET-P2MP routes; only IMET-IR routes are sent.
  - The **root-and-leaf** node imports the IMET-IR routes from the leaf nodes but it only sends BUM traffic to the P2MP tunnel as long as it is active.
  - If the P2MP tunnel goes operationally down, the **root-and-leaf** node starts sending BUM traffic to the evpn-mpls multicast destinations
3. When PE1/2/3 receive and import the IMET-P2MP or IMET-P2MP-IR from PE4, they join the mLDP P2MP tree signaled by PE4. They issue an LDP label-mapping message including the corresponding P2MP FEC.

As described in IETF Draft *draft-ietf-bess-evpn-etree*, mLDP and Ingress Replication (IR) can work in the same network for the same service; that is, EVI1 can have some nodes using mLDP (for example, PE1) and others using IR (for example, PE2). For scaling, this is significantly important in services that consist of a pair of root nodes sending BUM in P2MP tunnels and hundreds of leaf-nodes that only need to send BUM traffic to the roots. By using IMET-P2MP-IR routes from the roots, the operator makes sure the leaf-only nodes can send BUM traffic to the root nodes without the need to set up P2MP tunnels from the leaf nodes.

When both static and dynamic P2MP mLDP tunnels are used on the same router, Nokia recommends that the static tunnels use a tunnel ID lower than 8193. If a tunnel ID is statically configured with a value equal to or greater than 8193, BGP-EVPN may attempt to use the same tunnel ID for services with **enabled provider-tunnel**, and fail to set up an mLDP tunnel.

Inter-AS option C or seamless-MPLS models for non-segmented mLDP trees are supported with EVPN for BUM traffic. The leaf PE that joins an mLDP EVPN root PE supports Recursive and Basic Opaque FEC elements (types 7 and 1, respectively). Therefore, packet forwarding is handled as follows:

- The ABR or ASBR may leak the root IP address into the leaf PE IGP, which allows the leaf PE to issue a Basic opaque FEC to join the root.
- The ABR or ASBR may distribute the root IP using BGP label-ipv4, which results in the leaf PE issuing a Recursive opaque FEC to join the root.

For more information about mLDP opaque FECs, see the *7705 SAR Gen 2 Layer 3 Services Guide: IES and VPRN* and the *7705 SAR Gen 2 MPLS Guide*.

All-active multihoming and single-active with an ESI label multihoming are supported in EVPN-MPLS services together with P2MP mLDP tunnels. Both use an upstream-allocated ESI label, as described in RFC 7432 section 8.3.1.2, which is popped at the leaf PEs, resulting in the requirement that, in addition to the root PE, all EVPN-MPLS P2MP leaf PEs must support this capability (including the PEs not connected to the multihoming ES).

### 5.3.4 EVPN-VPWS for MPLS tunnels

This section provides information about EVPN-VPWS for MPLS tunnels.

#### 5.3.4.1 BGP-EVPN control plane for EVPN-VPWS

EVPN-VPWS for MPLS tunnels uses the RFC 8214 BGP extensions described in [EVPN-VPWS for VXLAN tunnels](#), with the following differences for the Ethernet AD per-EVI routes:

- The MPLS field encodes an MPLS label as opposed to a VXLAN VNI.
- The C flag is set if the control word is configured in the service.
- The F flag is set if the hash label is configured in the service.

#### 5.3.4.2 EVPN for MPLS tunnels in Epipe services (EVPN-VPWS)

The use and configuration of EVPN-VPWS services is described in [EVPN-VPWS for VXLAN tunnels](#) with the following differences when the EVPN-VPWS services use MPLS tunnels instead of VXLAN.

When MPLS tunnels are used, the **bgp-evpn>mpls** context must be configured in the Epipe. As an example, if Epipe 2 is an EVPN-VPWS service that uses MPLS tunnels between PE2 and PE4, this would be its configuration:

```
PE2>config>service>epipe(2)#
-----
bgp
exit
bgp-evpn
  evi 2
    local-attachment-circuit "AC-1"
    eth-tag 200
  exit
  remote-attachment-circuit "AC-2"
    eth-tag 200
  exit
  mpls bgp 1
```

```

    ecmp 2
    no shutdown
exit
sap 1/1/1:1 create

```

```

PE4>config>service>epipe(2)#
-----
bgp
exit
bgp-evpn
    evi 2
    local-attachment-circuit "AC-2"
    eth-tag 200
exit
    remote-attachment-circuit "AC-1"
    eth-tag 100
exit
    mpls bgp 1
        ecmp 2
        no shutdown
exit
    spoke-sdp 1:1

```

Where the following BGP-EVPN commands, specific to MPLS tunnels, are supported in the same way as in VPLS services:

- **mpls auto-bind-tunnel**
- **mpls control-word**
- **mpls entropy-label**
- **mpls force-vlan-vc-forwarding**
- **mpls shutdown**

EVPN-VPWS Epipes with MPLS tunnels can also be configured with the following characteristics:

- Access attachment circuits can be SAPs or spoke SDPs. Manually configured and BGP-VPWS spoke SDPs are supported. The VC switching configuration is not supported on BGP-EVPN-enabled pipes.
- EVPN-VPWS Epipes using null SAPs can be configured with **sap>ethernet>llf**. When enabled, upon removing the EVPN destination, the port is brought oper-down with flag LinkLossFwd, however the AD per EVI route for the SAP is still advertised (the SAP is kept oper-up). When the EVPN destination is created, the port is brought oper-up and the flag cleared.
- EVPN-VPWS Epipes for MPLS tunnels support **endpoints**. The parameter **endpoint endpoint name** is configurable along with **bgp-evpn>local-attachment-circuit** and **bgp-evpn>remote-attachment-circuit**. The following conditions apply to endpoints on EVPN-VPWS Epipes with MPLS tunnels:
  - Up to two explicit endpoints are allowed per Epipe service with BGP-EVPN configured.
  - A limited endpoint configuration is allowed in Epipes with BGP-EVPN. Specifically, neither active-hold-delay nor revert-time are configurable.
  - When **bgp-evpn>remote-attachment-circuit** is added to an explicit endpoint with a spoke SDP, the **spoke-sdp>precedence** command is not allowed. The spoke SDP always has a precedence of four, which is always higher than the EVPN precedence. Therefore, the EVPN-MPLS destination is used for transmission if it is created, and the spoke SDP is only used when the EVPN-MPLS destination is removed.
- EVPN-VPWS Epipes for MPLS tunnels support control word and ELs.

- When the control word is configured, the PE sets the C bit in its AD per-EVI advertisement and sends the control word in the datapath. In this case, the PE expects the control word to be received. If there is a mismatch between the received control word and the configured control word, the system does not set up the EVPN destination and the service does not come up.
- EVPN-VPWS Epipes support **force-qinq-vc-forwarding [c-tag-c-tag | s-tag-c-tag]** command under **bgp-evpn mpls** and the **qinq-vlan-translation s-tag.c-tag** command on ingress QinQ SAPs.  
When QinQ VLAN translation is configured at the ingress QinQ or dot1q SAP, the service-delimiting outer and inner VLAN values can be translated to the configured values. The **force-qinq-vc-forwarding s-tag-c-tag** command must be configured to preserve the translated QinQ tags in the payload when sending EVPN packets. This translation and preservation behavior is aligned with the “normalization” concept described in *draft-ietf-bess-evpn-vpws-fxc*. The VLAN tag processing described in [Epipe service pseudowire VLAN tag processing](#) applies to EVPN destinations in EVPN-VPWS services too.

The following features, described in [EVPN-VPWS for VXLAN tunnels](#), are also supported for MPLS tunnels:

- Advertisement of the Layer-2 MTU and consistency checking of the MTU of the AD per-EVI routes.
- Use of A/S PW and MC-LAG at access.
- EVPN multihoming, including:
  - Single-active and all-active
  - Regular or virtual ESs
  - All existing DF election modes

### 5.3.4.3 EVPN-VPWS services with local-switching support

Epipes with BGP-EVPN MPLS support the following configurations:

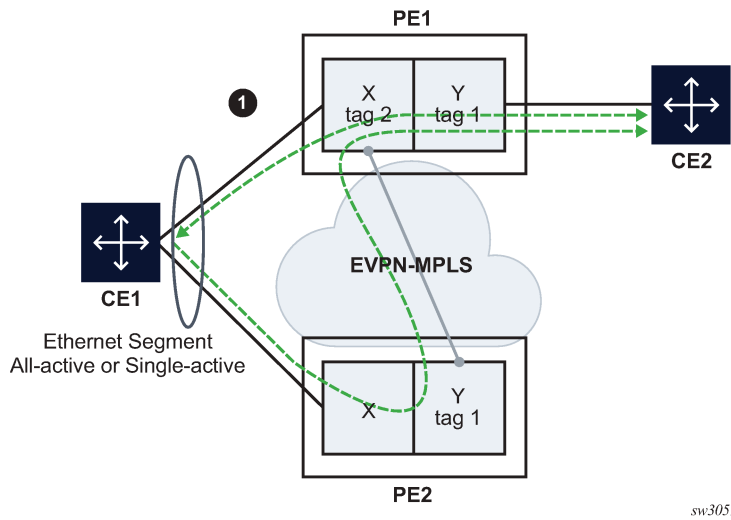
- up to two endpoints
- up to two SAPs, each associated with a different configured endpoint
- two pairs of local/remote attachment circuit Ethernet tags, also associated with different configured endpoints
- EVPN destinations used as Inter-Chassis Backup (ICB) links

The support of endpoints and up to two SAPs with local switching allows two-node and three-node topologies for EVPN-VPWS. See sections [EVPN-VPWS endpoints example 1](#), [EVPN-VPWS endpoints example 2](#), and [EVPN-VPWS endpoints example 3](#) for example topologies.

#### 5.3.4.3.1 EVPN-VPWS endpoints example 1

The following figure shows an example of EVPN-VPWS endpoints.

Figure 85: EVPN-VPWS endpoints example 1



In this example, PE1 is configured with the following Epipe services.

```

endpoint X create
exit
endpoint Y create
exit
bgp-evpn
  evi 350
    local-attachment-circuit "CE-1" endpoint "Y" create
      eth-tag 1
    exit
    remote-attachment-circuit "ICB-1" endpoint "Y" create
      eth-tag 2
    exit
    local-attachment-circuit "CE-2" endpoint "X" create
      eth-tag 2
    exit
    remote-attachment-circuit "ICB-2" endpoint "X" create
      eth-tag 1
    exit
  mpls bgp 1
    auto-bind-tunnel
      resolution any
    exit
    no shutdown
  exit
exit
sap lag-1:1 endpoint X create
exit
sap 1/1/1:1 endpoint Y create
exit

```

PE2 is configured with the following Epipe services.

```
bgp-evpn
  evi 350
    local-attachment-circuit "CE-1" create
    eth-tag 1
  exit
```

```
remote-attachment-circuit "ICB-1" create
  eth-tag 2
  exit
// implicit endpoint "Y"
mpls bgp 1
  auto-bind-tunnel
  resolution any
  exit
no shutdown
exit
  exit
sap lag-1:1 create
exit
// implicit endpoint "X"
```

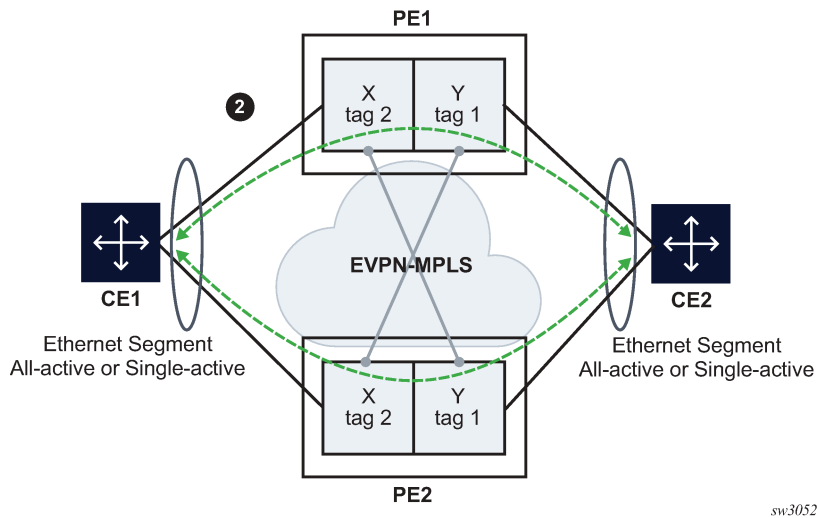
In this example, if we assume multihoming on CE1, the following applies:

- PE1 advertises two AD per-EVI routes, for tags 1 and 2, respectively. PE2 advertises only the route for tag 1.
  - AD per-EVI routes for tag 1 are advertised based on CE1 SAPs' states
  - AD per-EVI route for tag 2 is advertised based on CE2 SAP state
- PE1 creates endpoint X with sap lag-1:1 and ES-destination to tag 1 in PE2
- PE2 creates the usual destination to tag 2 in PE1
- In case of all-active MH:
  - traffic from CE1 to PE1 is forwarded to CE2 directly
  - traffic from CE1 to PE2 is forwarded to PE1 with the label that identifies CE2's SAP
  - traffic from CE2 is forwarded to CE1 directly because CE1's SAP is the endpoint Tx; in case of failure on CE1's SAP, PE1 changes the Tx object to the ES-destination to PE2
- In case of single-active MH, traffic flows in the same way, except that a non-DF SAP is operationally down and therefore cannot be an endpoint Tx object.

### 5.3.4.3.2 EVPN-VPWS endpoints example 2

The following figure shows an example of EVPN-VPWS endpoints.

Figure 86: EVPN-VPWS endpoints example 2



sw3052

In this example, PE1 is configured with the following Epipe services.

```

endpoint X create
exit
endpoint Y create
exit
bgp-evpn
  evi 350
    local-attachment-circuit "CE-1" endpoint "Y" create
      eth-tag 1
    exit
    remote-attachment-circuit "ICB-1" endpoint "Y" create
      eth-tag 2
    exit
    local-attachment-circuit "CE-2" endpoint "X" create
      eth-tag 2
    exit
    remote-attachment-circuit "ICB-2" endpoint "X" create
      eth-tag 1
    exit
  mpls bgp 1
    auto-bind-tunnel
    resolution any
    exit
    no shutdown
    exit
  exit
exit
sap lag-1:1 endpoint X create
exit
sap lag-2:1 endpoint Y create
exit

```

PE2 is configured with the following Epipe services.

```

endpoint X create
exit
endpoint Y create
exit
bgp-evpn

```

```
evi 350
local-attachment-circuit "CE-1" endpoint "Y" create
  eth-tag 1
exit
remote-attachment-circuit "ICB-1" endpoint "Y" create
  eth-tag 2
exit
local-attachment-circuit "CE-2" endpoint "X" create
  eth-tag 2
exit
remote-attachment-circuit "ICB-2" endpoint "X" create
  eth-tag 1
exit
mpls bgp 1
  auto-bind-tunnel
  resolution any
  exit
  no shutdown
  exit
exit
sap lag-1:1 endpoint X create
exit
sap lag-2:1 endpoint Y create
exit
```

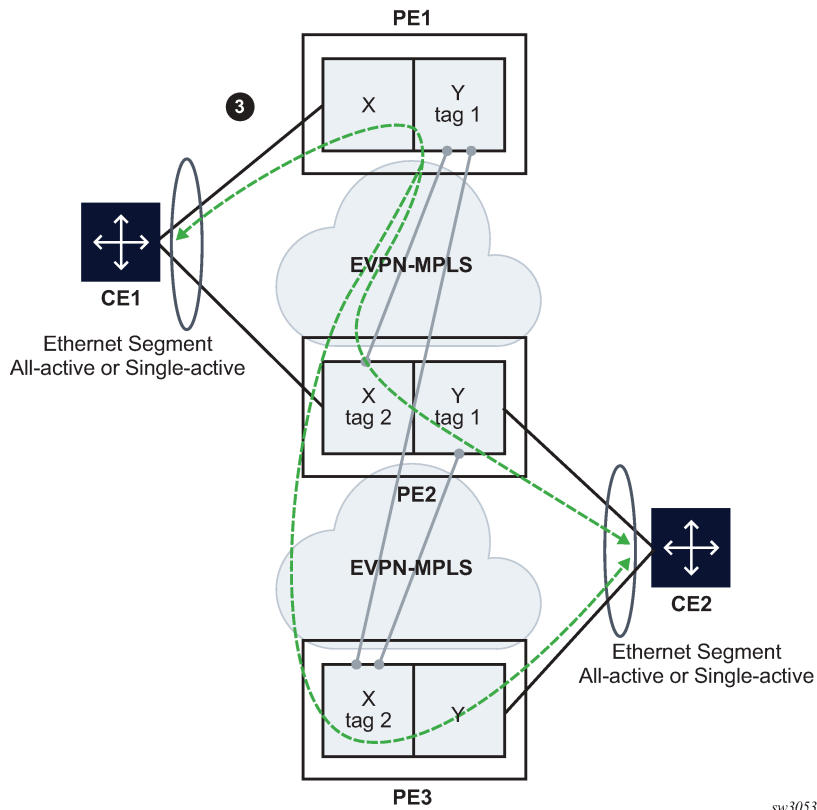
This example is similar to the [EVPN-VPWS endpoints example 1](#) example, except that the two PEs are multihomed to both CEs. In [EVPN-VPWS endpoints example 1](#), if CE2 goes down, then no traffic exists between PEs, because the two PEs lose all the objects in the endpoint connected to CE2. Traffic that arrives on EVPN is only forwarded to a SAP on a different endpoint.

### 5.3.4.3.3 EVPN-VPWS endpoints example 3

The following figure shows an example of EVPN-VPWS endpoints.



Figure 87: EVPN-VPWS endpoints example 3



In this example, PE1 is configured with the following Epipe services.

```

bgp-evpn
 evi 350
  local-attachment-circuit "CE-1"
    eth-tag 1
  exit
  remote-attachment-circuit "ICB-1"
    eth-tag 2
  exit
  // implicit endpoint "Y"
  mpls bgp 1
    auto-bind-tunnel
    resolution any
  exit
  no shutdown
  exit
  exit
  sap lag-1:1 create
  // implicit endpoint "X"
  exit

```

PE2 is configured with the following Epipe services.

```

endpoint X create
exit
endpoint Y create

```

```

exit
bgp-evpn
  evi 350
    local-attachment-circuit "CE-1" endpoint "Y"
      eth-tag 1
    exit
    remote-attachment-circuit "ICB-1" endpoint "Y"
      eth-tag 2
    exit
    local-attachment-circuit "CE-2" endpoint "X"
      eth-tag 2
    exit
    remote-attachment-circuit "ICB-2" endpoint "X"
      eth-tag 1
    exit
  mpls bgp 1
    auto-bind-tunnel
      resolution any
    exit
    no shutdown
  exit
exit
sap lag-1:1 endpoint X create
exit
sap lag-2:1 endpoint Y create
exit

```

PE3 is configured with the following Epipe services.

```

bgp-evpn
  evi 350
    local-attachment-circuit "CE-2"
      eth-tag 2
    exit
    remote-attachment-circuit "ICB-2"
      eth-tag 1
    exit
  // implicit endpoint "X"
  mpls bgp 1
    auto-bind-tunnel
      resolution any
    exit
    no shutdown
  exit
exit
sap lag-1:1 create
// implicit endpoint "Y"
exit

```

This example is similar to the [EVPN-VPWS endpoints example 2](#) example, except that a third node is added. Nodes PE1 and PE3 have implicit endpoints. Only node PE2 requires the configuration of endpoints.

#### 5.3.4.4 EVPN-VPWS FXC

FXC, specified in *draft-ietf-bess-evpn-vpws-fxc*, extends EVPN-VPWS services to support multiple SAPs at the access, in contrast with only one SAP in regular EVPN-VPWS services. Multiplexing multiple SAPs into a single EVPN-VPWS tunnel allows the user to save MPLS labels and reduce the number of EVPN Auto-Discovery per EVI routes.

Use the following command to turn an Epipe into a FXC service.

- **MD-CLI**

```
configure service epipe flexible-cross-connect true
```

- **classic CLI**

```
configure service epipe flexible-cross-connect create
```

Since there are multiple active egress SAPs in FXC services, EVPN labels are no longer enough to identify the egress SAP on the egress node. Therefore, payload VLAN tags are transmitted into EVPN-encapsulated FXC packets, and the router looks up these inner VLAN tags to forward the decapsulated traffic to the correct egress SAP.

The VLAN normalization configuration ensures that the inner VLAN tags are guaranteed to be unique in the context of the FXC service. Use commands under the following context for each SAP to configure normalized Q-tags if the service-delimiting tags on FXC SAPs are not unique.

```
configure service epipe sap qtag-normalization
```

VLAN normalization configuration and the two FXC modes of operation (default and VLAN aware mode) are described in [FXC VLAN normalization](#), [Default FXC mode](#), and [VLAN-aware bundle FXC mode](#).

#### 5.3.4.4.1 FXC VLAN normalization

Normalization is a key concept in EVPN-VPWS FXC services and refers to the process of making the SAP VLAN tags unique on the ingress FXC PE. The normalized VLAN tags are sent on the wire in EVPN packets so the egress PE can forward traffic to the correct egress SAP.

The FXC PEs build a table with <normalized-tags, sap> entries per service. An ILM lookup on the service label is performed first, which also indicates whether one or two tags are needed for the egress VLAN lookup (depending on the normalization mode supported by the egress SAP). The table is then looked up to find the egress SAP.

Use the following commands in the **configure service epipe** context to derive the normalized tags for a SAP:

- **MD-CLI**

```
sap qtag-normalization single-tag tag  
sap qtag-normalization double-tag s-tag  
sap qtag-normalization double-tag c-tag
```

- **classic CLI**

```
sap qtag-normalization tag create  
sap qtag-normalization s-tag.c-tag create
```

Where:

- The VLAN behavior adheres to the following rules:
  - Service-delimiting tags are popped at SAP ingress and pushed at SAP egress as usual.
  - When encapsulating into EVPN, service-delimiting tags are translated into normalized VLAN tags and pushed into the EVPN packet.

- The number of tags pushed on or popped from EVPN packets depends on whether single-tag normalization (one tag pushed/popped) or double-tag normalization (two tags pushed/popped) is configured.
- There is no dependency between the SAP type (**null**, **dot1q**, or **qinq**) and the normalization mode. A null, qinq, or dot1q SAP can be configured with single or double normalization mode.
- Normalized tags are also used for encoding the Ethernet Tag ID in the AD per-EVI route (in VLAN-aware FXC mode) as *s-tag.c-tag* (double-tag normalization) or tag (single-tag normalization), where:
  - in single-tag normalization mode, the tag takes the right-most 12 bits of the Ethernet Tag ID
  - in double-tag normalization mode, the S-tag value takes the left-most 12 bits and the C-tag takes the right-most 12 bits
- If no normalized mode is configured, the mode and normalized tag values are derived from the service-delimiting tags, if possible. If the SAP does have service-delimiting tags, configuration of the normalization mode and tags is optional, provided that the number of service-delimiting tags matches the normalization mode.
  - For example, if the SAP ID is 1/1/1:10 but the configured normalization is **double-tag**, the **s-tag** and **c-tag** values must be configured.
  - For SAPs that do not have service-delimiting tags, configuration of the normalization mode and tags is mandatory. The SAPs with no service-delimiting tags are:
    - Null
    - Dot1q:
      - 1/1/1:\*
      - 1/1/1:0
      - 1/1/1:cp-1
    - Qinq
      - 1/1/1:0.\*
      - 1/1/1:\*. \*
      - 1/1/1:\*.null
      - 1/1/1:null.null
      - 1/1/1:cp-1.\*
      - 1/1/1:cp-1.0

The following is an example of Q-tag normalization of a **double-tag** with **s-tag**=100 and **c-tag**=200 configuration.

#### Example: MD-CLI

```
*[ex:/configure service epipe "1"]
A:admin@PE-1# info
  sap 1/1/c1/1:10.20 {
    qtag-normalization {
      double-tag {
        s-tag 100
        c-tag 200
      }
    }
  }
```

```
}
```

### Example: classic CLI

```
A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1:10.20 qtag-normalization 100.200 create
```

The following is an example of Q-tag normalization of a **single-tag** with a **tag=300** configuration.

### Example: MD-CLI

```
[ex:/configure service epipe "1"]
A:admin@node-2# info
  sap 1/1/c1/1:300.400 {
    qtag-normalization {
      single-tag {
        tag 300
      }
    }
  }
}
```

### Example: classic CLI

```
A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1:300.400 qtag-normalization 300 create
```

The following is an example of Q-tag normalization of a **double-tag** with **s-tag=10** and **c-tag=0** for two tag configuration.

### Example: MD-CLI

```
[ex:/configure service epipe "1"]
A:admin@node-2# info
  flexible-cross-connect true
  sap 1/1/c1/1:10 {
    qtag-normalization {
      double-tag {
        s-tag 10
        c-tag 0
      }
    }
  }
}
```

### Example: classic CLI

```
A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1:10 qtag-normalization 10.0 create
```

In the following examples, the normalization mode is not configured but derived from the service-delimiting tag.

The following is an example of a normalized **s-tag.c-tag** (10.20) with a **double-tag** mode.

### Example: MD-CLI

```
[ex:/configure service epipe "1"]
```

```
A:admin@node-2# info
flexible-cross-connect true
sap 1/1/c1/1:10.20 {
}
```

### Example: classic CLI

```
A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1:10.20 create
```

The following is an example of a normalized *tag* (10) and the implicit normalization mode is **single-tag**.

### Example: MD-CLI

```
[ex:/configure service epipe "1"]
A:admin@node-2# info
flexible-cross-connect true
sap 1/1/c1/1:10 {
}

[ex:/configure service epipe "2"]
A:admin@node-2# info
flexible-cross-connect true
sap 1/1/c2/1:10.0 {
}

[ex:/configure service epipe "3"]
A:admin@node-2# info
flexible-cross-connect true
sap 1/1/c3/1:10.* {
}

[ex:/configure service epipe "4"]
A:admin@node-2# info
flexible-cross-connect true
sap 1/1/c4/1:10.cp-1 {
}
```

### Example: classic CLI

```
A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1:10 create

A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1:10.0 create

A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1:10.* create

A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1:10.cp-1 create
```

The following is an example showing the necessity of normalization mode and tags when there are no service-delimiting tags.

**Example: MD-CLI**

```
[ex:/configure service epipe "1"]
A:admin@node-2# info
flexible-cross-connect true
sap 1/1/c1/1 {
    qtag-normalization {
        single-tag {
            tag 100
        }
    }
}
```

**Example: classic CLI**

```
A:node-2>config>service>epipe# info
-----
sap 1/1/c1/1 qtag-normalization 100 create
```

**5.3.4.4.2 Default FXC mode**

SR OS supports the default FXC mode defined in *draft-ietf-bess-evpn-vpws-fxc*. Use the following command to enable the default mode:

- **MD- CLI**

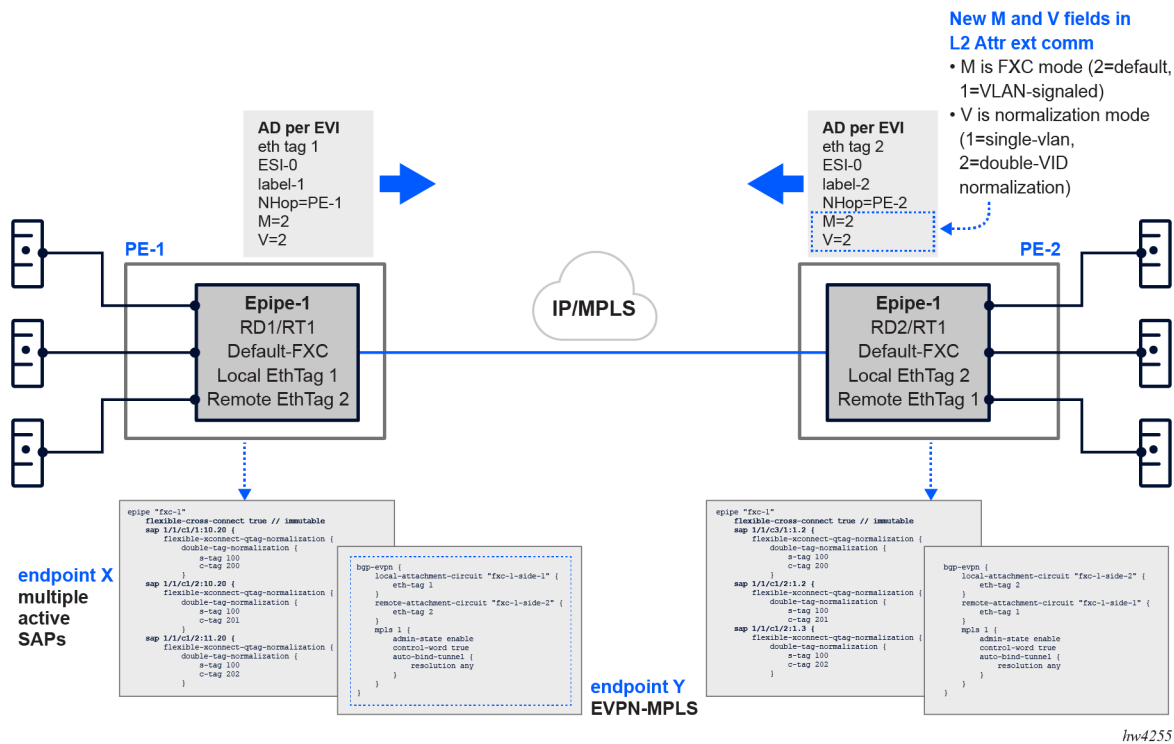
```
configure service epipe flexible-cross-connect true
```

- **classic CLI**

```
configure service epipe flexible-cross-connect
```

This command allows the user to configure multiple active SAPs and BGP-EVPN MPLS in the same Epipe service. From the perspective of EVPN, a local and remote Ethernet tag must be configured and, therefore, a single EVPN label is allocated per FXC. The default FXC mode is displayed in the following figure.

Figure 88: Default FXC mode



The AD per-EVI route advertised for the FXC default mode service contains two additional flags (compared to the route for regular Epipes):

- M-flag, with value 2, indicating the FXC default mode
- V-flag, which indicates the **qtag-normalization** mode. A value of 1 indicates **single-tag** normalization, while a value of 2 indicates **double-tag** normalization.

The following is an example of FXC mode.

### Example: MD-CLI

```
[ex:/configure service epipe "fxc-1"]
A:admin@node-2# info
flexible-cross-connect true // immutable
bgp 1 {
}
sap 1/1/c1/1:10.20 {
  qtag-normalization {
    double-tag {
      c-tag 200
      s-tag 100
    }
  }
}
sap 1/1/c2/1:10.20 {
  qtag-normalization {
    double-tag {
      c-tag 201
      s-tag 101
    }
  }
}
```



```

    }
  }
}
sap 1/1/c2/1:11.20 {
  qtag-normalization {
    double-tag {
      c-tag 202
      s-tag 102
    }
  }
}
}
bgp-evpn {
  local-attachment-circuit "fxc-1-side-1" {
    eth-tag 1
  }
  remote-attachment-circuit "fxc-1-side-2" {
    eth-tag 2
  }
  mpls 1 {
    admin-state enable
    control-word true
    auto-bind-tunnel {
      resolution any
    }
  }
}
}

```

### Example: classic CLI

```

A:node-2>config>service>epipe# info
-----
      bgp-evpn
        local-attachment-circuit fxc-1-side-1 bgp 1 create
        eth-tag 1
      exit
        remote-attachment-circuit fxc-1-side-2 bgp 1 create
        eth-tag 2
      exit
      evi 500
      mpls bgp 1
        control-word
        auto-bind-tunnel
        resolution any
      exit
      no shutdown
    exit
  exit
  sap 1/1/c1/1:10.20 qtag-normalization 100.200 create
  no shutdown
  exit
  sap 1/1/c2/1:10.20 qtag-normalization 101.201 create
  no shutdown
  exit
  sap 1/1/c2/1:11.20 qtag-normalization 102.202 create
  no shutdown
  exit
  no shutdown
-----

```

The following is an example of FXC mode where the remote PE is expected to be configured with the same normalized Q-tags on SAPs that may have the same service-delimiting VLANs as the local PE.

**Example: MD-CLI**

```
[ex:/configure service epipe "fxc-1"]
A:admin@node-2# info
flexible-cross-connect true // immutable
bgp 1 {
}
sap 1/1/c3/1:1.2 {
    qtag-normalization {
        double-tag {
            c-tag 200
            s-tag 100
        }
    }
}
sap 1/1/c2/1:1.2 {
    qtag-normalization {
        double-tag {
            c-tag 201
            s-tag 101
        }
    }
}
sap 1/1/c2/1:1.3 {
    qtag-normalization {
        double-tag {
            c-tag 202
            s-tag 102
        }
    }
}
bgp-evpn {
    evi 1
    local-attachment-circuit "fxc-1-side-2" {
        eth-tag 2
    }
    remote-attachment-circuit "fxc-1-side-1" {
        eth-tag 1
    }
    mpls 1 {
        admin-state enable
        control-word true
        auto-bind-tunnel {
            resolution any
        }
    }
}
```

**Example: classic CLI**

```
A:node-2>config>service>epipe# info
-----
    bgp-evpn
        local-attachment-circuit fxc-1-side-2 bgp 1 create
        eth-tag 2
    exit
    remote-attachment-circuit fxc-1-side-1 bgp 1 create
        eth-tag 1
    exit
    evi 1
    mpls bgp 1
        control-word
        auto-bind-tunnel
```

```

        resolution any
        exit
        no shutdown
    exit
exit
sap 1/1/c3/1:1.2 qtag-normalization 100.200 create
    no shutdown
exit
sap 1/1/c2/1:1.2 qtag-normalization 101.201 create
    no shutdown
exit
sap 1/1/c2/1:1.3 qtag-normalization 102.202 create
    no shutdown
exit
no shutdown
-----
```

Use the following command to display BGP EVPN-MPLS destination information.

```
show service id 500 evpn-mpls
```

Output example

```
=====
BGP EVPN-MPLS Dest (Instance 1)
=====
TEP Address                Egr Label          Num  Last Change
                          Transport:Tnl-id  Saps
-----
2001:db8::4                524283             2    09/13/2024 16:58:53
                          ldp:65542
-----
Number of entries : 1
=====

=====
BGP EVPN-MPLS Dest (Instance 2)
=====
TEP Address                Egr Label          Num  Last Change
                          Transport:Tnl-id  Saps
-----
No Matching Entries
=====

=====
BGP EVPN-MPLS Ethernet Segment Dest (Instance 1)
=====
Eth SegId                  Num Saps Last Change
-----
No Matching Entries
=====

=====
BGP EVPN-MPLS Ethernet Segment Dest (Instance 2)
=====
Eth SegId                  Num Saps Last Change
-----
No Matching Entries
=====
```

Use the following command to display FXC configuration that shows a single EVPN or ES destination (as in regular EVPN-VPWS).

```
show service id 500 evpn-mpls fxc
```

### Output example

```
=====
FXC SAP Connections (Instance 1)
=====
Dest Identifier          Sap (Normalized Tags)          Last Change
  Transport:Tnl-id
-----
mpls-1:2001:db8::4:524283  lag-1:500 (550.550)          09/13/2024
                               16:58:53
                               09/13/2024
                               16:58:53
-----
Number of Entries : 1
=====
```

While the default FXC mode uses a single label and single EVPN AD per-EVI route per Epipe service, which translates into significant resource savings in the network, an individual SAP going operationally down goes unnoticed. The remote PE continues sending traffic for the down SAP, only to be dropped at the egress PE where the SAP is operationally down.

Consequently, EVPN multihoming is supported only if all SAPs in the FXC belong to the same Ethernet segment (ES). While individual SAP failures are still unnoticed, physical port failures means the entire ES fails and the AD per-EVI and per-ES routes are withdrawn so that the remote PE can switch over to the redundant PE of the ES.

If signaling and fault propagation per SAP is needed in the FXC, the VLAN-aware bundle FXC mode must be configured.

#### 5.3.4.4.3 VLAN-aware bundle FXC mode

Use the following commands to enable the VLAN-aware bundle FXC mode:

- **MD-CLI**

```
configure service epipe flexible-cross-connect true
configure service epipe bgp-evpn local-attachment-circuit vlan-signaled-flexible-cross-
connect true
```

- **classic CLI**

```
configure service epipe flexible-cross-connect
configure service epipe flexible-cross-connect local-attachment-circuit vlan-signaled-
flexible-cross-connect
```

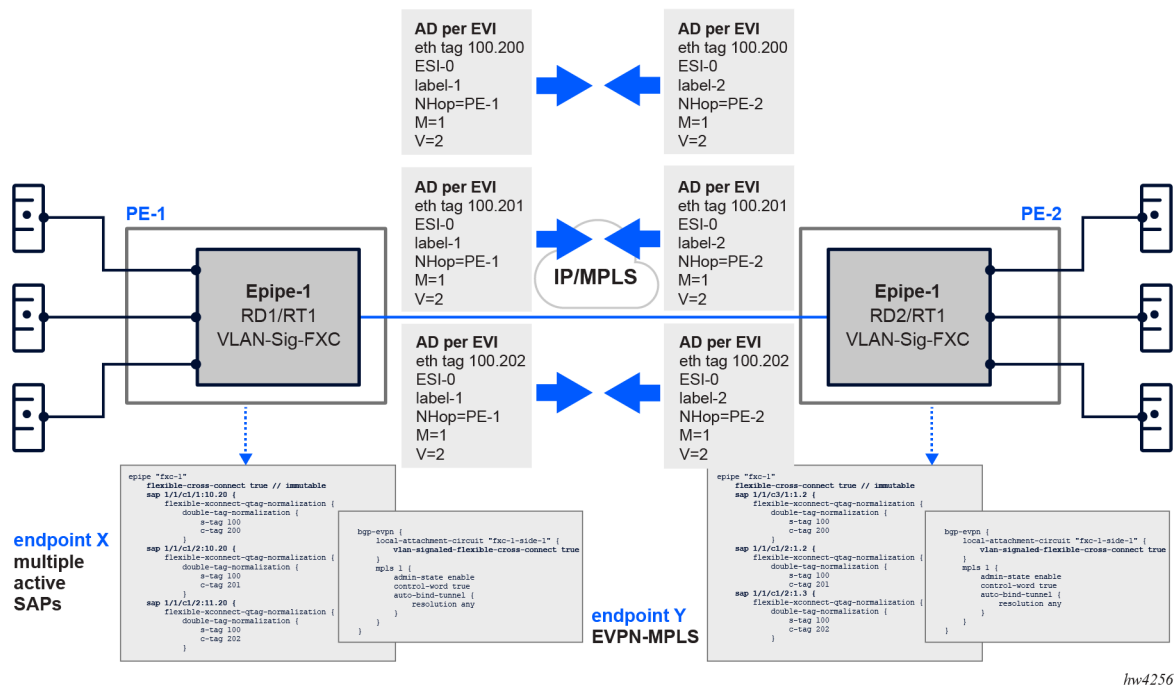
When the VLAN-aware bundle FXC mode is enabled, the following rules apply:

- Two EVPN labels are allocated for the Epipe, one per normalization mode (single- and double-tag normalization).

- The configuration of local and remote Ethernet tags is blocked; they are automatically derived from the normalized tags for each SAP. For example, suppose two SAPs with normalization 10.20 and 30.40 are configured in the service. This automatically creates a local Ethernet tag 10.20 (and same remote Ethernet tag) and a local Ethernet tag 30.40 (and same Ethernet tag).
- A separate AD per-EVI route is now advertised per SAP with the automatically generated local Ethernet tags, and flags M=1 (VLAN-signaled FXC) and V=1 or V=2 depending on the normalization type for the SAP.
- In this mode, traffic from a specific SAP with normalized tags 100.200 is discarded at the ingress PE when there is no AD per-EVI route for tags 100.200 at the Epipe service. The SAP (for which no remote peer matching tags are found) stays operationally up, however, traffic is dropped because there is no destination to send it to. SAP statistics show the discarded ingress packets. In this case, the **show service id evpn-mpls fxc** command displays the SAP associated with a destination with a value of none.
- Contrary to the default mode, in VLAN-aware bundle FXC mode, multiple ESs may exist in the same Epipe (with a single SAP per FXC per ES).

The following figure shows the VLAN-aware bundle or VLAN-signaled FXC mode.

Figure 89: VLAN-aware bundle or VLAN-signaled FXC mode



hw4256

Where node-1 is configured as shown in the following example.

### Example: MD-CLI

```
[ex:/configure service epipe "fxc-1"]
A:admin@node-1# info
flexible-cross-connect true // immutable
bgp 1 {
}
sap 1/1/c1/1:10.20 {
```

```

        qtag-normalization {
            double-tag {
                c-tag 200
                s-tag 100
            }
        }
    }
    sap 1/1/c2/1:10.20 {
        qtag-normalization {
            double-tag {
                c-tag 201
                s-tag 101
            }
        }
    }
    sap 1/1/c2/1:11.20 {
        qtag-normalization {
            double-tag {
                c-tag 202
                s-tag 102
            }
        }
    }
    bgp-evpn {
        evi 1
        local-attachment-circuit "fxc-1-side-1" {
            vlan-signaled-flexible-cross-connect true
        }
        mpls 1 {
            admin-state enable
            control-word true
            auto-bind-tunnel {
                resolution any
            }
        }
    }
}

```

### Example: classic CLI

```

A:node-1>config>service>epipe# info
-----
      bgp-evpn
        local-attachment-circuit fxc-1-side-1 bgp 1 create
        vlan-signaled-flexible-cross-connect
      exit
      evi 1
      mpls bgp 1
        control-word
        auto-bind-tunnel
        resolution any
      exit
      no shutdown
    exit
  exit
  sap 1/1/c1/1:10.20 qtag-normalization 100.200 create
    no shutdown
  exit
  sap 1/1/c2/1:10.20 qtag-normalization 101.201 create
    no shutdown
  exit
  sap 1/1/c2/1:11.20 qtag-normalization 102.202 create
    no shutdown
  exit

```

```
no shutdown
-----
```

Where node-2 in [Figure 89: VLAN-aware bundle or VLAN-signaled FXC mode](#) is configured as follows:

### Example: MD-CLI

```
[ex:/configure service epipe "fxc-1"]
A:admin@node-2# info
flexible-cross-connect true // immutable
bgp 1 {
}
sap 1/1/c3/1:1.2 {
  qtag-normalization {
    double-tag {
      c-tag 200
      s-tag 100
    }
  }
}
sap 1/1/c1/1:101.201 {
}
sap 1/1/c5/1:102.202 {
}
bgp-evpn {
  local-attachment-circuit "fxc-1-side-1" {
    vlan-signaled-flexible-cross-connect true
  }
  mpls 1 {
    admin-state enable
    control-word true
    auto-bind-tunnel {
      resolution any
    }
  }
}
```

### Example: classic CLI

```
A:node-2>config>service>epipe# info
-----
bgp-evpn
  local-attachment-circuit fxc-1-side-1 bgp 1 create
  vlan-signaled-flexible-cross-connect
  exit
  evi 1
  mpls bgp 1
    control-word
    auto-bind-tunnel
    resolution any
  exit
  no shutdown
  exit
exit
sap 1/1/c3/1:10.20 qtag-normalization 100.200 create
  no shutdown
exit
sap 1/1/c1/1:101.201 create
  no shutdown
exit
sap 1/1/c5/1:102.202 create
  no shutdown
```

```
exit
no shutdown
-----
```

In the preceding example configuration, if the SAP associated with normalized tags 100.200 on remote node-2 goes down, upon withdrawal of the AD per-EVI route, the local SAP associated with the same normalized tags is also removed from the output. Use the following command to display this information.

```
show service id 100 evpn-mpls fxc
```

Output example

```
=====
FXC SAP Connections (Instance 1)
=====
Dest Identifier          Sap (Normalized Tags)          Last Change
Transport:Tnl-id
-----
mpls-1:192.0.2.2:524281  1/1/c1/1:10.20(100.200)      09/13/2024
                                16:58:53
    ldp:65542
mpls-1:192.0.2.2:524281  1/1/c2/1:10.20(101.201)      09/13/2024
                                16:58:53
    ldp:65542
mpls-1:192.0.2.2:524281  1/1/c2/1:11.20(102.202)      09/13/2024
                                16:58:53
    ldp:65542

-----
Number of Entries : 3
-----
=====

1 2024/09/16 00:26:41.390 UTC MINOR: DEBUG #2001 Base Peer 1: 2001:db8::2
"Peer 1: 2001:db8::2: UPDATE
Peer 1: 2001:db8::2 - Received BGP UPDATE:
    Withdrawn Length = 0
    Total Path Attr Length = 34
    Flag: 0x90 Type: 15 Len: 30 Multiprotocol Unreachable NLRI:
        Address Family EVPN
        Type: EVPN-AD Len: 25 RD: 192.0.2.2:100 ESI: ESI-0, tag: 409800 Label: 0 (R
aw Label: 0x0) PathId:
"
```

```
show service id 100 evpn-mpls fxc
```

```
=====
FXC SAP Connections (Instance 1)
=====
Dest Identifier          Sap (Normalized Tags)          Last Change
Transport:Tnl-id
-----
mpls-1:192.0.2.2:524281  1/1/c1/1:10.20(101.201)      09/13/2024
                                16:58:53
    ldp:65542
mpls-1:192.0.2.2:524281  1/1/c2/1:11.20(102.202)      09/13/2024
                                16:58:53
    ldp:65542

-----
```



```
Number of Entries : 2
-----
=====
```

In addition, using the following command displays new information to identify the normalization mode (indicated in bold):

```
show service id 4 sap 1/1/cl/1:45 detail
```

### Output example

```
=====
Service Access Points(SAP)
=====
Service Id      : 4
SAP             : 1/1/cl/1:45  Encap           : q-tag
Description     : (Not Specified)
Admin State     : Up                Oper State      : Up
Flags           : None
Multi Svc Site  : None
Last Status Change : 04/02/2024 18:08:35
Last Mgmt Change  : 04/02/2024 18:08:26
Sub Type        : regular
Dot1Q Ethertype : 0x8100            QinQ Ethertype  : 0x8100
Split Horizon Group: (Not Specified)
Eth Seg Name     : vES-4
Admin MTU        : 9208            Oper MTU        : 9208
Ingr IP Fltr-Id  : n/a            Egr IP Fltr-Id  : n/a
Ingr Mac Fltr-Id : n/a            Egr Mac Fltr-Id : n/a
Ingr IPv6 Fltr-Id : n/a          Egr IPv6 Fltr-Id : n/a
qinq-pbit-marking : both
Endpoint        : N/A
Egr Agg Rate Limit : max
Q Frame-Based Acct : Disabled      Limit Unused BW  : Disabled
Vlan-translation  : None
Qinq-vlan-translation : None      Qinq-vlan-translation Ids : None
Acct. Pol        : None           Collect Stats     : Disabled
Application Profile: None
Transit Policy    : None

Oper Group       : og-4            Monitor Oper Grp  : (none)
Host Lockout Plcy : n/a
Ignore Oper Down  : Disabled
Lag Link Map Prof : (none)
Cflowd           : Disabled
Bandwidth         : Not-Applicable
Oper DCpu Prot Pol : _default-access-policy
Virtual Port      : (Not Specified)

FXC VLAN normalization mode : Double (Tags 9.10)

-----
ETH-CFM SAP specifics
-----
Tunnel Faults    : n/a            AIS              : Disabled
MC Prop-Hold-Timer : n/a
Squelch Levels   : None
Squelch Ctag Levels: None
Collect Lmm Stats : Disabled
LMM FC Stats     : None
LMM FC In Prof   : None
```

-----  
QoS

```

-----
Ingress qos-policy : 1                      Egress qos-policy : 1
Ingress FP QGrp    : (none)                  Egress Port QGrp   : (none)
Ing FP QGrp Inst   : (none)                  Egr Port QGrp Inst: (none)
Ing ip-match tag    : none                    Ing ipv6-match tag: none
I. Sched Pol       : (Not Specified)
E. Sched Pol       : (Not Specified)
I. Policer Ctl Pol : (Not Specified)
E. Policer Ctl Pol : (Not Specified)
I. QGrp Redir. List: (Not Specified)
E. QGrp Redir. List: (Not Specified)
Hw Agg Shaper Q Set: No
Hw Agg Shpr QSet Sz: 0
Hw Agg Shpr In-Use : No
Latency Budget     : 0 us

```

-----  
Sap Aggregate Stats

```

-----
                Packets                Octets
Ingress
Aggregate Offered : 0                    0
Aggregate Forwarded : 0                  0
Aggregate Dropped  : 0                    0

Egress
Aggregate Forwarded : 0                    0
Aggregate Dropped   : 0                    0

```

-----  
Sap Statistics

```

-----
Last Cleared Time : N/A

                Packets                Octets
CPM Ingress       : 0                    0

Forwarding Engine Stats
Dropped           : 3                    204
Received Valid    : 0                    0
Off. HiPrio       : 0                    0
Off. LowPrio      : 0                    0
Off. Uncolor      : 0                    0
Off. Managed      : 0                    0

Queueing Stats(Ingress QoS Policy 1)
Dro. HiPrio       : 0                    0
Dro. LowPrio      : 0                    0
For. InProf       : 0                    0
For. OutProf      : 0                    0

Queueing Stats(Egress QoS Policy 1)
Dro. In/InplusProf : 0                    0
Dro. Out/ExcProf   : 0                    0
For. In/InplusProf : 0                    0
For. Out/ExcProf   : 0                    0

```

-----  
Sap per Queue stats

```

-----
                Packets                Octets
Ingress Queue 1 (Unicast) (Priority)
Off. HiPrio       : 0                    0

```

```

Off. LowPrio      : 0
Dro. HiPrio      : 0
Dro. LowPrio     : 0
For. InProf      : 0
For. OutProf     : 0

Egress Queue 1
For. In/InplusProf : 0
For. Out/ExcProf  : 0
Dro. In/InplusProf : 0
Dro. Out/ExcProf  : 0
=====

```

Use the following command to display all the SAPs and their EVPN destinations.

```
show service id 100 evpn-mpls
```

### Output example

```

=====
BGP EVPN-MPLS Dest (Instance 1)
=====
TEP Address          Egr Label          Num Saps    Last Change
                    Transport:Tnl-id
-----
10.20.1.2            524285             3           08/15/2023 23:18:50
                    ldp:65543
10.20.1.2            524286             2           08/15/2023 23:18:50
                    ldp:65543
10.20.1.3            524285             1           08/15/2023 23:18:50
                    ldp:50000
-----
Number of entries : 3
=====

BGP EVPN-MPLS Ethernet Segment Dest (Instance 1)
=====
Eth SegId              Num Saps    Last Change
-----
00:00:11:11:00:00:11:11:11      1           08/15/2023 23:18:50
00:00:11:11:00:00:11:11:22:22    1           08/15/2023 23:18:50
-----
Number of entries : 2
=====

```

Use the following command to display all the FXC EVPN-MPLS connections.

```
show service id 100 evpn-mpls fxc
```

### Output example

```

=====
FXC EVPN-MPLS Connections
=====
Dest Identifier          Sap (Normalized Tags)  Last Change
Transport:Tnl-id
-----
mpls-1:10.20.1.2:524285  1/1/c1/1:5.6 (2.2)   08/15/2023 23:18:50

```

```

ldp:65543                lag-6:100 (3.4)      08/15/2023 23:18:50
                        1/1/c2/2:*.0 (100.1) 08/15/2023 23:18:50

mpls-1:10.20.1.2:524286  1/1/c2/3:1 (1)      08/15/2023 23:18:50
ldp:65543                lag-6:101 (101)    08/15/2023 23:18:50

mpls-1:10.20.1.3:524285  1/1/c2/3:2 (2)      08/15/2023 23:18:50
ldp:50000

eES:00:00:11:11:00:00:11:11:11 lag-6:200 (500)    08/15/2023 23:18:50

eES:00:00:11:11:00:00:11:11:22 lag-6:2.2 (2.2)    08/15/2023 23:18:50
-----
Number of entries : 5
=====

```

Use the following command to display all the FXC EVPN-MPLS connections for a specific SAP.

```
show service id 100 evpn-mpls fxc sap 1/1/c1/1:5.6
```

### Output example

```

=====
FXC EVPN-MPLS Connections
=====
Dest Identifier                Sap (Normalized Tags)  Last Change
Transport:Tnl-id
-----
mpls-1:10.20.1.2:524285      1/1/c1/1:5.6 (2.2)    08/15/2023 23:18:50
ldp:65543

```

## 5.3.5 EVPN for MPLS tunnels in routed VPLS services

EVPN-MPLS and IP-prefix advertisement (enabled by the **ip-route-advertisement** command) are fully supported in routed VPLS services and provide the same feature-set as EVPN-VXLAN. The following capabilities are supported in a service where **bgp-evpn mpls** is enabled:

- R-VPLS with VRRP support on the VPRN or IES interfaces
- R-VPLS support including **ip-route-advertisement** with regular interfaces  
This includes the advertisement and process of ip-prefix routes defined in IETF Draft *draft-ietf-bess-evpn-prefix-advertisement* with the appropriate encoding for EVPN-MPLS.
- R-VPLS support including **ip-route-advertisement** with **evpn-tunnel** interfaces
- R-VPLS with IPv6 support on the VPRN or IES IP interface

IES interfaces do not support either **ip-route-advertisement** or **evpn-tunnel**.

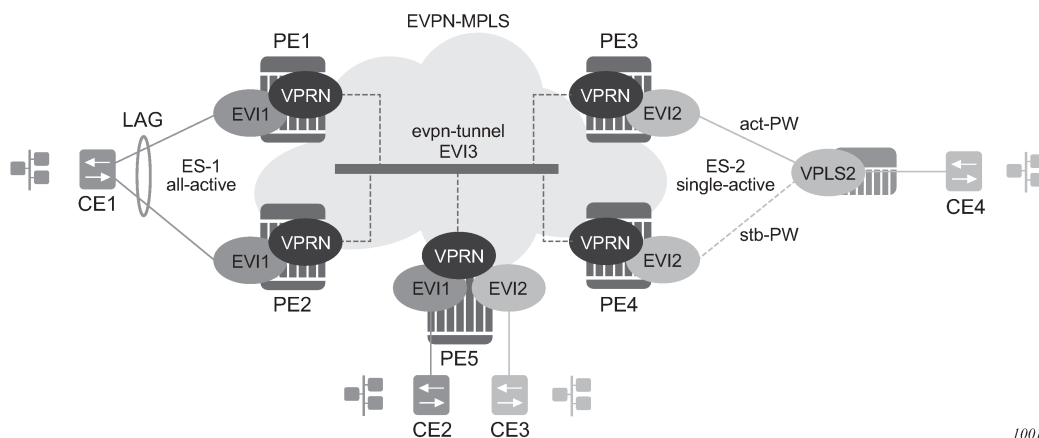
### 5.3.5.1 EVPN-MPLS multihoming and passive VRRP

SAP and spoke-SDP based ESs are supported on R-VPLS services where **bgp-evpn mpls** is enabled.

[Figure 90: EVPN-MPLS multihoming in R-VPLS services](#) shows an example of EVPN-MPLS multihoming in R-VPLS services, with the following assumptions:

- There are two subnets for a specific customer (for example, EVI1 and EVI2 in [Figure 90: EVPN-MPLS multihoming in R-VPLS services](#)), and a VPRN is instantiated in all the PEs for efficient inter-subnet forwarding.
- A “backhaul” R-VPLS with **evpn-tunnel** mode enabled is used in the core to interconnect all the VPRNs. EVPN IP-prefix routes are used to exchange the prefixes corresponding to the two subnets.
- An all-active ES is configured for EVI1 on PE1 and PE2.
- A single-active ES is configured for EVI2 on PE3 and PE4.

Figure 90: EVPN-MPLS multihoming in R-VPLS services



In the example in the preceding figure, the hosts connected to CE1 and CE4 can use regular VRRP for default gateway redundancy; however, this may not be the most efficient way to provide upstream routing.

For example, if PE1 and PE2 are using regular VRRP, the upstream traffic from CE1 may be hashed to the backup IRB VRRP interface, instead of being hashed to the master. The same may occur for single-active multihoming and regular VRRP for PE3 and PE4. The traffic from CE4 is sent to PE3, when PE4 may be the VRRP master. In that case, PE3 must send the traffic to PE4, instead of routing it directly.

Both cases use unnecessary bandwidth between the PEs to get to the master IRB interface. In addition, VRRP scaling is limited if aggressive keepalive timers are used.

Because of these issues, passive VRRP is the recommended method when EVPN-MPLS multihoming is used in combination with R-VPLS redundant interfaces.

Passive VRRP is a VRRP setting in which the transmission and reception of keepalive messages is completely suppressed, and therefore, the VPRN interface always behaves as the master. Passive VRRP is enabled by adding the **passive** keyword to the VRRP instance at creation, as shown in the following examples:

```
configure service vprn 1 interface int-1 vrrp 1 passive
```

```
configure service vprn 1 interface int-1 ipv6 vrrp 1 passive
```

For example, if PE1, PE2, and PE5 in [Figure 90: EVPN-MPLS multihoming in R-VPLS services](#) use passive VRRP, even if each individual R-VPLS interface has a different MAC/IP address, because they share the same VRRP instance 1 and the same backup IP, the three PEs will own the same virtual MAC and virtual IP address (for example, 00-00-5E-00-00-01 and 10.0.0.254). The virtual MAC is auto-derived

from 00-00-5E-00-00-VRID per RFC 3768. The following is the expected behavior when passive VRRP is used in this example:

- All R-VPLS IRB interfaces for EVI1 have their own physical MAC/IP address; they also own the same default gateway virtual MAC and IP address.
- All EVI1 hosts have a unique configured default gateway; for example, 10.0.0.254.
- When CE1 or CE2 send upstream traffic to a remote subnet, the packets are routed by the closest PE because the virtual MAC is always local to the PE.

For example, the packets from CE1 hashed to PE1 are routed at PE1. The packets from CE1 hashed to PE2 are routed directly at PE2.

- Downstream packets (for example, packets from CE3 to CE1), are routed directly by the PE to CE1, regardless of the PE to which PE5 routed the packets.

For example, the packets from CE3 sent to PE1 are routed at PE1. The packets from CE3 sent to PE2 are routed at PE2.

- In case of ES failure in one of the PEs, the traffic is forwarded by the available PE.

For example, if the packets routed by PE5 arrive at PE1 and the link to CE1 is down, PE1 sends the packets to PE2. PE2 forwards the packets to CE1 even if the MAC source address of the packets matches PE2's virtual MAC address. Virtual MACs bypass the R-VPLS interface MAC protection.

The following list summarizes the advantages of using passive VRRP mode versus regular VRRP for EVPN-MPLS multihoming in R-VPLS services:

- Passive VRRP does not require multiple VRRP instances to achieve default gateway load-balancing. Only one instance per R-VPLS, therefore only one default gateway, is needed for all the hosts.
- The convergence time for link/node failures is not impacted by the VRRP convergence, because all the nodes in the VRRP instance are master routers.
- Passive VRRP scales better than VRRP, as it does not use keepalive or BFD messages to detect failures and allow the backup to take over.

In EVPN all-active multi-homing scenarios with R-VPLS services where the advertisement of the ARP/ND entries is enabled, use the following command to avoid issues with MAC mobility caused by the MAC/IP advertisement route for the ARP/ND entry being sent with ESI=0:

- **MD-CLI**

```
configure service vpls bgp-evpn routes mac-ip arp-nd-only-with-fdb-advertisement true
```

- **classic CLI**

```
configure service vpls bgp-evpn arp-nd-only-with-fdb-advertisement
```

When this command is enabled, the local ARP/ND entries of VPRN interfaces using this VPLS are advertised in this BGP-EVPN service only when the corresponding local MAC is programmed in the FDB.

In an EVPN multi-homing scenario, this command prevents the router from advertising a MAC/IP advertisement route with the MAC and IP binding but without the correct ESI value (which is taken only when the MAC is properly programmed in the FDB against the ESI).

In addition, if an Ethernet Segment SAP receives a frame, the MAC address can be re-programmed as type learned, even if the MAC was previously programmed as type EVPN.

### 5.3.6 EVPN-MPLS routed VPLS multicast routing support

In an EVPN-MPLS VPRN routed VPLS, IPv4 multicast routing is supported through its IP interface, when the source of the multicast stream is on one side of its IP interface and the receivers are on either side of the IP interface. For example, the source for multicast stream G1 could be on the IP side sending to receivers on both other regular IP interfaces and the VPLS of the routed VPLS service, while the source for group G2 could be on the VPLS side sending to receivers on both the VPLS and IP side of the routed VPLS service.

See [IPv4 and IPv6 multicast routing support](#) for more information.

### 5.3.7 IGMP snooping in EVPN-MPLS

IGMP snooping is supported in EVPN-MPLS VPLS services. It is also supported in EVPN-MPLS VPRN and IES R-VPLS services. It is required in scenarios where the operator does not want to flood all of the IP multicast traffic to the access nodes or CEs, and only wants to deliver IP multicast traffic for which IGMP reports have been received.

The following points apply when IGMP snooping is configured in EVPN-MPLS VPLS services:

- IGMP snooping is enabled using the **configure service vpls igmp-snooping no shutdown** command.
- Queries and reports received on SAP or SDP bindings are snooped and properly handled; they are sent to SAP or SDP bindings as expected.
- Queries and reports on EVPN-MPLS destinations are handled as follows.
  - If received from SAP or SDP bindings, the queries and reports are sent to all EVPN-MPLS destinations, regardless of whether the service is using an ingress replication or mLDP provider tunnel.
  - If received on an EVPN-MPLS destination, the queries and reports are processed and propagated to access SAP or SDP bindings, regardless of whether the service is using an ingress replication or mLDP provider tunnel.
  - EVPN-MPLS destinations are is treated as a single IGMP snooping interface and is always added as an **mrouter**.
  - The debug trace output displays one copy of messages being sent to all EVPN-MPLS destinations (the trace does not show a copy for each destination) and displays messages received from all EVPN-MPLS destinations as coming from a single EVPN-MPLS interface.



**Note:** When IGMP snooping is enabled with P2MP LSPs, at least one EVPN-MPLS multicast destination must be established to enable the processing of IGMP messages by the system. The use of P2MP LSPs is not supported when sending IPv4 multicast into an EVPN-MPLS R-VPLS service from its IP interface.

In the following show command output, the EVPN-MPLS destinations are shown as part of the MFIB (when **igmp-snooping** is in a **no shutdown** state), and the EVPN-MPLS logical interface is shown as an **mrouter**.

```
*A:PE-2# show service id 2000 mfib
=====
Multicast FIB, Service 2000
=====
Source Address  Group Address          SAP or SDP Id          Svc Id  Fwd
```

```

-----
*          *          eMpls:192.0.2.3:262132      Local  Fwd
                  eMpls:192.0.2.4:262136      Local  Fwd
                  eMpls:192.0.2.5:262131      Local  Fwd
-----
Number of entries: 1
=====
*A:PE-2# show service id 2000 igmp-snooping base
=====
IGMP Snooping Base info for service 2000
=====
Admin State : Up
Querier      : 10.0.0.3 on evpn-mpls
-----
SAP or SDP          Oper MRtr Pim  Send Max   Max Max   MVR      Num
Id                  Stat Port Port Qrys Grps  Srcs Grp  From-VPLS Grps
                  Srcs
-----
sap:1/1/1:2000      Up   No   No   No   None  None None  Local    0
evpn-mpls           Up   Yes  N/A  N/A  N/A   N/A  N/A  N/A      N/A
=====

*A:PE-4# show service id 2000 igmp-snooping mrouter
=====
IGMP Snooping Multicast Routers for service 2000
=====
MRouter          SAP or SDP Id          Up Time          Expires          Version
-----
10.0.0.3         evpn-mpls          0d 00:38:49      175s             3
-----
Number of mrouter: 1
=====

```

### 5.3.7.1 Data-driven IGMP snooping synchronization with EVPN multihoming

When single-active multihoming is used, the IGMP snooping state is learned on the active multihoming object. If a failover occurs, the system with the newly active multihoming object must wait for IGMP messages to be received to instantiate the IGMP snooping state after the ES activation timer expires; this could result in an increased outage.

The outage can be reduced by using MCS synchronization, which is supported for IGMP snooping in both EVPN-MPLS and PBB-EVPN services (see [Multichassis synchronization for Layer 2 snooping states](#)). However, MCS only supports synchronization between two PEs, whereas EVPN multihoming is supported between a maximum of four PEs. Also, IGMP snooping state can be synchronized only on a SAP.

An increased outage would also occur when using all-active EVPN multihoming. The IGMP snooping state on an ES LAG SAP or virtual ES to the attached CE must be synchronized between all the ES PEs, as the LAG link used by the DF PE may not be the same as that used by the attached CE. MCS synchronization is not applicable to all-active multihoming as MCS only supports active/standby synchronization.

To eliminate any additional outage on a multihoming failover, IGMP snooping messages can be synchronized between the PEs on an ES using data-driven IGMP snooping state synchronization, which is supported in EVPN-MPLS services, PBB-EVPN services, EVPN-MPLS VPRN and IES R-VPLS services. The IGMP messages received on an ES SAP or spoke SDP are sent to the peer ES PEs with an ESI label

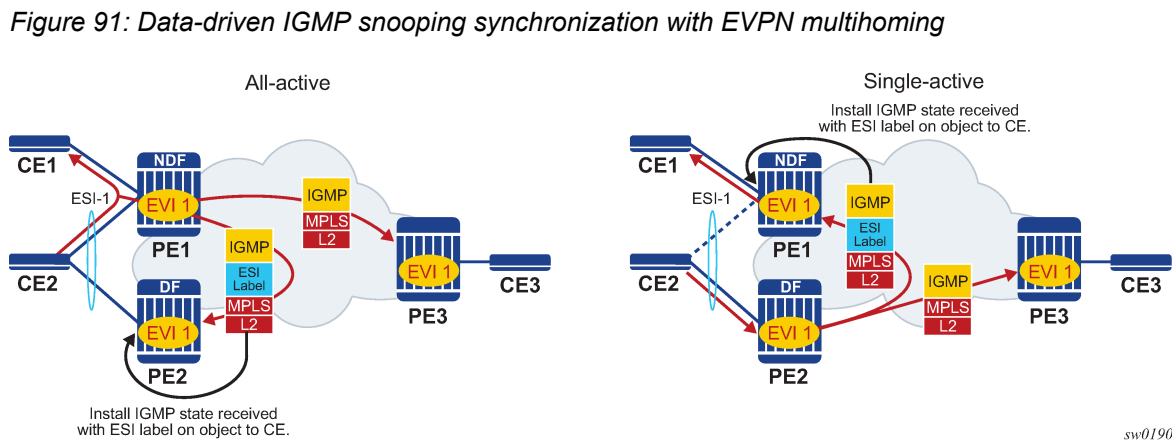


(for EVPN-MPLS) or ES B-MAC (for PBB-EVPN) and these are used to synchronize the IGMP snooping state on the ES SAP or spoke SDP on the receiving PE.

Data-driven IGMP snooping state synchronization is supported for both all-active multihoming and single-active with an ESI label multihoming in EVPN-MPLS, EVPN-MPLS VPRN and IES R-VPLS services, and for all-active multihoming in PBB-EVPN services. All PEs participating in a multihomed ES must be running an SR OS version supporting this capability. PBB-EVPN with IGMP snooping using single-active multihoming is not supported.

Data-driven IGMP snooping state synchronization is also supported with P2MP mLDP LSPs in both EVPN-MPLS and PBB-EVPN services. When P2MP mLDP LSPs are used in EVPN-MPLS services, all PEs (including the PEs not connected to a multihomed ES) in the EVPN-MPLS service must be running an SR OS version supporting this capability with IGMP snooping enabled and all network interfaces must be configured on FP3 or higher-based line cards.

**Figure 91: Data-driven IGMP snooping synchronization with EVPN multihoming** shows the processing of an IGMP message for EVPN-MPLS. In PBB-EVPN services, the ES B-MAC is used instead of the ESI label to synchronize the state.



Data-driven synchronization is enabled by default when IGMP snooping is enabled within an EVPN-MPLS service using all-active multihoming or single-active with an ESI label multihoming, or in a PBB-EVPN service using all-active multihoming. If IGMP snooping MCS synchronization is enabled on an EVPN-MPLS or PBB-EVPN (I-VPLS) multihoming SAP then MCS synchronization takes precedence over the data-driven synchronization and the MCS information is used. Mixing data-driven and MCS IGMP synchronization within the same ES is not supported.

When using EVPN-MPLS, the ES should be configured as **non-revertive** to avoid an outage when a PE takes over the DF role. The Ethernet A-D per ESI route update is withdrawn when the ES is down which prevents state synchronization to the PE with the ES down, as it does not advertise an ESI label. The lack of state synchronization means that if the ES comes up and that PE becomes DF after the ES activation timer expires, it may not have any IGMP snooping state until the next IGMP messages are received, potentially resulting in an additional outage. Configuring the ES as **non-revertive** can avoid this potential outage. Configuring the ES to be **non-revertive** would also avoid an outage when PBB-EVPN is used, but there is no outage related to the lack of the ESI label as it is not used in PBB-EVPN.

The following steps can be used when enabling IGMP snooping in EVPN-MPLS and PBB-EVPN services:

1. Upgrade SR OS on all ES PEs to a version supporting data-driven IGMP snooping synchronization with EVPN multihoming.

2. Enable IGMP snooping in the required services on all ES PEs. Traffic loss occurs until all ES PEs have IGMP snooping enabled and the first set of join/query messages are processed by the ES PEs.



**Note:** There is no action required on the non-ES PEs.

If P2MP mLDP LSPs are also configured, the following steps can be used when enabling IGMP snooping in EVPN-MPLS and PBB-EVPN services:

1. Upgrade SR OS on all PEs (both ES and non-ES) to a version supporting data-driven IGMP snooping synchronization with EVPN multihoming.
2. Enable IGMP snooping in EVPN-MPLS and PBB-EVPN services.
  - Perform the following steps for EVPN-MPLS:
    - Enable IGMP snooping on all non-ES PEs. Traffic loss occurs until the first set of join/query messages are processed by the non-ES PEs.
    - Then enable IGMP snooping on all ES PEs. Traffic loss occurs until all PEs have IGMP snooping enabled and the first set of join/query messages are processed by the ES PEs.
  - Perform the following steps for PBB-EVPN:
    - Enable IGMP snooping on all ES PEs. Traffic loss occurs until all PEs have IGMP snooping enabled and the first set of join/query messages are processed by the ES PEs.
    - There is no action required on the non-ES PEs.

To aid with troubleshooting, the debug packet output displays the IGMP packets used for the snooping state synchronization. An example of a join sent on ES esi-1 from one ES PE and the same join received on another ES PE follows.

```
6 2017/06/16 18:00:07.819 PDT MINOR: DEBUG #2001 Base IGMP
"IGMP: TX packet on svc 1
  from chaddr 5e:00:00:16:d8:2e
  send towards ES:esi-1
  Port   : evpn-mpls
  SrcIp  : 0.0.0.0
  DstIp  : 239.0.0.22
  Type   : V3 REPORT
    Num Group Records: 1
      Group Record Type: MODE_IS_EXCL (2), AuxDataLen 0, Num Sources 0
      Group Addr: 239.0.0.1

4 2017/06/16 18:00:07.820 PDT MINOR: DEBUG #2001 Base IGMP
"IGMP: RX packet on svc 1
  from chaddr d8:2e:ff:00:01:41
  received via evpn-mpls on ES:esi-1
  Port   : sap lag-1:1
  SrcIp  : 0.0.0.0
  DstIp  : 239.0.0.22
  Type   : V3 REPORT
    Num Group Records: 1
      Group Record Type: MODE_IS_EXCL (2), AuxDataLen 0, Num Sources 0
      Group Addr: 239.0.0.1
```

### 5.3.8 PIM snooping for IPv4 in EVPN-MPLS and PBB-EVPN services

**PIM snooping for VPLS** allows a VPLS PE router to build multicast states by snooping PIM protocol packets that are sent over the VPLS. The VPLS PE then forwards multicast traffic based on the multicast states. When all receivers in a VPLS are IP multicast routers running PIM, multicast forwarding in the VPLS is efficient when PIM snooping for VPLS is enabled.

PIM snooping for IPv4 is supported in EVPN-MPLS (for VPLS and R-VPLS) and PBB-EVPN I-VPLS (where BGP EVPN is running in the associated B-VPLS service) services. It is enabled using the following command (as IPv4 multicast is enabled by default):

```
configure service vpls <service-id> pim-snooping
```

PIM snooping on SAPs and spoke SDPs operates in the same way as in a plain VPLS service. However, EVPN-MPLS/PBB-EVPN B-VPLS destinations are treated as a single PIM interface, specifically:

- Hellos and join/prune messages from SAPs or SDPs are always sent to all EVPN-MPLS or PBB-EVPN B-VPLS destinations.
- As soon as a hello message is received from one PIM neighbor on an EVPN-MPLS or PBB-EVPN I-VPLS destination, then the single interface representing all EVPN-MPLS or PBB-EVPN I-VPLS destinations has that PIM neighbor.
- The EVPN-MPLS or PBB-EVPN B-VPLS destination split horizon logic ensures that IP multicast traffic and PIM messages received on an EVPN-MPLS or PBB-EVPN B-VPLS destination are not forwarded back to other EVPN-MPLS or PBB-EVPN B-VPLS destinations.
- The debug trace output displays one copy of messages being sent to all EVPN-MPLS or PBB-EVPN B-VPLS destinations (the trace does not show a copy for each destination) and displays messages received from all EVPN-MPLS or PBB-EVPN B-VPLS destinations as coming from a single EVPN-MPLS interface.

PIM snooping for IPv4 is supported in EVPN-MPLS services using P2MP LSPs and PBB-EVPN I-VPLS services with P2MP LSPs in the associated B-VPLS service. When PIM snooping is enabled with P2MP LSPs, at least one EVPN-MPLS multicast destination is required to be established to enable the processing of PIM messages by the system.

Multichassis synchronization (MCS) of PIM snooping for IPv4 state is supported for both SAPs and spoke SDPs which can be used with single-active multihoming. Care should be taken when using \*.null to define the range for a QinQ virtual ES if the associated SAPs are also being synchronized by MCS, as there is no equivalent MCS sync-tag support to the \*.null range.

PBB-EVPN services operate in a similar way to regular PBB services, specifically:

- The multicast flooding between the I-VPLS and the B-VPLS works in a similar way as for PIM snooping for IPv4 with an I-VPLS using a regular B-VPLS. The first PIM join message received over the local B-VPLS from a B-VPLS SAP or SDP or EVPN destination adds all of the B-VPLS SAP or SDP or EVPN components into the related multicast forwarding table associated with that I-VPLS context. The multicast packets are forwarded throughout the B-VPLS on the per ISID single tree.
- When a PIM router is connected to a remote I-VPLS instance over the B-VPLS infrastructure, its location is identified by the B-VPLS SAP, SDP or by the set of all EVPN destinations on which its PIM hellos are received. The location is also identified by the source B-MAC address used in the PBB header for the PIM hello message (this is the B-MAC associated with the B-VPLS instance on the remote PBB PE).

In EVPN-MPLS services, the individual EVPN-MPLS destinations appear in the MFIB but the information for each EVPN-MPLS destination entry is always identical, as shown below:

```
*A:PE# show service id 1 mfib
=====
Multicast FIB, Service 1
=====
Source Address  Group Address          Port Id                      Svc Id  Fwd
Blk
-----
*                239.252.0.1          sap:1/1/9:1                 Local   Fwd
                                     eMpls:1.1.1.2:262141        Local   Fwd
                                     eMpls:1.1.1.3:262141        Local   Fwd
-----
Number of entries: 1
=====
*A:PE#
```

Similarly for the PIM neighbors:

```
*A:PE# show service id 1 pim-snooping neighbor
=====
PIM Snooping Neighbors ipv4
=====
Port Id          Nbr DR Prty    Up Time          Expiry Time      Hold Time
Nbr Address
-----
SAP:1/1/9:1      1              0d 00:08:17      0d 00:01:29      105
10.0.0.1
EVPN-MPLS        1              0d 00:27:26      0d 00:01:19      105
10.0.0.2
EVPN-MPLS        1              0d 00:27:26      0d 00:01:19      105
10.0.0.3
-----
Neighbors : 3
=====
*A:PE#
```

A single EVPN-MPLS interface is shown in the outgoing interface, as can be seen in the following output:

```
*A:PE# show service id 1 pim-snooping group detail
=====
PIM Snooping Source Group ipv4
=====
Group Address    : 239.252.0.1
Source Address   : *
Up Time          : 0d 00:07:07
Up JP State      : Joined           Up JP Expiry       : 0d 00:00:37
Up JP Rpt        : Not Joined StarG Up JP Rpt Override : 0d 00:00:00
RPF Neighbor     : 10.0.0.1
Incoming Intf    : SAP:1/1/9:1
Outgoing Intf List : EVPN-MPLS, SAP:1/1/9:1
Forwarded Packets : 0              Forwarded Octets    : 0
-----
Groups : 1
=====
*A:PE#
```

An example of the debug trace output for a join received on an EVPN-MPLS destination is shown below:

```
A:PE1# debug service id 1 pim-snooping packet jp
```

```

A:PE1#
32 2016/12/20 14:21:22.68 CET MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: Join/Prune
[000 02:16:02.460] PIM-RX ifId 1071394 ifName EVPN-MPLS 10.0.0.3 -> 224.0.0.13
Length: 34
PIM Version: 2 Msg Type: Join/Prune Checksum: 0xd3eb
Upstream Nbr IP : 10.0.0.1 Resvd: 0x0, Num Groups 1, HoldTime 210
  Group: 239.252.0.1/32 Num Joined Srcs: 1, Num Pruned Srcs: 0
    Joined Srcs:
      10.0.0.1/32 Flag SWR  <*,G>

```

The equivalent output for PBB-EVPN services is similar to that above for EVPN-MPLS services, with the exception that the EVPN destinations are named "b-EVPN-MPLS".

### 5.3.8.1 Data-driven PIM snooping for IPv4 synchronization with EVPN multihoming

When single-active multihoming is used, PIM snooping for IPv4 state is learned on the active multihoming object. If a failover occurs, the system with the newly active multihoming object must wait for IPv4 PIM messages to be received to instantiate the PIM snooping for IPv4 state after the ES activation timer expires, which could result in an increased outage.

This outage can be reduced by using MCS synchronization, which is supported for PIM snooping for IPv4 in both EVPN-MPLS and PBB-EVPN services (see [Multichassis synchronization for Layer 2 snooping states](#)). However, MCS only supports synchronization between two PEs, whereas EVPN multihoming is supported between a maximum of four PEs.

An increased outage would also occur when using all-active EVPN multihoming. The PIM snooping for IPv4 state on an all-active ES LAG SAP or virtual ES to the attached CE must be synchronized between all the ES PEs, as the LAG link used by the DF PE may not be the same as that used by the attached CE. MCS synchronization is not applicable to all-active multihoming as MCS only supports active/standby synchronization.

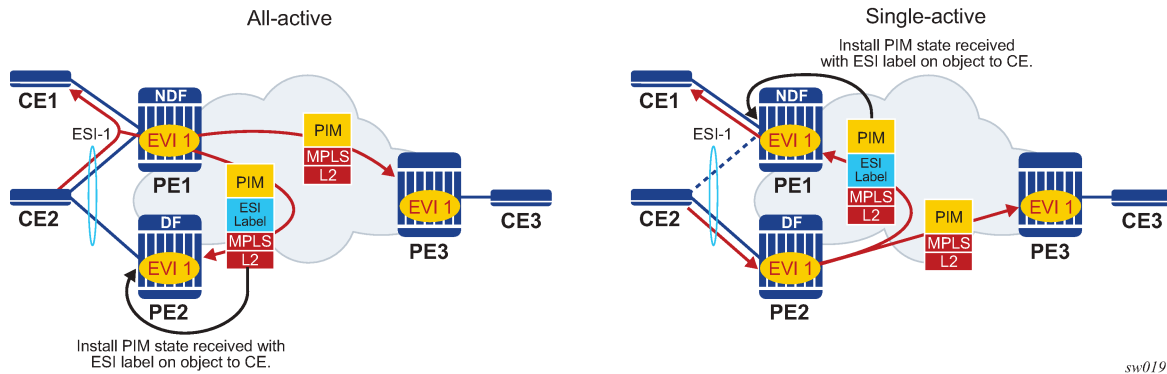
To eliminate any additional outage on a multihoming failover, snooped IPv4 PIM messages should be synchronized between the PEs on an ES using data-driven PIM snooping for IPv4 state synchronization, which is supported in both EVPN-MPLS and PBB-EVPN services. The IPv4 PIM messages received on an ES SAP or spoke SDP are sent to the peer ES PEs with an ESI label (for EVPN-MPLS) or ES B-MAC (for PBB-EVPN) and are used to synchronize the PIM snooping for IPv4 state on the ES SAP or spoke SDP on the receiving PE.

Data-driven PIM snooping state synchronization is supported for all-active multihoming and single-active with an ESI label multihoming in EVPN-MPLS services. All PEs participating in a multihomed ES must be running an SR OS version supporting this capability with PIM snooping for IPv4 enabled. It is also supported with P2MP mLDP LSPs in the EVPN-MPLS services, in which case all PEs (including the PEs not connected to a multihomed ES) must have PIM snooping for IPv4 enabled and all network interfaces must be configured on FP3 or higher-based line cards.

In addition, data-driven PIM snooping state synchronization is supported for all-active multihoming in PBB-EVPN services and with P2MP mLDP LSPs in PBB-EVPN services. All PEs participating in a multihomed ES, and all PEs using PIM proxy mode (including the PEs not connected to a multihomed ES) in the PBB-EVPN service must be running an SR OS version supporting this capability and must have PIM snooping for IPv4 enabled. PBB-EVPN with PIM snooping for IPv4 using single-active multihoming is not supported.

[Figure 92: Data-driven PIM snooping for IPv4 synchronization with EVPN multihoming](#) shows the processing of an IPv4 PIM message for EVPN-MPLS. In PBB-EVPN services, the ES B-MAC is used instead of the ESI label to synchronize the state.

Figure 92: Data-driven PIM snooping for IPv4 synchronization with EVPN multihoming



Data-driven synchronization is enabled by default when PIM snooping for IPv4 is enabled within an EVPN-MPLS service using all-active multihoming and single-active with an ESI label multihoming, or in a PBB-EVPN service using all-active multihoming. If PIM snooping for IPv4 MCS synchronization is enabled on an EVPN-MPLS or PBB-EVPN (I-VPLS) multihoming SAP or spoke SDP, then MCS synchronization takes preference over the data-driven synchronization and the MCS information is used. Mixing data-driven and MCS PIM synchronization within the same ES is not supported.

When using EVPN-MPLS, the ES should be configured as **non-revertive** to avoid an outage when a PE takes over the DF role. The Ethernet A-D per ESI route update is withdrawn when the ES is down, which prevents state synchronization to the PE with the ES down as it does not advertise an ESI label. The lack of state synchronization means that if the ES comes up and that PE becomes DF after the ES activation timer expires, it may not have any PIM snooping for IPv4 state until the next PIM messages are received, potentially resulting in an additional outage. Configuring the ES as **non-revertive** can avoid this potential outage. Configuring the ES to be **non-revertive** would also avoid an outage when PBB-EVPN is used, but there is no outage related to the lack of the ESI label as it is not used in PBB-EVPN.

The following steps can be used when enabling PIM snooping for IPv4 (using PIM snooping and PIM proxy modes) in EVPN-MPLS and PBB-EVPN services:

- PIM snooping mode
  1. Upgrade SR OS on all ES PEs to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
  2. Enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by the ES PEs.



**Note:** There is no action required on the non-ES PEs.

- PIM proxy mode
  - EVPN-MPLS
    1. Upgrade SR OS on all ES PEs to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
    2. Enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by the ES PEs.



**Note:** There is no action required on the non-ES PEs.

– PBB-EVPN

1. Upgrade SR OS on all PEs (both ES and non-ES) to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
2. Enable PIM snooping for IPv4 on all non-ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by each non-ES PE.
3. Enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by the ES PEs.

If P2MP mLDP LSPs are also configured, the following steps can be used when enabling PIM snooping or IPv4 (using PIM snooping and PIM proxy modes) in EVPN-MPLS and PBB-EVPN services.

• PIM snooping mode

1. Upgrade SR OS on all PEs (both ES and non-ES) to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
2. Then enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping enabled and the first set of join/hello messages are processed by the ES PEs.



**Note:** There is no action required on the non-ES PEs.

• PIM proxy mode

1. Upgrade SR OS on all PEs (both ES and non-ES) to a version supporting data-driven PIM snooping for IPv4 synchronization with EVPN multihoming.
2. Enable PIM snooping for IPv4 on all non-ES PEs. Traffic loss occurs until all PEs have PIM snooping for IPv4 enabled and the first set of join/hello messages are processed by each non-ES PE.
3. Enable PIM snooping for IPv4 on all ES PEs. Traffic loss occurs until all PEs have PIM snooping enabled and the first set of join/hello messages are processed by the ES PEs.

In the above steps, when PIM snooping for IPv4 is enabled, the traffic loss can be reduced or eliminated by configuring a larger hold-time (up to 300 seconds), during which multicast traffic is flooded.

To aid with troubleshooting, the debug packet output displays the PIM packets used for the snooping state synchronization. An example of a join sent on ES esi-1 from one ES PE and the same join received on another ES PE follows:

```
6 2017/06/16 17:36:37.144 PDT MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: pimVplsFwdJPToEvpn
Forwarding to remote peer on bgp-evpn ethernet-segment esi-1"
7 2017/06/16 17:36:37.144 PDT MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: Join/Prune
[000 00:19:37.040] PIM-TX ifId 1071394 ifName EVPN-MPLS-ES:esi-1 10.0.0.10 -> 22
10.0.0.13 Length: 34
PIM Version: 2 Msg Type: Join/Prune Checksum: 0xd2de
Upstream Nbr IP : 10.0.0.1 Resvd: 0x0, Num Groups 1, HoldTime 210
Group: 239.0.0.10/32 Num Joined Srcs: 1, Num Pruned Srcs: 0
Joined Srcs:
10.0.0.1/32 Flag SWR <*,G>
```



```

4 2017/06/16 17:36:37.144 PDT MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: pimProcessPdu
Received from remote peer on bgp-evpn ethernet-segment esi-1, will be applied on
lag-1:1
"
5 2017/06/16 17:36:37.144 PDT MINOR: DEBUG #2001 Base PIM[vpls 1 ]
"PIM[vpls 1 ]: Join/Prune
[000 00:19:30.740] PIM-RX ifId 1071394 ifName EVPN-MPLS-ES:esi-1 10.0.0.10 -> 22
10.0.0.13 Length: 34
PIM Version: 2 Msg Type: Join/Prune Checksum: 0xd2de
Upstream Nbr IP : 10.0.0.1 Resvd: 0x0, Num Groups 1, HoldTime 210
Group: 239.0.0.10/32 Num Joined Srcs: 1, Num Pruned Srcs: 0
Joined Srcs:
10.0.0.1/32 Flag SWR <*,G>

```

### 5.3.9 MPLS EL and hash label

The router supports the MPLS EL (RFC 6790) for EVPN VPLS and Epipe services. This label allows LSR nodes in a network to load-balance labeled packets with more granularity than by hashing on the standard label stack.



**Note:** The 7705 SAR Gen 2 does not support the hash label on EVPN-VPLS or EVPN-VPWS (Epipe) services.

To configure insertion of the EL on a BGP-EVPN VPLS or Epipe, use the **entropy-label** command in the **bgp-evpn>mpls** context. Use the **entropy-label** command under the **spoke-sdp** context to configure insertion of the EL on spoke SDPs bound to a BGP-EVPN VPLS. The EL is only inserted if the far end of the MPLS tunnel is also EL-capable. For more information, see the *7705 SAR Gen 2 MPLS Guide*.

### 5.3.10 Inter-AS Option B and Next-Hop-Self Route-Reflector for EVPN-MPLS

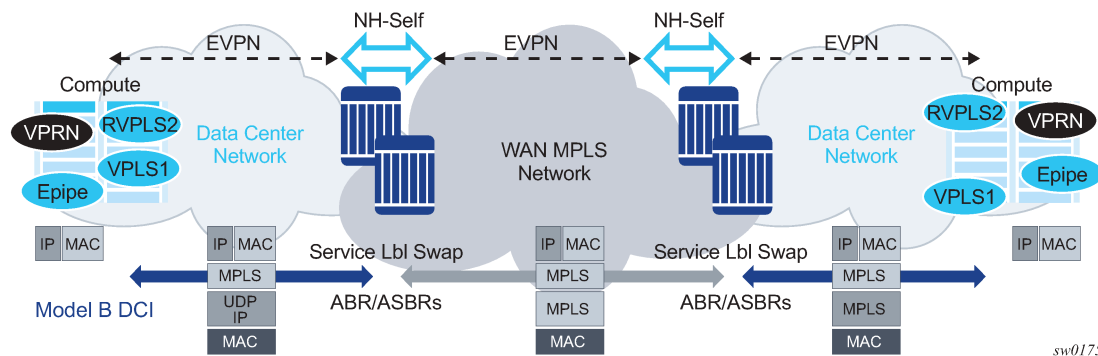
Inter-AS Option B and Next-Hop-Self Route-Reflector (VPN-NH-RR) functions are supported for the BGP-EVPN family in the same way both functions are supported for IP-VPN families.

A typical use case for EVPN Inter-AS Option B or EVPN VPN-NH-RR is Data Center Interconnect (DCI) networks, where cloud and service providers are looking for efficient ways to extend their Layer 2 and Layer 3 tenant services beyond the data center and provide a tighter DC-WAN integration. While the instantiation of EVPN services in the DGW to provide this DCI connectivity is a common model, some operators use Inter-AS Option B or VPN-NH-RR connectivity to allow the DGW to function as an ASBR or ABR respectively, and the services are only instantiated on the edge devices.

**Figure 93: EVPN inter-AS Option B or VPN-NH-RR model** shows a DCI example where the EVPN services in two DCs are interconnected without the need for instantiating services on the DC GWs.



Figure 93: EVPN inter-AS Option B or VPN-NH-RR model



The ASBRs or ABRs connect the DC to the WAN at the control plane and data plane levels where the following considerations apply:

- From a control plane perspective, the ASBRs or ABRs perform the following tasks:
  - accept EVPN-MPLS routes from a BGP peer  
EVPN-VXLAN routes are not supported.
  - extract the MPLS label from the EVPN NLRI or attribute and program a label swap operation on the IOM
  - re-advertise the EVPN-MPLS route to the BGP peer in the other Autonomous Systems (ASs) or IGP domains  
The re-advertised route has a Next-Hop-Self and a new label encoded for those routes that came with a label.
- From a data plan perspective, the ASBRs and ABRs terminate the ingress transport tunnel, perform an EVPN label swap operation, and send the packets on to an interface (if E-BGP is used) or a new tunnel (if IBGP is used).
- The ASBR or ABR resolves the EVPN routes based on the existing **bgp next-hop-resolution** command for **family vpn**, where **vpn** refers to EVPN, VPN-IPv4, and VPN-IPv6 families.

```
*A:ABR-1# configure router bgp next-hop-resolution labeled-routes transport-tunnel
family vpn resolution-filter
- resolution-filter
[no] bgp          - Use BGP tunnelling for next hop resolution
[no] ldp          - Use LDP tunnelling for next hop resolution
[no] rsvp         - Use RSVP tunnelling for next hop resolution
[no] sr-isis      - Use sr-isis tunnelling for next hop resolution
[no] sr-ospf      - Use sr-ospf for next hop resolution
[no] sr-te        - Use sr-te for next hop resolution
[no] udp          - Use udp for next hop resolution
```

For more information about the next-hop resolution of BGP-labeled routes, see the *7705 SAR Gen 2 Unicast Routing Protocols Guide*

Inter-AS Option B for EVPN services on ABRs and VPN-NH-RR on ABRs re-use the existing commands **enable-inter-as-vpn** and **enable-rr-vpn-forwarding** respectively. The two commands enable the ASBR or ABR function for both EVPN and IP-VPN routes. These two features can be used with the following EVPN services:

- EVPN-MPLS Epipe services (EVPN-VPWS)

- EVPN-MPLS VPLS services
- EVPN-MPLS R-VPLS services
- PBB-EVPN and PBB-EVPN E-Tree services
- EVPN-MPLS E-Tree services
- PE and ABR functions (EVPN services and **enable-rr-vpn-forwarding**), which are both supported on the same router
- PE and ASBR functions (EVPN services and **enable-inter-as-vpn**), which are both supported on the same router

The following sub-sections clarify some aspects of EVPN when used in an Inter-AS Option B or VPN-NH-RR network.

### 5.3.10.1 Inter-AS Option B and VPN-NH-RR procedures on EVPN routes

When **enable-rr-vpn-forwarding** or **enable-inter-as-vpn** is configured, only EVPN-MPLS routes are processed for label swap and the next hop is changed. EVPN-VXLAN routes are re-advertised without a change in the next hop.

The following shows how the router processes and re-advertises the different EVPN route types. For more information about the route fields, see the [BGP-EVPN control plane for MPLS tunnels](#) Guide.

- **Auto-discovery (AD) routes (type 1)**

For AD per EVI routes, the MPLS label is extracted from the route NLRI. The route is re-advertised with Next-Hop-Self (NHS) and a new label. No modifications are made for the remaining attributes.

For AD per ES routes, the MPLS label in the NLRI is zero. The route is re-advertised with NHS and the MPLS label remains zero. No modifications are made for the remaining attributes.

- **MAC/IP routes (type 2)**

The MPLS label (Label-1) is extracted from the NLRI. The route is re-advertised with NHS and a new Label-1. No modifications are made for the remaining attributes.

- **Inclusive Multicast Ethernet Tag (IMET) routes (type 3)**

Because there is no MPLS label present in the NLRI, the MPLS label is extracted from the PMSI Tunnel Attribute (PTA) if needed, and the route is then re-advertised with NHS, with the following considerations:

- For IMET routes with tunnel-type Ingress Replication, the router extracts the IR label from the PTA. The router programs the label swap and re-advertises the route with a new label in the PTA.
- For tunnel-type P2MP mLDP, the router re-advertises the route with NHS. No label is extracted; therefore, no swap operation occurs.
- For tunnel-type Composite, the IR label is extracted from the PTA, the swap operation is programmed and the route re-advertised with NHS. A new label is encoded in the PTA's IR label with no other changes in the remaining fields.
- For tunnel-type AR, the routes are always considered VXLAN routes and are re-advertised with the next-hop unchanged.

- **Ethernet-Segment (ES) routes (type 4)**

Because ES routes do not contain an MPLS label, the route is re-advertised with NHS and no modifications to the remaining attributes. Although an ASBR or ABR re-advertises ES routes, EVPN multihoming for ES PEs located in different ASs or IGMP domains is not supported.

- **IP-Prefix routes (type 5)**

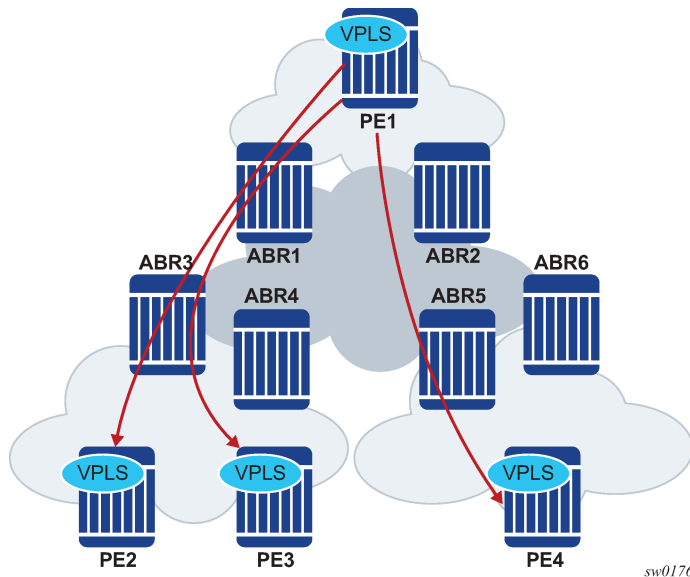
The MPLS label is extracted from the NLRI and the route is re-advertised with NHS and a new label. No modifications are made to the remaining attributes.

### 5.3.10.2 BUM traffic in inter-AS Option B and VPN-NH-RR networks

Inter-AS Option B and VPN-NH-RR support the use of non-segmented trees for forwarding BUM traffic in EVPN.

For ingress replication and non-segmented trees, the ASBR or ABR performs an EVPN BUM label swap without any aggregation or further replication. This concept is shown in [Figure 94: VPN-NH-RR and ingress replication for BUM traffic](#).

*Figure 94: VPN-NH-RR and ingress replication for BUM traffic*



In [Figure 94: VPN-NH-RR and ingress replication for BUM traffic](#), when PE2, PE3, and PE4 advertise their IMET routes, the ABRs re-advertise the routes with NHS and a different label. However, IMET routes are not aggregated; therefore, PE1 sets up three different EVPN multicast destinations and sends three copies of every BUM packet, even if they are sent to the same ABR. This example is also applicable to ASBRs and Inter-AS Option B.

P2MP mLDP may also be used with VPN-NH-RR, but not with Inter-AS Option B. The ABRs, however, do not aggregate or change the mLDP root IP addresses in the IMET routes. The root IP addresses must be leaked across IGP domains. For example, if PE2 advertises an IMET route with mLDP or composite tunnel type, PE1 is able to join the mLDP tree if the root IP is leaked into PE1's IGP domain.

### 5.3.11 ECMP for EVPN-MPLS destinations

ECMP is supported for EVPN route next hops that are resolved to EVPN-MPLS destinations as follows:

- **ECMP for Layer 2 unicast traffic on Epipe and VPLS services for EVPN-MPLS destinations**

This is enabled by the **configure service epipe bgp-evpn mpls auto-bind-tunnel ecmp number** and **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** commands and allows the resolution of an EVPN-MPLS next hop to a group of ECMP tunnels of type RSVP-TE, SR-TE or BGP.

- **ECMP for Layer 3 unicast traffic on R-VPLS services with EVPN-MPLS destinations**

This is enabled by the **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** and **configure service vpls allow-ip-int-bind evpn-mpls-ecmp** commands.

The VPRN unicast traffic (IPv4 and IPv6) is sprayed among "m" paths, with "m" being the lowest value of (16,n), where "n" is the number of ECMP paths configured in the **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** command.

CPM originated traffic is not sprayed and picks up the first tunnel in the set.

This feature is limited to FP3 and above systems.

- **ECMP for Layer 3 multicast traffic on R-VPLS services with EVPN-MPLS destinations**

This is enabled by the **configure service vpls allow-ip-int-bind ip-multicast-ecmp** and **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** commands. The VPRN multicast traffic (IPv4 and IPv6) are sprayed among up to "m" paths, with "m" being the lowest value of (16,n), and "n" is the number of ECMP paths configured in the **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp** command.

In all of these cases, the **configure service epipe bgp-evpn mpls auto-bind-tunnel ecmp number** and **configure service vpls bgp-evpn mpls auto-bind-tunnel ecmp number** commands determine the number of Traffic Engineering (TE) tunnels that an EVPN next hop can resolved to. TE tunnels refer to RSVP-TE or SR-TE types. For shortest path tunnels, such as, ldp, sr-isis, sr-ospf, udp, and so on, the number of tunnels in the ECMP group are determined by the **configure router ecmp** command.

Weighted ECMP for Layer 2 unicast traffic on Epipe and VPLS services for EVPN-MPLS destinations is supported. Packets are sprayed across the LSPs according to the outcome of the hash algorithm and the configured load balancing weight of each LSP when both:

- the Epipe or VPLS service directly uses an ECMP set of RSVP or SR-TE LSPs with the **configure router mpls lsp load-balancing-weight** command configured
- the **configure service epipe bgp-evpn mpls auto-bind-tunnel weighted-ecmp** or **configure service vpls bgp-evpn mpls auto-bind-tunnel weighted-ecmp** commands are configured

If the service uses a BGP tunnel which uses an ECMP set of RSVP or SR-TE LSPs with a load-balancing-weight configured, the router performs weighted ECMP regardless of the setting of **weighted-ecmp** under the **auto-bind-tunnel** context.

### 5.3.12 IPv6 tunnel resolution for EVPN MPLS services

EVPN MPLS services can be deployed in a pure IPv6 network infrastructure, where IPv6 addresses are used as next hops of the advertised EVPN routes, and EVPN routes received with IPv6 next hops are resolved to tunnels in the IPv6 tunnel table.

To change the default **system-ipv4** address that is advertised as the next hop for a local EVPN MPLS service, configure the **route-next-hop {system-ipv4 | system-ipv6 | ip-address}** CLI command using the **config service vpls bgp-evpn mpls route-next-hop {system-ipv4 | system-ipv6 | ip-address}** or the **config service epipe bgp-evpn mpls route-next-hop {system-ipv4 | system-ipv6 | ip-address}** context.

The configured IP address is used as a next hop for the MAC/IP, IMET, and AD per-EVI routes advertised for the service. This configured next hop can be overridden by a policy using the **next-hop-self** command.

In the case of Inter-AS model B or next-hop-self route-reflector scenarios, at the ASBR/ABR the following guidelines apply.

- A route received with an IPv4 next hop can be readvertised to a neighbor with an IPv6 next hop. The **advertise-ipv6-next-hops evpn** command must be configured on that neighbor.
- A route received with an IPv6 next hop can be readvertised to a neighbor with an IPv4 next hop. The **no advertise-ipv6-next-hops evpn** command must be configured on that neighbor.

## 5.4 General EVPN topics

This section provides information about general topics related to EVPN.

### 5.4.1 ARP/ND snooping and proxy support

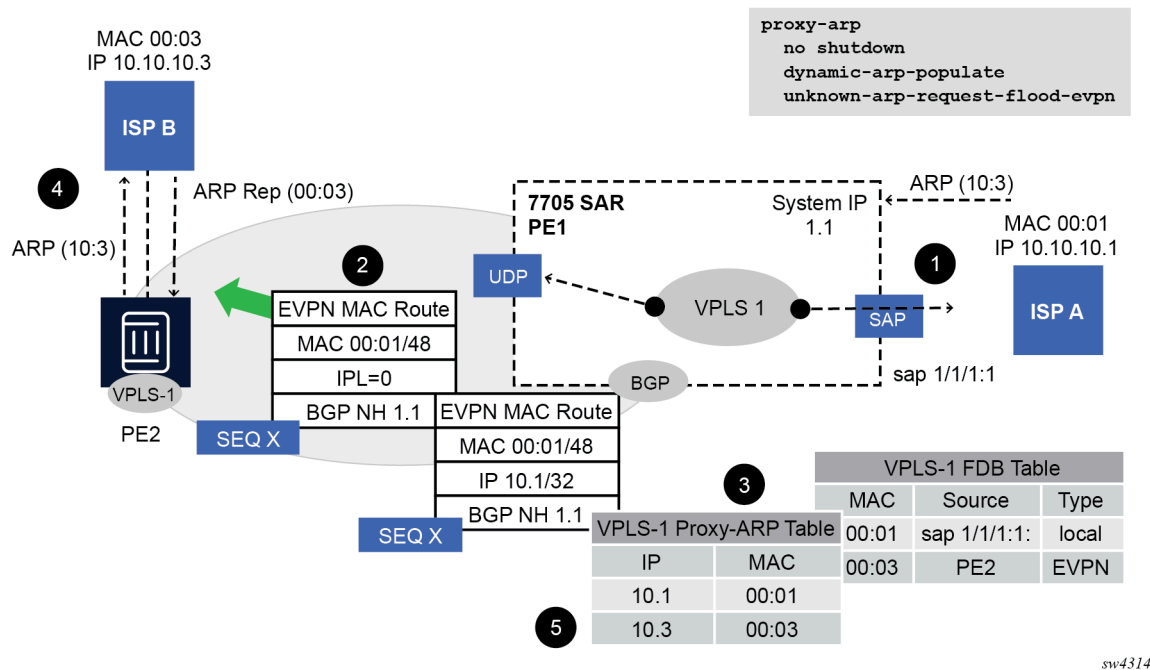
VPLS services support proxy-ARP (Address Resolution Protocol) and proxy-ND (Neighbor Discovery) functions that can be enabled or disabled independently per service. When enabled (proxy-ARP/proxy-ND **no shutdown**), the system populates the corresponding proxy-ARP/proxy-ND table with IP-MAC entries learned from the following sources:

- EVPN-received IP-MAC entries
- User-configured static IP-MAC entries
- Snooped dynamic IP-MAC entries (learned from ARP/GARP/NA messages received on local SAPs/SDP bindings)

In addition, any ingress ARP or ND frame on a SAP or SDP binding is intercepted and processed. ARP requests and Neighbor Solicitations are answered by the system if the requested IP address is present in the proxy table.

[Figure 95: Proxy-ARP example usage in an EVPN network](#) shows an example of how proxy-ARP is used in an EVPN network. Proxy-ND would work in a similar way. The MAC address notation in the diagram is shortened for readability.

Figure 95: Proxy-ARP example usage in an EVPN network



PE1 is configured as follows:

```
*A:PE1>config>service>vpls# info
-----
vxlan instance 1 vni 600 create
exit
bgp
  route-distinguisher 192.0.2.71:600
  route-target export target:64500:600 import target:64500:600
exit
bgp-evpn
  vxlan bgp 1 vxlan-instance 1
  no shutdown
exit
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 95: Proxy-ARP example usage in an EVPN network shows the following steps, assuming proxy-ARP is no shutdown on PE1 and PE2, and the tables are empty:

1. ISP-A sends ARP-request for (10.10.)10.3.

2. PE1 learns the MAC 00:01 in the FDB as usual and advertises it in EVPN without any IP. Optionally, the MAC can be configured as a CStatic mac, in which case it is advertised as protected. If the MAC is learned on a SAP or SDP binding where **auto-learn-mac-protect** is enabled, the MAC is also advertised as protected.
3. The ARP-request is sent to the CPM where:
  - An ARP entry (IP 10.1'MAC 00:01) is populated into the proxy-ARP table.
  - EVPN advertises MAC 00:01 and IP 10.1 in EVPN with the same SEQ number and Protected bit as the previous route-type 2 for MAC 00:01.
  - A GARP is also issued to other SAPs/SDP bindings (assuming they are not in the same split horizon group as the source). If **garp-flood-evpn** is enabled, the GARP message is also sent to the EVPN network.
  - The original ARP-request can still be flooded to the EVPN or not based on the **unknown-arp-request-flood-evpn** command.
4. Assuming PE1 was configured with **unknown-arp-request-flood-evpn**, the ARP-request is flooded to PE2 and delivered to ISP-B. ISP-B replies with its MAC in the ARP-reply. The ARP-reply is finally delivered to ISP-A.
5. PE2 learns MAC 00:01 in the FDB and the entry 10.1'00:01 in the proxy-ARP table, based on the EVPN advertisements.
6. When ISP-B replies with its MAC in the ARP-reply:
  - MAC 00:03 is learned in FDB at PE2 and advertised in EVPN.
  - MAC 00:03 and IP 10.3 are learned in the proxy-ARP table and advertised in EVPN with the same SEQ number as the previous MAC route.
  - ARP-reply is unicasted to MAC 00:01.
7. EVPN advertisements are used to populate PE1's FDB (MAC 00:03) and proxy-ARP (IP 10.3—>MAC 00:03) tables as mentioned in 5.

From this point onward, the PEs reply to any ARP-request for 00:01 or 00:03, without the need for flooding the message in the EVPN network. By replying to known ARP-requests / Neighbor Solicitations, the PEs help to significantly reduce the flooding in the network.

Use the following commands to customize proxy-ARP/proxy-ND behavior:

- **dynamic-arp-populate** and **dynamic-nd-populate**  
Enables the addition of dynamic entries to the proxy-ARP or proxy-ND table (disabled by default). When executed, the system populates proxy-ARP/proxy-ND entries from snooped GARP/ARP/NA messages on SAPs/SDP bindings in addition to the entries coming from EVPN (if EVPN is enabled). These entries are shown as **dynamic**.
- **static <IPv4-address> <mac-address>** and **static <IPv4-address> <mac-address> and static <ipv6-address> <mac-address> {host | router}**  
Configures static entries to be added to the table.  
 **Note:** A static IP-MAC entry requires the addition of the MAC address to the FDB as either learned or CStatic (conditional static mac) to become active (**Status** —> active).
- **age-time <60 to 86400>** (seconds)



Specifies the aging timer per proxy-ARP/proxy-ND entry. When the aging expires, the entry is flushed. The age is reset when a new ARP/GARP/NA for the same IP MAC is received.

- **send-refresh <120 to 86400>** (seconds)

If enabled, the system sends ARP-request/Neighbor Solicitation messages at the configured time, so that the owner of the IP can reply and therefore refresh its IP MAC (proxy-ARP entry) and MAC (FDB entry).

- **table-size [1 to 16384]**

Enables the user to limit the number of entries learned on a specified service. By default, the table-size limit is 250.

Use the following commands to configure whether unknown ARP-requests, NS, or the unsolicited GARPs and NA messages are flooded in an EVPN network:

– **MD-CLI**

```
configure service vpls proxy-arp evpn flood unknown-arp-req
configure service vpls proxy-arp evpn flood gratuitous-arp
configure service vpls proxy-nd evpn flood unknown-neighbor-solicitation
configure service vpls proxy-nd evpn flood unknown-neighbor-advertise-host
configure service vpls proxy-nd evpn flood unknown-neighbor-advertise-router
```

– **classic CLI**

```
configure service vpls proxy-arp unknown-arp-request-flood-evpn
configure service vpls proxy-arp garp-flood-evpn
configure service vpls proxy-nd unknown-ns-flood-evpn
configure service vpls proxy-nd host-unsolicited-na-flood-evpn
configure service vpls proxy-nd router-unsolicited-na-flood-evpn
```

The preceding commands control whether ARP/ND messages are flooded to EVPN destinations only. That is, when flooding (for a specific message type) is configured to be blocked when sending to EVPN, it is still flooded to other local SAPs or SDP-binds. Use the following commands if the desired behavior is to block flooding in the service, even to local SAPs and SDP-binds:

– **MD-CLI**

```
configure service vpls proxy-arp flood received-unknown-arp-req
configure service vpls proxy-arp flood received-gratuitous-arp
configure service vpls proxy-nd flood received-unknown-neighbor-solicitation
configure service vpls proxy-nd flood received-unknown-neighbor-advertise-host
configure service vpls proxy-nd flood received-unknown-neighbor-advertise-router
```

– **classic CLI**

```
configure service vpls proxy-arp received-unknown-arp-request-flood
configure service vpls proxy-arp received-garp-flood
configure service vpls proxy-nd received-unknown-ns-flood
configure service vpls proxy-nd received-host-unsolicited-na-flood
configure service vpls proxy-nd received-router-unsolicited-na-flood
```

- **dup-detect [anti-spoof-mac <mac-address>] window <minutes> num-moves <count> hold-down <minutes> | max>**

Enables a mechanism that detects duplicate IPs and ARP/ND spoofing attacks. The working of the **dup-detect** command can be summarized as follows:



- Attempts (relevant to dynamic and EVPN entry types) to add the same IP (different MAC) are monitored for <window> minutes and when <count> is reached within that **window**, the proxy-ARP/proxy-ND entry for the IP is suspected and marked as **duplicate**. An alarm is also triggered.
- The condition is cleared when hold-down time expires (**max** does not expire) or a **clear** command is issued.
- If the **anti-spoof-mac** is configured, the proxy-ARP/proxy-ND offending entry's MAC is replaced by this <mac-address> and advertised in an unsolicited GARP/NA for local SAP or SDP bindings and in EVPN to remote PEs.
- This mechanism assumes that the same **anti-spoof-mac** is configured in all the PEs for the same service and that traffic with destination **anti-spoof-mac** received on SAPs/SDP bindings are dropped. An ingress MAC filter has to be configured to drop traffic to the **anti-spoof-mac**.

**Table 15: Proxy-arp entry combinations** shows the combinations that produce a **Status = Active** proxy-arp entry in the table. The system replies to proxy-ARP requests for active entries. Any other combination results in a **Status = inActiv** entry. If the service is not active, the proxy-arp entries are not active either, regardless of the FDB entries



**Note:** A static entry is active in the FDB even when the service is down.

*Table 15: Proxy-arp entry combinations*

| Proxy-arp entry type | FDB entry type (for the same MAC)              |
|----------------------|--|
| Dynamic              | learned  |
| Static               | learned  |
| Dynamic              | CStatic/Static                                 |
| Static               | CStatic/Static                                 |
| EVPN                 | EVPN, learned/CStatic/Static with matching ESI |
| Duplicate            | —  |

When proxy-ARP/proxy-ND is enabled on services with all-active multihomed Ethernet Segments, a proxy-arp entry type **evpn** may be associated with learned/CStatic/Static FDB entries (because for example, the CE can send traffic for the same MAC to all the multihomed PEs in the ES). If this is the case, the entry is active if the ESI of the EVPN route and the FDB entry match, or inactive otherwise, as per [Table 15: Proxy-arp entry combinations](#).

#### 5.4.1.1 Proxy-ARP/ND periodic refresh, unsolicited refresh and confirm-messages

When proxy-ARP/proxy-ND is enabled, the system starts populating the proxy table and responding to ARP-requests/NS messages. To keep the active IP→MAC entries alive and ensure that all the host/routers in the service update their ARP/ND caches, the system may generate the following three types of ARP/ND messages for a specified IP→MAC entry:

- **periodic refresh messages (ARP-requests or NS for a specified IP)**

These messages are activated by the **send-refresh** command and their objective is to keep the existing FDB and Proxy-ARP/ND entries alive, to minimize EVPN withdrawals and re-advertisements.

- **unsolicited refresh messages (unsolicited GARP or NA messages)**  
These messages are sent by the system when a new entry is learned or updated. Their objective is to update the attached host/router caches.
- **confirm messages (unicast ARP-requests or unicast NS messages)**  
These messages are sent by the system when a new MAC is learned for an existing IP. The objective of the confirm messages is to verify that a specified IP has really moved to a different part of the network and is associated with the new MAC. If the IP has not moved, it forces the owners of the duplicate IP to reply and cause **dup-detect** to kick in.

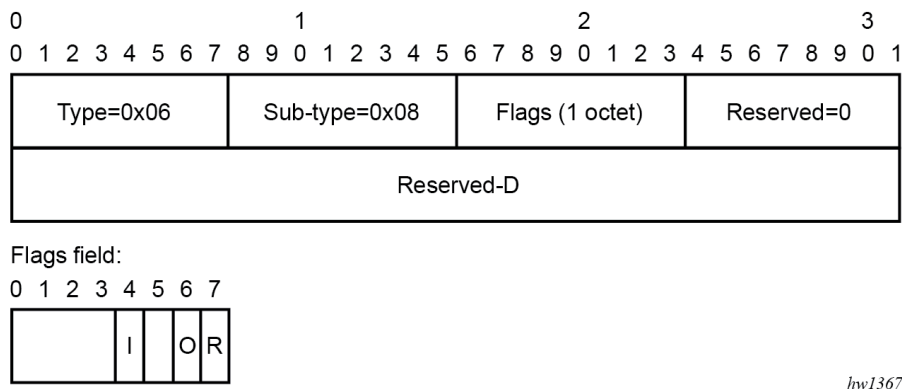
5.4.1.2 Advertisement of Proxy-ARP/ND flags in EVPN

When a dynamic or static Proxy-ARP/ND entry is learned (or configured), a few property flags are created along with it. Those flags are:

- The Router Flag (R) - used in IPv6 Neighbor Advertisement messages to indicate if the proxy-ND entry belongs to an IPv6 router or an IPv6 host.
- The Override Flag (O) - used in IPv6 Neighbor Advertisement messages to indicate whether the resolved entry should override a potential ND entry that the solicitor may already have for the same IPv6 address.
- The Immutable Flag (I) - used to indicate that the proxy-ARP/ND entry cannot change its binding to a different MAC addresses. This Flag is always set for static proxy-ARP/ND entries or configured dynamic IP addresses that are associated with a **mac-list**.

In order for the ingress and egress EVPN PEs to install the proxy-ARP/ND entries with the same property flags, RFC 9047 describes how those flags (R, O and I) can be conveyed in the EVPN ARP/ND extended community advertised along with the EVPN MAC/IP Advertisement routes. The format of the EVPN ARP/ND extended community follows:

Figure 96: Format of EVPN ARP/ND extended community



By default, the router does not advertise the ARP/ND extended community. Use the following command to configure the router to advertise all the proxy ARP/ND MAC/IP Advertisement routes with the extended community:

```
configure service vpls bgp-evpn arp-nd-extended-community
```

### 5.4.1.3 Proxy-ARP/ND and flag processing

#### Proxy-ND and the Router Flag

RFC 4861 describes the use of the (R) or Router flag in NA messages as follows:

- A node capable of routing IPv6 packets must reply to NS messages with NA messages where the R flag is set (R=1).
- Hosts must reply with NA messages where R=0.

The R flag in NA messages impacts how the hosts select their default gateways when sending packets off-link. The proxy-ND function on the router does one of the following, depending on whether it can provide the appropriate R flag information:

- provides the appropriate R flag information in the proxy-ND NA replies, if possible
- floods the received NA messages, if it cannot provide the appropriate R flag when replying

The use of the R flag (only present in NA messages and not in NS messages) makes the procedure for learning proxy-ND entries and replying to NS messages different from the procedures for proxy-ARP in IPv4. The NA messages snooping determines the router or host flag to add to each entry, and that determines the flag to use when responding to an NS message.

The procedure to add the R flag to a specified entry is as follows:

- Dynamic entries are learned based on received NA messages. The R flag is also learned and added to the proxy-ND entry so that the appropriate R flag is used in response to NS requests for a specified IP.
- Static entries are configured as host or router using the following command.

– **MD-CLI**

```
configure service vpls proxy-nd static-neighbor ip-address type
```

– **classic CLI**

```
configure service vpls proxy-nd static
```

- EVPN entries are learned from BGP and the following command determines the R flag added to them;

– **MD-CLI**

```
configure service vpls proxy-nd evpn advertise-neighbor-type
```

– **classic CLI**

```
configure service vpls proxy-nd evpn-nd-advertise
```

in case the following command is not configured (if configured, the signaled flag value determines the flag of the entry).

- **MD-CLI**

```
configure service vpls bgp-evpn routes mac-ip arp-nd-extended-community
```

- **classic CLI**

```
configure service vpls bgp-evpn arp-nd-extended-community-advertisement
```

- In addition, the EVPN ND advertisement indicates what static and dynamic IP → MAC entries the system advertises in EVPN.
  - If you specify the router option for EVPN ND advertisement, the system should flood the received unsolicited NA messages for hosts. This is controlled by the following command:

- **MD-CLI**

```
configure service vpls proxy-nd evpn flood unknown-neighbor-advertise-host
```

- **classic CLI**

```
configure service vpls proxy-nd host-unsolicited-na-flood-evpn
```

- The opposite is also true so that the host option for EVPN ND advertisement is configured with the following command:

- **MD-CLI**

```
configure service vpls proxy-nd evpn flood unknown-neighbor-advertise-router
```

- **classic CLI**

```
configure service vpls proxy-nd router-unsolicited-na-flood-evpn
```

- The router-host option for EVPN ND advertisement allows the router to advertise both types of entries in EVPN at the same time. That is, static and dynamic entries with the **router** or **host** flag are advertised in EVPN with the corresponding flag in the ARP/ND extended community. This option can be enabled only if the ARP/ND extended community is configured.

EVPN proxy-ND MAC/IP Advertisement routes received without the EVPN ARP/ND extended communities create an entry with type Router (which is the default value). Entries created as duplicate are advertised in EVPN with an R flag value that depends on the configuration of the EVPN ND advertisement command. If the **host** option is configured for the EVPN ND advertisement, the duplicate entry is treated as a host. If the **router** or **router-host** option is configured for the EVPN ND advertisement, the duplicate entry behaves as a router.

## Proxy-ARP/ND and the Immutable Flag

The I bit or Immutable flag in the ARP/ND extended community is advertised and used as follows:

- Any static proxy-ARP/ND entry is advertised with I=1 if you enable ARP/ND extended community advertisement.
- Any configured **dynamic** IP address (associated with a mac-list) proxy-ARP/ND entry is advertised with I=1 if you enable ARP/ND extended community
- Duplicate entries are advertised with I=1 as well (in addition to O=1 and R=0 or 1 based on the configuration).

- The setting of the I bit is independent of the static bit associated with the FDB entry, and it is only used with proxy-ARP/ND advertisements.

The I bit in the ARP/ND extended community is processed on reception as follows:

- A PE receiving an EVPN MAC/IP Advertisement route containing an IP-MAC and the I flag set, installs the IP-MAC entry in the ARP/ND or proxy-ARP/ND table as an immutable binding.
- This immutable binding entry overrides an existing non-immutable binding for the same IP-MAC. In general, the ARP/ND extended community command changes the selection of ARP/ND entries when multiple routes with the same IP address exist. This preferred order of ARP/ND entries selection is as follows:
  1. Local immutable ARP/ND entries (static and dynamic)
  2. EVPN immutable ARP/ND entries
  3. Remaining ARP/ND entries
- The absence of the EVPN ARP/ND Extended Community in a MAC/IP Advertisement route indicates that the IP→MAC entry is not an immutable binding.
- Receiving multiple EVPN MAC/IP Advertisement routes with the I flag set to 1 for the same IP but a different MAC address is considered a misconfiguration or a transient error condition. If this happens in the network, a PE receiving multiple routes (with the I flag set to 1 for the same IP and a different MAC address) selects one of them based on the previously described selection rules.

### Proxy-ND and the Override Flag

The O bit or Override flag in the ARP/ND extended community is advertised and used as follows:

- The O flag is learned for dynamic entries (being 0 or 1) and added to the proxy-ND table. If the ARP/ND extended community is configured, the O flag associated with the entry is advertised along with the EVPN MAC/IP Advertisement route. Static and duplicate entries are always advertised with O=1.
- Upon receiving an EVPN MAC/IP Advertisement route, the received O flag is stored in the entry created in the proxy-ND table, and used when replying to local NS messages for the IP address.

#### 5.4.1.4 Proxy-ARP/ND mac-List for dynamic entries

SR OS supports the association of configured MAC lists with a configured dynamic proxy-ARP or proxy-ND IP address. The actual proxy-ARP or proxy-ND entry is not created until an ARP or Neighbor Advertisement message is received for the IP and one of the MACs in the associated MAC-list. This is in accordance with IETF RFC 9161, which states that a proxy-ARP or proxy-ND IP entry can be associated with one MAC among a list of allowed MACs.

The following example shows the use of MAC lists for dynamic entries.

```
A:PE-2>config>service#
  proxy-arp-nd
    mac-list ISP-1 create
      mac 00:de:ad:be:ef:01
      mac 00:de:ad:be:ef:02
      mac 00:de:ad:be:ef:03

A:PE-2>config>service>vpls>proxy-arp#
  dynamic 1.1.1.1 create
    mac-list ISP-1
    resolve 30
```

```
A:PE-2>config>service>vpls>proxy-nd#
dynamic 2001:db8:1000::1 create
mac-list ISP-1
resolve 30
```

where:

- A dynamic IP (**dynamic ip create**) is configured and associated with a MAC list (**mac-list name**).
- The MAC list is created in the **config>service** context and can be reused by multiple configured dynamic IPs as follows:
  - in different services
  - in the same service, for proxy-ARP and proxy-ND entries
- If the MAC list is empty, the proxy-ARP or proxy-ND entry is not created for the configured IP.
- The same MAC list can be applied to multiple configured dynamic entries even within the same service.
- The new proxy-ARP and proxy-ND entries behave as dynamic entries and are displayed as type **dyn** in the **show** commands.

The following output example displays the entry corresponding to the configured dynamic IP.

```
show service id 1 proxy-arp detail
```

### Output example

```
-----
Proxy Arp
-----
Admin State       : enabled
Dyn Populate      : enabled
Age Time          : 900 secs
Table Size        : 250
Static Count      : 0
Dynamic Count     : 1
Dup Detect
-----
Detect Window     : 3 mins
Hold down         : 9 mins
Anti Spoof MAC    : None
EVPN
-----
Garp Flood        : enabled
Static Black Hole : disabled
Req Flood         : enabled
-----
=====
VPLS Proxy Arp Entries
=====
IP Address      Mac Address      Type Status      Last Update
-----
1.1.1.1         00:de:ad:be:ef:01  dyn active      02/23/2016 09:05:49
-----
Number of entries : 1
=====
```

```
show service proxy-arp-nd mac-list "ISP-1" associations
```

## Output example

```

=====
MAC List Associations
=====
Service Id          IP Addr
-----
1                   1.1.1.1
1                   2001:db8:1000::1
-----
Number of Entries: 2
=====

```

Although no new proxy-ARP or proxy-ND entries are created when a dynamic IP is configured, the router triggers the following resolve procedure:

1. The router sends a resolve message with a configurable frequency of 1 to 60 minutes; the default value is five minutes.



**Note:** The resolve message is an ARP-request or NS message flooded to all the non-EVPN endpoints in the service.

2. The router sends resolve messages at the configured frequency until a dynamic entry for the IP is created.



**Note:** The dynamic entry is created only if an ARP, GARP, or NA message is received for the configured IP, and the associated MAC belongs to the configured MAC list of the IP. If the MAC list is empty, the proxy-ARP or proxy-ND entry is not created for the configured IP.

After a dynamic entry (with a MAC address included in the list) is successfully created, its behavior (for send-refresh, age-time, and other activities) is the same as a configured dynamic entry with the following exceptions.

- Regular dynamic entries may override configured dynamic entries, but static or EVPN entries cannot override configured dynamic entries.
- If the corresponding MAC is flushed from the FDB after the entry is successfully created, the entry becomes inactive in the proxy-ARP or proxy-ND table and the resolve process is restarted.
- If the MAC list is changed, all the IPs that point to the list delete the proxy entries and the resolve process is restarted.
- If there is an existing configured dynamic entry and the router receives a GARP, ARP, or NA for the IP with a MAC that is not contained in the MAC list, the message is discarded and the proxy-ARP or proxy-ND entry is deleted. The resolve process is restarted.
- If there is an existing configured dynamic entry and the router receives a GARP, ARP, or NA for the IP with a MAC contained in the MAC list, the existing entry is overridden by the IP and new MAC, assuming the confirm procedure passes.
- The dup-detect and confirm procedures work for the configured dynamic entries when the MAC changes are between MACs in the MAC list. Changes to an off-list MAC cause the entry to be deleted and the resolve process is restarted.

Configured **dynamic** entries are advertised as immutable if you enable advertisement of ARP/ND extended community. The following considerations about IP duplication and immutable configured **dynamic** entries apply:

- The CPM drops received dynamic ARP/ND messages without learning them, if they match a dynamic (immutable) entry.
- If there is a local configured dynamic address (irrespective of whether there is an entry for it or not), a received EVPN immutable entry for the same IP address is not installed. Therefore the IP duplication mechanisms do not apply to immutable entries.

Configured **dynamic** entries also allow other non-configured dynamic entries to be learned in the proxy-ARP or proxy-ND table. If a more restrictive configuration is needed, add the following configuration command:

- The following command prevents the router from learning dynamic proxy-ARP or proxy-ND entries.

```
configure service vpls proxy-arp restrict-non-configured-ip-address
configure service vpls proxy-nd restrict-non-configured-ip-address
```

However, if the following command is used, the subsequent rules must be followed to create the proxy-ARP or proxy-ND entry:

– **MD-CLI**

```
configure service vpls proxy-arp dynamic-arp ip-address
configure service vpls proxy-nd dynamic-arp ip-address
```

– **classic CLI**

```
configure service vpls proxy-arp dynamic create
configure service vpls proxy-nd dynamic create
```

Adhere to the following rules when creating the proxy-ARP or proxy-ND entry:

- The MAC address of the entry must be associated with a MAC address contained in the MAC list (if configured).
- The following command checks that the reply for a configured dynamic IP address is coming on that SAP. Otherwise, the entry is not created:

– **MD-CLI**

```
configure service vpls proxy-arp dynamic-arp ip-address sap
configure service vpls proxy-nd dynamic-neighbor ip-address sap
```

– **classic CLI**

```
configure service vpls proxy-arp dynamic sap
configure service vpls proxy-nd dynamic sap
```

- In addition, the following optional command is supported so that the router can reply with a configured sponge MAC to requests for an unauthorized IP address. Only the configured dynamic-ARP or dynamic-neighbor IP addresses that pass the MAC list or the SAP ID check are considered authorized IP addresses.

```
configure service vpls proxy-arp restrict-non-configured-ip-address sponge-mac
configure service vpls proxy-nd restrict-non-configured-ip-address sponge-mac
```



## 5.4.2 BGP-EVPN MAC mobility

EVPN defines a mechanism to allow the smooth mobility of MAC addresses from an NVE to another NVE. The 7705 SAR Gen 2 supports this procedure as well as the MAC-mobility extended community in MAC advertisement routes as follows:

- The router honors and generates the SEQ 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 detect num-moves** value is reached).
- A SEQ number = zero or no MAC mobility *ext-community* are interpreted as sequence zero.
- In case of mobility, the following MAC selection procedure is performed.
  - If a PE has two or more active remote EVPN routes for the same MAC (VNI can be the same or different), the highest SEQ number is selected. The tie-breaker is the lowest IP (BGP NH IP).
  - If a PE has two or more active EVPN routes and it is the originator of one of them, the highest SEQ number is selected. The tie-breaker is the lowest IP (BGP NH IP of the remote route is compared to the local system address).



**Note:** When EVPN multihoming is used in EVPN-MPLS, the ESI is compared to determine whether a MAC received from two different PEs should be processed within the context of MAC mobility or multihoming. Two MAC routes that are associated with the same remote or local ESI but different PEs are considered reachable through all those PEs. Mobility procedures are not triggered if the MAC route still belongs to the same ESI.

## 5.4.3 BGP-EVPN MAC duplication

EVPN defines a mechanism to protect the EVPN service from control plane churn as a result of loops or accidental duplicated MAC addresses. The 7705 SAR Gen 2 supports an enhanced version of this procedure as described in this section.

In a scenario where two or more hosts are misconfigured using the same (duplicate) MAC address, the duplicate MAC address is learned by the PEs in the VPLS. As a result, the traffic originating from the hosts triggers continuous MAC moves among the PEs attached to the hosts. It is important to recognize such a situation and avoid incrementing the sequence number (in the MAC Mobility attribute) to infinity.

To remedy such situation, a router that detects a MAC mobility event by way of local learning starts a **window <in-minutes>** timer (default value of window = 3) and if it detects **num-moves <num>** before the timer expires (default value of num-moves = 5), it concludes that a duplicate MAC situation has occurred. The window and number of moves are configured using the following commands.

```
configure service vpls bgp-evpn mac-duplication detect window
configure service vpls bgp-evpn mac-duplication detect num-moves
```

The router then alerts the operator with a trap message when a duplicate MAC situation occurs.

```
10 2014/01/14 01:00:22.91 UTC MINOR: SVCNMR #2331 Base
"VPLS Service 1 has MAC(s) detected as duplicates by EVPN mac-
duplication detection."
```

Use the following command in the BGP EVPN Table section to display the offending MAC address.

```
show service id svc-id bgp-evpn
```

### Example: Displaying duplicated MAC addresses

```
=====
BGP EVPN Table
=====
EVI : 1000
Creation Origin : manual

Adv L2 Attributes : Disabled
Ignore Mtu Mismatch: Disabled

MAC/IP Routes
MAC Advertisement : Enabled           Unknown MAC Route : Disabled
CFM MAC Advertise : Disabled
ARP/ND Ext Comm Adv: Disabled

Multicast Routes
Sel Mcast Advert : Disabled
Ing Rep Inc McastAd: Enabled

IP Prefix Routes
IP Route Advert : Disabled

MAC Duplication Detection
Num. Moves : 5                       Window : 3
Retry : 9                           Number of Dup MACs : 1
Black Hole : Enabled
Local Learned Trusted MAC
MAC time : 1                         MAC move factor : 3

-----
Detected Duplicate MAC Addresses      Time Detected
-----
00:de:fe:ca:da:04                    05/18/2023 09:55:22
-----
=====

-----
Local Learned Trusted MAC
-----
MAC Address Time Detected
-----
-----
```

After a duplicate MAC is detected, the router stops sending and processing any BGP MAC advertisement routes for that MAC address until one of the following occurs.

1. The MAC is flushed because of a local event (SAP or SDP-binding associated with the MAC fails) or the reception of a remote update with better SEQ number (as a result of MAC flush at the remote router).
2. The **retry in-minutes** timer expires, which flushes the MAC and restart the process.



**Note:** The other routers in the VPLS instance forward the traffic for the duplicate MAC address to the router advertising the best route for the MAC.

The values of **num-moves** and window are configurable to allow for the required flexibility in different environments. In scenarios where BGP **configure router bgp rapid-update evpn** is configured, the operator may want to configure a shorter window timer than in scenarios where BGP updates are sent every (default) **min-route-advertisement** interval.

MAC duplication is always enabled in EVPN VPLS services. The preceding example shows the output for BGP-EVPN MAC duplication detection configuration per VPLS service under the following context.

```
configure service vpls bgp-evpn mac-duplication
```

The following example shows a MAC duplication detection configuration.

#### Example: MD-CLI

```
[ex:/configure service vpls "bd-1000-mac-dup-mpls" bgp-evpn mac-duplication]
A:admin@node-2# info detail
retry 9
detect {
    num-moves 5
    window 3
}
```

#### Example: classic CLI

```
A:node-2>config>service>vpls>bgp-evpn>mac-duplication# info detail
-----
detect num-moves 5 window 3 trusted-mac-move-factor 3
retry 9
```

### 5.4.4 Conditional static MAC and protection

RFC 7432, MAC mobility Extended Community section defines the use of the sticky bit to signal static MAC addresses. These addresses must be protected to prevent attempts to dynamically learn them in a different place in the EVPN-MPLS/VXLAN VPLS service.

Any conditional static MAC address that is defined in an EVPN-MPLS/VXLAN VPLS service is advertised by BGP-EVPN as a static address (that is, with the sticky bit set). Local static MACs or remote MACs with the sticky bit set are considered "protected". A packet entering a SAP/SDP-binding is discarded if its source MAC address matches a "protected" MAC.

```
A:node2config>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:node-2# 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
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric  : None
Connector    : None
Community    : target:65000:1000
Cluster      : No Cluster Members
Originator Id : None
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
IP Address    : ::
Mac Address   : 00:ca:ca:ca:ca:00
Neighbor-AS   : N/A
Source Class  : 0
Interface Name : NotAvailable
Aggregator    : None
MED           : 0
mac-mobility:Seq: 0/Static
Peer Router Id : 192.0.2.63
Tag           : 1063
RD            : 65063:1000
Mac Mobility   : Seq:0
Dest Class    : 0
-----
Routes : 1
=====

```

Local static MACs or remote MACs with sticky bit are considered as "protected". A packet entering a SAP / SDP binding is discarded if its source MAC address matches one of these 'protected' MACs.

### 5.4.5 Auto-learn MAC protect and restricting protected source MACs

Auto-learn MAC protect, together with the ability to restrict where the protected source MACs are allowed to enter the service, can be enabled within an EVPN-MPLS and EVPN-VXLAN VPLS and routed VPLS services, but not in PBB-EVPN services. The protection, using the **auto-learn-mac-protect** command (described in [Auto-learn MAC protect](#)), and the restrictions, using the **restrict-protected-src [discard-frame]** command, operate in the same way as in a non-EVPN VPLS service.

- When **auto-learn-mac-protect** is enabled on an object, source MAC addresses learned on that object are marked as protected within the FDB.
- When **restrict-protected-src** is enabled on an object and a protected source MAC is received on that object, the object is automatically shutdown (requiring the user to **shutdown** then **no shutdown** the object to make it operational again).
- When **restrict-protected-src discard-frame** is enabled on an object and a frame with a protected source MAC is received on that object, that frame is discarded.

In addition, the following behavioral differences are specific to EVPN services:

- An implicit **restrict-protected-src discard-frame** command is enabled by default on SAPs, mesh-SDPs and spoke SDPs. As this is the default, it is not possible to configure this command in an EVPN service. This default state can be seen in the show output for these objects, for example on a SAP:

```

*A:PE# show service id 1 sap 1/1/9:1 detail
=====
Service Access Points(SAP)
=====
Service Id      : 1
SAP             : 1/1/9:1
Encap           : q-tag
...
RestMacProtSrc Act : none (oper: Discard-frame)

```

- A **restrict-protected-src discard-frame** can be optionally enabled on EVPN-MPLS/VXLAN destinations within EVPN services. When enabled, frames that have a protected source MAC address are discarded if received on any EVPN-MPLS/VXLAN destination in this service, unless the MAC address is learned and protected on an EVPN-MPLS/VXLAN destination in this service. This is enabled as follows:

```
configure
service
  vpls <service id>
    bgp-evpn
      mpls bgp <instance>
        [no] restrict-protected-src discard-frame
      vxlan instance <instance> vni <vni-id>
        [no] restrict-protected-src discard-frame
```

- Auto-learned protected MACs are advertised to remote PEs in an EVPN MAC/IP advertisement route with the sticky bit set.
- The source MAC protection action relating to the **restrict-protected-src [discard-frame]** commands also applies to MAC addresses learned by receiving an EVPN MAC/IP advertisement route with the sticky bit set from remote PEs. This causes remotely configured conditional static MACs and auto-learned protected MACs to be protected locally.
- In all-active multihoming scenarios, if **auto-learn-mac-protect** is configured on all-active SAPs and **restrict-protected-src discard-frame** is enabled on EVPN-MPLS/VXLAN destinations, traffic from the CE that enters one multihoming PE and needs to be switched through the other multihoming PE is discarded on the second multihoming PE. Each multihoming PE protects the CE's MAC on its local all-active SAP, which results in any frames with the CE's MAC address as the source MAC being discarded as they are received on the EVPN-MPLS/VXLAN destination from the other multihoming PE.

Conditional static MACs, EVPN static MACs and locally protected MACs are marked as protected within the FDB, as shown in the example output.

```
*A:PE# show service fdb-mac
=====
Service Forwarding Database
=====
```

| ServId | MAC               | Source-Identifier      | Type<br>Age   | Last Change       |
|--------|-------------------|------------------------|---------------|-------------------|
| 1      | 00:00:00:00:00:01 | sap:1/1/9:1            | LP/30         | 01/05/16 11:58:22 |
| 1      | 00:00:00:00:00:02 | vxlan-1:<br>10.1.1.2:1 | EvpnS:P       | 01/05/16 11:58:23 |
| 1      | 00:00:00:00:01:01 | sap:1/1/9:1            | CStatic:<br>P | 01/04/16 20:05:02 |
| 1      | 00:00:00:00:01:02 | vxlan-1:<br>10.1.1.2:1 | EvpnS:P       | 01/04/16 20:18:02 |

```
-----
No. of Entries: 4
-----
Legend: L=Learned O=0am P=Protected-MAC C=Conditional S=Static
=====
```

In this output:

- the first MAC is locally protected using the **auto-learn-mac-protect** command
- the second MAC has been protected using the **auto-learn-mac-protect** command on a remote PE

- the third MAC is a locally configured conditional static MAC
- the fourth MAC is a remotely configured conditional static MAC

The command **auto-learn-mac-protect** can be optionally extended with an exclude-list by using the following command:

#### **auto-learn-mac-protect [exclude-list name]**

This list refers to a **mac-list <name>** created under the **config>service** context and contains a list of MACs and associated masks.

When **auto-learn-mac-protect [exclude-list name]** is configured on a service object, dynamically learned MACs are excluded from being learned as protected if they match a MAC entry in the MAC list. Dynamically learned MAC SAs are protected only if they are learned on an object with ALMP configured and one of the following conditions is true:

- there is no exclude list associated with the same object
- there is an exclude-list but the MAC does not match any entry

The MAC lists can be used in multiple objects of the same or different service. When empty, ALMP does not exclude any learned MAC from protection on the object. This extension allows the mobility of specific MACs in objects where MACs are learned as protected.

## 5.4.6 Blackhole MAC and its application to proxy-ARP/proxy-ND duplicate detection

A blackhole MAC is a local FDB record that is similar to a conditional static MAC. It is associated with a blackhole (similar to a VPRN blackhole static-route in VPRNs) instead of a SAP or SDP-binding.

Use the following syntax to configure a blackhole MAC.

```
config>service>vpls# static-mac mac
mac ieee-address [create] black-hole
```

The static blackhole MAC can have security applications (for example, replacement of MAC filters) for specific MACs. When used in combination with **restrict-protected-src**, the static blackhole MAC provides a simple and scalable way to filter MAC DA or SA in the data plane, regardless of whether the frame arrived at the system using SAP or SDP-bindings or EVPN endpoints.

For example, if a specified **static-mac mac 00:00:ca:fe:ca:fe create black-hole** is added to a service, the following behavior occurs.

- The configured MAC is created as a static MAC with a **black-hole** source identifier.

### Example

```
*A:PE1# show service id 1 fdb detail
=====
Forwarding Database, Service 1
=====
```

| ServId | MAC               | Source-Identifier                  | Type Age  | Last Change       |
|--------|-------------------|------------------------------------|-----------|-------------------|
| 1      | 00:ca:ca:ba:ca:01 | eES: 01:00:00:00:00:71:00:00:00:01 | Evpn      | 06/29/15 23:21:34 |
| 1      | 00:ca:ca:ba:ca:06 | eES: 01:74:13:00:74:13:00:00:74:13 | Evpn      | 06/29/15 23:21:34 |
| 1      | 00:ca:00:00:00:00 | sap:1/1/1:2                        | CStatic:P | 06/29/15 23:20:58 |
| 1      | 00:ca:fe:ca:fe:00 | black-hole                         | CStatic:P | 06/29/15 23:20:00 |

```

1          00:ca:fe:ca:fe:69 eMpls:          EvpnS:P    06/29/15 20:40:13
              192.0.2.69:262133
-----
No. of MAC Entries: 5
-----
Legend:  L=Learned O=Oam P=Protected-MAC C=Conditional S=Static
=====

```

- After it is successfully added to the FDB, the blackhole MAC is treated like any other protected MAC.
  - The blackhole MAC is added as protected (CStatic:P) and advertised in EVPN as static.
  - SAP or SDP-bindings or EVPN endpoints where the **restrict-protected-src discard-frame** is enabled discard frames where MAC SA is equal to blackhole MAC.
  - SAP or SDP-bindings where **restrict-protected-src (no discard-frame)** is enabled go operationally down if a frame with MAC SA is equal to blackhole MAC is received.
- After the blackhole MAC is successfully added to the FDB, any frame that arrives at any SAP, SDP-binding, or EVPN endpoint with MAC DA equal to blackhole MAC is discarded.

Blackhole MACs can also be used in services with proxy-ARP/ND enabled to filter traffic with destination to **anti-spoof-mac**. The **anti-spoof-mac** function provides a way to attract traffic to a specified IP when a duplicate condition is detected for that IP address. However, the system still needs to drop the traffic addressed to the **anti-spoof-mac** by using either a MAC filter or a blackhole MAC.

The user does not need to configure MAC filters when configuring a **static-black-hole** MAC address for the **anti-spoof-mac** function. To use a blackhole MAC entry for the **anti-spoof-mac** function in a proxy-ARP/ND service, configure the following options:

- the **static-black-hole** option for the **anti-spoof-mac**

### Example

```

*A:PE1# config>service>vpls>proxy-arp#
dup-detect window 3 num-moves 5 hold-down max anti-spoof-
mac 00:66:66:66:66:00 static-black-hole

```

- a static blackhole MAC using the same MAC address used for the **anti-spoof-mac**

### Example

```

*A:PE1# config>service>vpls#
static-mac mac 00:66:66:66:66:00 create black-hole

```

When this configuration is complete, the behavior of the **anti-spoof-mac** function changes as follows:

- In the EVPN, the MAC is advertised as Static. Locally, the MAC is shown in the FDB as "CStatic" and associated with a blackhole.
- The combination of the **anti-spoof-mac** and the **static-black-hole** ensures that any frame that arrives at the system with MAC DA = **anti-spoof-mac** is discarded, regardless of the ingress endpoint type (SAP, SDP-binding, or EVPN) and without the need for a filter.
- Instead of discarding traffic, if the user wants to redirect it using MAC DA as the **anti-spoof-mac**, then redirect filters should be configured on SAPs or SDP-bindings (instead of the **static-black-hole** option).

When the **static-black-hole** option is not configured with the **anti-spoof-mac**, the behavior of the **anti-spoof-mac** function, as described in [ARP/ND snooping and proxy support](#), remains unchanged, and the following applies:

- the **anti-spoof-mac** is not programmed in the FDB
- any attempt to add a Static MAC (or any other MAC) with the **anti-spoof-mac** value is rejected by the system
- a MAC filter is needed to discard traffic with MAC DA = **anti-spoof-mac**

### 5.4.7 Blackhole MAC for EVPN loop detection

SR OS can combine a blackhole MAC address concept and the EVPN MAC duplication procedures to provide loop protection in EVPN networks. The feature is compliant with the MAC mobility and multihoming functionality in RFC 7432, and the Loop Protection section in *draft-ietf-bess-rfc7432bis*. Use the following command to enable the feature:

- **MD-CLI**

```
configure service vpls bgp-evpn mac-duplication blackhole enable
```

- **classic CLI**

```
configure service vpls bgp-evpn mac-duplication black-hole-dup-mac
```

If enabled, there are no apparent changes in the MAC duplication; however, if a duplicated MAC is detected (for example, M1), then the router performs the following:

1. adds M1 to the duplicate MAC list
2. programs M1 in the FDB as a Protected MAC associated with a blackhole endpoint (where type is set to **EvpnD:P** and Source-Identifier is **black-hole**)

While the MAC type value remains **EvpnD:P**, the following additional operational details apply.

- Incoming frames with MAC DA = M1 are discarded by the ingress IOM, regardless of the ingress endpoint type (SAP, SDP, or EVPN), based on an FDB MAC lookup.
- Incoming frames with MAC SA = M1 are discarded by the ingress IOM or cause the router to bring down the SAP or SDP binding, depending on the **restrict-protected-src** setting on the SAP, SDP, or EVPN endpoint.

The following example shows an EVPN-MPLS service where blackhole is enabled and MAC duplication programs the duplicate MAC as a blackhole.

```
19 2016/12/20 19:45:59.69 UTC MINOR: SVCMGR #2331 Base
"VPLS Service 1000 has MAC(s) detected as duplicates by EVPN mac-duplication
detection."
```

#### Example: MD-CLI

```
[ex:/configure service vpls "bd-1000"]
A:admin@node-2# info
  admin-state enable
  service-id 1000
  customer "1"
  bgp 1 {
  }
  bgp-evpn {
    evi 1000
    mac-duplication {
      blackhole true
```



```

    detect {
        num-moves 5
        window 3
    }
}
mpls 1 {
    admin-state enable
    ingress-replication-bum-label true
    auto-bind-tunnel {
        resolution any
    }
}
}
sap 1/1/1:1000 {
}
spoke-sdp 56:1000 {
}

```

### Example: classic CLI

```

A:node-2# configure service vpls 1000
A:node-2>config>service>vpls# info
-----
    bgp
    exit
    bgp-evpn
        evi 1000
        mac-duplication
            detect num-moves 5 window 3
            retry 6
            black-hole-dup-mac
        exit
        mpls bgp 1
            ingress-replication-bum-label
            auto-bind-tunnel
                resolution any
            exit
            no shutdown
        exit
    exit
    stp
        shutdown
    exit
    sap 1/1/1:1000 create
        no shutdown
    exit
    spoke-sdp 56:1000 create
        no shutdown
    exit
    no shutdown
-----

```

### Example

The following command displays BGP EVPN table values.

```
show service id 1000 bgp-evpn
```

### Output example

```
=====
BGP EVPN Table
```

```

=====
EVI : 1000
Creation Origin : manual

Adv L2 Attributes : Disabled
Ignore Mtu Mismatch: Disabled

MAC/IP Routes
MAC Advertisement : Enabled Unknown MAC Route : Disabled
CFM MAC Advertise : Disabled
ARP/ND Ext Comm Adv: Disabled

Multicast Routes
Sel Mcast Advert : Disabled
Ing Rep Inc McastAd: Enabled

IP Prefix Routes
IP Route Advert : Disabled

MAC Duplication Detection
Num. Moves : 5 Window : 3
Retry : 9 Number of Dup MACs : 1
Black Hole : Enabled
Local Learned Trusted MAC
MAC time : 1 MAC move factor : 3

-----
Detected Duplicate MAC Addresses Time Detected
-----
00:de:fe:ca:da:04 05/18/2023 09:55:22
-----
=====

-----
Local Learned Trusted MAC
-----
MAC Address Time Detected
-----

=====
BGP EVPN MPLS Information
=====
Admin Status : Enabled                      Bgp Instance : 1
Force Vlan Fwding : Disabled
Force Qinq Fwding : none
Route NextHop Type : system-ipv4
Control Word : Disabled
Max Ecmp Routes : 1
Entropy Label : Disabled
Default Route Tag : none
Split Horizon Group: (Not Specified)
Ingress Rep BUM Lbl: Enabled
Ingress Ucast Lbl : 524262                  Ingress Mcast Lbl : 524261
RestProtSrcMacAct : none
Evpn Mpls Encap : Enabled                   Evpn MplsUdp : Disabled
Oper Group :
MH Mode : network
Evi 3-byte Auto-RT : Disabled
Dyn Egr Lbl Limit : Disabled
Hash Label : Disabled

```

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

=====
BGP EVPN MPLS Auto Bind Tunnel Information
=====
Allow-Flex-Algo-Fallback : false
Resolution : any                               Strict Tnl Tag : false
Max Ecmp Routes : 1
Bgp Instance : 1
Filter Tunnel Types : (Not Specified)
Weighted Ecmp : false
=====
=====
```

Example

The following command displays Forwarding Database details.


```
show service id 1000 fdb detail
```

Output example

```
=====
Forwarding Database, Service 1000
=====
ServId  MAC                Source-Identifier Type      Last Change
      Transport:Tnl-Id                Age
-----
1000  00:de:fe:da:da:04  black-hole      EvpnD:P  05/18/23 10:04:49
-----
No. of MAC Entries: 1
Legend:L=Learned O=0am P=Protected-MAC C=Conditional S=Static Lf=Leaf T=Trusted
=====
```

If the **retry** time expires, the MAC is flushed from the FDB and the process starts again. The following command clears the duplicate blackhole MAC address.

```
clear service id evpn mac-dup-detect
```



**Note:** The **clear service id 1000 fdb** command clears learned MAC addresses; blackhole MAC addresses are not cleared.

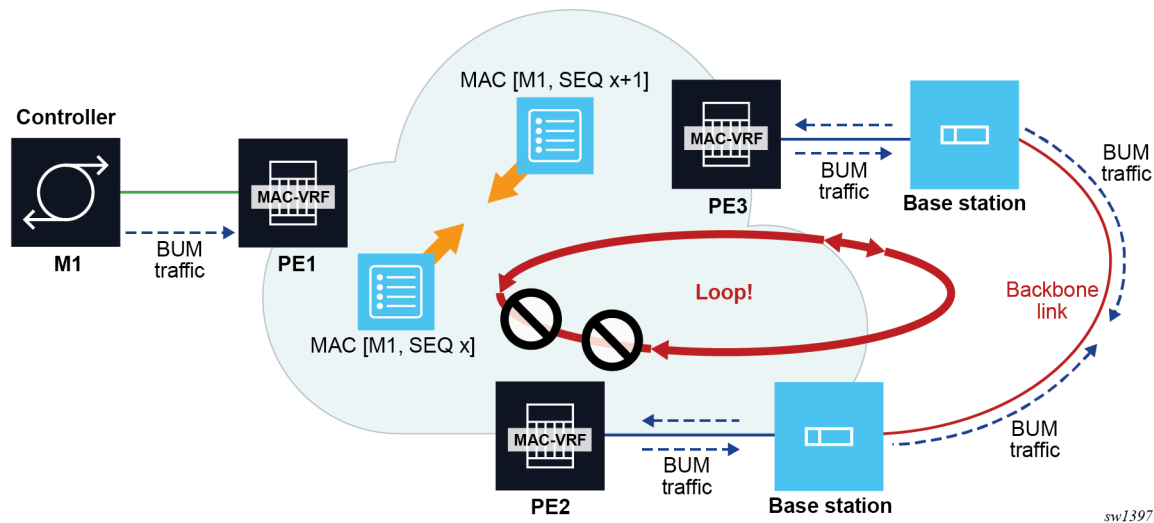
Support for the **blackhole enable** and **black-hole-dup-mac** commands and the preceding associated loop detection procedures is as follows:

- not supported on B-VPLS, I-VPLS, or M-VPLS services
- fully supported on EVPN-VXLAN VPLS/R-VPLS services, EVPN-MPLS VPLS services (including EVPN E-Tree) and EVPN-SRv6 VPLS services
- fully supported with EVPN MAC mobility and EVPN multihoming

### 5.4.7.1 Deterministic EVPN loop detection with trusted MACs

The EVPN loop detection procedure, described in the preceding section, is compliant with *draft-ietf-bess-rfc7432bis* and is an efficient way of detecting and blocking loops in EVPN networks. Contrary to other intrusive methods that inject Ethernet beacons into the customer network and detect loops depending on whether the beacon messages get back to the PEs, the EVPN loop detection mechanism is non intrusive because it relies entirely on learning the same MAC on different nodes. However, the mechanism lacks determinism, as shown in the following figure.

Figure 97: EVPN non-intrusive loop detection mechanism



Assume PE1, PE2 and PE3 are attached to the same EVPN VPLS service, and there is an accidental backdoor link between the Base Stations connected to PE2 and PE3. When the Controller with MAC M1 issues a broadcast frame, PE1 forwards it to PE2 and PE3, and the frame is looped back via the backdoor link. The mac-duplication procedure kicks in and M1 is detected as duplicate and turned into a blackhole MAC in the FDB, effectively solving the loop. However, the user does not know beforehand if M1 is blackholed in PE1, PE2, PE3 or multiple of them at the same time. If M1 is blackholed in PE1, this represents an issue for the hosts connected to other PEs (not shown) attached to the same service. Therefore, in the example, we want to influence the mac-duplication procedure so that M1 gets blackholed in PE2, PE3, or both, but not in PE1. To make the procedure more deterministic, the Trusted MAC concept is used.

A trusted MAC on a specific PE and VPLS service is dynamically learned and stays in the FDB as type learned without being flushed or a change in its type for a configurable number of minutes, as shown in the following command.

```
configure service vpls bgp-evpn mac-duplication trusted-mac-time
```

If the MAC moves from a SAP to another SAP in the same service and PE, the MAC does not reset its trusted MAC timer.

Trusted MACs are affected by the mac-duplication procedures in a different way from the non-trusted MACs. Trusted MACs require a different number of moves (during the mac-duplication window) to be

declared as duplicate, specified by **num-moves <number> \* trusted-mac-move-factor <number>** in the following context.

```
configure service vpls bgp-evpn mac-duplication detect
```

While non-trusted MACs are detected as duplicate after **num-moves**, trusted MACs need more moves to be declared as duplicate.

The following example shows the configuration of three PEs, as shown in [Figure 97: EVPN non-intrusive loop detection mechanism](#).

### Example: MD-CLI

```
// Applicable to PE1, PE2 and PE3

[ex:/configure service vpls "bd-1000" bgp-evpn mac-duplication]
A:admin@node-2# info
  blackhole true
  trusted-mac-time 5 // value 1..15, default: 5
  detect {
    num-moves 5
    window 3
    trusted-mac-move-factor 3 // value 1..10, default: 1
  }
```

### Example: classic CLI

```
// Applicable to PE1, PE2 and PE3

A:node-2>config>service>vpls>bgp-evpn>mac-duplication# info
-----
  detect num-moves 5 window 3 trusted-mac-move-factor 3 // value 1..10, default: 1
  black-hole-dup-mac
  trusted-mac-time 5 // value 1..15, default: 5
```

Based on the preceding configuration, recall the example described at the beginning of this section and assume M1 is a trusted MAC in PE1 (it has been dynamically learned for 5 minutes), then M1 requires 15 moves to be declared as duplicate (therefore a blackhole MAC) in PE1, whereas M1 only need 5 moves to be declared as duplicate in PE2 and PE3. This procedure guarantees that M1 does not get blackholed in the PE of its location (PE1).

The trusted MACs are shown in the following FDB show command example output with a "T" in the Type field.

```
show service id 1000 fdb detail
```

### Output example

```
=====
Forwarding Database, Service 1000
=====
ServId  MAC                Source-Identifrier  Type  Last Change
      Transport:Tnl-Id      Age
-----
1000    00:de:fe:da:da:04  sap:1/1/1:1000    LT/0   05/18/23 10:54:54
-----
No. of MAC Entries: 1
-----
```

Legend: L=Learned O=Oam P=Protected-MAC C=Conditional S=Static Lf=Leaf T=Trusted  
 =====

### 5.4.8 CFM interaction with EVPN services

Ethernet Connectivity and Fault Management (ETH-CFM) allows users to validate and measure Ethernet Layer 2 services using standard IEEE 802.1ag and ITU-T Y.1731 protocols. Each tool performs a unique function and adheres to that tool's specific PDU and frame format and the associate rules governing the transmission, interception, and process of the PDU. Detailed information describing the ETH-CFM architecture, the tools, and various functions is located in the various OAM and Diagnostics guides and is not repeated here.

EVPN provides powerful solution architectures. ETH-CFM is supported in Layer 2 EVPN architectures. Because the destination Layer 2 MAC address, unicast or multicast, is ETH-CFM tool dependent (that is, Ethernet continuity check (ETH-CC) is sent as a Layer 2 multicast and ETH-DM is sent as a Layer 2 unicast), the ETH-CFM function is allowed to multicast and broadcast to the virtual EVPN connections.

The MEP does not populate the local Layer 2 MAC address FDB with the MAC related to the MEP. 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 FDBs to learn the remote MEP Layer 2 MAC addresses, configure the **cfm-mac-advertisement** command under the **config>service>vpls>bgp-evpn** context. This allows the MEP Layer 2 IEEE MAC addresses to be exchanged with peers. This command tracks configuration changes and sends the required updates via the EVPN notification process related to a change.

Up MEP, Down MEP, and MIP creation is supported on the SAP, spoke, and mesh connections within the EVPN service. There is no support for the creation of ETH-CFM Management Points (MPs) on the virtual connection. VirtualMEP (vMEP) is supported with a VPLS context and the applicable EVPN Layer 2 VPLS solution architectures. The vMEP follows the same rules as the general MPs. When a vMEP is configured within the supported EVPN service, the ETH-CFM extraction routines are installed on the SAP, Binding, and EVPN connections within an EVPN VPLS Service. The vMEP extraction within the EVPN-PBB context requires the **vmep-extensions** parameter to install the extraction on the EVPN connections.

When MPs are used in combination with EVPN multihoming, observe the following considerations:

- behavior of operationally down MEPs on SAPs or SDP-bindings with EVPN multihoming, as follows:
  - **All-active multihoming**

No ETH-CFM is expected to be used in this case, because the two (or more) SAPs/SDP-bindings on the PEs are oper-up and active; however, the CE has a single LAG and responds as though it is connected to a single system. In addition, the **cfm-mac-advertisement** command can lead to traffic loops in all-active multihoming.
  - **Single-active multihoming**

Operationally down MEPs defined on single-active ES SAPs or SDP-bindings do not send CCMs when the PE is non-DF for the ES and fault-propagation is configured. For single-active multihoming, the behavior is equivalent to MEPs defined on BGP-MH SAPs or binds.
- behavior for operationally up MEPs on ES SAPs or SDP-bindings with EVPN multihoming, as follows:
  - **All-active multihoming**

Operationally up MEPs defined on non-DF ES SAPs can send CFM packets; however, they cannot receive CCMs (the SAP is removed from the default multicast list) or unicast CFM packets (because

the MEP MAC is not installed locally in the FDB; unicast CFM packets are treated as unknown and not sent to the non-DF SAP MEP).

- **Single-active multihoming**

Operationally up MEPs should be able to send or receive CFM packets normally.

- Operationally up MEPs defined on LAG SAPs require the command `process_cpm_traffic_on_sap_down` so that they can process CFM when the LAG is down and act as regular Ethernet ports.

Because of the preceding considerations, Nokia recommends the use of ETH-CFM in EVPN multihomed SAPs or SDP-bindings only on operationally down MEPs and single-active multihoming, in which case ETH-CFM is used to notify the CE of the DF or non-DF status.

### 5.4.9 Multi-instance EVPN: Two instances of different encapsulation in the same VPLS/R-VPLS/Epipe service

SR OS supports a maximum of two BGP instances in the same VPLS or R-VPLS, where the two instances can be:

- one EVPN-VXLAN instance and one EVPN-MPLS instance in the same VPLS or R-VPLS service
- two EVPN-VXLAN instances in the same VPLS or R-VPLS service
- two EVPN-MPLS instances in the same VPLS or R-VPLS service
- one EVPN-MPLS instance and one EVPN-SRv6 instance in the same VPLS service
- one EVPN-VXLAN instance and one EVPN-SRv6 instance in the same VPLS service

In this case, the procedures are compliant with RFC 9014.

SR OS also supports up to two BGP instances in the same Epipe. These two instances can be an EVPN-MPLS instance and an EVPN-SRv6 instance in the same Epipe service.

The procedures to support two BGP instances in the same Epipe adhere to *draft-sr-bess-evpn-vpws-gateway*.

#### 5.4.9.1 EVPN-VXLAN to EVPN-MPLS interworking

This section describes the configuration aspects of a VPLS/R-VPLS with EVPN-VXLAN and EVPN-MPLS.

In a service where EVPN-VXLAN and EVPN-MPLS are configured together, the **configure service vpls bgp-evpn vxlan bgp 1** and **configure service vpls bgp-evpn mpls bgp 2** commands allow the user to associate EVPN-MPLS to a different instance from that associated with EVPN-VXLAN, and have both encapsulations simultaneously enabled in the same service. At the control plane level, EVPN MAC/IP advertisement routes received in one instance are consumed and readvertised in the other instance as long as the route is the best route for a specific MAC. Inclusive multicast routes are independently generated for each BGP instance. In the data plane, the EVPN-MPLS and EVPN-VXLAN destinations are instantiated in different implicit Split Horizon Groups (SHGs) so that traffic can be forwarded between them.

The following example shows a VPLS service with two BGP instances and both VXLAN and MPLS encapsulations configured for the same BGP-EVPN service.

```
*A:PE-1>config>service>vpls# info
-----
description "evpn-mpls and evpn-vxlan in the same service"
```

```

vxlan instance 1 vni 7000 create
exit
bgp
    route-distinguisher 10:2
    route-target target:64500:1
exit
bgp 2
    route-distinguisher 10:1
    route-target target:64500:1
exit
bgp-evpn
    evi 7000
    incl-mcast-orig-ip 10.12.12.12
    vxlan bgp 1 vxlan-instance 1
        no shutdown
    mpls bgp 2
        control-word
        auto-bind-tunnel
        resolution any
    exit
    force-vlan-vc-forwarding
    no shutdown
    exit
exit
no shutdown

```

The following list describes the preceding example:

- **bgp 1** or **bgp** is the default BGP instance
- **bgp 2** is the additional instance required when both **bgp-evpn vxlan** and **bgp-evpn mpls** are enabled in the service
- The commands supported in instance 1 are also available in instance 2 with the following considerations:
  - **pw-template-binding**  
The pw-template-binding can only exist in instance 1; it is not supported in instance 2.
  - **route-distinguisher**  
The operating route-distinguisher in both BGP instances must be different.
  - **route-target**  
The route target in both instances can be the same or different.
  - **vsi-import and vsi-export**  
Import and export policies can also be defined for either BGP instance.
- MPLS and VXLAN can use either BGP instance, and the instance is associated when **bgp-evpn mpls** or **bgp-evpn vxlan** is created. The **bgp-evpn vxlan** command must include not only the association to a BGP instance, but also to a **vxlan-instance** (because the VPLS services support two VXLAN instances).



**Note:** The **bgp-evpn vxlan no shutdown** command is only allowed if **bgp-evpn mpls shutdown** is configured, or if the BGP instance associated with the MPLS has a different route distinguisher than the VXLAN instance.

The following features are not supported when two BGP instances are enabled on the same VPLS/R-VPLS service:



- SDP bindings
- M-VPLS, I-VPLS, B-VPLS, or E-Tree VPLS
- Proxy-ARP and proxy-ND
- BGP Multihoming
- IGMP, MLD, and PIM snooping
- BGP-VPLS or BGP-AD (SDP bindings are not created)

The **service>vpls>bgp-evpn>ip-route-advertisement** command is not supported on R-VPLS services with two BGP instances.

### 5.4.9.2 EVPN-SRv6 to EVPN-MPLS or EVPN-VXLAN interworking

EVPN-SRv6 and EVPN-MPLS or EVPN-VXLAN can be simultaneously configured in the same VPLS service (but not R-VPLS), in different instances. In addition, EVPN-SRv6 and EVPN-MPLS can be simultaneously configured in the same Epipe service, so that border routers can stitch SRv6 and MPLS domains for point-to-point services.

#### 5.4.9.2.1 VPLS services

EVPN-SRv6 and EVPN-VXLAN instances in the same VPLS service follow the same configuration rules as described in [EVPN-VXLAN to EVPN-MPLS interworking](#), and the same processing of MAC/IP Advertisement routes and Inclusive Multicast Ethernet Tag routes is applied.

The following example shows a VPLS service with two BGP instances, with both VXLAN and SRv6 encapsulations configured under BGP-EVPN.

#### Example: MD-CLI

```
A:node-2>config>service>vpls "evpn-srv6-vxlan-1"> info
  admin-state enable
  description "evpn-srv6 and evpn-vxlan in the same service"
  vxlan {
    instance 1 {
      vni 12340
    }
  }
  segment-routing-v6 1 {
    locator "loc-1" {
      function {
        end-dt2u {
        }
        end-dt2m {
        }
      }
    }
  }
}
  bgp 1 {
    route-distinguisher "12340:1"
    route-target {
      export "target:64500:12340"
      import "target:64500:12340"
    }
  }
  bgp 2 {
```

```

    route-distinguisher "12340:2"
    route-target {
        export "target:64500:12341"
        import "target:64500:12341"
    }
}
bgp-evpn {
    evi 12340
    incl-mcast-orig-ip 10.12.12.12
    segment-routing-v6 2 {
        admin-state enable
        ecmp 4
        force-vc-forwarding vlan
        srv6 {
            default-locator "loc-1"
        }
    }
}
vxlan 1 {
    admin-state enable
    vxlan-instance 1
}
}

```

### Example: classic CLI

```

A:node-2>config>service>vpls# info
-----
description "evpn-srv6 and evpn-vxlan in the same service"
vxlan instance 1 vni 12340 create
exit
segment-routing-v6 1 create
    locator "loc-1"
        function
            end-dt2u
            end-dt2m
        exit
    exit
exit
bgp
    route-distinguisher 12340:1
    route-target export target:64500:12340 import target:64500:12340
exit
bgp 2
    route-distinguisher 12340:2
    route-target export target:64500:12341 import target:64500:12341
exit
bgp-evpn
    incl-mcast-orig-ip 10.12.12.12
    evi 12340
    vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
    segment-routing-v6 bgp 2 srv6-instance 1 default-locator "loc-1" create
        ecmp 4
        force-vlan-vc-forwarding
        route-next-hop 2001:db8::76
        no shutdown
    exit
exit
stp
    shutdown
exit
no shutdown

```

-----

When an EVPN-SRv6 instance and an EVPN-MPLS instance are both configured in the same VPLS service, each instance can be configured in a different or the same split horizon group. The former option allows the interconnection of domains of different encapsulation, and the rules of configuration and route processing described in [EVPN-VXLAN to EVPN-MPLS interworking](#) apply. The latter option is used for domains where MPLS and SRv6 PE are attached to the same service, typically for migration purposes.

When the EVPN-SRv6 and the EVPN-MPLS instances are configured in the same split horizon group:

- MAC/IP Advertisement routes are not redistributed between the two instances
- Two BUM EVPN destinations to the same far-end PE (identified by the originator IP of the Inclusive Multicast Ethernet Tag routes) cannot be created. An EVPN-MPLS BUM destination is removed if there is another BUM destination to the same far end with an SRv6 encapsulation. This is to prevent BUM traffic duplication between multi-instance nodes
- SAPs are supported, but SDP binds are not supported

The following example shows a VPLS service with two BGP instances, with both MPLS and SRv6 encapsulations configured under BGP-EVPN, with the same split horizon group.

### Example: MD-CLI

```
configure service vpls "evpn-srv6-mpls-1" >info
admin-state enable
  description "evpn-srv6 and evpn-mpls in the same service"
  segment-routing-v6 1 {
    locator "loc-1" {
      function {
        end-dt2u {
        }
        end-dt2m {
        }
      }
    }
  }
  bgp 1 {
    route-distinguisher "12341:1"
    route-target {
      export "target:64500:12342"
      import "target:64500:12342"
    }
  }
  bgp 2 {
    route-distinguisher "12341:2"
    route-target {
      export "target:64500:12343"
      import "target:64500:12343"
    }
  }
  bgp-evpn {
    evi 12340
    incl-mcast-orig-ip 10.12.12.12
    segment-routing-v6 2 {
      admin-state enable
      ecmp 4
      force-vc-forwarding vlan
      srv6 {
        default-locator "loc-1"
      }
    }
  }
}
```

```

    }
    route-next-hop {
        ip-address 2001:db8::76
    }
}
mpls 1 {
    admin-state enable
    force-vlan-vc-forwarding vlan
    split-horizon-group "SHG-1"
    ingress-replication-bum-label true
    ecmp 4
    mh-mode access
    auto-bind-tunnel {
        resolution any
    }
}
}
split-horizon-group "SHG-1" {
}

```

### Example: classic CLI

```

A:node-2>config>service>vpls# info
-----
description "evpn-srv6 and evpn-mpls in the same service"
split-horizon-group "SHG-1" create
exit
segment-routing-v6 1 create
    locator "loc-1"
        function
            end-dt2u
            end-dt2m
        exit
    exit
exit
bgp
    route-distinguisher 12341:1
    route-target export target:64500:12342 import target:64500:12342
exit
bgp 2
    route-distinguisher 12341:2
    route-target export target:64500:12343 import target:64500:12343
exit
bgp-evpn
    incl-mcast-orig-ip 10.12.12.12
    evi 12341
    mpls bgp 1
        mh-mode access
        force-vlan-vc-forwarding
        split-horizon-group "SHG-1"
        ingress-replication-bum-label
        ecmp 4
        auto-bind-tunnel
            resolution any
        exit
    no shutdown
exit
segment-routing-v6 bgp 2 srv6-instance 1 default-locator "loc-1" create
    ecmp 4
    force-vlan-vc-forwarding
    route-next-hop 2001:db8::76
    split-horizon-group "SHG-1"
    no shutdown

```

```

        exit
    exit
    stp
        shutdown
    exit
    no shutdown
-----

```

### 5.4.9.2.2 Epipe services

The following example shows an Epipe service with two BGP instances, with both MPLS and SRv6 encapsulations configured under BGP-EVPN. This is the gateway configuration when it is stitching MPLS and SRv6 domains for E-Line services.

#### Example: MD-CLI

```

[ex:/configure service epipe "multi-inst-evpn-vpws-100"]
A:admin@node-2# info
  admin-state enable
  service-id 100
  customer "1"
  segment-routing-v6 1 {
    locator "LOC-2-16bits" {
      function {
        end-dx2 {
        }
      }
    }
  }
  bgp 1 {
    route-distinguisher "23.23.23.1:100"
  }
  bgp 2 {
    route-distinguisher "23.23.23.2:100"
  }
  endpoint "mpls" {
  }
  endpoint "srv6" {
  }
  bgp-evpn {
    evi 100
    local-attachment-circuit "mpls" {
      endpoint "mpls"
      eth-tag 1
    }
    local-attachment-circuit "srv6" {
      endpoint "srv6"
      eth-tag 1
      bgp 2
    }
    remote-attachment-circuit "mpls" {
      endpoint "mpls"
      eth-tag 1
    }
    remote-attachment-circuit "srv6" {
      endpoint "srv6"
      eth-tag 1
      bgp 2
    }
  }
  mpls 1 {
    admin-state enable
  }

```

```

        ecmp 2
        domain-id "64500:1"
        auto-bind-tunnel {
            resolution any
        }
    }
    segment-routing-v6 2 {
        admin-state enable
        source-address 2001:db8::2
        mh-mode access
        domain-id "64500:2"
        srv6 {
            instance 1
            default-locator "LOC-2-16bits"
        }
    }
}

```

### Example: classic CLI

```

A:node-2# configure service epipe 100
A:node-2>config>service>epipe# info
-----
endpoint "mpls" create
exit
endpoint "srv6" create
exit
segment-routing-v6 1 create
    locator "LOC-2-16bits"
    function
    end-dx2
    exit
exit
exit
bgp 1
    route-distinguisher 23.23.23.1:100
exit
bgp 2
    route-distinguisher 23.23.23.2:100
exit
bgp-evpn
    local-attachment-circuit mpls bgp 1 endpoint mpls create
        eth-tag 1
    exit
    local-attachment-circuit srv6 bgp 2 endpoint srv6 create
        eth-tag 1
    exit
    remote-attachment-circuit mpls bgp 1 endpoint mpls create
        eth-tag 1
    exit
    remote-attachment-circuit srv6 bgp 2 endpoint srv6 create
        eth-tag 1
    exit
evi 100
mpls bgp 1
    domain-id 64500:1
    ecmp 2
    auto-bind-tunnel
        resolution any
    exit
    no shutdown
exit

```

```

create      segment-routing-v6 bgp 2 srv6-instance 1 default-locator "LOC-2-16bits"
            domain-id 64500:2
            mh-mode access
            source-address 2001:db8::2
            no shutdown
            exit
        exit
        no shutdown
-----

```

Where:

- The **epipe bgp** command supports up to two instances, where the default value is 1, and the accepted values are now in the range 1..2.
- The *bgp-instance* of 1 or 2 can be matched under the following contexts:

```

configure service epipe bgp-evpn mpls
configure service epipe bgp-evpn segment-routing-v6

```

MPLS and SRv6 can be configured in Epipes with one or two instances, and they can indistinctly use instance “1” or “2”. The preceding example shows an Epipe service with MPLS configured in *bgp-instance 1* and segment-routing-v6 configured in *bgp-instance 2*.

- The *bgp-instance 2* requires the support of the explicit route distinguisher (RD) configuration because the EVI-based autoderivation of the RD only applies to *bgp-instance 1*. The route target EVI-based autoderivation applies to both instances.
- The following command can also be associated with a BGP instance.

```

configure service epipe bgp-evpn local-attachment-circuit

```

By default, all local attachment circuits in the service are associated with *bgp-instance 1*. When the local attachment circuits are associated with different BGP instances, no local SAPs or spoke-SDPs are supported in the service (this is blocked by the CLI).

- The BGP instances are configured with a D-PATH **domain-id**. The D-PATH attribute is described in section [BGP D-PATH attribute for Layer 3 loop protection](#) and can also be used in multi-instance Epipe services. D-PATH is used in the EVPN-VPWS AD per-EVI routes for best-path selection and loop avoidance in case of redundant gateways. In the preceding example, configuring **segment-routing-v6 bgp 2 domain-id 64500:2** means that the received EVPN AD per-EVI route in the SRv6 instance is redistributed to the MPLS instance with a D-PATH attribute where domain 64500:2 is prepended.
- When configuring an SRv6 and an MPLS instance in an Epipe service, one of the two instances must be configured as **mh-mode access**, with the other one configured as **mh-mode network** (default value for SRv6 and MPLS instances). The command is added under the MPLS and SRv6 instances (not in VXLAN instances).

As in the case of any Epipe service, two explicit or implicit endpoints exist, where traffic always flows from one endpoint to the other endpoint. The preceding example uses the configuration of two explicit endpoints, however one implicit endpoint plus one explicit endpoint can also be configured, and the behavior would be identical. In other words, the preceding configuration is also valid if the **endpoint "mpls"** is not configured. In this case, the **local-attachment-circuit** and **remote-attachment-circuit** associated with **bgp 1** would be part of an implicit end-point.

### 5.4.9.3 BGP-EVPN routes in services configured with two BGP instances

The following sections describe BGP-EVPN routes in EVPN and VPLS services configured with two BGP instances.

#### 5.4.9.3.1 VPLS services

From a BGP perspective, the two BGP instances configured in the service are independent of each other. The redistribution of routes between the BGP instances is resolved at the EVPN application layer.

By default, if EVPN-VXLAN and EVPN-MPLS are both enabled in the same service, BGP sends the generated EVPN routes twice: with the RFC 9012 BGP encapsulation extended community set to VXLAN and a second time with the encapsulation type set to MPLS.

Usually, a DCGW peers a pair of Route Reflectors (RRs) in the DC and a pair of RRs in the WAN. For this reason, the user needs to add router policies so that EVPN-MPLS routes are only sent to the WAN RRs and EVPN-VXLAN routes are only sent to the DC RRs. The following examples show how to configure router policies.

#### Example: MD-CLI

```
[ex:/configure router "Base" bgp]
A:admin@node-2# info
  vpn-apply-export true
  vpn-apply-import true
  group "WAN" {
    type internal
    family {
      evpn true
    }
    export {
      policy ["allow only mpls"]
    }
  }
  group "DC" {
    type internal
    family {
      evpn true
    }
    export {
      policy ["allow only vxlan"]
    }
  }
  neighbor "192.0.2.2" {
    group "WAN"
  }
  neighbor "192.0.2.75" {
    group "DC"
  }

*[ex:/configure policy-options]
A:admin@PE-76# info
  community "mpls" {
    member "bgp-tunnel-encap:MPLS" { }
  }
  community "vxlan" {
    member "bgp-tunnel-encap:VXLAN" { }
  }
  policy-statement "allow only mpls" {
```



```

    entry 10 {
      from {
        family [evpn]
        community {
          name "vxlan"
        }
      }
      action {
        action-type reject
      }
    }
  }
  policy-statement "allow only vxlan" {
    entry 10 {
      from {
        family [evpn]
        community {
          name "mpls"
        }
      }
      action {
        action-type reject
      }
    }
  }
}

```

### Example: classic CLI

```

config>router>bgp#
vpn-apply-import
vpn-apply-export
group "WAN"
family evpn
type internal
export "allow only mpls"
neighbor 192.0.2.6
group "DC"
family evpn
type internal
export "allow only vxlan"
neighbor 192.0.2.2

```

```

A:node-2>config>router>policy-options# info
-----
community "vxlan" members "bgp-tunnel-encap:VXLAN"
community "mpls" members "bgp-tunnel-encap:MPLS"
policy-statement "allow only mpls"
  entry 10
    from
      family evpn
      community vxlan
    action drop
  exit
exit
policy-statement "allow only vxlan"
  entry 10
    from
      family evpn
      community mpls
    action drop
  exit

```

```

        exit
    exit

```

In a BGP instance, the EVPN routes are imported based on the route-targets and regular BGP selection procedures, regardless of their encapsulation.

The BGP-EVPN routes are generated and redistributed between BGP instances based on the following rules:

- Auto-discovery (AD) routes (type 1) are not generated by services with two BGP EVPN instances, unless a local Ethernet segment is present on the service. However, AD routes received from the EVPN-MPLS peers are processed for aliasing and backup functions as usual.
- MAC/IP routes (type 2) received in one of the two BGP instances are imported and the MACs added to the FDB according to the existing selection rules. If the MAC is installed in the FDB, it is readvertised in the other BGP instance with the new BGP attributes corresponding to the BGP instance (route target, route distinguisher, and so on). The following considerations apply to these routes:
  - The **mac-advertisement** command governs the advertisement of any MACs (even those learned from BGP).
  - A MAC route is redistributed only if it is the best route based on the EVPN selection rules.
  - If a MAC route is the best route and has to be redistributed, the MAC/IP information, along with the MAC mobility extended community, is propagated in the redistribution.
  - The router redistributes any MAC route update for which any attribute has changed. For example, a change in the SEQ or sticky bit in one instance is updated in the other instance for a route that is selected as the best MAC route.
- EVPN inclusive multicast routes are generated independently for each BGP instance with the corresponding BGP encapsulation extended community (VXLAN or MPLS). Also, the following considerations apply to these routes:
  - Ingress Replication (IR) and Assisted Replication (AR) routes are supported in the EVPN-VXLAN instance. If AR is configured, the AR IP address must be a loopback address different from the **system-ip** and the configured **originating-ip** address.
  - IR, P2MP mLDP, and composite inclusive multicast routes are supported in the EVPN-MPLS instance.
  - The modification of the **incl-mcast-orig-ip** command is supported, subject to the following considerations:
    - The configured IP in the **incl-mcast-orig-ip** command is encoded in the **originating-ip** field of the inclusive multicast Routes for IR, P2MP, and composite tunnels.
    - The **originating-ip** field of the AR routes is still derived from the **service>system>vxlan>assisted-replication-ip** configured value.
  - EVPN handles the inclusive multicast routes in a service based on the following rules:
    - For IR routes, the EVPN destination is set up based on the NLRI next hop.
    - For P2MP mLDP routes, the PMSI Tunnel Attribute **tunnel-id** is used to join the mLDP tree.
    - For composite P2MP-IR routes, the PMSI Tunnel Attribute **tunnel-id** is used to join the tree and create the P2MP bind. The NLRI next-hop is used to build the IR destination.
    - For AR routes, the NLRI next-hop is used to build the destination.
    - The following applies if a router receives two inclusive multicast routes in the same instance:

- If the routes have the same **originating-ip** but different route distinguishers and next-hops, the router processes both routes. In the case of IR routes, it sets up two destinations: one to each next-hop.
  - If the routes have the same **originating-ip**, different route distinguishers, but same next hops, the router sets up only one binding for IR routes.
  - The router ignores inclusive multicast routes received with its own **originating-ip**, regardless of the route distinguisher.
- IP-Prefix routes (type 5) are not generated or imported by a service with two BGP instances.

The rules in this section can be extrapolated to VPLS services where SRv6 and MPLS or VXLAN are configured in different instances of the same VPLS with different split horizon groups.

### 5.4.9.3.2 Epipe services

Single-instance EVPN-VPWS services do not generate AD per-EVI routes if they do not have a local SAP/ spoke SDP configured and in the **oper-up** state. Multi-instance EVPN-VPWS do not allow local SAP/ Spoke-SDPs, therefore they do not generate AD per-EVI routes for the configured **local-attachment-circuit eth-tags**. These services "redistribute" AD per-EVI routes received in one instance into the other. The redistribution rules follow *draft-sr-bess-evpn-vpws-gateway* as follows:

- Upon receiving an AD per-EVI route in *bgp-instance 1*, if the route is selected to be installed and the route does not contain a local **domain-id** in its D-PATH attribute (local means the **domain-id** is configured in the Epipe), an AD per-EVI route is triggered in *bgp-instance 2*, using the **eth-tag**, RD, RT and properties of *bgp-instance 2*.
- The EVPN Layer 2 attributes extended community is regenerated for the redistributed route. The value of the P and B flags is set to 0 when redistributing routes.
- The encapsulation-specific attributes of the redistributed route are regenerated based on the encapsulation of the BGP instance in which the route is advertised.
- The redistributed route carries the Communities, Extended Communities, and Large Communities of the source route when the following command is configured:

– **MD-CLI**

```
configure service system bgp evpn ad-per-evi-routes attribute-propagation true
```

– **classic-CLI**

```
configure service system bgp-evpn ad-per-evi-routes attribute-propagation
```

The exception are RTs (which are re-originated), EVPN Extended Communities, and BGP Encapsulation Extended Communities [RFC 9012]. EVPN Extended Communities and BGP Encapsulation Extended Communities are not propagated across domains.

- The redistributed AD per-EVI route updates the D-PATH attribute of the received route or adds the D-PATH attribute if the received route did not contain a D-PATH.
- The ESI of the redistributed AD per-EVI route is always zero.
- AD per-ES and ES routes are never redistributed.

### 5.4.9.3.2.1 Route selection of AD per-EVI routes

The redistribution of attributes, as well as the BGP best-path selection for AD per-EVI routes is controlled by the commands shown in the following examples.

#### Example: MD-CLI

```
[ex:/configure service system bgp evpn ad-per-evi-routes]
A:admin@PE-2# tree detail
+-- attribute-propagation <boolean>
+-- bgp-path-selection <boolean>
+-- d-path-ignore <boolean>
```

#### Example: Classic CLI

```
*A:PE-2>config>service>system>bgp-evpn>ad-per-evi-routes# tree detail
ad-per-evi-routes
|
+---attribute-propagation
| no attribute-propagation
|
+---bgp-path-selection
| no bgp-path-selection
|
+---d-path-ignore
| no d-path-ignore
```

Where both (**bgp-path-selection** and **attribute-propagation**) are disabled by default, and the router enforces that **bgp-path-selection** can only be enabled if **attribute-propagation** is enabled before.

If **bgp-path-selection false** (default) is configured, in case of multiple AD per-EVI routes for the same Ethernet tag are received in the same Epipe BGP instance, the lowest IP route is selected. Those routes may have zero ESI, or different non-zero ESI.

When multiple non-zero ESI AD per-EVI routes are received and the ESI matches on all of them, the **bgp-path-selection** command impacts the following procedures:

- The command influences the selection of AD per-EVI routes to create the ES destination. If disabled, the lowest IP address routes are selected, up to the number of configured ECMP paths. If enabled, the routes are selected based on BGP best-path selection.
- The command influences the selection of the best AD per-EVI route of the ES for the purpose of attribute propagation. If enabled, the attributes of the best-path route are propagated.

The best-path selection tie-breaking rules are included below for reference:

1. High Local Pref wins
2. Shortest D-PATH wins (if d-path-ignore false)
3. Lowest left-most D-PATH domain-id wins (if d-path-ignore false)
4. Shortest AS\_PATH wins
5. Lowest Origin wins
6. Lowest MED wins
7. EBGP wins

8. Lowest tunnel-table cost to the next-hop
9. Lowest next-hop type wins (resolution in TTM wins vs RTM)
10. Lowest next-hop type wins
11. Lowest router ID wins (applicable to ibgp peers only)
12. Shortest cluster\_list length wins (applicable to ibgp peers only)
13. Lowest IP address
14. Next-hop check (IPv4 NH wins, then lowest NH wins)
15. RD check (lowest RD wins)
16. Path-Id (add path)

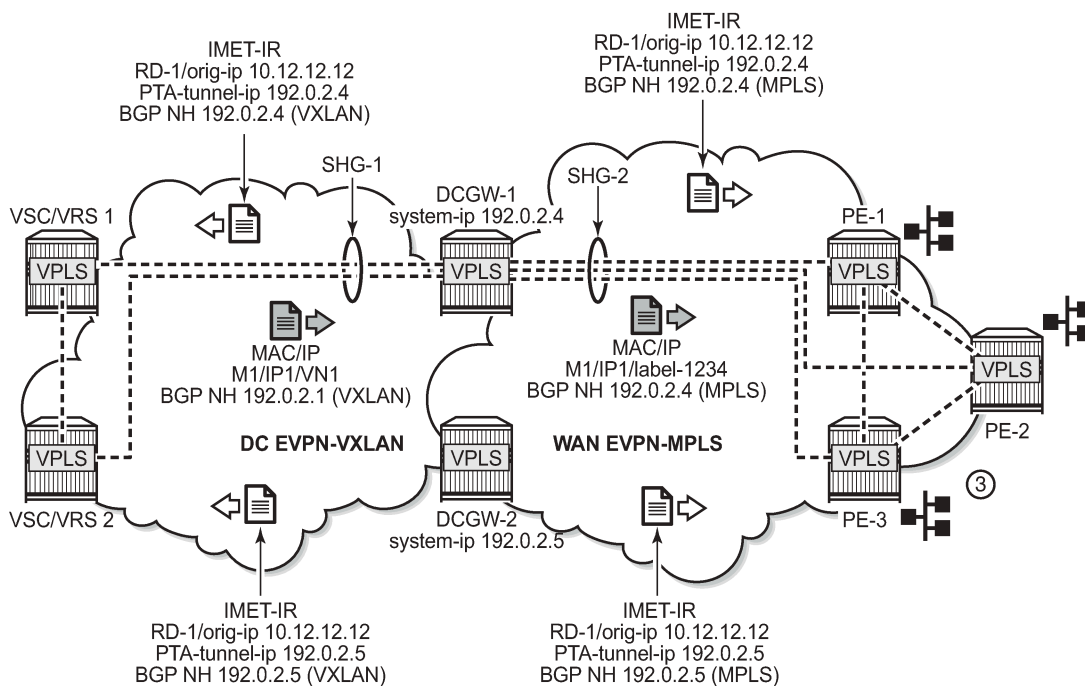
#### 5.4.9.4 Anycast redundant solution for dual BGP-instance services

The following sections describe the anycast redundant solution for dual BGP instances in VPLS and Epipe services.

##### 5.4.9.4.1 VPLS services

The following figure shows the anycast mechanism used to support gateway redundancy for dual BGP-instance services. The example shows two redundant DC gateways (DCGWs) where the VPLS services contain two BGP instances: one each for EVPN-VXLAN and EVPN-MPLS.

Figure 98: Multihomed anycast solution



The example shown in [Figure 98: Multihomed anycast solution](#) depends on the ability of the two DCGWs to send the same inclusive multicast route to the remote PE or NVEs, such that:

- The remote PE or NVEs create a single BUM destination to one of the DCGWs (because the BGP selects only the best route to the DCGWs).
- The DCGWs do not create a destination between each other.

This solution avoids loops for BUM traffic, and known unicast traffic can use either DCGW router, depending on the BGP selection. The following CLI example output shows the configuration of each DCGW.

### Example: MD-CLI

```
/* bgp configuration on DCGW1 and DCGW2 */

[ex:/configure router "Base" bgp]
A:admin@DCGW# info
  vpn-apply-export true
  vpn-apply-import true
  group "DC" {
    type internal
    family {
      evpn true
    }
  }
  group "WAN" {
    type internal
    family {
      evpn true
    }
  }
  neighbor "192.0.2.2" {
    group "DC"
  }
  neighbor "192.0.2.6" {
    group "WAN"
  }
}

/* vpls service configuration in DCGW1 */

[ex:/configure service vpls "1"]
A:admin@DCGW1# info
  admin-state enable
  customer "1"
  vxlan {
    instance 1 {
      vni 1
    }
  }
  bgp 1 {
    route-distinguisher "64501:12"
    route-target {
      export "target:64500:1"
      import "target:64500:1"
    }
  }
  bgp 2 {
    route-distinguisher "64502:12"
    route-target {
      export "target:64500:1"
      import "target:64500:1"
    }
  }
}
```

```

    }
    bgp-evpn {
        evi 1
        incl-mcast-orig-ip 10.12.12.12
        vxlan 1 {
            admin-state enable
            vxlan-instance 1
        }
        mpls 2 {
            admin-state enable
            auto-bind-tunnel {
                resolution any
            }
        }
    }
}

/* vpls service configuration in DCGW2 */

[ex:/configure service vpls "1"]
A:admin@DCGW2# info
  admin-state enable
  customer "1"
  vxlan {
    instance 1 {
      vni 1
    }
  }
  bgp 1 {
    route-distinguisher "64501:12"
    route-target {
      export "target:64500:1"
      import "target:64500:1"
    }
  }
  bgp 2 {
    route-distinguisher "64502:12"
    route-target {
      export "target:64500:1"
      import "target:64500:1"
    }
  }
  }
  bgp-evpn {
    evi 1
    incl-mcast-orig-ip 10.12.12.12
    vxlan 1 {
      admin-state enable
      vxlan-instance 1
    }
    mpls 2 {
      admin-state enable
      auto-bind-tunnel {
        resolution any
      }
    }
  }
}

```

### Example: classic CLI

```

/* bgp configuration on DCGW1 and DCGW2 */
config>router>bgp#
  group "WAN"
  family evpn
  type internal

```

```

neighbor 192.0.2.6
group "DC"
family evpn
type internal
neighbor 192.0.2.2
/* vpls service configuration */
DCGW-1# config>service>vpls(1)#
-----
bgp
  route-distinguisher 64501:12
  route-target target:64500:1
exit
bgp 2
  route-distinguisher 64502:12
  route-target target:64500:1
exit
vxlان instance 1 vni 1 create
exit
bgp-evpn
  evi 1
  incl-mcast-orig-ip 10.12.12.12
  vxlan bgp 1 vxlan-instance 1
  no shutdown
  mpls bgp 2
  no shutdown
  auto-bind-tunnel
  resolution any
exit
<snip>
DCGW-2# config>service>vpls(1)#
-----
bgp
  route-distinguisher 64501:12
  route-target target:64500:1
exit
bgp 2
  route-distinguisher 64502:12
  route-target target:64500:1
exit
vxlان instance 1 vni 1 create
exit
bgp-evpn
  evi 1
  incl-mcast-orig-ip 10.12.12.12
  vxlan bgp 1 vxlan-instance 1
  no shutdown
  mpls bgp 2
  no shutdown
  auto-bind-tunnel
  resolution any
<snip>

```

Based on the preceding configuration example, the behavior of the DCGWs in this scenario is as follows:

- DCGW-1 and DCGW-2 send inclusive multicast routes to the DC RR and WAN RR with the same route key. For example:
  - DCGW-1 and DCGW-2 both send an IR route to DC RR with RD=64501:12, orig-ip=10.12.12.12, and a different next hop and tunnel ID
  - DCGW-1 and DCGW-2 both send an IR route to WAN RR with RD=64502:12, orig-ip=10.12.12.12, and different next hop and tunnel ID



- DCGW-1 and DCGW-2 both receive MAC/IP routes from DC and WAN that are redistributed to the other BGP instances, assuming that the route is selected as best route and the MAC is installed in the FDB.

As described in section [BGP-EVPN routes in services configured with two BGP instances](#), router peer policies are required so that only VXLAN or MPLS routes are sent or received for a specific peer.

- Configuration of the same **incl-mcast-orig-ip** address in both DCGWs enables the anycast solution for BUM traffic for all the following reasons:
  - The configured **originating-ip** is not required to be a reachable IP address and this forces the remote PE or NVEs to select only one of the two DCGWs.
  - The BGP next hops are allowed to be the **system-ip** or even a loopback address. In both cases, the BGP next hops are not required to be reachable in their respective networks.

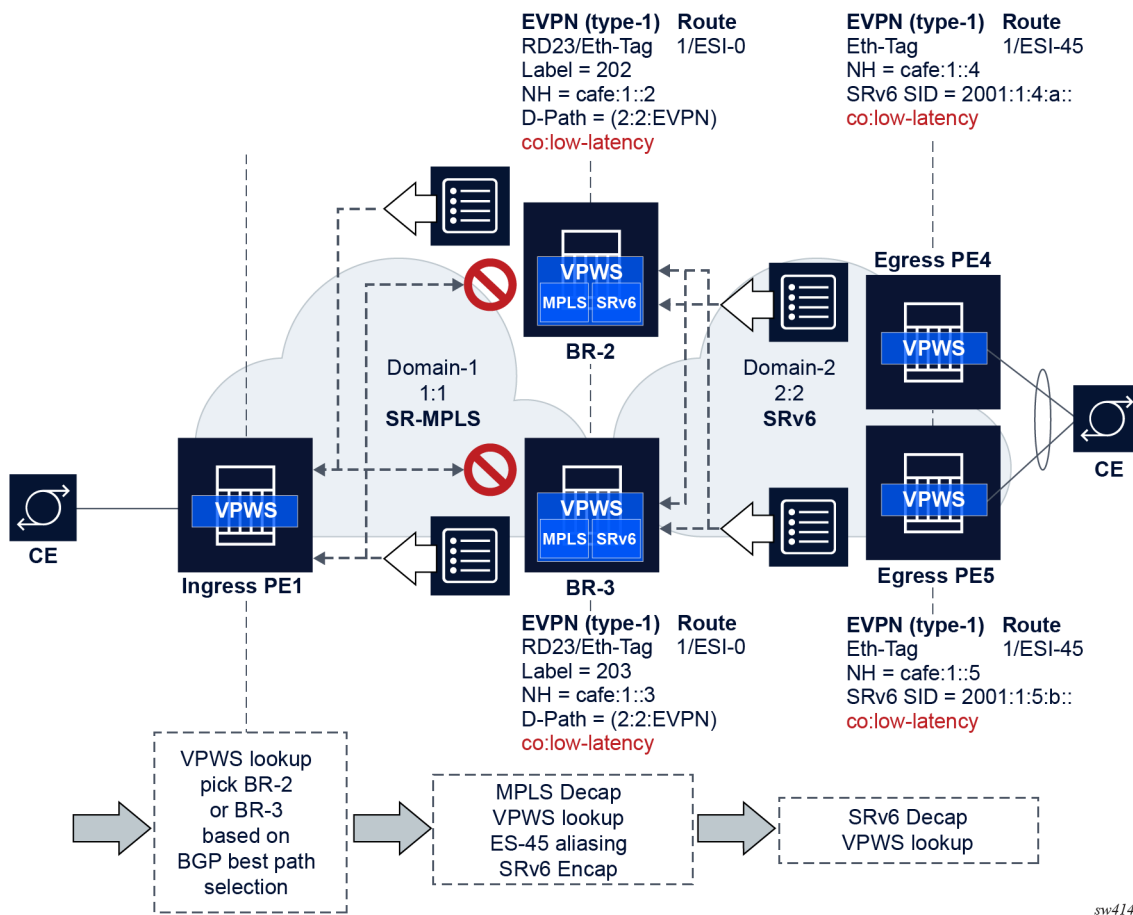
In the example shown in [Figure 98: Multihomed anycast solution](#), PE-1 picks up DCGW-1's inclusive multicast route (because of its lower BGP next hop) and creates a BUM destination to 192.0.2.4. When sending BUM traffic for VPLS-1, it only sends the traffic to DCGW-1. In the same way, the DCGWs do not set up BUM destinations between each other as they use the same **originating-ip** in their inclusive multicast routes.

The remote PE or NVEs perform a similar BGP selection for MAC/IP routes, as a specific MAC is sent by the two DCGWs with the same route key. A PE or NVE sends known unicast traffic for a specific MAC to only one DCGW.

#### 5.4.9.4.2 Epipe services

The anycast redundancy solution can also be used for gateways that stitch SRv6 to MPLS domains for EVPN-VPWS services. The principle is similar to the one described in [VPLS services](#). The following figure shows an example.

Figure 99: Multihomed anycast solution for Epipe services



The configuration on the two gateways (BR-2 and BR-3 in the preceding example) must generate AD per-EVI routes with the same route key (including the same RD) from both gateways so that the ingress PEs select one of the two gateways based on BGP best-path selection.



**Note:** The D-PATH configuration (domain ID) is also needed, and **mh-mode** is configured on both gateways.

The following is an example of the (identical) configuration in BR-2 and BR-3.

### Example: MD-CLI

```
[ex:/configure service epipe "1"]
A:admin@BR-2/BR-3# info
  admin-state enable
  service-id 1
  customer "1"
  segment-routing-v6 1 {
    locator "LOC-1" {
      function {
        end-dx2 {
        }
      }
    }
  }
```

```

    }
  }
  bgp 1 {
    route-distinguisher 2323:1
  }
  bgp 2 {
    route-distinguisher 2323:2
  }
  endpoint "MPLS" {
  }
  endpoint "SRv6" {
  }
  bgp-evpn {
    evi 1
    local-attachment-circuit "gw-mpls" { // implicitly associated to bgp 1
      eth-tag 1
    }
    endpoint "MPLS"
  }
    remote-attachment-circuit "ac-1-mpls" {
      eth-tag 1
    }
    endpoint "MPLS"
  }
    local-attachment-circuit "gw-srv6" { // associated to bgp 2
      eth-tag 1
    }
    endpoint "SRv6"
  }
  bgp 2
  }
    remote-attachment-circuit "ac-2-srv6" {
      eth-tag 1
    }
    endpoint "SRv6"
  }
    bgp 2
  }
  mpls 1 {
    admin-state enable
    ecmp 2
    domain 64500:1
    mh-mode access
    auto-bind-tunnel {
      resolution any
    }
    route-next-hop {
      ip-address 2001:db8::2
    }
  }
  segment-routing-v6 2 {
    admin-state enable
    source-address 2001:db8::2
    ecmp 2
    domain 64500:2
    mh-mode network // default
    srv6 {
      instance 1
      default-locator "LOC-1"
    }
    route-next-hop {
      system-ipv6
    }
  }
}

```

### Example: classic CLI

```
*A:BR-2/BR-3# configure service epipe 1
```

```

*A:BR-2/BR-3>config>service>epipe# info
-----
    endpoint "MPLS" create
    exit
    endpoint "SRv6" create
    exit
    segment-routing-v6 1 create
        locator "LOC-1"
            function
                end-dx2
            exit
        exit
    exit
    bgp 1
        route-distinguisher 2323:1
    exit
    bgp 2
        route-distinguisher 2323:2
    exit
    bgp-evpn
        local-attachment-circuit "gw-mpls" bgp 1 endpoint "MPLS" create
            eth-tag 1
        exit
        local-attachment-circuit "gw-srv6" bgp 2 endpoint "SRv6" create
            eth-tag 1
        exit
        remote-attachment-circuit "ac-1-mpls" bgp 1 endpoint "MPLS" create
            eth-tag 1
        exit
        remote-attachment-circuit "ac-2-srv6" bgp 2 endpoint "SRv6" create
            eth-tag 1
        exit
    evi 1
        mpls bgp 1
            domain-id 64500:1
            ecmp 2
            auto-bind-tunnel
                resolution any
            exit
            no shutdown
        exit
        segment-routing-v6 bgp 2 srv6-instance 1 default-locator "LOC-1" create
            domain-id 64500:2
            mh-mode access
            source-address 2001:db8::3
            no shutdown
        exit
    exit
    no shutdown
-----

```

In this example:

- The anycast gateways attached to the same two domains redistribute the EVPN AD per-EVI routes between domains, where ESI is always reset to zero.
- The redundant gateways may set the same Ethernet Tag ID in the redistributed A-D per-EVI route (the example shows the same **eth-tag** values, but the gateways could also use other values).
- The anycast gateways process the received D-PATH attribute and update the D-PATH (with the source domain ID) when redistributing the AD per-EVI route to the next domain. The D-PATH attribute avoids control plane loops.

The following considerations related to the use of D-PATH in this configuration apply:

- Based on the domain configuration, when an AD per-EVI route is imported in domain X and redistributed into domain Y, the domain ID of X is prepended to the D-PATH in the redistributed AD per-EVI route.
- When two AD per-EVI routes for the same Ethernet tag (same route key) are received on the same services from different next hops, D-PATH is considered in the BGP best-path selection, unless **d-path-ignore true** is configured.
- When two AD per-EVI routes for the same service are received with different RDs and the same Ethernet tag from different next hops, D-PATH is considered in the BGP best-path selection, unless **d-path-ignore true** is configured, and assuming **bgp-path-selection true** is configured.
- If **d-path-ignore false** is configured, the router compares the D-PATH attribute received in VPWS AD per-EVI routes with the same key (same or different RDs) as follows:
  - The routes with the shortest D-PATH are preferred, therefore routes not tied to the shortest D-PATH are removed. Routes without D-PATH are considered zero-length D-PATH.
  - The routes with the numerically lowest left-most domain ID are preferred, therefore routes not tied to the numerically lowest left-most domain ID are removed from consideration.

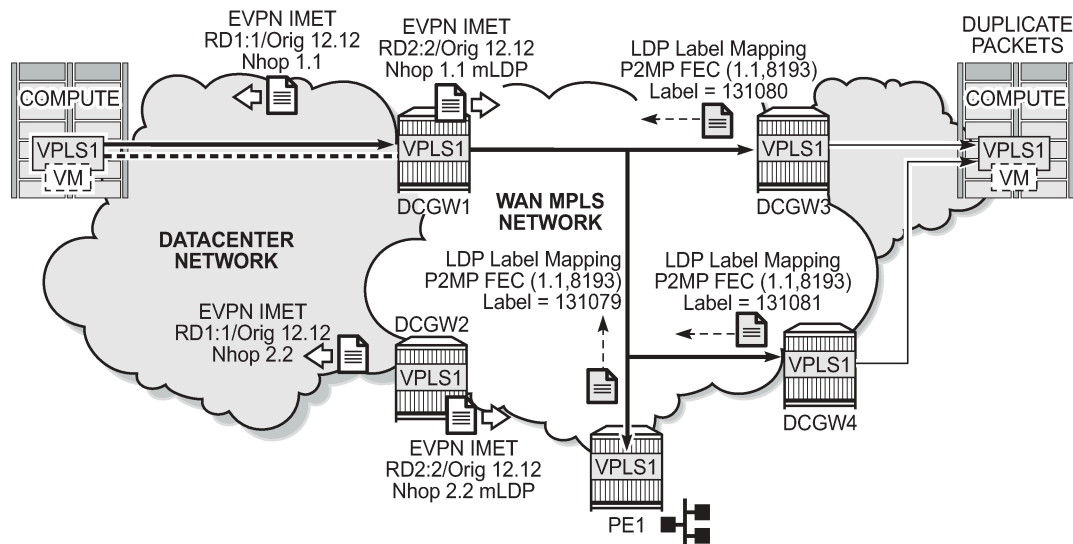
#### 5.4.9.5 Using P2MP mLDP in redundant anycast DCGWs

Figure 100: Anycast multihoming and mLDP shows an example of a common BGP EVPN service configured in redundant anycast DCGWs and mLDP used in the MPLS instance.



**Note:** Packet duplication may occur if the service configuration is not performed carefully.

Figure 100: Anycast multihoming and mLDP



No3492

When mLDP is used with multiple anycast multihoming DCGWs, the same originating IP address must be used by all the DCGWs. Failure to do so may result in packet duplication.

In the example shown in [Figure 100: Anycast multihoming and mLDP](#), each pair of DCGWs (DCGW1/DCGW2 and DCGW3/DCGW4) is configured with a different originating IP address, which causes the following behavior:

1. DCGW3 and DCGW4 receive the inclusive multicast routes with the same route key from DCGW1 and DCGW2.
2. Both DCGWs (DCGW3 and DCGW4) select only one route, which is generally the same, for example, DCGW1's inclusive multicast route.
3. As a result, DCGW3 and DCGW4 join the mLDP tree with root in DCGW1, creating packet duplication when DCGW1 sends BUM traffic.
4. Remote PE nodes with a single BGP-EVPN instance join the mLDP tree without any problem.

To avoid the packet duplication shown in [Figure 100: Anycast multihoming and mLDP](#), Nokia recommends to configure the same originating IP address in all four DCGWs (DCGW1/DCGW2 and DCGW3/DCGW4). However, the route distinguishers can be different per pair.

The following behavior occurs if the same originating IP address is configured on the DCGW pairs shown in [Figure 100: Anycast multihoming and mLDP](#).



**Note:** This configuration allows the use of mLDP as long as BUM traffic is not required between the two DCs. Ingress replication must be used if BUM traffic between the DCs is required.

- DCGW3 and DCGW4 do not join any mLDP tree sourced from DCGW1 or DCGW2, which prevents any packet duplication. This is because a router ignore inclusive multicast routes received with its own **originating-ip**, regardless of the route-distinguisher.
- PE1 joins the mLDP trees from the two DCs.

#### 5.4.9.6 I-ES solution for dual BGP instance services

SR OS supports Interconnect ESs (I-ES) for VXLAN as per *RFC9014*. An I-ES is a virtual ES that allows DCGWs with two BGP instances to handle VXLAN access networks as any other type of ES. I-ESs support the RFC 7432 multihoming functions, including single-active and all-active, ESI-based split-horizon filtering, DF election, and aliasing and backup on remote EVPN-MPLS PEs.

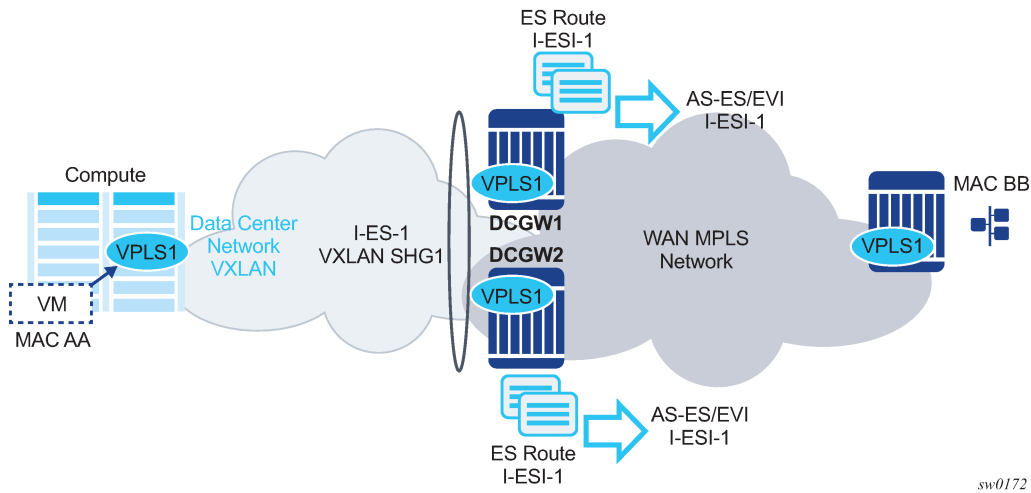
In addition to the EVPN multihoming features, the main advantages of the I-ES redundant solution compared to the redundant solution described in [Anycast redundant solution for dual BGP-instance services](#) are as follows:

- The use of I-ES for redundancy in dual BGP-instance services allows local SAPs on the DCGWs.
- P2MP mLDP can be used to transport BUM traffic between DCs that use I-ES without any risk of packet duplication. As described in [Using P2MP mLDP in redundant anycast DCGWs](#), packet duplication may occur in the anycast DCGW solution when mLDP is used in the WAN.

Where EVPN-MPLS networks are interconnected to EVPN-VXLAN networks, the I-ES concept applies only to the access VXLAN network; the EVPN-MPLS network does not modify its existing behavior.

[Figure 101: The Interconnect ES concept](#) shows the use of I-ES for Layer 2 EVPN DCI between VXLAN and MPLS networks.

Figure 101: The Interconnect ES concept



The following example shows how I-ES-1 would be provisioned on DCGW1 and the association between I-ES to a specified VPLS service. A similar configuration would occur on DCGW2 in the I-ES.

New I-ES configuration:

```
DCGW1#config>service>system>bgp-evpn#
ethernet-segment I-ES-1 virtual create
esi 01:00:00:00:12:12:12:12:00
service-carving
mode auto
multi-homing all-active
network-interconnect-vxlan 1
service-id
service-range 1 to 1000
no shutdown
```

Service configuration:

```
DCGW1#config>service>vpls(1)#
vxlan instance 1 vni 1 instance 1 create
exit
bgp
route-distinguisher 1:1
bgp 2
route-distinguisher 2:2
bgp-evpn
evi 1
vxlan bgp 1 vxlan-instance 1
no shutdown
exit
mpls bgp 2
auto-bind-tunnel resolution any
no shutdown
...
DCGW1#config>service>vpls(2)#
vxlan instance 1 vni 2 create
exit
bgp
```

```

route-distinguisher 3:3
bgp 2
route-distinguisher 4:4
bgp-evpn
evi 2
vxlان bgp 1 vxlan-instance 1
no shutdown
exit
mpls bgp 2
auto-bind-tunnel resolution any
no shutdown
sap 1/1/1:1 create
exit

```

The above configuration associates I-ES-1 to the VXLAN instance in services VPLS1 and VPLS 2. The I-ES is modeled as a virtual ES, with the following considerations:

- The commands **network-interconnect-vxlan** and **service-id service-range svc-id [to svc-id]** are required within the ES.
  - The **network-interconnect-vxlan** parameter identifies the VXLAN instance associated with the virtual ES. The value of the parameter must be set to 1. This command is rejected in a non-virtual ES.
  - The **service-range** parameter associates the specific service range to the ES. The ES must be configured as **network-interconnect-vxlan** before any service range can be added.
  - The ES parameters **port**, **lag**, **sdp**, **vc-id-range**, **dot1q**, and **qinq** cannot be configured in the ES when a **network-interconnect-vxlan** instance is configured. The **source-bmac-lsb** option is blocked, as the I-ES cannot be associated with an I-VPLS or PBB-Epipe service. The remaining ES configuration options are supported.
  - All services with two BGP instances associate the VXLAN destinations and ingress VXLAN instances to the ES.
- Multiple services can be associated with the same ES, with the following considerations:
  - In a DC with two DCGWs (as in [Figure 101: The Interconnect ES concept](#)), only two I-ESs are needed to load-balance, where one half of the dual BGP-instance services would be associated with one I-ES (for example, I-ES-1, in the above configuration) and one half to another I-ES.
  - Up to eight service ranges per VXLAN instance can be configured. Ranges may overlap within the same ES, but not between different ESs.
  - The service range can be configured before the service.
- After the I-ES is configured using **network-interconnect-vxlan**, the ES operational state depends exclusively on the ES administrative state. Because the I-ES is not associated with a physical port or SDP, when testing the non-revertive service carving manual mode, an ES **shutdown** and **no shutdown** event results in the node sending its own administrative preference and DP bit and taking control if the preference and DP bit are higher than the current DF. This is because the peer ES routes are not present at the EVPN application layer when the ES is configured for **no shutdown**; therefore, the PE sends its own administrative preference and DP values. For I-ESs, the non-revertive mode works only for node failures.
- A VXLAN instance may be placed in MhStandby under any of the following situations:
  - if the PE is single-active NDF for that I-ES
  - if the VXLAN service is added to the I-ES and either the ES or BGP-EVPN MPLS is shut down in all the services included in the ES



The following example shows the change of the MhStandby flag from false to true when BGP-EVPN MPLS is shut down for all the services in the I-ES.

```
A:PE-4# show service id 500 vxlan instance 1 oper-flags
=====
VPLS VXLAN oper flags
=====
MhStandby                               : false
=====
A:PE-4# configure service vpls 500 bgp-evpn vxlan shutdown
*A:PE-4# show service id 500 vxlan instance 1 oper-flags
=====
VPLS VXLAN oper flags
=====
MhStandby                               : true
=====
```

#### 5.4.9.6.1 BGP-EVPN routes on dual BGP-instance services with I-ES

The configuration of an I-ES on DCGWs with two BGP instances has the following impact on the advertisement and processing of BGP-EVPN routes.

- For EVPN MAC/IP routes, the following considerations apply:
  - If **bgp-evpn>vxlan>no auto-disc-route-advertisement** and **mh-mode access** are configured on the access instance:
    - MAC/IP routes received in the EVPN-MPLS BGP instance are readvertised in the EVPN-VXLAN BGP instance with the ESI set to zero.
    - EVPN-VXLAN PEs and NVEs in the DC receive the same MAC from two or more different MAC/IP routes from the DCGWs, which perform regular EVPN MAC/IP route selection.
    - MAC/IP routes received in the EVPN-VXLAN BGP instance are readvertised in the EVPN-MPLS BGP instance with the configured non-zero I-ESI value, assuming the VXLAN instance is not in an MhStandby operational state; otherwise the MAC/IP routes are dropped.
    - EVPN-MPLS PEs in the WAN receive the same MAC from two or more DCGWs set with the same ESI. In this case, regular aliasing and backup functions occur as usual.
  - If **bgp-evpn>vxlan>auto-disc-route-advertisement** and **mh-mode access** are configured, the following differences apply to the above:
    - MAC/IP routes received in the EVPN-MPLS BGP instance are readvertised in the EVPN-VXLAN BGP instance with the ESI set to the I-ESI.
    - In this case, EVPN-VXLAN PEs and NVEs in the DC receive the same MAC from two or more different MAC/IP routes from the DCGWs, with the same ESI, therefore they can perform aliasing.
- ES routes are exchanged for the I-ES. The routes should be sent only to the MPLS network and not to the VXLAN network. This can be achieved by using router policies.
- AD per-ES and AD per-EVI routes are also advertised for the I-ES, and are sent only to the MPLS network and not to the VXLAN if **bgp-evpn>vxlan>no auto-disc-route-advertisement** is configured. For ES routes, router polices can be used to prevent these routes from being sent to VXLAN peers. If **bgp-evpn>vxlan>auto-disc-route-advertisement** is configured, AD routes must be sent to the VXLAN peers so that they can apply backup or aliasing functions.

In general, when I-ESs are used for redundancy, the use of router policies is needed to avoid control plane loops with MAC/IP routes. Consider the following to avoid control plane loops:

- **loops created by remote MACs**

Remote EVPN-MPLS MAC/IP routes are readvertised into EVPN-VXLAN routes with an SOO (Site Of Origin) EC added by a BGP peer or VSI export policy identifying the DCGW pair. The other DCGW in the pair drops EVPN-VXLAN MAC/IP routes tagged with the pair SOO. Router policies are needed to add SOO and drop routes received with self SOO.

When remote EVPN-VXLAN MAC/IP routes are readvertised into EVPN-MPLS, the DCGWs automatically drop EVPN-MPLS MAC/IP routes received with their own non-zero I-ESI.

- **loops created by local SAP MACs**

Local SAP MACs are learned and MAC/IP routes are advertised into both BGP instances. The MAC/IP routes advertised in the EVPN-VXLAN instance are dropped by the peer based on the SOO router policies as described above for loops created by remote MACs. The DCGW local MACs are always learned over the EVPN-MPLS destinations between the DCGWs.

The following describes the considerations for BGP peer policies on DCGW1 to avoid control plane loops. Similar policies would be configured on DCGW2.

- Avoid sending service VXLAN routes to MPLS peers and service MPLS routes to VXLAN peers.
- Avoid sending AD and ES routes to VXLAN peers. If **bgp-evpn>vxlan>auto-disc-route-advertisement** is configured AD routes must be sent to the VXLAN peers.
- Add SOO to VXLAN routes sent to the ES peer.
- Drop VXLAN routes received from the ES peer.

The following shows the CLI configuration:

```
A:DCGW1# configure router bgp
A:DCGW1>config>router>bgp# info
-----
    family vpn-ipv4 evpn
    vpn-apply-import
    vpn-apply-export
    rapid-withdrawal
    rapid-update vpn-ipv4 evpn
    group "wan"
        type internal
        export "allow only mpls"
        neighbor 192.0.2.4
        exit
        neighbor 192.0.2.5
        exit
    exit
    group "internal"
        type internal
        neighbor 192.0.2.1
            export "allow only vxlan"
        exit
        neighbor 192.0.2.3
            import "drop S00-DCGW-23"
            export "add S00 to vxlan routes"
        exit
    exit
    no shutdown
-----
A:DCGW1>config>router>bgp# /configure router policy-options
A:DCGW1>config>router>policy-options# info
```

```
-----
community "mpls" members "bgp-tunnel-encap:MPLS"
community "vxlan" members "bgp-tunnel-encap:VXLAN"
community "S00-DCGW-23" members "origin:64500:23"
```

```
// This policy prevents the router from sending service VXLAN routes to MPLS peers. //
```

```
policy-statement "allow only mpls"
  entry 10
    from
      community "vxlan"
      family evpn
    exit
    action drop
    exit
  exit
exit
```

This policy ensures the router only exports routes that include the VXLAN encapsulation.

```
policy-statement "allow only vxlan"
  entry 10
    from
      community "vxlan"
      family evpn
    exit
    action accept
    exit
  exit
  default-action drop
  exit
exit
```

This import policy avoids importing routes with a self SOO.

```
policy-statement "drop S00-DCGW-23"
  entry 10
    from
      community "S00-DCGW-23"
      family evpn
    exit
    action drop
    exit
  exit
exit
```

This import policy adds SOO only to VXLAN routes. This allows the peer to drop routes based on the SOO, without affecting the MPLS routes.

```
policy-statement "add S00 to vxlan routes"
  entry 10
    from
      community "vxlan"
      family evpn
    exit
    action accept
      community add "S00-DCGW-23"
    exit
  exit
```

```

default-action accept
exit
exit

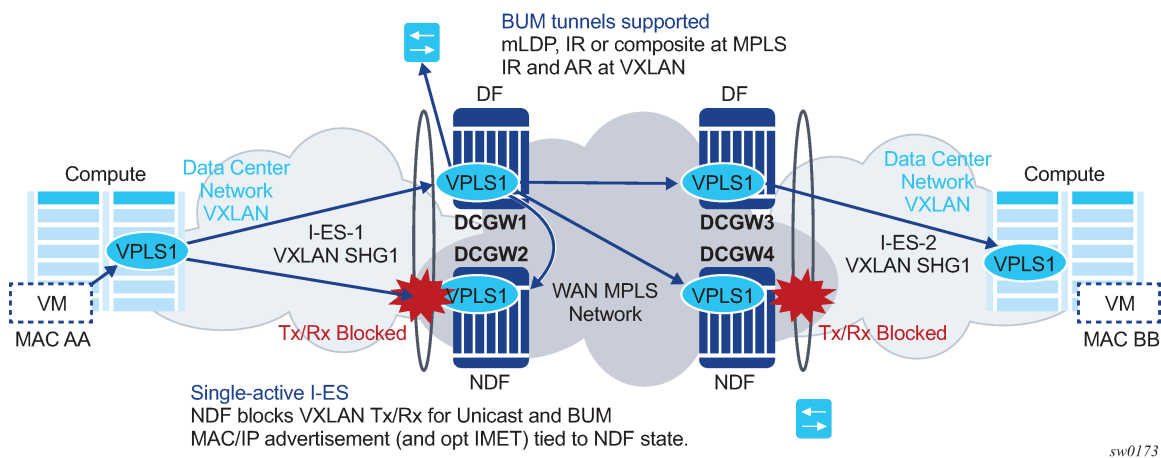
```

#### 5.4.9.6.2 Single-active multihoming on I-ES

When an I-ES is configured as single-active and configured as **no shutdown** with at least one associated service, the DCGWs send ES and AD routes as for any ES. It also runs DF election as normal, based on the ES routes, with the candidate list being pruned by the AD routes.

Figure 102: I-ES — single-active shows the expected behavior for a single-active I-ES.

Figure 102: I-ES — single-active



As shown in Figure 102: I-ES — single-active, the Non-Designated Forwarder (NDF) for a specified service carries out the following tasks:

- From a datapath perspective, the VXLAN instance on the NDF goes into an MhStandby operational state and blocks ingress and egress traffic on the VXLAN destinations associated with the I-ES.
- The MAC/IP routes and the FDB process
  - MAC/IP routes associated with the VXLAN instance and readvertised to EVPN-MPLS peers are withdrawn.
  - MAC/IP routes corresponding to local SAP MACs or EVPN-MPLS binding MACs are withdrawn if they were advertised to the EVPN-VXLAN instance.
  - Received MAC/IP routes associated with the VXLAN instance are not installed in the FDB. MAC/IP routes show as "used" in BGP; however, only the MAC/IP route received from MPLS (from the ES peer) is programmed.
- The Inclusive Multicast Ethernet Tag (IMET) routes process
  - IMET-AR-R routes (IMET-AR with replicator role) must be withdrawn if the VXLAN instance goes into an MhStandby operational state. Only the DF advertises the IMET-AR-R routes.
  - IMET-IR advertisements in the case of the NDF (or MhStandby) are controlled by the command **config>service>vpls>bgp-evpn>vxlan [no] send-imet-ir-on-ndf**.

By default, the command is enabled and the router advertises IMET-IR routes, even if the PE is NDF (MhStandby). This attracts BUM traffic, but also speeds up convergence in the case of a DF switchover. The command is supported for single-active and all-active.

If the command is disabled, the router withdraws the IMET-IR routes when the PE is NDF and do not attract BUM traffic.

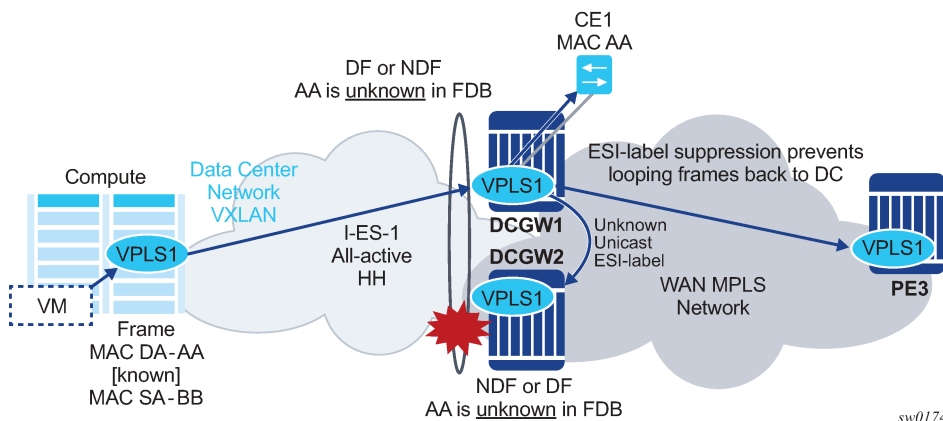
The I-ES DF PE for the service continues advertising IMET and MAC/IP routes for the associated VXLAN instance as usual, as well as forwarding on the DF VXLAN bindings. When the DF DCGW receives BUM traffic, it sends the traffic with the egress ESI label if needed.

### 5.4.9.6.3 All-active multihoming on I-ES

The same considerations for ES and AD routes, and DF election apply for all-active multihoming as for single-active multihoming; the difference is in the behavior on the NDF DCGW. The NDF for a specified service performs the following tasks:

- From a datapath perspective, the NDF blocks ingress and egress paths for broadcast and multicast traffic on the VXLAN instance bindings associated with the I-ES, while unknown and known unicast traffic is still allowed. The unknown unicast traffic is transmitted on the NDF if there is no risk of duplication. For example, unknown unicast packets are transmitted on the NDF if they do not have an ESI label, do not have an EVPN BUM label, and they pass a MAC SA suppression. In the example in [Figure 103: All-active multihoming and unknown unicast on the NDF](#), the NDF transmits unknown unicast traffic. Regardless of whether DCGW1 is a DF or NDF, it accepts the unknown unicast packets and floods to local SAPs and EVPN destinations. When sending to DGW2, the router sends the ESI label identifying the I-ES. Because of the ESI-label suppression, DCGW2 does not send unknown traffic back to the DC.

Figure 103: All-active multihoming and unknown unicast on the NDF



- The MAC/IP routes and the FDB process
  - MAC/IP routes associated with the VXLAN instance are advertised normally.
  - MACs are installed as normal in the FDB for received MAC/IP routes associated with the VXLAN instance.
- The IMET routes process

- As with single-active multihoming, IMET-AR-R routes must be withdrawn on the NDF (MhStandby state). Only the DF advertises the IMET-AR-R routes.
- The IMET-IR advertisements in the case of the NDF (or MhStandby) are controlled by the command **config>service>vpls>bgp-evpn>vxlan [no] send-imet-ir-on-ndf**, as in single-active multihoming.

The behavior on the non-DF for BUM traffic can also be controlled by the command

**config>service>vpls>vxlan>rx-discard-on-ndf {bm | bum | none}**, where the default option is **bm**.

However, the user can change this option to discard all BUM traffic, or forward all BUM traffic (none).

The I-ES DF PE for the service continues advertising IMET and MAC/IP routes for the associated VXLAN instance as usual. When the DF DCGW receives BUM traffic, it sends the traffic with the egress ESI label if needed.

#### 5.4.10 Multi-instance EVPN: Two instances of the same encapsulation in the same VPLS/R-VPLS service



**Note:** VXLAN is not supported on the 7705 SAR Gen 2. The configuration documented in this section is provided for reference only. The configuration for MPLS is similar to VXLAN.

As described in [Multi-instance EVPN: Two instances of different encapsulation in the same VPLS/R-VPLS/Epipse service](#), two BGP instances are supported in VPLS services, where one instance can be associated with the EVPN-VXLAN and the other instance with the EVPN-MPLS. In addition, both BGP instances in a VPLS/R-VPLS service can also be associated with EVPN-VXLAN, or both instances can be associated with EVPN-MPLS.

For example, a VPLS service can be configured with two VXLAN instances that use VNI 500 and 501 respectively, and those instances can be associated with different BGP instances:

```
*A:PE-2# configure service vpls 500
*A:PE-2>config>service>vpls# info
-----
vxlan instance 1 vni 500 create
exit
vxlan instance 2 vni 501 create
exit
bgp
  route-distinguisher 192.0.2.2:500
  vsi-export "vsi-500-export"
  vsi-import "vsi-500-import"
exit
bgp 2
  route-distinguisher 192.0.2.2:501
  vsi-export "vsi-501-export"
  vsi-import "vsi-501-import"
exit
bgp-evpn
  incl-mcast-orig-ip 23.23.23.23
  evi 500
  vxlan bgp 1 vxlan-instance 1
  no shutdown
exit
  vxlan bgp 2 vxlan-instance 2
  no shutdown
exit
exit
stp
shutdown
```

```
exit
no shutdown
-----
```

From a data plane perspective, each VXLAN instance is instantiated in a different implicit SHG, so that traffic can be forwarded between them.

In addition, multi-instance EVPN-VXLAN services support:

- assisted-replication for IPv4 VTEPs in both VXLAN instances, where a single assisted-replication IPv4 address can be used for both instances
- non-system IP and IPv6 termination, where a single **vxlan-src-vtep ip-address** can be configured for each service, and therefore used for the two instances

For example, a VPLS service can be configured with two EVPN-MPLS instances that are associated with two BGP instances as follows.

```
*A:PE-2# configure service vpls 700
*A:PE-2>config>service>vpls# info
-----
description "two bgp-evpn mpls instances"
bgp
  route-distinguisher auto-rd
  vsi-export "vsi-700-export"
  vsi-import "vsi-700-import"
exit
bgp 2
  route-distinguisher auto-rd
  vsi-export "vsi-701-export"
  vsi-import "vsi-701-import"
exit
bgp-evpn
  evi 700
  mpls bgp 1
    mh-mode access
    ingress-replication-bum-label
    auto-bind-tunnel
    resolution any
    exit
    no shutdown
  exit
  mpls bgp 2
    mh-mode network
    ingress-replication-bum-label
    auto-bind-tunnel
    resolution any
    exit
    no shutdown
  exit
exit
stp
  shutdown
exit
no shutdown
-----
```

Multi-instance EVPN-MPLS VPLS/R-VPLS services have the same limitations as any multi-instance service, as described in [Multi-instance EVPN: Two instances of the same encapsulation in the same VPLS/R-VPLS service](#). In addition, services with two EVPN-MPLS instances do not support SAPs.

The **mh-mode {network|access}** command in the **vpls>bgp-evpn>mpls** context determines which instance is considered access and which instance is considered network.

- The default form of the **bgp-evpn>mpls** command is **mh-mode network** and only one instance can be configured. The other instance must be configured as **mh-mode access**.
- The use of **provider-tunnel** is supported if there is one instance configured as **network**, and the P2MP tunnel is implicitly associated with the network instance.

Multi-instance EVPN-MPLS VPLS/R-VPLS services support:

- all of the **auto-bind-tunnel resolution** options in each of the two instances. This includes resolution of IPv4 next-hops to TTMv4 entries and resolution of IPv6 next-hops to TTMv6 entries.
- different address families in different instances. For instance, **mpls bgp 1** may resolve routes to TTMv4 and **mpls bgp 2** to TTMv6, or the reverse. In a VPLS service with two EVPN-VXLAN instances, it is not possible to have an instance with routes resolved to IPv4 VXLAN tunnels and the other instance with routes resolved to IPv6 VXLAN tunnels.
- an explicit **split-horizon-group** in each instance; however, the same split-horizon-group cannot be configured on the two instances of the same VPLS service
- a **restrict-protected-src discard-frame** per instance. If a MAC is protected in one instance and a frame arrives at the other instance with the protected MAC as source MAC, the frame is discarded if **restrict-protected-src discard-frame** is configured.

At the control plane level for two EVPN-VXLAN or two EVPN-MPLS instances, the processing of MAC/IP routes and inclusive multicast routes is described in [BGP-EVPN routes in services configured with two BGP instances](#) with the differences between the two scenarios described in [BGP-EVPN routes in multi-instance EVPN services with the same encapsulation](#).

#### 5.4.10.1 BGP-EVPN routes in multi-instance EVPN services with the same encapsulation

If two BGP instances with the same encapsulation (VXLAN or MPLS) are configured in the same VPLS/R-VPLS service, different import route targets in each BGP instance are mandatory (although this is not enforced).

[BGP-EVPN routes in services configured with two BGP instances](#) describes the use of policies to avoid sending WAN routes (routes meant to be redistributed from DC to WAN) to the DC again and DC routes (routes meant to be redistributed from WAN to DC) to the WAN again. Those policies are based on export policy statements that match on the RFC 9012 BGP encapsulation extended community (MPLS and VXLAN respectively).

When the two BGP instances are of the same encapsulation (VXLAN or MPLS), the policies matching on different BGP encapsulation extended community are not feasible because both instances advertise routes with the same encapsulation value. Because the export route targets in the two BGP instances must be different, the policies, to avoid sending WAN routes back to the WAN and DC routes back to the DC, can be based on export policies that prevent routes with a DC route target from being sent to the WAN peers (and opposite for routes with a WAN route target).

In scaled scenarios, matching based on route targets, does not scale well. An alternative and preferred solution is to configure a **default-route-tag** that identifies all the EVPN instances connected to the DC (or one domain), and a different **default-route-tag** in all the EVPN instances connected to the WAN (or the other domain). [Anycast redundant solution for multi-instance EVPN services with the same encapsulation](#) shows an example that demonstrates the use of **default-route-tags**.



Other than the specifications described in this section, the processing of MAC/IP routes and inclusive multicast Ethernet tag routes in multi-instance EVPN services of the same encapsulation follow the rules described in [BGP-EVPN routes in services configured with two BGP instances](#).

### 5.4.10.2 Anycast redundant solution for multi-instance EVPN services with the same encapsulation

The solution described in [Anycast redundant solution for dual BGP-instance services](#) is also supported in multi-instance EVPN VPLS/R-VPLS services with the same encapsulation.

The following CLI example output shows the configuration of DCGW-1 and DCGW-2 in [Figure 98: Multihomed anycast solution](#) where VPLS 500 is a multi-instance EVPN-VXLAN service and BGP instance 2 is associated with VXLAN instead of MPLS.

Different default-route-tags are used in BGP instance 1 and instance 2, so that in the export route policies, DC routes are not advertised to the WAN, and WAN routes are not advertised to the DC, respectively.

```
*A:DCGW-1(and DCGW-2)>config>service>vpls(500)# info
-----
vxlan instance 1 vni 500 create
exit
vxlan instance 2 vni 501 create
exit
bgp
  route-distinguisher 192.0.2.2:500
  route-target target:64500:500
exit
bgp 2
  route-distinguisher 192.0.2.2:501
  route-target target:64500:501
exit
bgp-evpn
  incl-mcast-orig-ip 23.23.23.23
  evi 500
  vxlan bgp 1 vxlan-instance 1
  default-route-tag 500
  no shutdown
exit
  vxlan bgp 2 vxlan-instance 2
  default-route-tag 501
  no shutdown
exit
exit
stp
shutdown
exit
no shutdown
-----
config>router>bgp#
vpn-apply-import
vpn-apply-export
group "WAN"
  family evpn
  type internal
  export "allow only mpls"
  neighbor 192.0.2.6
group "DC"
  family evpn
  type internal
  export "allow only vxlan"
```

```
neighbor 192.0.2.2
config>router>policy-options# info
-----
    policy-statement "allow only mpls"
      entry 10
        from
          family evpn
          tag 500
        action drop
      exit
    exit
  exit
  policy-statement "allow only vxlan"
    entry 10
      from
        family evpn
        tag 501
      action drop
    exit
  exit
exit
exit
```

The same Anycast redundant solution can be applied to VPLS/R-VPLS with two instances of EVPN-MPLS encapsulation. The configuration would be identical, other than replacing the VXLAN aspects with the EVPN-MPLS-specific parameters.

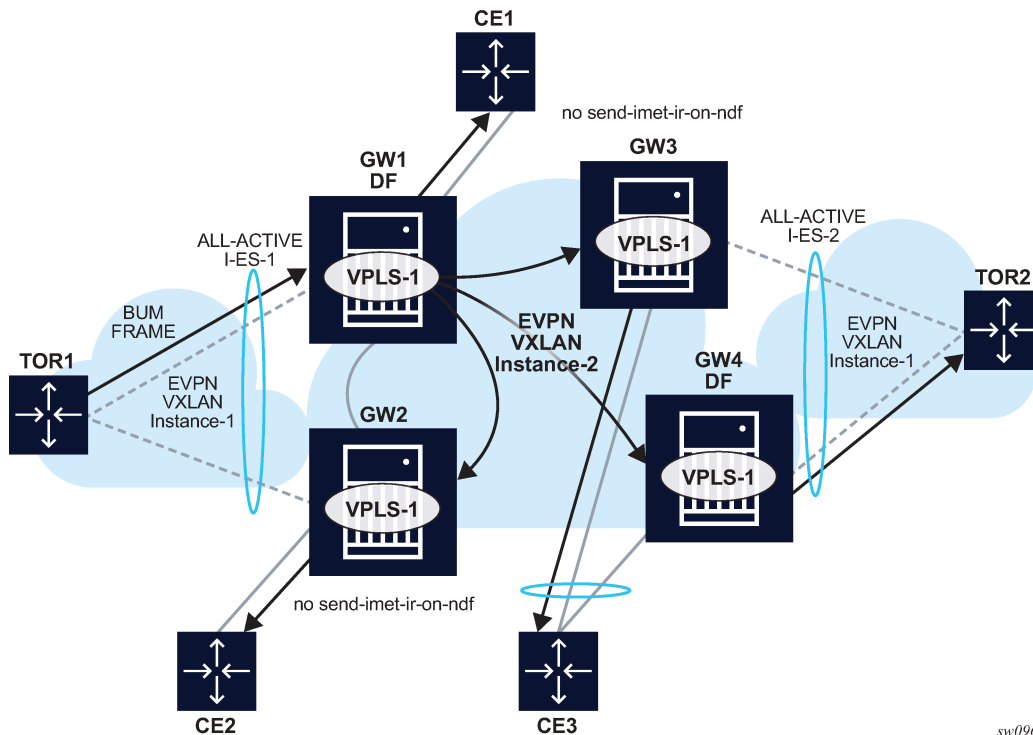
For a full description of this solution, see the [Anycast redundant solution for dual BGP-instance services](#)

### 5.4.10.3 I-ES solution for dual BGP EVPN instance services with the same encapsulation

The I-ES of network-interconnect VXLAN Ethernet segment is described in [I-ES solution for dual BGP instance services](#). I-ES's are also supported on VPLS and R-VPLS services with two EVPN-VXLAN instances.

[Figure 104: I-ES in dual EVPN-VXLAN services](#) shows the use of an I-ES in a dual EVPN-VXLAN instance service.

Figure 104: I-ES in dual EVPN-VXLAN services



sw0969

Similar to (single-instance) EVPN-VXLAN all-active multihoming, the BUM forwarding procedures follow the "Local Bias" behavior.

At the ingress PE, the forwarding rules for EVPN-VXLAN services are as follows:

- The **no send-imet-ir-on-ndf** or **rx-discard-on-ndf bum** command must be enabled so that the NDF does not forward any BUM traffic.
- BUM frames received on any SAP or I-ES VXLAN binding are flooded to:
  - local non-ES and single-active DF ES SAPs
  - local all-active ES SAPs (DF and NDF)
  - EVPN-VXLAN destinations

BUM received on an I-ES VXLAN binding follows SHG rules, for example, it can only be forwarded to EVPN-VXLAN destinations that belong to the other VXLAN instance (instance 2), which is a different SHG.

- As an example, in [Figure 104: I-ES in dual EVPN-VXLAN services](#):
  - GW1 and GW2 are configured with **no send-imet-ir-on-ndf**.
  - TOR1 generates BUM traffic that only reaches GW1 (DF).
  - GW1 forwards to CE1 and EVPN-VXLAN destinations.

The forwarding rules at the egress PE are as follows:

- The source VTEP is looked up for BUM frames received on EVPN-VXLAN.

- If the source VTEP matches one of the PEs with which the local PE shares an ES \_AND\_ a VXLAN service:
  - Then the local PE does not forward to the shared local ESs (this includes port, lag, or network-interconnect-vxlan ESs). It forwards though to non-shared ES SAPs unless they are in NDF state.
  - Else, the local PE forwards normally to local ESs unless they are in NDF state.
- Because there is no multicast label or multicast B-MAC in VXLAN, the only way the egress PE can identify BUM traffic is by looking at the customer MAC DA. Therefore, BM or unknown MAC DAs identify BUM traffic.
- As an example, in [Figure 104: I-ES in dual EVPN-VXLAN services](#):
  - GW2 receives BUM on EVPN-VXLAN. GW2 identifies the source VTEP as a PE with which the I-ES-1 is shared, therefore it does not forward the BUM frames to the local I-ES. It forwards to the non-shared ES and local SAPs though (CE2).
  - GW3 receives BUM on EVPN-VXLAN, however the source VTEP does not match any PE with which GW3 shares an ES. Hence GW3 forwards to all local ESs that are DF, in other words, CE3.

The following configuration example shows how I-ES-1 would be provisioned on DCGW1 and the association between I-ES to a specified VPLS service. A similar configuration would occur on DCGW2 in the I-ES.

I-ES configuration:

```
*A:GW1>config>service>system>bgp-evpn>eth-seg# info
-----
esi 00:23:23:23:23:23:00:00:01
service-carving
  mode manual
  manual
    preference non-revertive create
    value 150
  exit
  evi 101 to 200
exit
exit
multi-homing all-active
network-interconnect-vxlan 1
service-id
  service-range 1
  service-range 1000 to 1002
  service-range 2000
exit
no shutdown
```

Service configuration:

```
*A:GW1>config>service>vpls# info
-----
vxlan instance 1 vni 1000 create
  rx-discard-on-ndf bum
exit
vxlan instance 2 vni 1002 create
exit
bgp
  route-target export target:64500:1000 import target:64500:1000
exit
bgp 2
  route-distinguisher auto-rd
```

```

        route-target export target:64500:1002 import target:64500:1002
    exit
    bgp-evpn
        evi 1000
        vxlan bgp 1 vxlan-instance 1
            ecmp 2
            default-route-tag 100
            auto-disc-route-advertisement
            no shutdown
        exit
        vxlan bgp 2 vxlan-instance 2
            ecmp 2
            default-route-tag 200
            auto-disc-route-advertisement
            mh-mode network
            no shutdown
        exit
    exit
    no shutdown

```

Multi-instance EVPN VPLS/R-VPLS services with two EVPN-MPLS instances do not support I-ESs.

For information about how the EVPN routes are processed and advertised in an I-ES, see the [I-ES solution for dual BGP instance services](#).

### 5.4.11 EVPN IP-VRF-to-IP-VRF models

SR OS supports the three IP-VRF-to-IP-VRF models defined in RFC 9136 for EVPN services. Those three models are known as:

- interface-ful IP-VRF-to-IP-VRF with Supplementary Bridge Domain Integrated Routing Bridging (SBD IRB)
- interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB
- interface-less IP-VRF-to-IP-VRF

The three models refer to different control and data plane procedures to advertise and process the EVPN IP-Prefix routes. The interface-less model also supports the advertisement and processing of host routes by using EVPN MAC/IP Advertisement routes (instead of IP-Prefix routes), as specified in the symmetric IRB model in RFC 9135.

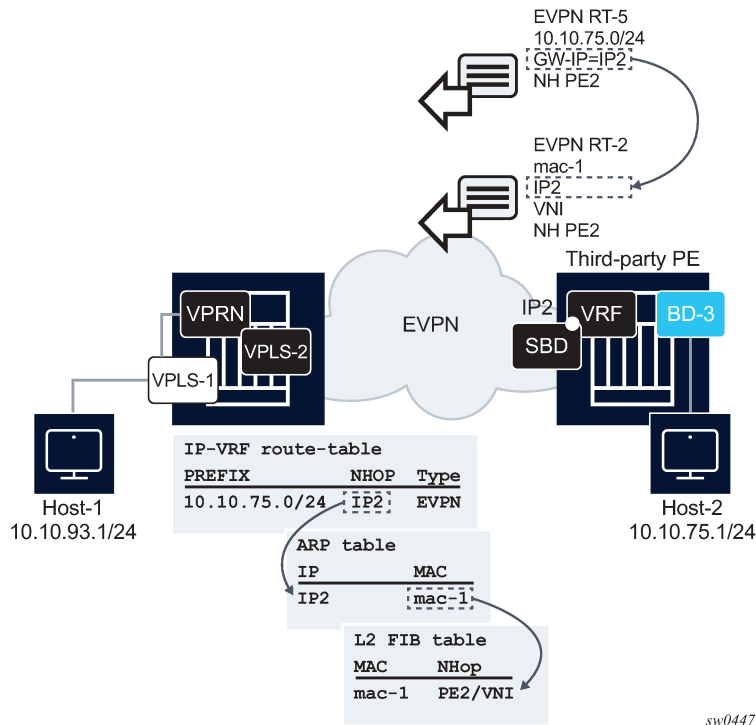
SR OS supports the preceding models for IPv4 and IPv6 prefixes. Vendors may choose different models depending on the use cases that they intend to address. When a third-party vendor is connected to an SR OS router, it is important to know which of the three models the third-party vendor implements.

#### 5.4.11.1 Interface-ful IP-VRF-to-IP-VRF with SBD IRB model

The SBD is equivalent to an R-VPLS that connects all the PEs that are attached to the same tenant VPRN. Interface-ful refers to the use of a full IRB interface between the VPRN and the SBD (an interface object with MAC and IP addresses, over which interface parameters can be configured).

The following figure shows an example of the interface-ful IP-VRF-to-IP-VRF with SBD IRB model.

Figure 105: Interface-ful IP-VRF-to-IP-VRF with SBD IRB model



In the preceding figure, an SR OS router and a third-party router are using the interface-ful IP-VRF-to-IP-VRF with SBD IRB model. The two routers are attached to a VPRN for the same tenant, and those VPRNs are connected by R-VPLS-2, or SBD. Both routers exchange IP prefix routes with a non-zero gateway IP, which is the IP address of the SBD IRB. The SBD IRB MAC and IP are advertised in a MAC/IP route. On reception, the IP prefix route creates a route-table entry in the VPRN, where the gateway IP must be recursively resolved to the information provided by the MAC/IP route and installed in the ARP and FDB tables.

This model is detailed in [EVPN for VXLAN in IRB backhaul R-VPLS services and IP prefixes](#). The following is an example of the configuration of SBD and VPRN, as shown in [Figure 105: Interface-ful IP-VRF-to-IP-VRF with SBD IRB model](#).

```
A:node-2>config>service

vpls 2 customer 1 name "sbd" create
allow-ip-int-bind
exit
bgp
exit
bgp-evpn
evi 2
ip-route-advertisement
mpls bgp 1
auto-bind-tunnel resolution any
no shutdown

vprn 1 customer 1 name "vprn1" create
route-distinguisher auto-rd
interface "sbd" create
address 192.168.0.1/16
```

```

ipv6
 30::3/64
exit
vpls "sbd"

```

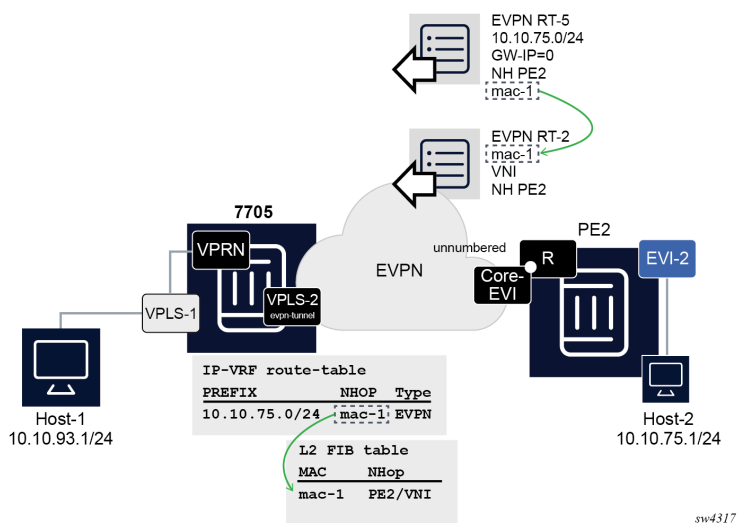
The interface-ful IP-VRF-to-IP-VRF with SBD IRB model is also supported for IPv6 prefixes. There are no configuration differences except the ability to configure an IPv6 address and interface.

#### 5.4.11.2 Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model

Interface-ful refers to the use of a full IRB interface between the VPRN and the SBD. However, the SBD IRB is unnumbered in this model, which means no IP address is configured on it. In SR OS, an unnumbered SBD IRB is equivalent to an R-VPLS that is linked to a VPRN interface through an EVPN tunnel. See [EVPN for VXLAN in EVPN tunnel R-VPLS services](#) for more information.

The following figure shows an example of the interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model.

Figure 106: Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model



In the preceding figure, an SR OS router and a third-party router are using the interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model. The IP prefix routes are expected to have a zero gateway IP, and the MAC in the router's MAC extended community is used for the recursive resolution to a MAC/IP route.

The corresponding configuration of the VPRN and SBD in the example could be:

```

A:node-2>config>service

vpls 2 customer 1 name "sbd" create
allow-ip-int-bind
exit
bgp
exit
bgp-evpn
evi 2
ip-route-advertisement
mpls bgp 1

```

```
auto-bind-tunnel resolution any
no shutdown

vprn 1 customer 1 create
route-distinguisher auto-rd
interface "sbd" create
  ipv6
  exit
vpls "sbd"
  evpn-tunnel ipv6-gateway-address mac
```

The **evpn-tunnel** command controls the use of the router's MAC extended community and the zero gateway IP in the IPv4-prefix route. For IPv6, the **ipv6-gateway-address mac** command allows the router to advertise the IPv6-prefix routes with a router's MAC extended community and zero gateway IP.

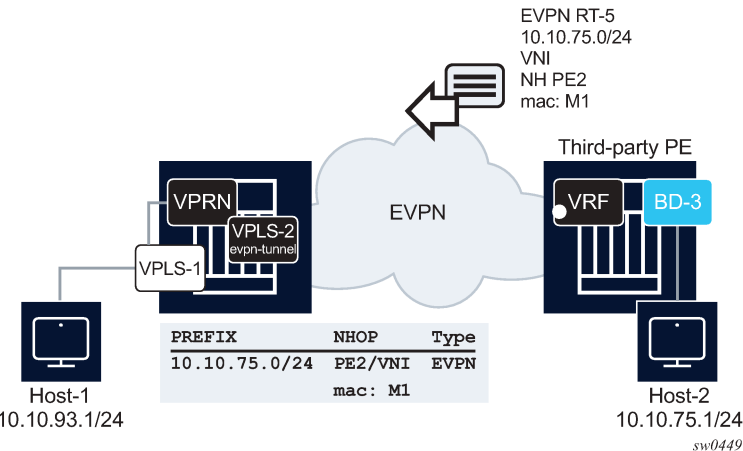
5.4.11.3 Interoperable interface-less IP-VRF-to-IP-VRF model (Ethernet encapsulation)

The interface-less model does not require a Supplementary Broadcast Domain (SBD) connecting the VPRNs for the tenant, and no recursive resolution is required upon receiving an IP prefix route. In other words, the next-hop of the IP prefix route is directly resolved to an EVPN tunnel without the need for any other route. The standard specification RFC 9136 supports two variants of this model that are not interoperable.

- EVPN IFL for Ethernet Network Virtualization Overlay (NVO) tunnels**  
Ethernet NVO indicates that the EVPN packets contain an inner Ethernet header. This is the case for tunnels such as VXLAN  
  
In the Ethernet NVO option, the ingress PE uses the received router's MAC extended community address (received along with the route type 5) as the inner destination MAC address for the EVPN packets sent to the prefix.
- EVPN IFL for IP NVO tunnels**  
IP NVO indicates that the EVPN packets contain an inner IP packet, but without Ethernet header. This is similar to the IPVPN packets exchanged between PEs.

The following figure shows an example of the interface-less IP-VRF-to-IP-VRF model.

Figure 107: Interface-less IP-VRF-to-IP-VRF model





SR OS supports the interoperable Interface-less IP-VRF-to-IP-VRF Model for Ethernet NVO tunnels. In the preceding figure, this interoperable is shown on the left-most PE router. The following is the model implementation.

- There is no datapath difference between this model and the existing R-VPLS EVPN tunnel model or the model described in [Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model](#).
- This model is enabled by configuring the **config service vprn if vpls evpn-tunnel** (with **ipv6-gateway-address mac** for IPv6) and **bgp-evpn ip-route-advertisement** commands. In addition, because the SBD IRB MAC/IP route is no longer needed, the **config service vpls bgp-evpn no mac-advertisement** command prevents the advertisement of the MAC/IP route.
- The following IP prefix routes are processed as follows.
  - On transmission, there is no change in the IP prefix route processing compared to the configuration of the [Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model](#).
    - IPv4 or IPv6 prefix routes are advertised, based on the information in the route-table for IPv4 and IPv6, with GW-IP=0 and the corresponding MAC extended community.
    - If the **bgp-evpn no mac-advertisement** command is configured, no MAC/IP route is sent for the R-VPLS.
  - The received IPv4 or IPv6 prefix routes are processed as follows.
    - Upon receiving an IPv4 or IPv6 prefix route with a MAC extended community for the router, an internal MAC/IP route is generated with the encoded MAC and the RD, Ethernet tag, ESI, Label/VNI and next hop from the IP prefix route itself.
    - If no competing received MAC/IP routes exists for the same MAC, this IP prefix-derived MAC/IP route is selected and the MAC installed in the R-VPLS FDB with type "Evpn".
    - After the MAC is installed in FDB, there are no differences between this interoperable interface-less model and the interface-ful with unnumbered SBD IRB model. Therefore SR OS is compatible with the received IP prefix routes for both models.

The following is an example of a typical configuration of a PE's SBD and VPRN that work in interface-less model for IPv4 and IPv6.

```
A:node-2>config>service
vpls 2 customer 1 name "sbd" create
    allow-ip-int-bind
    exit
    bgp
    exit
    bgp-evpn
        evi 2
        no mac-advertisement
        ip-route-advertisement
        mpls bgp 1
            auto-bind-tunnel resolution any
            no shutdown
vprn 1 customer 1 create
    route-distinguisher auto-rd
    interface "sbd" create
        ipv6
        exit
        vpls "sbd"
            evpn-tunnel ipv6-gateway-address mac
```

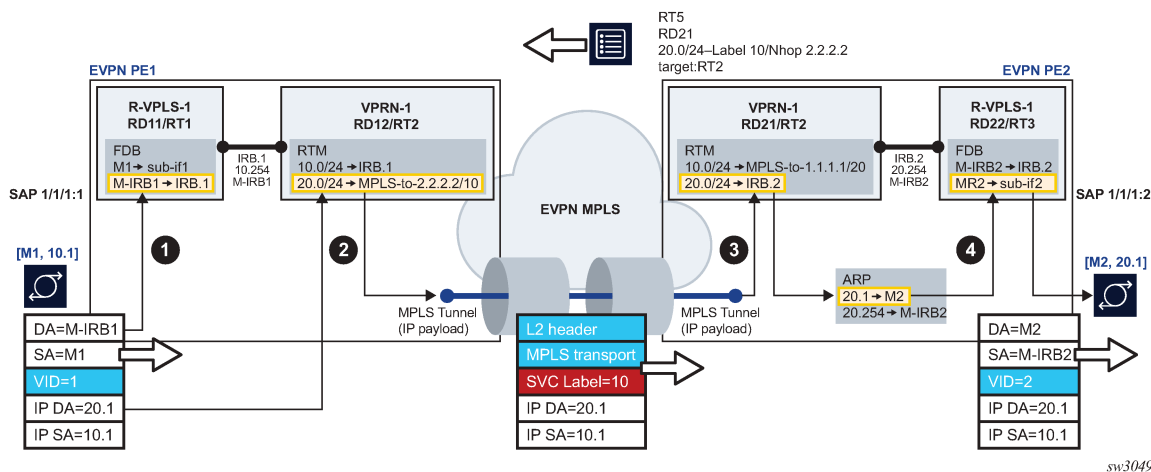
### 5.4.11.4 Interface-less IP-VRF-to-IP-VRF model (IP encapsulation) for MPLS tunnels

In addition to the Interface-ful and interoperable Interface-less models described in [Interface-ful IP-VRF-to-IP-VRF with SBD IRB model](#), [Interface-ful IP-VRF-to-IP-VRF with unnumbered SBD IRB model](#), and [Interoperable interface-less IP-VRF-to-IP-VRF model \(Ethernet encapsulation\)](#) sections, SR OS also supports the Interface-less Model (EVPN IFL) with IP encapsulation for MPLS tunnels. The RFC 9136 standard specification refers to this model as the EVPN IFL model for IP NVO tunnels.

Compared to the Ethernet NVO model in which the ingress PE pushes an inner Ethernet header, the IP packet in this EVPN IFL model is directly encapsulated with an EVPN service label and the transport labels.

The following figure shows the EVPN IFL model with IP encapsulation for MPLS tunnels.

Figure 108: Interface-less IP-VRF-to-IP-VRF model for IP encapsulation in MPLS tunnels



EVPN IFL uses EVPN IP Prefix routes to exchange prefixes between PEs without the need for an R-VPLS service, termed Supplementary Broadcast Domain (SBD) in the standards, and any destination MAC lookup. The EVPN IFL uses the same datapath that is used for IP-VPN services in the VPRN.

In the example shown in [Figure 108: Interface-less IP-VRF-to-IP-VRF model for IP encapsulation in MPLS tunnels](#), the following applies.

- PE2 advertises IP Prefix 20.0/24 (shorthand for 20.0.0.0/24) in an EVPN IP Prefix route that no longer contains a router's MAC extended community. In step 1, per usual processing, arriving frames with IP destination 20.0.0.1 on PE1's R-VPLS-1 are processed for a route lookup on VPRN-1.
- In step 2, in contrast to and as opposed to the other EVPN Layer 3 models, the lookup yields a route-table entry that does not point at an SBD R-VPLS, but points instead to an MPLS tunnel terminated on PE2. PE1 then pushes the EVPN service label received on the IP Prefix route at the top of the IP packet, and the packet is sent on the wire without an inner Ethernet header.
- In step 3, the MPLS tunnel is terminated on PE2 and the EVPN label identifies the VPRN-1 service for a route lookup.
- In Step 4, the processing corresponds to the regular R-VPLS forwarding that occurs in the other EVPN Layer 3 models.

Use the `vprn>bgp-evpn>mpls` context to configure a VPRN service for EVPN IFL with MPLS encapsulation. This context which is similar to the existing contexts in VPLS and Epipe services, enables

the use of EVPN IFL in the VPRN service. When this context is enabled, no R-VPLS with **evpn-tunnel** should be added to the VPRN; that is, the user must ensure that no SBD is configured. For example, in [Figure 108: Interface-less IP-VRF-to-IP-VRF model for IP encapsulation in MPLS tunnels](#), PE1 and PE2 VPRN-1 services are configured as follows.

```
[ex:configure service vprn "vprn-1"]
A:admin@PE1# info
  admin-state enable
  ecmp 2
  bgp-evpn {
    mpls 1 {
      admin-state enable
      route-distinguisher "192.0.2.1:12"
      vrf-target {
        community "target:64500:2"
      }
      auto-bind-tunnel {
        resolution any
      }
    }
  }
  interface "irb-1" {
    ipv4 {
      primary {
        address 10.0.0.254
        prefix-length 24
      }
    }
    vpls "r-vpls-1" {
    }
  }
}
```

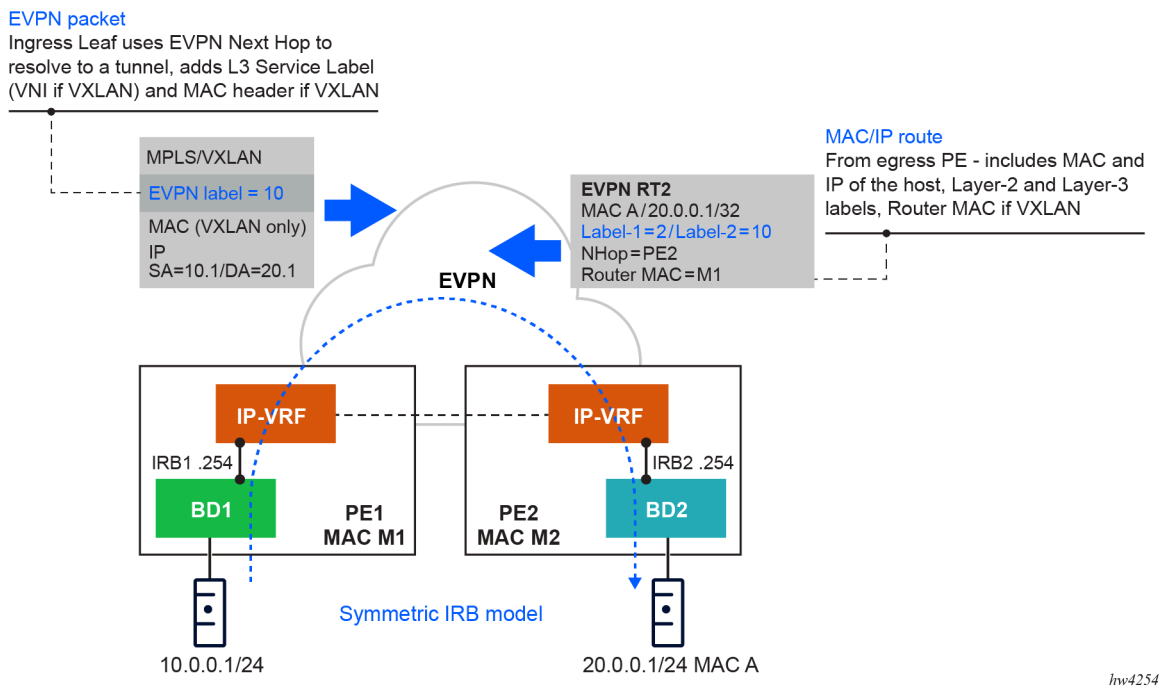
```
[ex:configure service vprn "vprn-1"]
A:admin@PE2# info
  admin-state enable
  ecmp 2
  bgp-evpn {
    mpls 1 {
      admin-state enable
      route-distinguisher "192.0.2.2:21"
      vrf-target {
        community "target:64500:2"
      }
      auto-bind-tunnel {
        resolution any
      }
    }
  }
  interface "irb-2" {
    ipv4 {
      primary {
        address 20.0.0.254
        prefix-length 24
      }
    }
    vpls "r-vpls-1" {
    }
  }
}
```

### 5.4.11.5 EVPN MAC/IP advertisement route install into the route table

SR OS supports the symmetric IRB model specified in RFC 9135, by which, when enabled, the router advertises the learned MAC and IP addresses of a host in an EVPN MAC/IP Advertisement route. This route contains not only the label and route target of the R-VPLS service where the host is learned, but also the label and the route target of the VPRN attached to the R-VPLS service. This is possible because the MAC/IP Advertisement route contains two labels in the NLRI: label-1 (for Layer 2 purposes) and label-2 (for Layer 3 purposes).

On reception, the remote router receives the EVPN MAC/IP Advertisement route and programs the IP address as a host route (/32 or /128) in the VPRN route table. Although host routes can be advertised in IP-Prefix routes, the symmetric IRB model allows the advertisement of the MAC and host IP information in a single NLRI. The following figure shows this model.

Figure 109: Symmetric IRB model



The symmetric IRB behavior is enabled by the following command.

```
configure service interface vpls evpn arp advertise interface-less-routing bgp-evpn-instance
configure service interface vpls evpn nd advertise interface-less-routing bgp-evpn-instance
```

The **bgp-evpn-instance** refers to the VPRN EVPN instance that is used to grab the second route target and second label. A typical configuration example of a VPRN using this command follows.

- **MD-CLI**

```
[ex:/configure service vprn "EVPN-IFL-HOST-200"]
A:admin@node-2# info
  admin-state enable
  service-id 200
```

```

customer "1"
  bgp-evpn {
    mpls 1 {
      admin-state enable
      route-distinguisher "192.0.2.2:200"
      evi 200
      vrf-target {
        community "target:64500:200"
      }
      auto-bind-tunnel {
        resolution any
      }
    }
  }
  interface "BD-201" {
    ipv4 {
      primary {
        address 10.0.0.254
        prefix-length 24
      }
      neighbor-discovery {
        learn-unsolicited true
        proactive-refresh true
        static-neighbor 10.0.0.100 {
          mac-address 00:de:ad:be:ef:00
        }
      }
      vrrp 1 {
        backup [10.0.0.254]
        owner true
        passive true
      }
    }
    vpls "BD-201" {
      evpn {
        arp {
          learn-dynamic false
          advertise static {
            route-tag 200
            interface-less-routing {
            }
          }
          advertise dynamic {
            route-tag 200
            interface-less-routing {
            }
          }
        }
      }
    }
  }
}

```

- **Classic CLI**

```

A:node-2# configure service vprn 200
A:node-2>config>service>vprn# info
-----
      interface "BD-201" create
        address 10.0.0.254/24
        arp-learn-unsolicited
        arp-proactive-refresh
        static-arp 10.0.0.100 00:de:ad:be:ef:00
        vrrp 1 owner passive
          backup 10.0.0.254

```

```

        exit
        vpls "BD-201"
        evpn
            arp
                no learn-dynamic
                advertise static route-tag 200 interface-less-routing bgp-evpn-
instance 1
                advertise dynamic route-tag 200 interface-less-routing bgp-evpn-
instance 1
        exit
    exit
    exit
    exit
    bgp-evpn
    mpls
        auto-bind-tunnel
        resolution any
    exit
    evi 200
        route-distinguisher 192.0.2.2:200
        vrf-target target:64500:200
        no shutdown
    exit
    exit
    no shutdown
-----

```

The IP addresses received in MAC/IP Advertisement routes with non-zero label-2 and the VPRN route target are programmed in the route table of the receiving router with route owner EVPN-IFL-HOST, as follows.

Use the following command to display route table information.

```
show router 200 route-table 10.0.0.202/32
```

### Output example

```

=====
Route Table (Service: 200)
=====
Dest Prefix[Flags]                Type   Proto   Age      Pref
  Next Hop[Interface Name]                Metric
-----
10.0.0.202/32                    Remote EVPN-IFL* 00h04m11s 170
  2001:db8::2 (tunneled)                10
-----
No. of Routes: 1
Flags: n = Number of times nexthop is repeated
      B = BGP backup route available
      L = LFA nexthop available
      S = Sticky ECMP requested
=====
* indicates that the corresponding row element may have been truncated.

```

Use the following command to display extensive route table information.

```
show router 200 route-table 10.0.0.202/32 extensive
```

## Output example

```

=====
Route Table (Service: 200)
=====
Dest Prefix      : 10.0.0.202/32
Protocol        : EVPN-IFL-HOST
Age             : 00h04m18s
Preference      : 170
Indirect Next-Hop : 2001:db8::2
Label           : 524280
VPN Next-Hop Index : 10
QoS             : Priority=n/c, FC=n/c
Source-Class    : 0
Dest-Class      : 0
ECMP-Weight     : N/A
Resolving Next-Hop : 2001:db8::2 (LDP tunnel)
Metric          : 10
ECMP-Weight     : N/A
-----
No. of Destinations: 1
=====

```

Route policies can match EVPN-IFL-HOST routes installed in the VPRN route table. In addition to the supported policy qualifiers for MAC/IP Advertisement routes (route type, IP-prefix list, communities, or family), the qualifier **from protocol evpn-ifl-host** can be used in VRF export or peer export policies. Use the following command to configure the qualifier:

- **MD-CLI**

```
configure policy-options policy-statement entry from protocol name
```

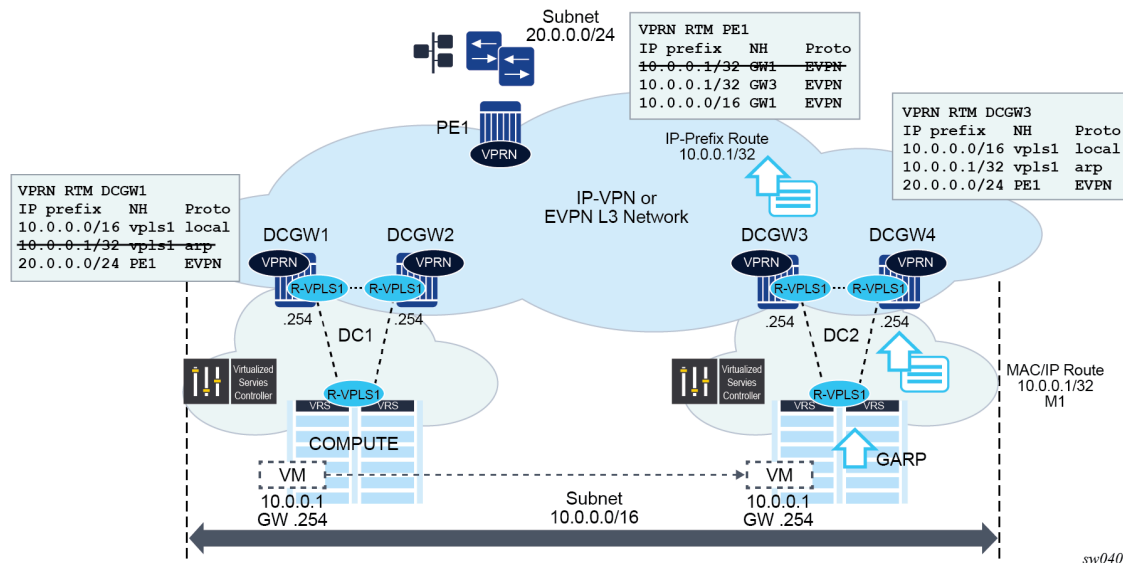
- **classic CLI**

```
configure router policy-options policy-statement entry from protocol evpn-ifl-host
```

### 5.4.12 ARP-ND host routes for extended Layer 2 data centers

SR OS supports the creation of host routes for IP addresses that are present in the ARP or neighbor tables of a routing context. These host routes, also called ARP-ND routes, can be advertised using EVPN or IP-VPN families. The following figure shows a typical use case where ARP-ND routes are used for extension of Layer 2 data centers (DCs).

Figure 110: Extended Layer 2 data centers



Subnet 10.0.0.0/16 in the preceding figure is extended throughout two DCs. The DC gateways are connected to the users of subnet 20.0.0.0/24 on PE1 using IP-VPN (or EVPN). If the virtual machine VM 10.0.0.1 is connected to DC1, PE1 performs a longest prefix match (LPM) lookup on the VPRN's route table when it needs to send traffic to host 10.0.0.1. If the only IP prefix advertised by the four DC GWs was 10.0.0.0/16, PE1 could send the packets to the DC where the VM is not present.

To provide efficient downstream routing to the DC where the VM is located, DCGW1 and DCGW2 must generate host routes for the VMs to which they connect. When the VM moves to the other DC, DCGW3 and DCGW4 must be able to learn the VM's host route and advertise it to PE1. DCGW1 and DCGW2 must withdraw the route for 10.0.0.1, because the VM is no longer in the local DC.

In this case, SR OS is able to learn the VM's host route from the generated ARP/ND messages when the VM boots or moves.

A route owner type called "ARP-ND" is supported in the base or VPRN route table. The ARP-ND host routes have a preference of 1 in the route table and are automatically created from the ARP/ND neighbor entries in the router instance.

The following commands enable ARP-ND host routes to be created in the applicable route tables:

- **config service vprn/ies interface arp-host-route populate {evpn | dynamic | static}**
- **config service vprn/ies interface ipv6 nd-host-route populate {evpn | dynamic | static}**

When the **config service vprn/ies interface arp-host-route populate** command is enabled, the EVPN, dynamic, and static ARP entries of the routing context create ARP-ND host routes in the route table. Similarly, ARP-ND host routes are created in the IPv6 route table from static, dynamic, and EVPN neighbor entries if the **config service vprn/ies interface ipv6 nd-host-route populate** command is enabled.

The **arp-host-route populate** and **nd-host-route populate** commands are used with the following features:

- **adding ARP-ND hosts**

A route tag can be added to ARP-ND hosts using the **route-tag** commands. This tag can be matched on BGP VRF export and peer export policies.



- **keeping entries active**

The ARP-ND host routes are kept in the route table as long as the corresponding ARP or neighbor entry is active. Even if there is no traffic destined for them, the **arp-proactive-refresh** and **nd-proactive-refresh** commands configure the node to keep the entries active by sending an ARP refresh message 30 seconds before the **arp-timeout** or starting a confirmation message (NUD) when the stale time expires.

- **speeding up learning**

Configure the **arp-learn-unsolicited** and **nd-learn-unsolicited** commands to speed up the learning of the ARP-ND host routes. When **arp-learn-unsolicited** is enabled, received unsolicited ARP messages (typically GARPs) create an ARP entry; consequently, an ARP-ND route is created if **arp-host-route populate** is enabled. Similarly, unsolicited Neighbor Advertisement messages create a stale neighbor. If **nd-host-route populate** is enabled, NUD is sent for all the neighbor entries created as stale, and, if confirmed, the corresponding ARP-ND routes are added to the route table.



**Note:** The ARP-ND host routes are created in the route table but not in the routing context FIB, which helps preserve the FIB scale in the router.

In [Figure 110: Extended Layer 2 data centers](#), enabling the **arp-host-route populate** command on the DCGWs allows them to learn or advertise the ARP-ND host route 10.0.0.1/32 when the VM is locally connected and to remove or withdraw the host routes when the VM is no longer present in the local DC.

ARP-ND host routes installed in the route table can be exported to VPN IPv4, VPN IPv6, or EVPN routes. No other BGP families or routing protocols are supported.

### 5.4.13 EVPN host mobility procedures within the same R-VPLS service

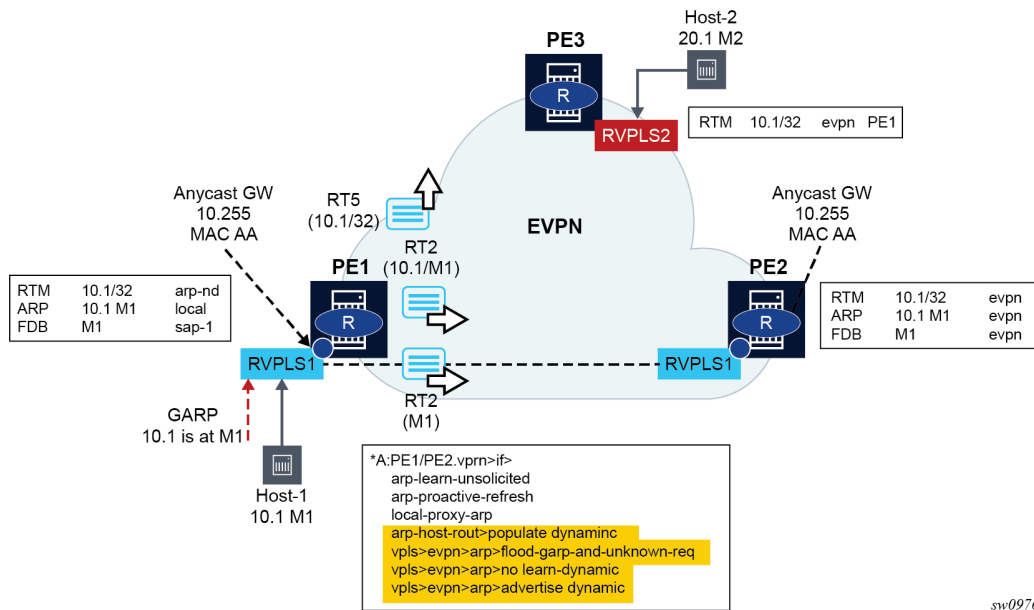
EVPN host mobility is supported in SR OS in accordance with Section 4 of *draft-ietf-bess-evpn-inter-subnet-forwarding*. When a host moves from a source PE to a target PE, it can behave in one of the following ways.

- The host initiates an ARP request or GARP upon moving to the target PE.
- The host sends a data packet without first initiating an ARP request or GARP.
- The host is silent.

#### 5.4.13.1 EVPN host mobility configuration

The following figure shows an example of a host connected to a source PE, PE1, that moved to the target, PE2. The figure shows the expected configuration on the VPRN interface, where R-VPLS 1 is attached (for both PE1 and PE2). PE1 and PE2 are configured with an anycast gateway, that is, a VRRP passive instance with the same backup MAC and IP in both PEs.

Figure 111: Host mobility within the same R-VPLS – initial phase



sw0976

In this initial phase:

- PE1 learns Host-1 IP to MAC (10.1-M1) in the ARP table and generates a host route (RT5) for 10.1/32 because Host-1 is locally connected to PE1. In particular:
  - **arp-learn-unsolicited** triggers the learning of 10.1-M1 upon receiving a GARP from Host-1 or any other ARP
  - **arp-proactive-refresh** triggers the refresh of the host-1 ARP entry 30 seconds before the entry ages out
  - **local-proxy-arp** ensures that PE1 replies to any received ARP request on behalf of other hosts in the R-VPLS
  - **arp-host-route populate dynamic** ensures that only the dynamically learned ARP entries create a host route, for example, 10.1
  - **no flood-garp-and-unknown-req** suppresses ARP flooding (from the CPM) within the R-VPLS1 context, and because ARP entries are synchronized via EVPN, reduces significantly the unnecessary ARP flooding
  - **advertise dynamic** triggers the advertisement of MAC/IP routes for the dynamic ARP entries, including the IP and MAC addresses, for example, 10.1-M1; a MAC/IP route for M1-only that has been previously advertised as M1 is learned on the FDB as local or dynamic
- PE2 learns Host-1 10.1-M1 in the ARP and FDB tables as EVPN type. PE2 must not learn 10.1-M1 as dynamic so that PE2 is prevented from advertising an RT5 for 10.1/32. If PE2 advertises 10.1/32, PE3 could select PE2 as the next hop to reach Host-1, creating an undesired hair-pinning forwarding behavior. PE2 is expected to have the same configuration as PE1, including the following commands, as well as those described for PE1:
  - **no learn-dynamic** prevents PE2 from learning ARP entries from ARP traffic received on an EVPN tunnel

- **populate dynamic**, as in PE1, ensures PE2 only creates route-table ARP-ND host routes for dynamic entries. Therefore, 10.1-M1 does not create a host route as long as it is learned via EVPN only.

The configuration described in this section and the cases described in [Host mobility case 1](#), [Host mobility case 2](#), and [Host mobility case 3](#) apply to IPv4 hosts; however, the functionality is also supported for IPv6 hosts. The IPv6 configuration requires equivalent commands that use the prefix “**nd-**” instead of “**arp-**”. The only exception is the **flood-garp-and-unknown-req** command, which does not have an equivalent command for ND.

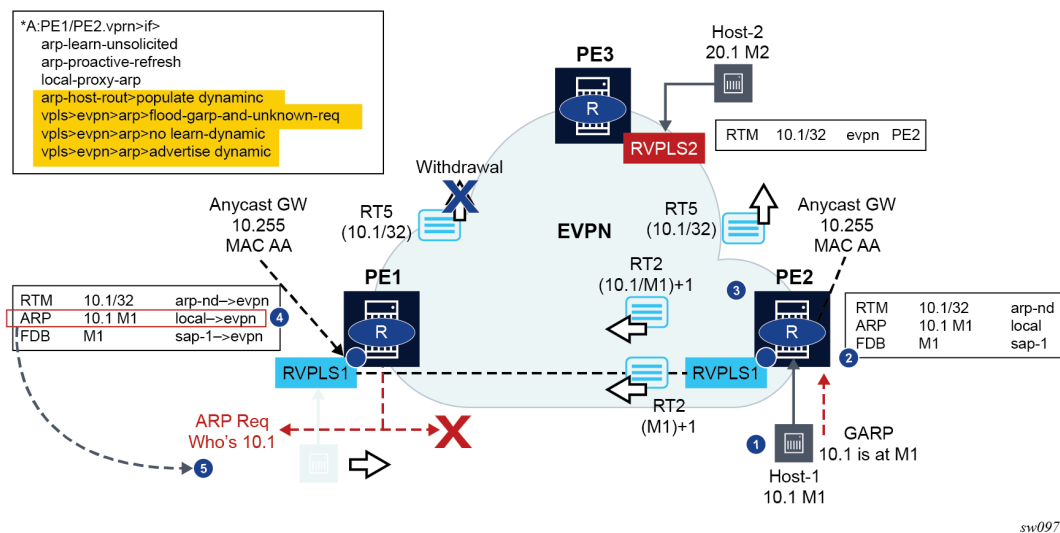
### 5.4.13.1.1 Host mobility case 1

#### Host initiates an ARP/GARP upon moving to the target PE

[Figure 112: Host mobility within the same R-VPLS – move with GARP](#) shows the case where the host initiates an ARP/GARP upon moving to the target PE. The following is the expected behavior based on the configuration described in [EVPN host mobility configuration](#).

1. Host-1 moves from PE1 to PE2 and issues a GARP with 10.1-M1.
2. Upon receiving the GARP, PE2 updates its FDB and ARP table.
3. The route-table entry for 10.1/32 changes from EVPN to type ARP-ND (based on **populate dynamic**); consequently, PE2 advertises a RT5 with 10.1/32. Also, M1 is now learned in the FDB and ARP as local; consequently, MAC/IP routes with a higher sequence number are advertised (one MAC/IP route with M1 only and another MAC/IP route with 10.1-M1).
4. Upon receiving the routes, PE1 performs the following actions:
  - updates its FDB and withdraws its RT2(M1) based on the higher SEQ number
  - updates its ARP entry 10.1-M1 from **dynamic** to type **evpn**
  - as a result of **populate dynamic**, removes its ARP-ND host from the route-table and withdraws its RT5 for 10.1/32
5. The move of 10.1-M1 from **dynamic** to **evpn** triggers an ARP request from PE1 asking for 10.1. The **no flood-garp-and-unknown-req** command prevents PE1 from flooding the ARP request to PE2.

Figure 112: Host mobility within the same R-VPLS – move with GARP



After step 5 is complete, the process ends if no host replies to the PE1 ARP request; however, if a host replied to the ARP for 10.1, the process starts again.

#### 5.4.13.1.2 Host mobility case 2

**Host sends a data packet upon a move to target PE**

In this case, the host does not send a GARP/ARP packet when moving to the target PE. Only regular data packets are sent. [Figure 113: Host mobility within the same R-VPLS – move with data packet](#) shows the case where a host sends a data packet upon a move to the target PE.

1. Host-1 moves from PE1 to PE2 and issues a non-ARP frame with MAC SA=M1.
2. When receiving the frame, PE2 updates its FDB and starts the mobility procedures for M1 (because it was previously learned from EVPN). At the same time, PE2 also creates a short-lived dynamic ARP entry for the host, and triggers an ARP request for it.
3. PE2 advertises a RT2 with M1 only, and a higher sequence number.
4. PE1 receives the RT2, updates its FDB, and withdraws its RT2s for M1 (this includes the RT2 with M1-only and the RT2 with 10.1-M1).
5. PE1 issues an ARP request for 10.1, triggered by the update on M1.

In this case, the PEs are configured with **flood-garp-and-unknown-req** and, therefore, the generated ARP request is flooded to local SAP and SDP-binds and EVPN destinations. When the ARP request is arrives at PE2, it is flooded to the PE2 SAP and SDP-binds and received by Host-1.

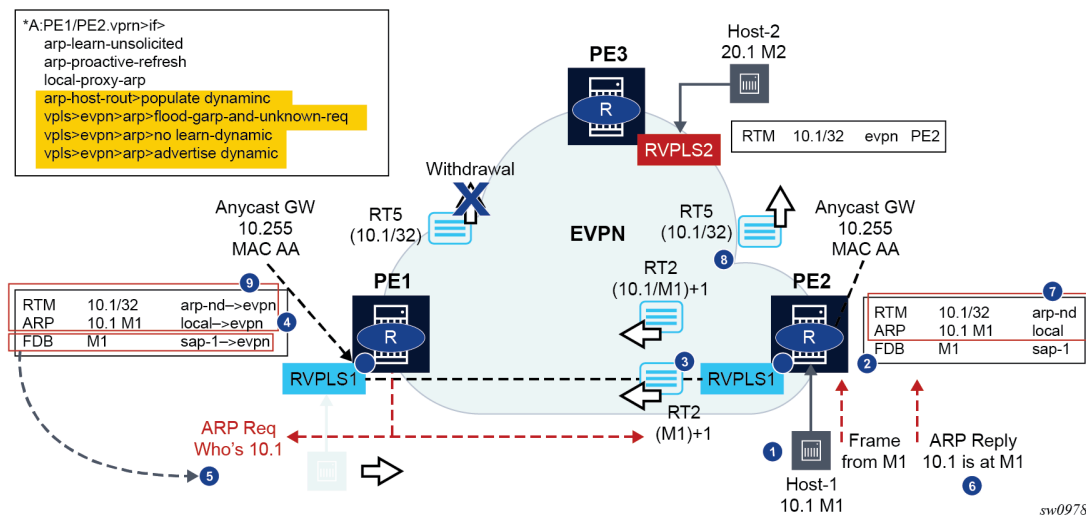
- Host-1 sends an ARP reply that is snooped by PE2 and triggers a similar process described in [Host mobility case 1](#) (as shown in [Figure 113: Host mobility within the same R-VPLS – move with data packet](#)).

Because passive VRRP is used in this scenario, the ARP reply uses the anycast backup MAC that is consumed by PE2.

7. Upon receiving the ARP reply, PE2 updates its ARP table to **dynamic**.

8. Because the route-table entry for 10.1/32 changes from EVPN to type ARP-ND (based on **populate dynamic**), PE2 advertises a RT5 with 10.1/32. Also, M1 is now learned in ARP as local, therefore a RT2 for 10.1-M1 is sent (the sequence number follows the RT2 with M1 only).
9. Upon receiving the route, PE1 performs the following actions:
  - updates the ARP entry 10.1-M1, from type **local** to type **evpn**
  - as a result of **populate dynamic**, removes its ARP-ND host from the route-table and withdraws its RT5 for 10.1/32

Figure 113: Host mobility within the same R-VPLS – move with data packet



### 5.4.13.1.3 Host mobility case 3

#### Silent host upon a move to the target PE

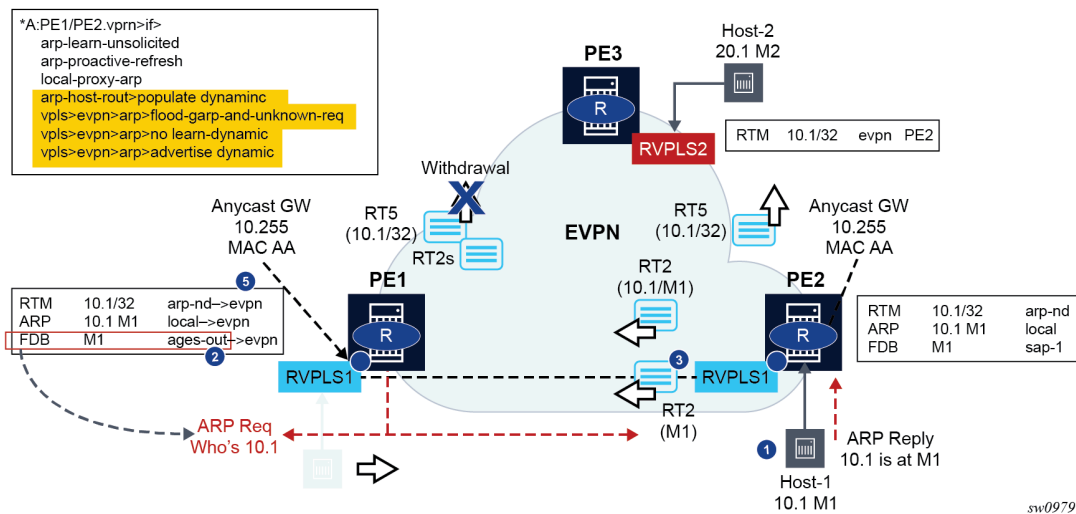
This case assumes the host moves but it stays silent after the move. [Figure 114: Host mobility within the same R-VPLS – silent host](#) shows a silent host upon a move to the target PE.

1. Host-1 moves from PE1 to PE2 but remains silent.
2. Eventually, M1 ages out in PE1's FDB and the RT2s for M1 are withdrawn. This update on M1 triggers PE1 to issue an ARP request for 10.1.

The **flood-garp-and-unknown-req** command is configured. The ARP request makes it to PE2 and Host-1.

3. Host-1 sends an ARP reply that is consumed by PE2. FDB and ARP tables are updated.
4. The FDB and ARP updates trigger RT2s with M1-only and with 10.1-M1. Because an ARP-ND dynamic host route is also created in the route-table, an RT5 with 10.1/32 is triggered.
5. Upon receiving the routes, PE1 updates the FDB and ARP tables. The update on the ARP table from **dynamic** to **evpn** removes the host route from the route-table and withdraws the RT5 route.

Figure 114: Host mobility within the same R-VPLS – silent host



#### 5.4.14 BGP and EVPN route selection for EVPN routes

When two or more EVPN routes are received at a PE, BGP route selection typically takes place when the route key or the routes are equal. When the route key is different, but the PE has to make a selection (for example, the same MAC is advertised in two routes with different RDs), BGP hands over the routes to EVPN and the EVPN application performs the selection.

The following EVPN and BGP selection criteria apply for EVPN routes.

- **EVPN route selection for MAC routes**

When two or more routes with the same *mac-length/mac* but different route key are received, BGP transfers the routes to EVPN. EVPN selects the route based on the following tie-breaking order:

1. conditional static MACs (local protected MACs)
2. auto-learned protected MACs (locally learned MACs on SAPs or mesh/spoke-SDPs as a result of the configuration of **auto-learn-mac-protect**)
3. EVPN ES PBR MACs
4. EVPN static MACs (remote protected MACs)
5. data plane learned MACs (regular learning on SAPs or SDP bindings) and EVPN MACs with higher SEQ numbers. Learned MACs and EVPN MACs are considered equal if they have the same SEQ number.
6. EVPN MACs with higher SEQ number
7. EVPN E-Tree root MACs
8. EVPN non-RT-5 MACs (this tie-breaking rule is only observed if the selection algorithm is comparing received MAC routes and internal MAC routes derived from the MACs in IP-Prefix routes, such as RT-5 MACs)
9. lowest IP (next-hop IP of the EVPN NLRI)
10. lowest Ethernet tag (that is zero for MPLS and may be different from zero for VXLAN)

11. lowest RD

12. lowest BGP instance (this tie-breaking rule is only considered if the preceding rules fail to select a unique MAC and the service has two BGP instances of the same encapsulation)

- **ES PBR MAC routes**

When a PBR filter with a forward action to an ESI and Service Function IP (SF-IP) exists, a MAC route is created by the system. This MAC route is compared to other MAC routes received from BGP.

When static, EVPN, or dynamic ARP resolves for an SF-IP and the system has an AD EVI route for the ESI, a MAC route is created by ES PBR with the <MAC Address = ARPed MAC Address, VTEP = AD EVI VTEP, VNI = AD EVI VNI, RD = ES PBR RD (special RD), Static = 1> and installed in EVPN.

- This MAC route does not add anything back to ARP; however, it goes through the MAC route selection in EVPN and triggers the FDB addition if it is the best route.
- In terms of priority, this route priority is lower than local static but higher than remote EVPN static.
- If there are two competing ES PBR MAC routes, then the selection goes through the rest of checks (Lowest IP > Lowest RD).

- **EVPN route selection for EVPN AD per-EVI routes**

See [Route selection of AD per-EVI routes](#).

- 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 follows regular BGP selection.



**Note:**

- If BGP runs through the selection criteria and a specified and valid EVPN route is not selected in favor of another EVPN route, the non-selected route is displayed using the **show router bgp routes evpn evpn-type detail** command with a tie-breaker reason.
- Protected MACs do not overwrite EVPN static MACs. If a MAC is in the FDB and protected because it was received with the sticky/static bit set in a BGP EVPN update, and a frame is received with the source MAC on an object configured with **auto-learn-mac-protect**, that frame is dropped as a result of the implicit **restrict-protected-src discard-frame**. The reverse is not true; when a MAC is learned and protected using **auto-learn-mac-protect**, its information is not overwritten with the contents of a BGP update containing the same MAC address.

#### 5.4.15 LSP tagging for BGP next-hops or prefixes and BGP-LU

It is possible to constrain the tunnels used by the system for resolution of BGP next-hops or prefixes and BGP labeled unicast routes using LSP administrative tags. See "LSP Tagging and Auto-Bind Using Tag Information" in the *7705 SAR Gen 2 MPLS Guide* for further details.

### 5.4.16 Oper-groups interaction with EVPN services

Operational groups, also referred to as oper-groups, are supported in EVPN services. In addition to supporting SAP and SDP-binds, oper-groups can also be configured under the following objects:

- EVPN-MPLS instances
- Ethernet segments

These oper-groups can be monitored in LAGs or service objects. Oper-groups are particularly useful for the following applications:

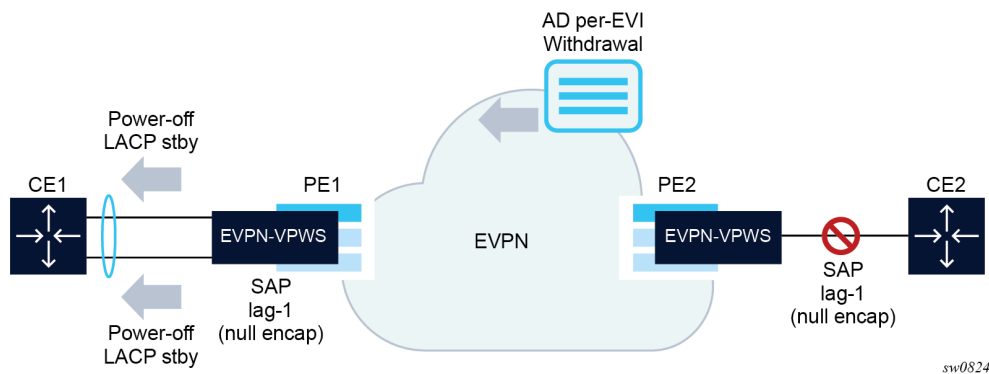
- Link Loss Forwarding (LLF) for EVPN VPWS services
- core isolation blackhole avoidance
- LAG standby signaling to CE on non-DF EVPN PEs (single-active)

#### 5.4.16.1 LAG-based LLF for EVPN-VPWS services

SR OS uses the Eth-CFM fault-propagation to support CE-to-CE fault propagation in EVPN-VPWS services. That is, upon detecting a CE failure, an EVPN-VPWS PE withdraws the corresponding Auto-Discovery per-EVI route, which then triggers a down maintenance endpoint (MEP) on the remote PE that signals the fault to the connected CE. In cases where the CE connected to EVPN-VPWS services does not support Eth-CFM, the fault can be propagated to the remote CE by using LAG standby-signaling, which can be LACP-based or simply power-off.

The following figure shows an example of link loss forwarding for EVPN-VPWS.

Figure 115: Link loss forwarding for EVPN-VPWS



In this example, PE1 is configured as follows:

```
A:PE1>config>lag(1)# info
-----
mode access
encap-type null
port 1/1/1
port 1/1/2
standby-signaling power-off
monitor-oper-group "llf-1"
no shutdown
-----
```



```
*A:PE1>config>service>epipe# info
-----
bgp
exit
bgp-evpn
  evi 1
    local-attachment-circuit ac-1
      eth-tag 1
    exit
  remote-attachment-circuit ac-2
    eth-tag 2
  exit
  mpls bgp 1
    oper-group "llf-1"
    auto-bind-tunnel
    resolution any
  exit
  no shutdown
exit
sap lag-1 create
no shutdown
exit
no shutdown
```

The following applies to the PE1 configuration.

- The EVPN Epipe service is configured on PE1 with a null LAG SAP and the oper-group "llf-1" under **bgp-evpn>mpls**. This is the only member of oper-group "llf-1".



**Note:** Do not configure the oper-group under **config>service>epipe**, because circular dependencies are created when the access SAPs go down as a result of the LAG **monitor-oper-group** command.

- The oper-group monitors the status of the BGP-EVPN instance in the Epipe service. The status of the BGP-EVPN instance is determined by the existence of an EVPN destination at the Epipe.
- The LAG, in **access** mode and encap-type **null**, is configured using the command **monitor-oper-group** "llf-1".



**Note:** The **configure lag monitor-oper-group name** command is only supported in **access** mode. Any encapsulation type can be used.

As shown in [Figure 115: Link loss forwarding for EVPN-VPWS](#), upon failure on CE2, the following events occur.

- PE2 withdraws the EVPN route.
- The EVPN destination is removed in PE1 and oper-group "llf-1" also goes down.
- Because lag-1 is monitoring "llf-1", the oper-group that is becoming inactive triggers standby signaling on the LAG; that is, power-off or LACP out-of-sync signaling to the CE1.

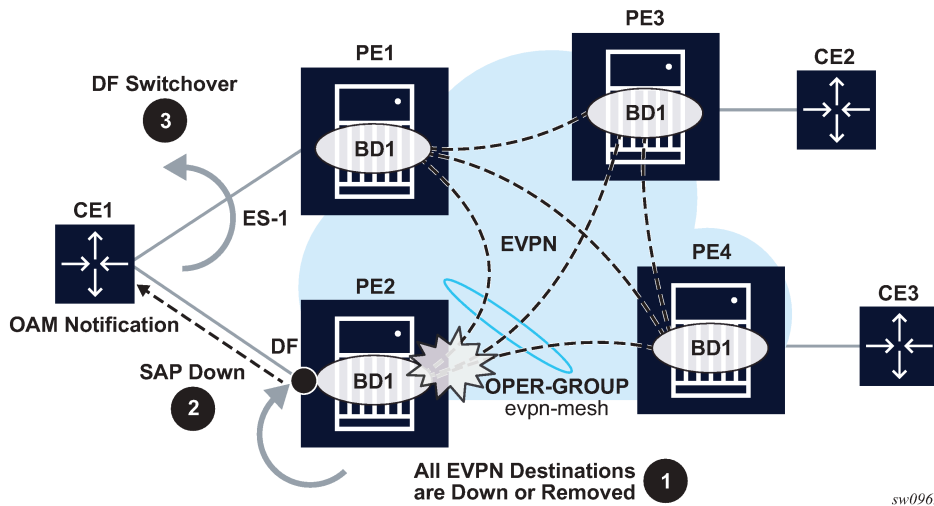
Because the SAP or port is down as a result of the LAG monitoring of the oper-group, PE1 does not trigger an AD per-EVI route withdrawal, even if the SAP is brought operationally down.

- After CE2 recovers and PE2 re-advertises the AD per-EVI route, PE1 creates the EVPN destination and oper-group "llf-1" comes up. As a result, the monitoring LAG stops signaling standby and the LAG is brought up.

### 5.4.16.2 Core isolation blackhole avoidance

The following figure shows how black holes can be avoided when a PE becomes isolated from the core.

Figure 116: Core isolation blackhole avoidance



In this example, consider PE2 and PE1 are single-active multihomed to CE1. If PE2 loses all its core links, PE2 must somehow notify CE1 so that PE2 does not continue attracting traffic and so that PE1 can take over. This notification is achieved by using oper-groups under the BGP-EVPN instance in the service. The following is an example output of the PE2 configuration.

```
*[ex:configure service vpls "evl1"]
A:admin@PE-2# info
  admin-state enable
  bgp-evpn {
    evi 1
    mpls 1 {
      admin-state enable
      oper-group "evpn-mesh"
      auto-bind-tunnel {
        resolution any
      }
    }
  }
  sap lag-1:351 {
    monitor-oper-group "evpn-mesh"
  }
*[ex:configure service oper-group "evpn-mesh"]
A:admin@PE-2# info detail
  hold-time {
    up 4
  }
}
```

With the PE2 configuration and [Figure 116: Core isolation blackhole avoidance](#) example, the following steps occur.

1. PE2 loses all its core links, therefore, it removes its EVPN-MPLS destinations. This causes oper-group "evpn-mesh" to go down.

2. Because PE2 is the DF in the Ethernet Segment (ES) ES-1 and sap lag-1:351 is monitoring the oper-group, the SAP becomes operationally down. If ETH-CFM fault propagation is enabled on a down MEP configured on the SAP, CE1 is notified of the failure.
3. PE1 takes over as the DF, based on the withdrawal of the ES (and AD) routes from PE2, and CE1 begins sending traffic immediately to PE1 only, therefore, avoiding a traffic black hole.

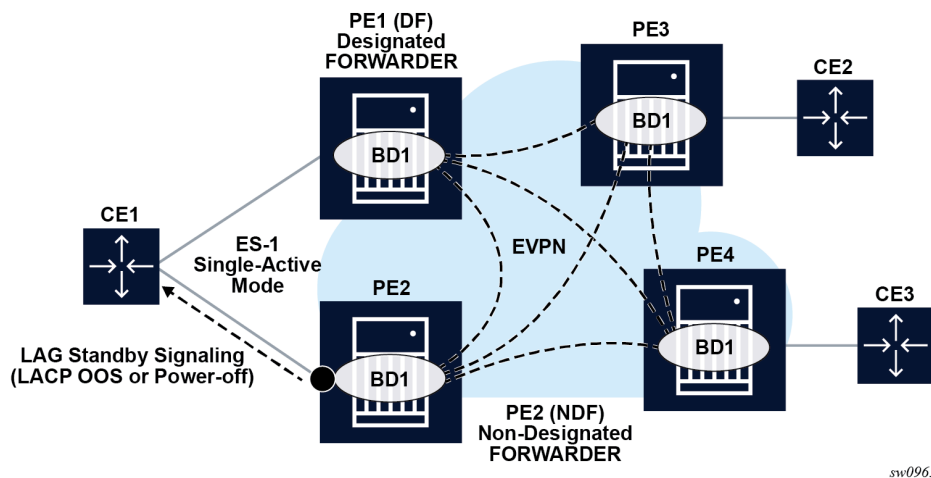
Generally, when oper-groups are associated with EVIs, the following applies.

- The oper-group state is determined by the existence of at least one EVPN destination in the EVPN instance.
- The oper-group that is configured under a BGP EVPN instance cannot be configured under any other object (for example, SAP, SDP binding, and so on) of the same or different service.
- The status of an oper-group associated with an EVPN instance does not go down if all the EVPN destinations are operationally down because of a control word or MTU mismatch.
- The status of an oper-group associated with an EVPN instance goes down in the following cases:
  - the service admin-state is disabled (only for VPLS services, not for Epipes)
  - the BGP EVPN VXLAN or MPLS admin-state are disabled
  - there are no EVPN destinations associated with the instance

### 5.4.16.3 LAG or port standby signaling to the CE on non-DF EVPN PEs (single-active)

As described in [EVPN for MPLS tunnels](#), EVPN single-active multihoming PEs that are elected as non-DF must notify their attached CEs so the CE does not send traffic to the non-DF PE. This can be performed on a per-service basis that is based on the ETH-CFM and fault-propagation. However, sometimes ETH-CFM is not supported in multihomed CEs and other notification mechanisms are needed, such as LACP standby or power-off. This scenario is shown in the following figure.

Figure 117: LACP standby signaling from the non-DF



As shown in the preceding figure, the multihomed PEs are configured with multiple EVPN services that use ES-1. ES-1 and its associated LAG is configured as follows.

```
*[ex:configure lag 1]
```

```

A:admin@PE-2# info
  admin-state enable
  standby-signaling {power-off|lacp}
  monitor-oper-group "DF-signal-1"
  mode access
  port 1/1/c2/1 {
  }
<snip>
ex:configure service system bgp evpn]
A:admin@PE-2# info
  ethernet-segment "ES-1" {
    admin-state enable
    esi 0x01010000000000000000
    multi-homing-mode single-active
    oper-group "DF-signal-1"
    association {
      lag 1 {
      }
    }
  }
<snip>

```

The following applies when the operational group is configured on the ES and monitored on the associated LAG.

- The operational group status is driven by the ES DF status (defined by the number of DF SAPs or oper-up SAPs owned by the ES).
- The operational group goes down if all the SAPs in the ES go down (this happens in PE2 in [Figure 117: LACP standby signaling from the non-DF](#)). The ES operational group goes up when at least one SAP in the ES goes up.
- As a result, if PE2 becomes non-DF on all the SAPs in the ES, they all go oper-down, including the ES-1 operational group.
- Because LAG-1 is monitoring the operational group, when its status goes down, LAG-1 signals LAG standby state to the CE. The standby signaling can be configured as LACP or power-off.
- The ES and AD routes for the ES are not withdrawn because the router recognizes that the LAG becomes standby as a result of the ES operational group.

If the single-active ES is associated with a port instead of a LAG, the **config>port>monitor-oper-group DF-signal-1** command can be configured. In this case, the port monitors the ES operational group and the following rules apply:

- As in the case of the LAG, if the ES goes non-DF, its operational group also goes down.
- The port that is monitoring the ES operational group signals standby state by powering off the port itself.
- As in the case of the LAG, the ES and AD routes for the ES are not withdrawn because the router recognizes that the port is in standby state because of the ES operational group.

Operational groups cannot be assigned to ESs that are configured as **virtual**, **all-active** or **service-carving mode auto**.

#### 5.4.16.4 AC-influenced DF election capability on an ES with oper-group

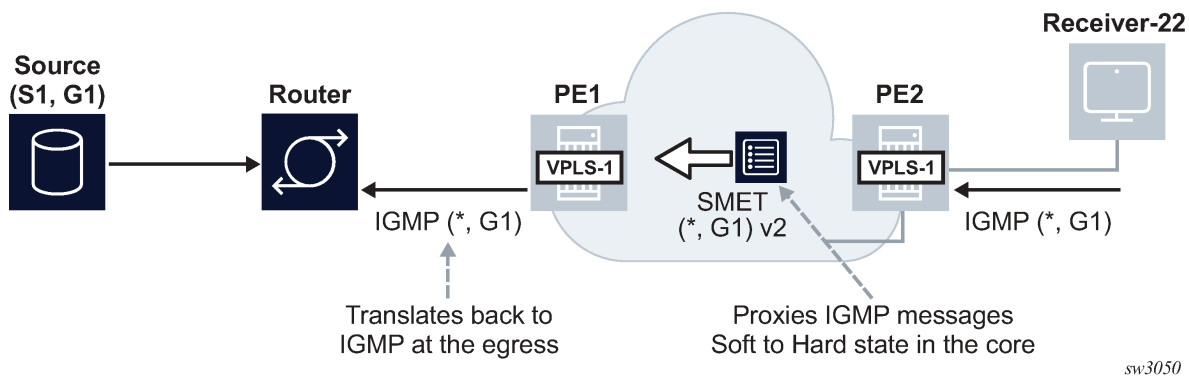
The AC-influenced designated forwarder (AC-DF) election capability, as described in RFC 8584, is supported in SR OS. By default, the **ac-df-capability** command is set to the **include** option. This configuration addresses the need to consider EVPN Auto-discovery per EVI/ES (AD per EVI/ES) routes for a specific PE, which ensures that the PE is included on the candidate DF list.

Configuring **ac-df-capability** to **exclude** disables the AC-DF capability. When **ac-df-capability exclude** is configured on a specific ES, the presence or absence of the AD per EVI/ES routes from the ES peers does not modify the DF Election candidate list for the ES. The **exclude** option is recommended in ESs that use an **oper-group**, which is monitored by the access LAG, to signal standby **lACP** or **power-off**, as described in [LAG or port standby signaling to the CE on non-DF EVPN PEs \(single-active\)](#). All PE routers attached to the same ES must be configured consistently for the specific **ac-df-capability**.

#### 5.4.17 EVPN Layer-2 multicast (IGMP/MLD proxy)

SR OS supports EVPN Layer-2 multicast as described in the EVPN IGMP/MLD Proxy specification RFC 9251. When this is enabled in a VPLS service with active IGMP or MLD snooping, IGMP or MLD messages are no longer sent to EVPN destinations. SMET routes (EVPN routes type 6) are advertised instead, so that the interest in a specific (S,G) can be signaled to the rest of the PEs attached to the same VPLS (also known as a Broadcast Domain (BD)). The following figure shows this scenario.

Figure 118: SMET routes replace IGMP/MLD reports



A VPLS service supporting EVPN-based proxy-IGMP/MLD functionality is configured as follows.

```
vpls 1 name "evi-1" customer 1 create
  bgp
  exit
  bgp-evpn
    evi 1
    sel-mcast-advertisement
    vxlan
      shutdown
    exit
  mpls
    auto-bind-tunnel
    resolution any
    exit
    no shutdown
  exit
exit
igmp/mld-snooping
  evpn-proxy
    no shutdown
  exit
sap lag-1:101 create
  igmp-snooping
    send-queries
  exit
```

```
no shutdown
exit
```

Where:

- The **sel-mcast advertise** (MD-CLI) or **sel-mcast-advertisement** (classic CLI) command allows the advertisement of SMET routes.
- The received SMET routes are processed regardless of the command.
- The **evpn-proxy** command in either the **igmp-snooping** or **mld-snooping** contexts:
  - triggers an IMET route update with the multicast flags EC and the proxy bits set. The multicast flags extended community carries a flag for IGMP proxy, that is set if **igmp-snooping evpn-proxy** is administratively enabled. Similarly, the MLD proxy flag is set if **mld-snooping evpn-proxy** is administratively enabled.
  - no longer turns EVPN MPLS into an Mrouter port, when used in EVPN MPLS service
  - enables EVPN proxy (IGMP or MLD snooping must be administratively disabled)

When the VPLS service is configured as an EVPN proxy service, IGMP or MLD queries or reports are no longer forwarded to EVPN destinations of PEs that support EVPN proxy. The reports are also no longer processed when received from PEs that support EVPN proxy.

The IGMP or MLD snooping function works in the following manner when the **evpn-proxy** command is enabled:

- IGMP or MLD works in proxy mode despite its configuration as IGMP or MLD snooping.
- Received IGMP or MLD join or leave messages on SAP or SDP bindings are processed by the proxy database to summarize the IGMP or MLD state in the service based on the group joined (each join for a group lists all sources to join). The proxy database can be displayed using the following command.

```
show service id igmp-snooping proxy-db
```

Output example

|   |         |             |             |
|---|---------|-------------|-------------|
| =====   |         |             |             |
| IGMP Snooping Proxy-reporting DB for service 4000 |         |             |             |
| =====   |         |             |             |
| Group Address                                     | Mode    | Up Time     | Num Sources |
| -----   |         |             |             |
| 239.0.0.1   | exclude | 0d 00:53:00 | 0           |
| 239.0.0.2   | include | 0d 00:53:01 | 1           |
| -----   |         |             |             |
| Number of groups: 2                               |         |             |             |
| =====   |         |             |             |

- When **evpn-proxy** is enabled, an additional EVPN proxy database is created to hand the version flags over to BGP and generate the SMET routes with the correct IGMP or MLD version flags. This EVPN proxy database is populated with local reports received on SAP or SDP binds but not with received SMET routes (the regular proxy database includes reports from SMETs too, without the version). The EVPN proxy database can be displayed using the following command.

```
show service id igmp-snooping evpn-proxy-db
```

Output example

|       |  |  |  |
|-------|--|--|--|
| ===== |  |  |  |
|-------|--|--|--|

IGMP Snooping Evpn-Proxy-reporting DB for service 4000

| Group               | Address | Mode    | Up Time     | Num Sources | V1 | V2 | V3 |
|---------------------|---------|---------|-------------|-------------|----|----|----|
| 239.0.0.1           |         | exclude | 0d 00:53:55 | 0           |    |    | V3 |
| 239.0.0.2           |         | include | 0d 00:53:55 | 1           |    |    | V3 |
| Number of groups: 2 |         |         |             |             |    |    |    |

- The EVPN proxy database or proxy database processes IGMP or MLD reports as follows:
  - The EVPN proxy database result is communicated to the EVPN layer so that the corresponding SMET routes and flags are sent to the BGP peers. If multiple versions exist on the EVPN proxy database, multiple flags are set in the SMET routes.
  - The regular proxy database result is conveyed to the local Mrouter ports on SAP or SDP binds by IGMP or MLD reports and they are never sent to EVPN destinations of PEs with **evpn-proxy** configured.
- IGMP or MLD messages received on local SAP or SDP bind Mrouter ports (which have a default \*.\* entry) and queries are not processed by the proxy database. Instead, they are forwarded to local SAP or SDP binds but never to EVPN destinations of PEs with **evpn-proxy** configured (they are, however, still sent to non-EVPN proxy PEs).
- IGMP or MLD reports or queries are not received from EVPN PEs with **evpn-proxy** configured, but they are received and processed from EVPN PEs with **evpn-proxy** command disabled. A PE determines if a specified remote PE, in the same BD, supports EVPN proxy based on the received igmp-proxy and mldproxy flags along with the IMET routes.
- The Layer-2 MFIB OIF list for an (S,G) is built out of the local IGMP or MLD reports and remote SMET routes.
  - For backward compatibility, PEs that advertise IMET routes without the multicast flags EC or with the EC but without the proxy bit set, are considered as Mrouters. For example, its EVPN binds are added to all OIF lists and reports are sent to them.
  - Even if MLD snooping is shut down and only IGMP snooping is enabled, the MFIB shows the EVPN binds added to \*,\* for MAC scope. If MLD snooping is enabled, the EVPN binds are not added as Mrouter ports for MAC scope.
- When SMET routes are received for a specific (S,G), the corresponding reports are sent to local SAP or SDP binds connected to queriers. The report version is set based on the local version of the querier.

The IGMP or MLD EVPN proxy functionality is supported in VPLS services with EVPN-VXLAN or EVPN MPLS, and along with ingress replication or mLDp provider-tunnel trees.

In addition, EVPN proxy VPLS services support EVPN multihoming with multicast state synchronization using EVPN routes type 7 and 8. No additional command is needed to trigger the advertisement and processing of the multicast synch routes. In VPLS services, BGP sync routes are advertised or processed whenever the **evpn-proxy** command is enabled and there is a local ES in the service.

#### 5.4.18 EVPN-VPWS PW headend functionality

EVPN-VPWS is often used as an aggregation technology to connect access devices to the residential or business PE in the service provider network. The PE receives tagged traffic inside EVPN-VPWS circuits and maps each tag to a different service in the core, such as ESM services, Epipse services, or VPRN services.

SR OS implements this PW headend functionality by using PW ports that use multihomed Ethernet Segments (ESs) for redundancy. ESs can be associated with PW ports in two different modes of operation.

- PW port-based ESs with multihoming procedures on PW SAPs
- PW port-based ESs with multihoming procedures on stitching Epipe

## PW port-based ESs with multihoming procedures on PW SAPs

PW ports in ESs and virtual ESs (vESs) are supported for EVPN-VPWS MPLS services. In addition to LAG, port, and SDP objects, PW port ID can be configured in an Ethernet Segment. In this mode of operation, PW port-based ESs only support **all-active** configuration mode, and not **single-active** configuration mode.

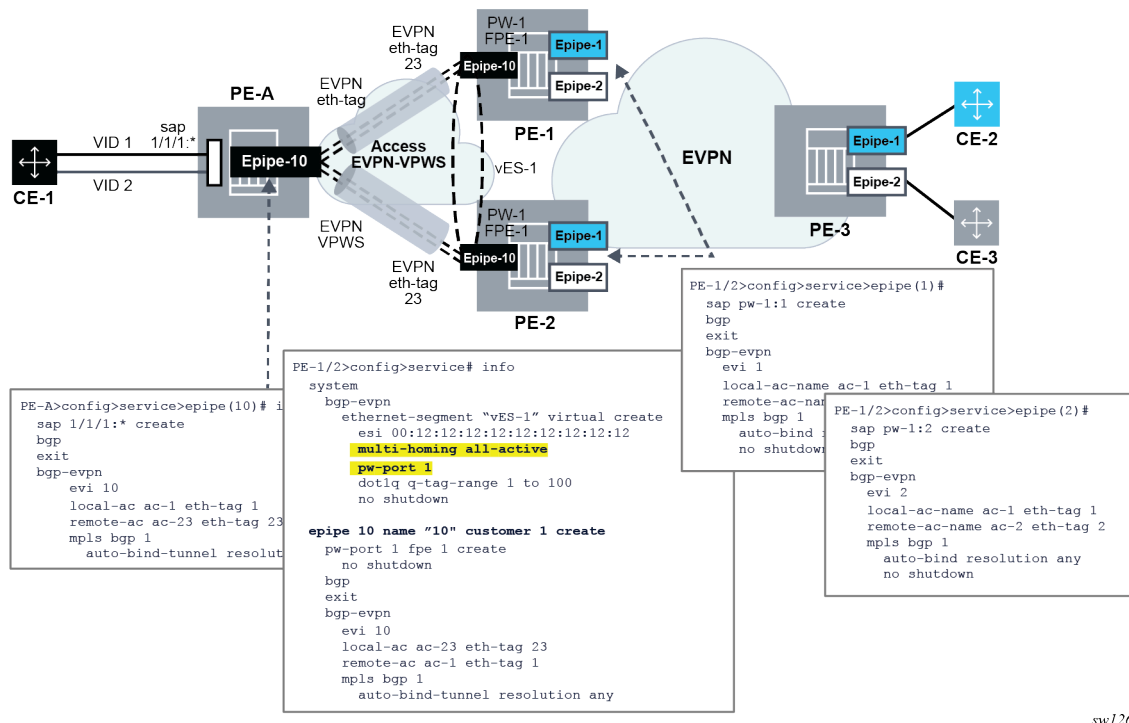
The following requirements apply:

- Port-based or FPE-based PW ports can be used in ESs
- PW port scenarios supported along with ESs are as follows:
  - port-based PW port
  - FPE-based PW port, where the stitching service uses a spoke SDP to the access CE
  - FPE-based PW port, where the stitching service uses EVPN-VPWS (MPLS) to the access CE

For all the preceding scenarios, fault-propagation to the access CE only works in the case of physical failures. Administrative shutdown of individual Epipes, PW SAPs, ESs or BGP-EVPN may result in traffic black holes.

The following figure shows the use of PW ports in ESs. In this example, an FPE-based PW port is associated with the ES, where the stitching service itself also uses EVPN-VPWS.

Figure 119: ES FPE-based PW port access using EVPN-VPWS



sw1260



In this example, the following conditions apply:

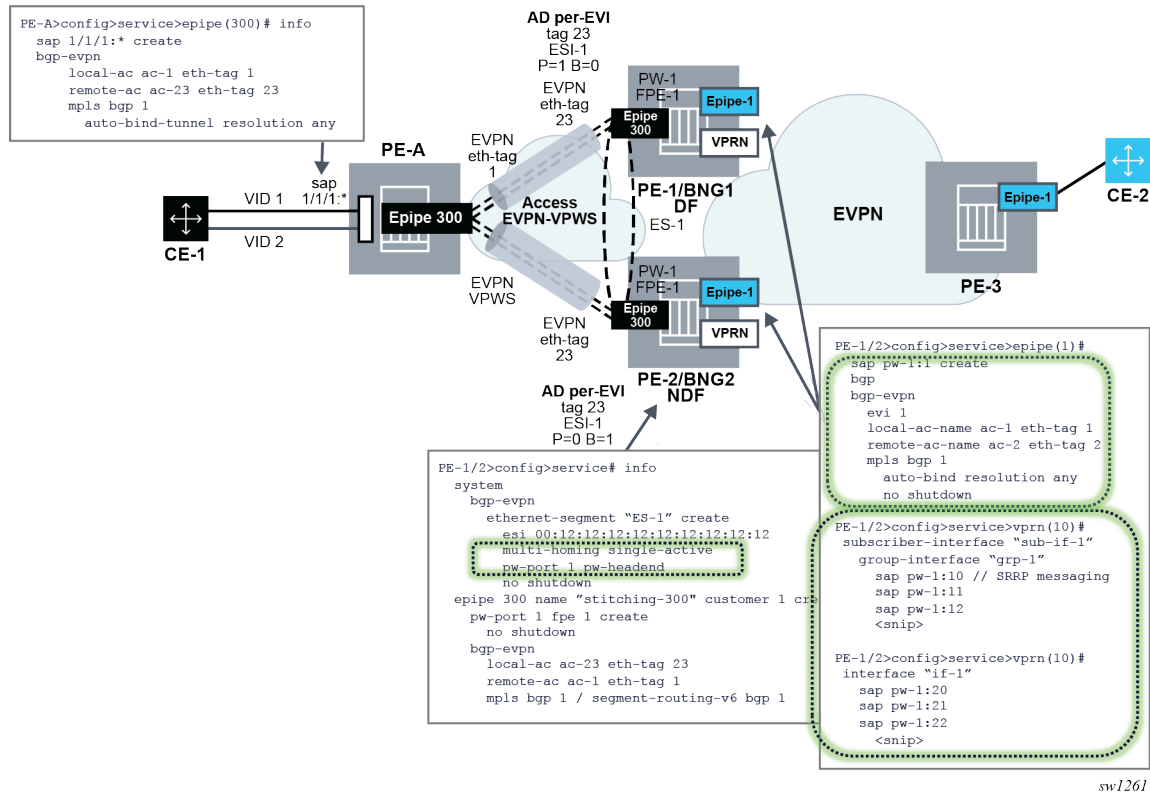
- Redundancy is driven by EVPN all-active multihoming. ES-1 is a virtual ES configured on the FPE-based PW port on PE-1 and PE-2.
- The access network between the access PE (PE-A) and the network PEs (PE-1 and PE-2), uses EVPN-VPWS to backhaul the traffic. Therefore, PE-1 and PE-2 use EVPN-VPWS in the PW port stitching service, where:
  - PE-1 and PE-2 apply the same Ethernet tag configuration on the stitching service (Epipe 10)
  - Optionally PE-1 and PE-2 can use the same RD on the stitching service
  - AD per-EVI routes for the stitching service Ethernet tags are advertised with ESI=0
- Forwarding in the CE-1 to CE-2 or CE-3 direction, works as follows:
  - PE-A forwards traffic based on the selection of the best AD per-EVI route advertised by PE-1 and PE-2 for the stitching Epipe 10. This selection can be either BGP-based if PE-2 and PE-3 use the same RD in the stitching service, or EVPN-based if different RD is used.
  - When the PE-1 route is selected, PE-1 receives the traffic on the local PW-SAP for Epipe 1 or Epipe 2, and forwards it based on the customer EVPN-VPWS rules in the core.
- Forwarding in the CE-2 or CE-3 to CE-1 direction, works as follows:
  - PE-3 forwards the traffic based on the configuration of ECMP and aliasing rules for Epipe 1 and Epipe 2.
  - PE-3 can send the traffic to PE-2 and PE-2 to PE-A, following different directions.
- If the user needs the traffic to follow a symmetric path in both directions, then the AD per-EVI route selection on PE-A and PE-3 can be handled so that the same PE (PE-1 or PE-2) is selected for both directions.
- For this example, the solution provides redundancy in case of node failures in PE-1 or PE-2. However, the administrative shutdowns, configured in some objects, are not propagated to PE-A, leading to traffic blackholing. As a result, black holes may be caused by the following events in PE-1 or PE-2:
  - Epipe 1 or Epipe 2 service shutdown
  - Epipe 1 or Epipe 2 BGP-EVPN MPLS shutdown
  - vES-1 shutdown
  - BGP shutdown

### PW port-based ESs with multihoming on stitching Epipe

The solution described in [PW port-based ESs with multihoming procedures on PW SAPs](#) provides PW-headend redundancy where the access PE selects one of the PW-headend PE devices based on BGP best path selection, and the traffic from the core to the access may follow an asymmetric path. This is because the multihoming procedures are actually run on the PW SAPs of the core services, and the AD per-EVI routes advertised in the context of the stitching Epipe use an ESI=0.

SR OS also supports a different mode of operation called **pw-port headend** which allows running the multihoming procedures in the stitching Epipe and, therefore, use regular EVPN-VPWS primary or backup signaling to the access PE. The mode of operation is supported in a single-active mode shown in the following figure.

Figure 120: ES FPE-based pw-port headend



The following configuration triggers the needed behavior:

```

// ES and stitching Epipe config

PE-1/2>config>service# info
system
  bgp-evpn
    ethernet-segment "ES-1" create
      esi 00:12:12:12:12:12:12:12:12
      multi-homing single-active
      pw-port 1 pw-headend
      no shutdown
    epipe 300 name "stitching-300" customer 1 create
      pw-port 1 fpe 1 create
      no shutdown
    bgp-evpn
      local-attachment-circuit ac-23 eth-tag 23
      remote-attachment-circuit ac-1 eth-tag 1
      mpls bgp 1
      auto-bind-tunnel resolution any

// Services config

epipe 10
  sap pw-1:10 create
  bgp-evpn
    mpls bgp 1

```

```

epipe 11
  sap pw-1:10 create
  bgp-evpn
  mpls bgp 1

```

The configuration and functionality are divided in four aspects.

### Configuration of single-active multihoming on ESs associated with PW ports of type pw-headend

In this mode, PW Ports are associated with single-active non-virtual Ethernet Segments. The **pw-headend** keyword is needed when associating the PW port.

```

PE-1/2>config>service# info
system
  bgp-evpn
    ethernet-segment "ES-1" create
      esi 00:12:12:12:12:12:12:12:12
      multi-homing single-active
      pw-port 1 pw-headend
      no shutdown

```

The **pw-port id pw-headend** command indicates to the system that the multihoming procedures are run in the PW port stitching Epipe and the routes advertised in the context of the stitching Epipe contains the ESI of the ES.

### Configuration of the PW port stitching Epipe

A configuration example of the stitching Epipe follows.

```

epipe 300 name "stitching-300" customer 1 create
  pw-port 1 fpe 1 create
  no shutdown
  bgp-evpn
    local-attachment-circuit ac-23 eth-tag 23
    remote-attachment-circuit ac-1 eth-tag 1
    mpls bgp 1
    auto-bind-tunnel resolution any

```

The preceding example shows the configuration of a stitching EVPN VPWS Epipe with MPLS transport, however SRv6 transport is also supported.

When the ES is configured with a PW port in **pw-headend** mode, the stitching Epipe associated with the PW port is now running the ES and DF election procedures. Therefore, the following actions apply:

- an AD per-ES route is advertised with:
  - the RD or RT of the stitching Epipe
  - the configured ESI of the ES associated with the PW port
  - the ESI-label extended community with the multihomed mode indication and ESI label
- an AD per EVI route is advertised with:
  - the RD or RT of the stitching Epipe
  - the configured ESI where the PW port resides
  - the P/B bits according to the DF election procedures

- the non-DF drives the PW port operationally down with a flag MHStandby. As a result, all the PW SAPs contained in the PW port are brought operationally down. Optionally, the **config>service>epipe>pw-port>oper-up-on-mhstandby** command can be configured so that the PW port stays operationally up even if it is in MHStandby state (that is, the PE is non-DF). This command may speed up convergence in case a significant number of PW SAPs are configured in the same PW port.

### Configuration of the PW port-contained PW SAPs and edge services

The edge services that contain the PW SAPs of the **pw-headend pw-port** command are configured without any other additional commands. These PW SAPs can be configured on Epipes, VPRN interfaces, or subscriber interfaces, VPLS (capture SAPs). As an example, if the PW SAP is configured on an Epipe EVPN-VPWS service:

```
epipe 10
  sap pw-1:10 create
  bgp-evpn
  mpls bgp 1
```

The behavior of the PW SAPs when the PW port is configured with the **pw-headend** keyword follows:

- The PW SAP is brought operationally down if the PW port is down. The PW port goes down with the reason MHStandby if the PE is a non-DF, or with reason stitching-svc-down if the EVPN destination is removed from the stitching Epipe.
- If the PW SAP is configured in an EVPN-VPWS edge service as in the preceding example, the following actions are performed:
  - An AD per ES route is advertised for the EVPN-VPWS service with the RD or RT of the service Epipe, the configured ESI of the ES associated with the PW port, and the ESI-label extended community with the multihomed mode indication of the ES and ESI label (label is the same value as in the AD per ES for the stitching Epipe). If the PW port is only down because of the MHStandby flag, the AD per ES route for the Epipe service is still advertised.
  - In addition, an AD per EVI route is advertised with the RD or RT of the service Epipe, the configured ESI of the ES associated with the PW port, and the P/B flags of the ES:
    - P=1/B=0 on the DF
    - P=0/B=1 on backup
    - P=0/B=0 on non-DFs and non-backup
  - If the PW port is down only because of MHStandby, the AD per EVI route for the service Epipe is still advertised.

### Some considerations and dependencies between the PW port and the service Epipe PW SAPs

- If all the PW SAPs associated with the FPE PW port are brought down, the following rules apply:
  - state of the PW port does not change
  - does not trigger any AD per-ES/EVI or ES route withdraw toward the CE from the stitching Epipe
- Any event that brings down the PW port (except for MHStandby) triggers:
  - an AD per-EVI/ES route withdrawal within the context of the stitching Epipe
  - an ES route withdrawal
  - an AD per-EVI/ES routes withdrawal within the context of the service Epipes

- the **pw-port>monitoring-oper-group** command can also modify the state of the PW port driven by the state of the operational group
- An individual PW SAP going administrative or operationally down while the PW port is still operationally up, the following actions may be performed:
  - may create black holes for that particular service
  - triggers the withdrawal of the AD per-EVI routes for the service Epipe (not the AD per-ES route, which is kept advertised if the PW port is up)
  - if the PW SAP is administratively not shutdown, the service Epipe AD per-ES/EVI routes mirror the AD per-ES/EVI routes of the stitching service and they are advertised if the routes for the stitching Epipe are advertised

The PW SAP can also be configured on VPRN services (under regular interfaces or subscriber interfaces) and works without any special consideration, other than that a PW port in non-DF state brings down the PW SAP and, therefore, the interface. Similarly, VPLS services with capture PW SAPs support this mode of operation too.

## 5.4.19 Interaction of EVPN and other features

This section describes the interaction of EVPN with other features.

### 5.4.19.1 Interaction of EVPN-MPLS with existing VPLS features

When enabling existing VPLS features in an EVPN-MPLS enabled service, the following considerations apply.

- EVPN-MPLS is only supported in regular VPLS. Other VPLS types, such as **m-vpls**, are not supported with EVPN-MPLS.
- In general, no router-generated control packets are sent to the EVPN destination bindings, except for proxy-ARP/proxy-ND confirm messages and Eth-CFM for EVPN-MPLS.
- For xSTP and M-VPLS services:
  - xSTP can be configured in BGP-EVPN services. BPDUs are not sent over the EVPN bindings.
  - BGP-EVPN is blocked in M-VPLS services; however, a different M-VPLS service can manage a SAP or spoke-SDP in a BGP-EVPN-enabled service.
  - xSTP is not supported in BGP-EVPN services that use Ethernet segments for multihoming; an M-VPLS must not drive the state of a BGP-EVPN service that uses Ethernet segments.
- For BGP-EVPN-enabled VPLS services, **mac-move** can be used in SAPs/SDP-bindings; however, the MACs being learned through BGP-EVPN are not considered.



**Note:** MAC duplication already provides a protection against MAC moves between EVPN and SAPs/SDP-bindings.

- The **disable-learning** command and other FDB-related tools work only for data plane learned MAC addresses.
- The **mac-protect** command cannot be used in conjunction with EVPN.



**Note:** EVPN provides its own protection mechanism for static MAC addresses.

- MAC OAM tools (**mac-ping**, **mac-trace**, **mac-populate**, **mac-purge**, and **cpe-ping**) are not supported for BGP-EVPN services
- EVPN multihoming and BGP-MH can be enabled in the same VPLS service, as long as they are not enabled in the same SAP-SDP or spoke-SDP. There is no limitation on the number of BGP-MH sites supported per EVPN-MPLS service.
- SAPs/SDP-bindings that belong to a specified ES but are configured on non-BGP-EVPN-MPLS-enabled VPLS or Epipe services are kept down using the **StandByForMHProtocol** flag.
- CPE ping is not supported on EVPN services but it is supported in PBB-EVPN services (including I-VPLS and PBB-Epipe). CPE ping packets are not sent over EVPN destinations.
- Other features not supported in conjunction with BGP-EVPN:
  - subscriber management commands under service, SAP, and SDP-binding interfaces
  - BPDU translation
  - L2PT termination
  - MAC-pinning
- Other features not supported in conjunction with the **bgp-evpn mpls** command are:
  - SPB configuration and attributes

#### 5.4.19.2 Interaction of EVPN-MPLS with existing VPRN or IES features

When enabling existing VPRN features on interfaces linked to EVPN-MPLS R-VPLS interfaces, consider that the following are not supported:

- the **arp-populate** and **authentication-policy** commands
- dynamic routing protocols such as IS-IS, RIP, and OSPF

When enabling existing IES features on interfaces linked to EVPN-MPLS R-VPLS interfaces, the following commands are not supported:

- **if vpls evpn-tunnel**
- **bgp-evpn ip-route-advertisement**
- **arp-populate**
- **authentication-policy**
- dynamic routing protocols such as IS-IS, RIP, and OSPF

#### 5.4.20 Interaction of EVPN with BGP owners in the same VPRN service

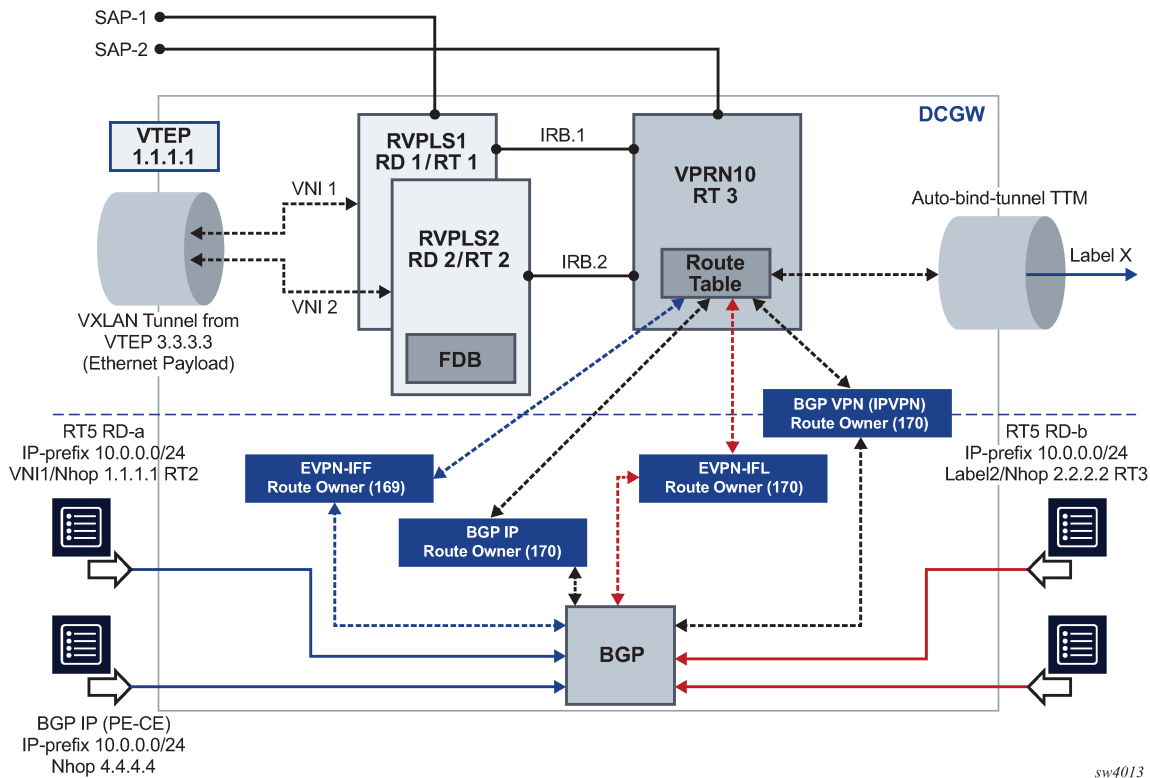
SR OS allows multiple BGP owners in the same VPRN service to receive or advertise IP prefixes contained in the VPRN's route table. Specifically, the same VPRN route table can simultaneously install and process IPv4 or IPv6 prefixes for the following owners:

- VPN-IP (also referred to as IPVPN routes)

- IP (also referred to as BGP PE-CE routes)

The following figure shows the service architecture and the concept of different owners supported on the same VPRN.

Figure 121: Different owners supported on the same VPRN



In the example shown in the preceding figure, VPRN 10 is configured with regular interfaces and R-VPLS interfaces and receives the same prefix 10.0.0.0/24 via the four owners.

EVPN-IFL routes are EVPN IP-Prefix (or type 5) routes that are imported and exported based on the VPRN **bgp-evpn mpls** configuration, as described in [Interface-less IP-VRF-to-IP-VRF model \(IP encapsulation\) for MPLS tunnels](#).

EVPN-IFF routes are EVPN IP-Prefix (or type 5) routes that are imported and exported based on the configuration of the R-VPLS services attached to the VPRN. EVPN-IFF routes are advertised and processed if the R-VPLS services are configured with the **configure service vpls bgp-evpn ip-route-advertisement** command. Although installed in the VPRN service, EVPN-IFF routes use the route distinguisher and route targets determined by the configuration in the R-VPLS, and are supported in R-VPLS services with VXLAN or MPLS encapsulations. See [Interface-ful IP-VRF-to-IP-VRF with SBD IRB model](#) for more information about EVPN-IFF routes.

In addition to EVPN-IFL and EVPN-IFF routes, BGP IP and VPN-IP families are supported on the same VPRN.

### 5.4.20.1 BGP path attribute propagation

A VPRN can receive and install routes for a specific BGP for a specific BGP owner. The routes may be re-exported in the context of the same VPRN and to the same BGP owner or a different one. For example, an EVPN-IFL route can be received from peer N, installed in VPRN 1, and re-exported to peer M using family VPN-IPv4.

When re-exporting BGP routes, the original BGP path attributes are preserved without any configuration in the following cases:

- EVPN-IFL or EVPN-IFL-HOST route re-exported into an IPVPN route, and an IPVPN route re-exported into an EVPN-IFL route
- EVPN-IFL or EVPN-IFL-HOST route re-exported into a BGP IP route (PE-CE), and a BGP IP routes re-exported into an EVPN-IFL route
- IPVPN route re-exported into a BGP IP route (PE-CE), and the other way around
- EVPN-IFL or EVPN-IFL-HOST, IPVPN or BGP IP routes re-exported into a route of the same owner. For example, EVPN-IFL to EVPN-IFL, when the **allow-export-bgp-vpn** command is configured. Received EVPN-IFL-HOST routes are re-exported into an EVPN-IFL route when **allow-export-bgp-vpn** command is enabled.

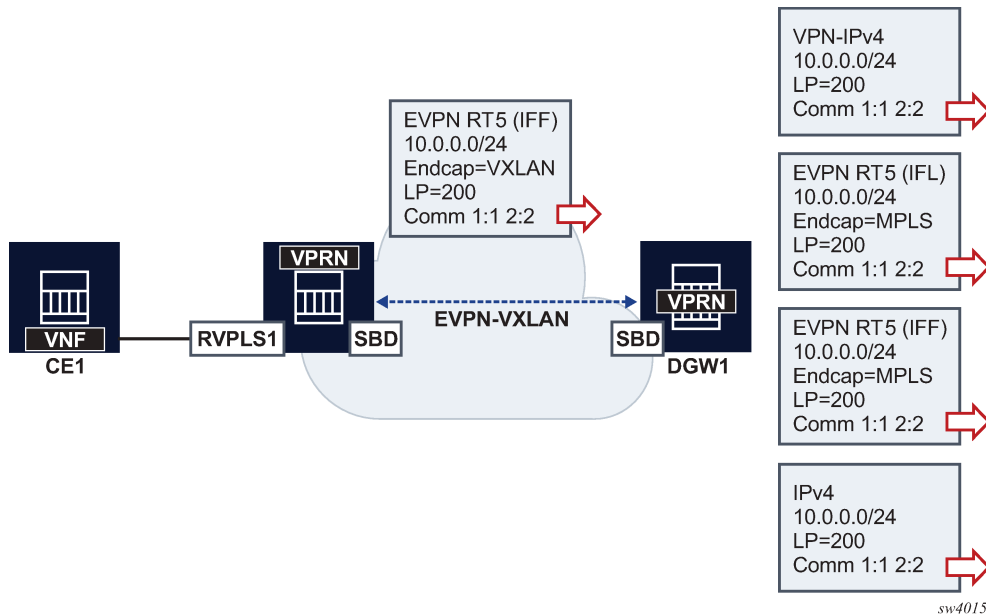


**Note:** **allow-export-bgp-vpn** must never be used in a VPRN service with a route distinguisher that is used in other PEs attached to the same service. If the same route distinguisher is used in this case, constant route flaps occur.

BGP path attributes to or from EVPN-IFF are not preserved by default. If BGP Path Attribute propagation is required, the **configure service system bgp-evpn ip-prefix-routes iff-attribute-uniform-propagation** command must be configured. The following figure shows an example of BGP Path Attribute propagation from EVPN-IFF to the other BGP owners in the VPRN when the **iff-attribute-uniform-propagation** command is configured.



Figure 122: BGP path attribute propagation when *iff-attribute-uniform-propagation* is configured



In the example in [Figure 122: BGP path attribute propagation when iff-attribute-uniform-propagation is configured](#), DGW1 propagates the received LP and communities on an EVPN-IFF route, when advertising the same prefix into any type of BGP owner route, including VPN-IPv4/6, EVPN-IFL, EVPN-IFF, IPv4, or IPv6. If the **iff-attribute-uniform-propagation** command is not configured on DGW1, no BGP path attributes are propagated, but are re-originated instead. The propagation in the opposite direction follows the same rules; configuration of the **iff-attribute-uniform-propagation** command is required.

When propagating BGP path attributes, the following criteria are considered.

- The propagation is compliant with the uniform propagation described in *draft-ietf-bess-evpn-ipvpn-interworking*.
- The following extended communities are filtered or excluded when propagating attributes:
  - all extended communities of type 0x06 (EVPN type). In particular, all those that are supported by routes type 5:
    - MAC Mobility extended community (sub-type 0x00)
    - EVPN Router's MAC extended community (sub-type 0x03)
  - BGP encapsulation extended community
  - all Route Target extended communities
- The BGP Path Attribute propagation within the same owner is supported in the following cases:
  - EVPN-IFF to EVPN-IFF (route received on R-VPLS and advertised in a different R-VPLS context), assuming the **iff-attribute-uniform-propagation** command is configured
  - EVPN-IFL or EVPN-IFL-HOST to EVPN-IFL (route received on a VPRN and re-advertised based on the configuration of **vprn>allow-export-bgp-vpn**)
  - VPN-IPv4/6 to VPN-IPv4/6 (route received on a VPRN and re-advertised based on the configuration of **vprn>allow-export-bgp-vpn**)
- The propagation is supported for iBGP and eBGP as follows:

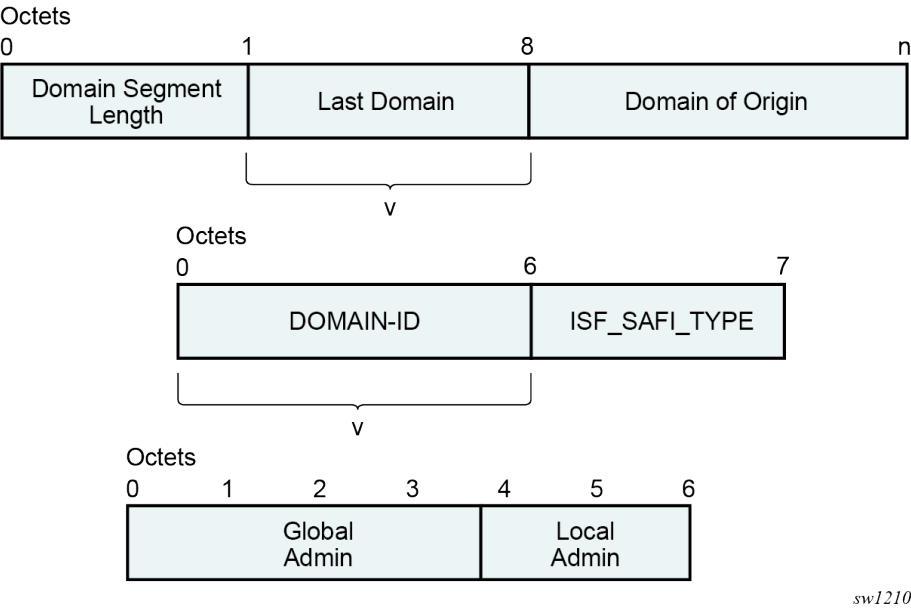
- iBGP-only attributes can only be propagated to iBGP peers
- non-transitive attributes are propagated based on existing rules
- when peering an eBGP neighbor, the AS\_PATH is prepended by the VPRN ASN
- If ECMP is enabled in the VPRN and multiple routes of the same BGP owner with different Route Distinguishers are installed in the route table, only the BGP path attributes of the best route are subject for propagation.

5.4.20.2 BGP D-PATH attribute for Layer 3 loop protection

The SR OS has a full implementation of the D-PATH attribute as described in *draft-ietf-bess-evpn-ipvpn-interworking*.

D-PATH is composed of a sequence of domain segments (similar to AS\_PATH). Each domain segment is graphically represented as shown in the following figure.

Figure 123: D-PATH attribute



Where:

- Each domain segment is comprised of <domain\_segment\_length, domain\_segment\_value>, where the domain segment value is a sequence of one or more domains.
- Each domain is represented by <DOMAIN-ID:ISF\_SAFI\_TYPE>, where the newly added domain is added by a GW, is always prepended at the left of the existing last domain.
- The supported ISF\_SAFI\_TYPE values are:
  - 0 = Local ISF route
  - 1 = safi 1 (typically identifies PE-CE BGP domains)
  - 70 = evpn
  - 128 = safi 128 (IPVPN domains)

- Labeled unicast IP routes do not support D-PATH.
- The D-PATH attribute is only modified by a gateway and not by an ABR/ASBR or RR. A gateway is defined as a PE where a VPRN is instantiated, and that VPRN advertises or receives routes from multiple BGP owners (for example, EVPN-IFL and BGP-IPVPN) or multiple instances of the same owner (for example, VPRN with two BGP-IPVPN instances)

Suppose a router receives prefix P in an EVPN-IFL instance with the following D-PATH from neighbor N.

```
+-----+-----+
|Seg Len=1 | 65000:1:128|
+-----+-----+
```

If the router imports the route in VPRN-1, BGP-EVPN SRv6 instance with domain 65000:2, the router readvertises the route to its BGP-IPVPN MPLS instance as follows:

```
+-----+-----+-----+
|Seg Len=2 |65000:2:70|65000:1:128|
+-----+-----+-----+
```

If the router imports the route in VPRN-1, BGP-EVPN SRv6 instance with domain 65000:3, the router readvertises the route to its BGP-EVPN MPLS instance as follows:

```
+-----+-----+-----+
|Seg Len=2 |65000:3:70|65000:1:128|
+-----+-----+-----+
```

If the router imports the route in VPRN-1, BGP-EVPN MPLS instance with domain 65000:4, the router readvertises the route to its PE-CE BGP neighbor as follows:

```
+-----+-----+-----+
|Seg Len=2 |65000:4:70|65000:1:128|
+-----+-----+-----+
```

When a BGP route of families that support D-PATH is received and must be imported in a VPRN, the following rules apply:

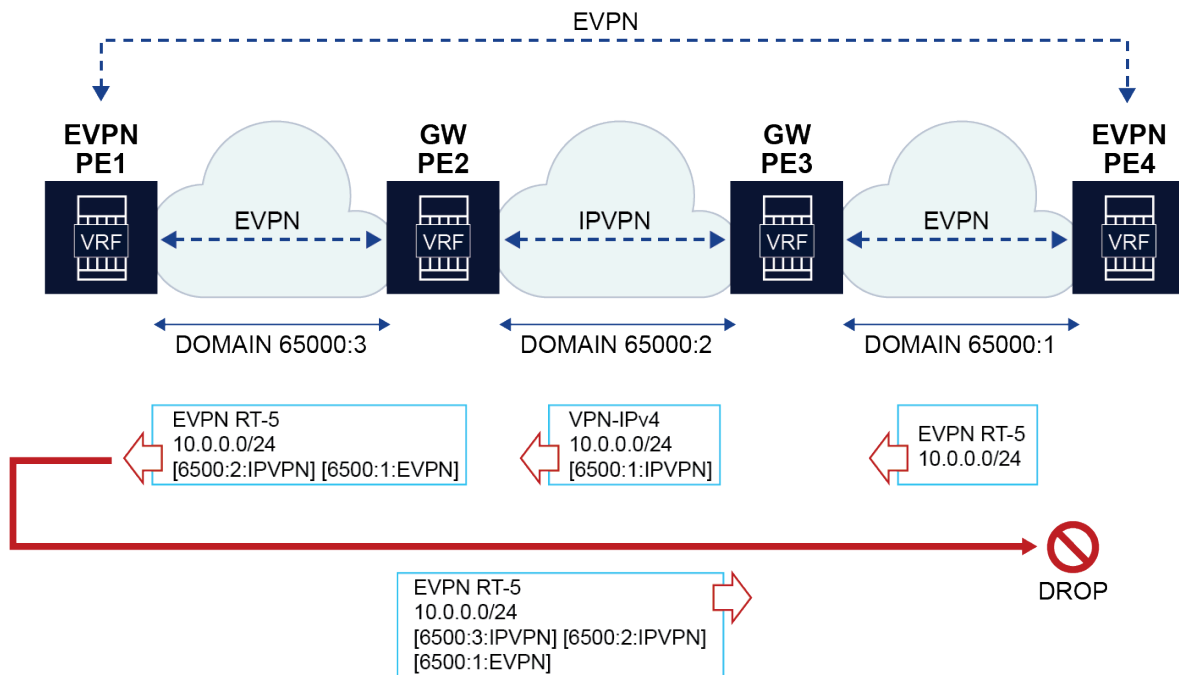
- All domain IDs included in the D-PATH are compared with the local domain IDs configured in the VPRN. The local domain IDs for the VPRN include a list of (up to four) domain IDs configured at the **vprn** or **vprn bgp instance** level, including the domain IDs in local attached R-VPLS instances.
- If one or more D-PATH domain IDs match any local domain IDs for the VPRN, the route is not installed in the VPRN's route table.
- In the case where the IP-VPN or EVPN route matches the import route target in multiple VRFs, the D-PATH loop detection works per VPRN. For example, for each VPRN, BGP checks if the received domain IDs match any locally configured (maximum 4) domain IDs for that VPRN. A route may have a looped domain for one VPRN and not the other. In this case, BGP installs a route only in the VPRN route table that does not have a loop; the route is not installed in the VPRN that has the loop.
- A route that is not installed in any VPRN RTM (due to the domain ID matching any of the local domain IDs in the importing VPRNs) is still kept in the RIB-IN. The route is displayed in the **show router bgp routes** command with a **DPath Loop VRFs** field, indicating the VPRN in which the route is not installed due to a loop.
- Route target-based leaking between VPRNs and D-PATH loop detection is described in the following example.

Consider an EVPN-IFL or EVPN-IFL-HOST route to prefix P imported in VPRN 20 (configured with domain 65000:20) is leaked into VPRN 30.

When the route to prefix P is readvertised in the context of VPRN 30, which is enabled for BGP-IPVPN MPLS and BGP-EVPN MPLS, the readvertised BGP-IPVPN and BGP-EVPN routes have a D-PATH with a prepended domain 65000:20:0. That is, leaked routes are readvertised with the domain ID of the VPRN of origin and an ISF\_SAFI\_TYPE = 0, as described in *draft-ietf-bess-evpn-ipvpn-interworking*.

In the D-PATH example shown in the following figure, the different gateway PEs along the domains modify the D-PATH attribute by adding the source domain and family. If PE4 receives a route for the prefix with the domain of PE4 included in the D-PATH, PE4 does not install the route in order to avoid control plane loops.

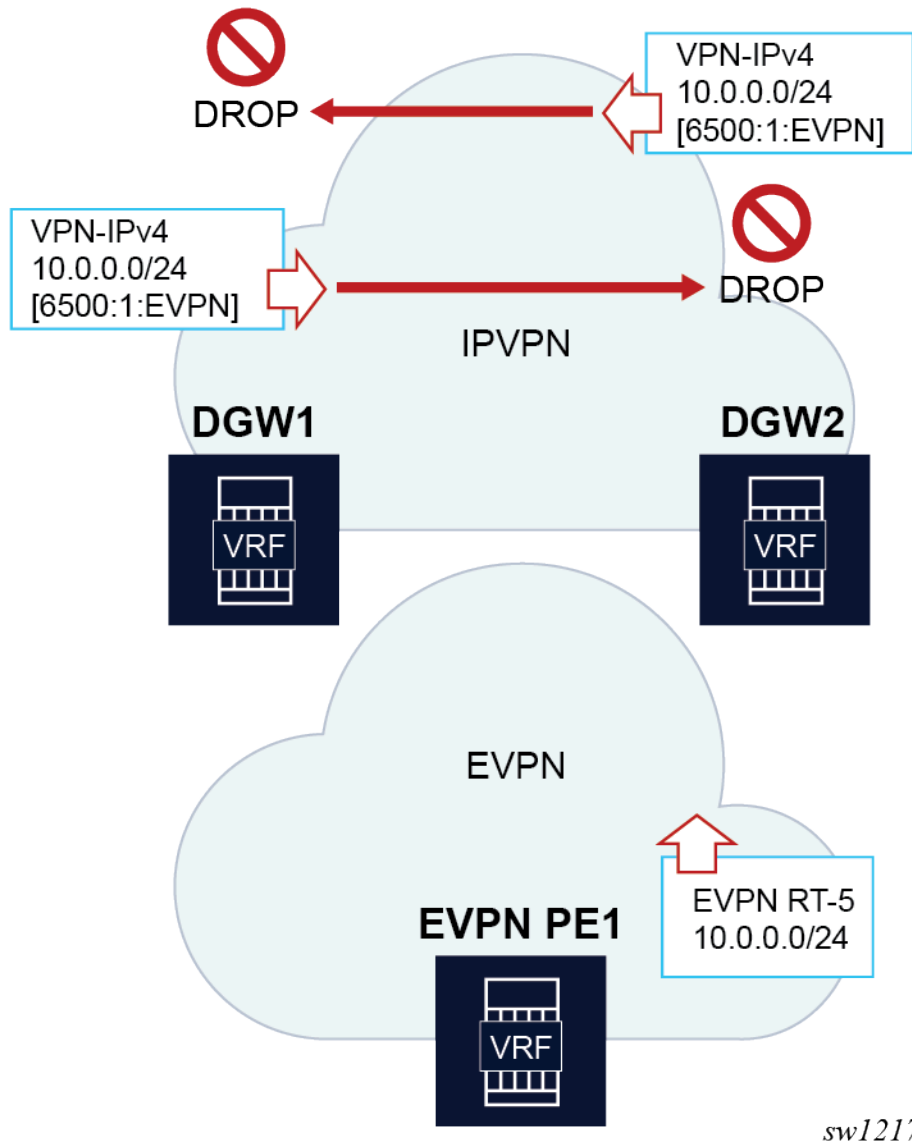
Figure 124: D-PATH attribute example



sw1216

In the D-PATH example shown in the following figure, DGW1 and DGW2 rely on the D-PATH attribute to automatically discard the prefixes received from the peer gateway in IPVPN and avoid loops by reinjecting the route back into the EVPN domain.

Figure 125: D-PATH attribute example two



**Note:** While site-of-origin extended communities and policies can be used in [Figure 125: D-PATH attribute example two](#), the D-PATH method works across multiple domains and does not require policies.

#### 5.4.20.2.1 BGP D-PATH configuration

The D-PATH attribute is modified on transmission or processed on reception based on the local VPRN or R-VPLS configuration. The domain ID is configured per-BGP instance and the ISF\_SAFI\_TYPE automatically derived from the instance type that imported the original route.

The **domain-id** is configured at **service bgp instance** level as a six-byte value that includes a global **admin** value and a local **admin** value, for example, 65000:1. Domain ID configuration is supported on:

- VPRN BGP-EVPN MPLS and SRv6 instances (EVPN-IFL)
- VPRN BGP-IPVPN MPLS and SRv6 instance
- R-VPLS BGP-EVPN MPLS and VXLAN instances (EVPN-IFF only – the R-VPLS is configured with the **evpn-tunnel** command)
- VPRN BGP neighbors (PE-CE)
- VPRN level (for local routes)

The following is an example CLI configuration:

```
// domain-id configuration

*[ex:configure service vprn "blue" bgp-evpn mpls 1]
*[ex:configure service vprn "blue" bgp-evpn segment-routing-v6 1]
*[ex:configure service vprn "blue" bgp-ipvpn mpls 1]
*[ex:configure service vprn "blue" bgp-ipvpn segment-routing-v6 1]
*[ex:configure service vprn "blue" bgp]
*[ex:configure service vpls "blue" bgp-evpn routes ip-prefix]
+-- domain-id <global-field:local-field>

*[ex:configure service vprn "blue"]
A:admin@PE-2#
+-- local-routes-domain-id <global-field:local-field>
// used as the domain-id for non-bgp routes in the VPRN.

// Example 'a'

*[ex:configure service vprn "blue" bgp-ipvpn mpls 1]
    domain-id 65000:1
```

In the preceding "example 'a'", if a VPN-IPv4 route is received from a neighbor, imported in VPRN "blue" and exported to another neighbor as EVPN, the router prepends a D-PATH segment <65000:1:IPVPN> to the advertised EVPN RT5.

```
// Example 'b'

*[ex:configure service vprn "blue"]
    local-routes-domain-id 65000:10
```

In the preceding "example 'b'", the **local-routes-domain-id** is configured at the **vprn** level. When configured, local routes (direct, static, IGP routes) are advertised with a D-PATH that contains the **vprn>local-routes-domain-id**.

The following additional considerations apply:

- If **vprn>local-routes-domain-id** is not configured, the local routes are advertised into the BGP instances with no D-PATH.
- If a VPRN BGP instance is not configured with a domain ID, the following handling applies.
  - Routes imported in the VPRN BGP instance are readadvertised in a different instance without modifying the D-PATH.
  - Routes exported in the VPRN BGP instance are advertised with the D-PATH modified to include the domain ID of the instance that imported the route in the first place.

- Up to a maximum of four domain IDs per VPRN are supported. This includes domain IDs configured in the associated R-VPLS services.
- Modifying the domain IDs list initiates a route refresh for all address families associated with the VPRN.

### 5.4.20.2.2 BGP D-PATH and BGP best path selection

D-PATH is also considered for the BGP best path selection, as described in *draft-ietf-bess-evpn-ipvpn-interworking*.

As D-PATH is introduced in networks, not all the PEs may support D-PATH for BGP path selection. In order to guarantee compatibility in networks with PEs that do not support D-PATH the following command determines if the D-PATH should be considered for BGP best-path selection.

```
ex:/configure]
A:admin@PE-3#
router "Base" bgp best-path-selection d-path-length-ignore <boolean> // default: false
service vprn <string> bgp best-path-selection d-path-length-ignore <boolean> // default: false
service vprn <string> d-path-length-ignore <boolean> // default: false

configure service system bgp evpn ip-prefix-routes d-path-length-ignore <boolean> // default:
false
```

The following conditions apply to the **d-path-length-ignore** command usage:

- When **d-path-length-ignore** is configured at the base router level (or **vprn>bgp** level for PE-CE routes), BGP ignores the D-PATH domain segment length for best path selection purposes. This ignores **d-path-length** when comparing two VPN routes or two IFL routes within the same RD. These VPN or IFL routes are processed in main BGP instance.
- When **d-path-length-ignore** is configured at the VPRN router level, the VPRN RTM ignores the D-PATH domain segment length for best path selection purposes (for routes in VPRN).
- When **d-path-length-ignore** is configured at the **service system bgp evpn ip-prefix-routes** context, EVPN ignores the D-PATH length when **iff-bgp-path-selection** is enabled.
- When **d-path-length-ignore** is not configured, the D-PATH length is considered in the BGP best path selection process (at the BGP, the RTM, and IFF levels, respectively).

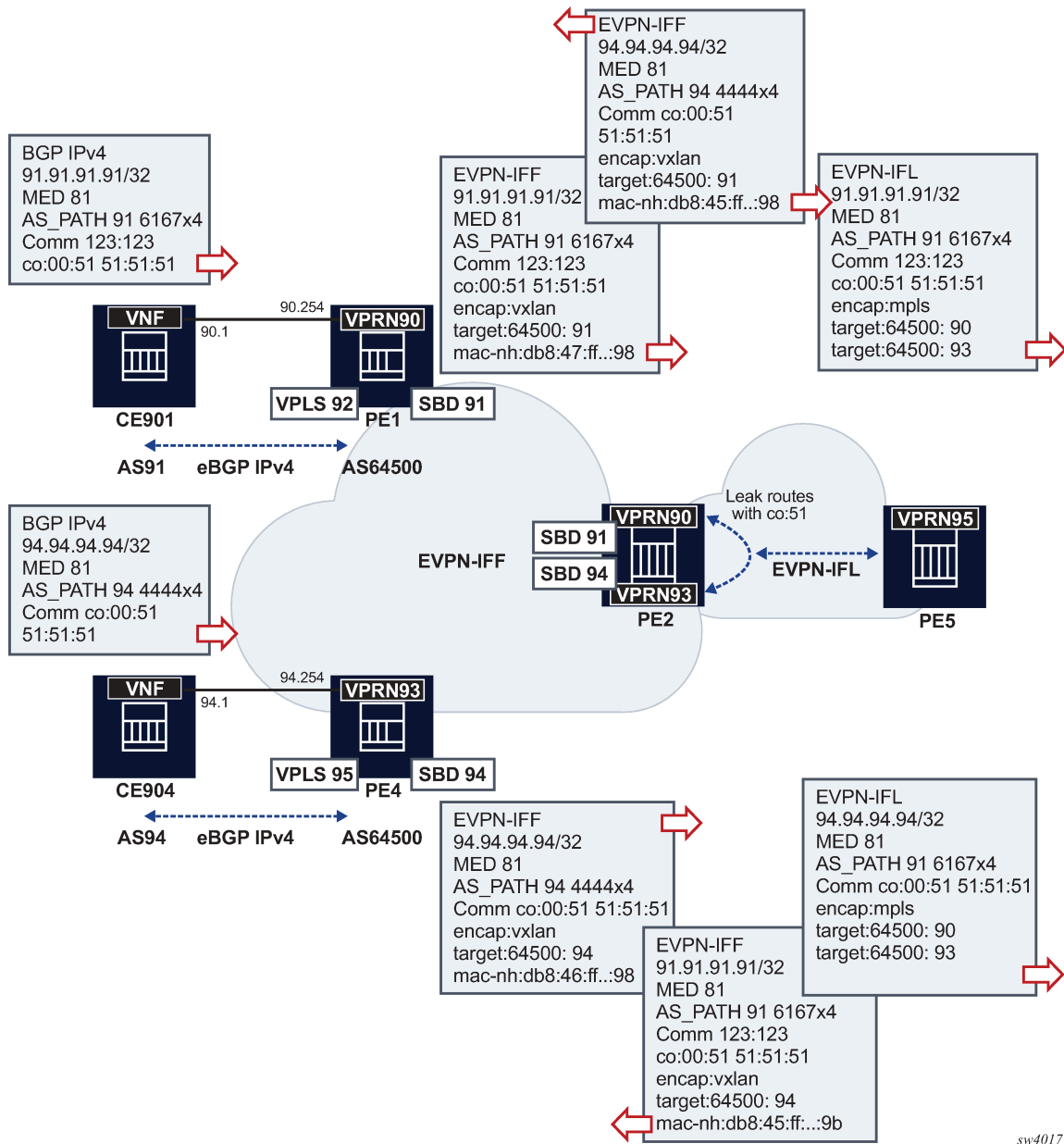
### 5.4.20.3 Configuration examples

This section describes configuration examples for stitching IPVPN and EVPN-IFL domains and the propagation of BGP path attributes for EVPN-IFF.

#### 5.4.20.3.1 Propagation of BGP path attributes for EVPN-IFF

In this configuration example, the DCGW PE2 re-exports EVPN-IFF routes into EVPN-IFF (leaked) routes and EVPN-IFL routes. The BGP path attributes are propagated as shown in [Figure 126: Propagation of BGP path attributes for EVPN-IFF](#). As described in [BGP path attribute propagation](#), EVPN extended communities, BGP encapsulation extended community and route targets are not propagated but instead, re-originated.

Figure 126: Propagation of BGP path attributes for EVPN-IFF



sw4017

The following is an example configuration for PE4 and PE2 (PE1 has equivalent configuration as PE4).

```
// PE4 services for EVPN-IFF
A:PE-4>config>service>vprn# /configure service vprn 93
A:PE-4>config>service>vprn# info
-----
router-id 4.4.4.4
autonomous-system 64500
interface "evi-95" create
  address 94.0.0.254/24
  vrrp 1 owner passive
  backup 94.0.0.254
```



```

        exit
        vpls "evi-95"
        exit
    exit
    interface "evi-94" create
        vpls "evi-94"
        evpn-tunnel
        exit
    exit
    bgp
        min-route-advertisement 1
        group "pe-ce"
        family ipv4
        type external
        export "export-al-to-vnf"
        neighbor 94.0.0.1
            local-as 64500
            peer-as 94
        exit
    exit
    no shutdown
    exit
    no shutdown
-----
A:PE-4>config>service>vprn# /configure service vpls 95
A:PE-4>config>service>vpls# info
-----
        allow-ip-int-bind
        exit
        stp
            shutdown
        exit
        sap 1/1/cl/1:90 create
            no shutdown
        exit
        no shutdown
-----
A:PE-4>config>service>vpls# /configure service vpls 94
A:PE-4>config>service>vpls# info
-----
        allow-ip-int-bind
        exit
        vxlan instance 1 vni 94 create
        exit
        bgp
        exit
        bgp-evpn
            no mac-advertisement
            ip-route-advertisement
            evi 94
            vxlan bgp 1 vxlan-instance 1
                no shutdown
            exit
        exit
        stp
            shutdown
        exit
        no shutdown
-----
// PE2 config
A:PE-2# configure service vprn 90
A:PE-2>config>service>vprn# info
-----
        interface "evi-91" create

```

```

        vpls "evi-91"
        evpn-tunnel
    exit
exit
bgp-evpn
mpls
    auto-bind-tunnel
    resolution any
    exit
    route-distinguisher 192.0.2.2:90
    vrf-export "leak-color-51-into-93"
    vrf-target import target:64500:90
    no shutdown
    exit
exit
no shutdown
-----
A:PE-2>config>service>vprn# /configure service vpls 91
A:PE-2>config>service>vpls# info
-----
    allow-ip-int-bind
    exit
    vxlan instance 1 vni 91 create
    exit
    bgp
    exit
    bgp-evpn
        no mac-advertisement
        ip-route-advertisement
        evi 91
        vxlan bgp 1 vxlan-instance 1
        no shutdown
    exit
exit
stp
    shutdown
exit
no shutdown
-----
A:PE-2>config>service>vpls# /configure service vprn 93
A:PE-2>config>service>vprn# info
-----
    interface "evi-94" create
        vpls "evi-94"
        evpn-tunnel
    exit
exit
bgp-evpn
mpls
    auto-bind-tunnel
    resolution any
    exit
    route-distinguisher 192.0.2.2:93
    vrf-export "leak-color-51-into-90"
    vrf-target import target:64500:93
    no shutdown
    exit
exit
no shutdown
-----
A:PE-2>config>service>vprn# /configure service vpls 94
A:PE-2>config>service>vpls# info
-----
    allow-ip-int-bind

```

```

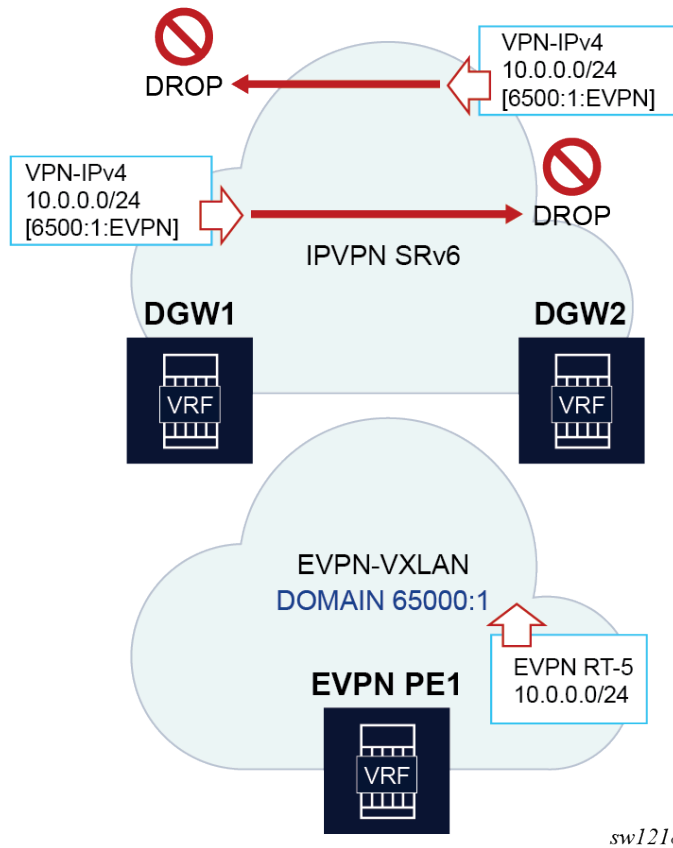
        exit
        vxlan instance 1 vni 94 create
        exit
        bgp
        exit
        bgp-evpn
        no mac-advertisement
        ip-route-advertisement
        evi 94
        vxlan bgp 1 vxlan-instance 1
        no shutdown
        exit
    exit
    stp
        shutdown
    exit
    no shutdown
-----
A:PE-2>config>service>vpls# /show router policy "leak-color-51-into-90"
    entry 10
        from
            community "color-51"
        exit
        action accept
            community add "RT64500:90" "RT64500:93"
        exit
    exit
    default-action accept
        community add "RT64500:93"
    exit
A:PE-2>config>service>vpls# /show router policy "leak-color-51-into-93"
    entry 10
        from
            community "color-51"
        exit
        action accept
            community add "RT64500:90" "RT64500:93"
        exit
    exit
    default-action accept
        community add "RT64500:90"
    exit

```

### 5.4.20.3.2 D-PATH configuration

The example in the following figure shows a typical Layer 3 EVPN DC gateway scenario where EVPN-IFF routes are translated into IPVPN routes, and vice versa. Because redundant gateways are used, this scenario is subject to Layer 3 routing loops, and the D-PATH attribute helps preventing these loops in an automatic way, without the need for extra routing policies to tag or drop routes.

Figure 127: Use of D-PATH for Layer 3 DC gateway redundancy



The following is the configuration of the VPRN or R-VPLS services in DGW1 and DGW2 in the preceding figure.

```
A:DGW1# configure service vprn 20
A:DGW1>config>service>vprn# info
-----
    interface "sbd-1" create
        vpls "sbd-1"
        evpn-tunnel
    exit
exit
segment-routing-v6 1 create
    locator "LOC-1"
        function
        end-dt46
    exit
exit
exit
bgp-ipvpn
    segment-routing-v6
        route-distinguisher 192.0.2.1:20
        srv6-instance 1 default-locator "LOC-1"
        source-address 2001:db8::1
        vrf-target target:64500:20
        domain-id 65000:2
        no shutdown
    exit
```

```

        exit
        no shutdown
*A:DGW1# configure service vpls "sbd-1"
*A:DGW1>config>service>vpls# info
-----
        allow-ip-int-bind
        exit
        vxlan instance 1 vni 1 create
        exit
        bgp
        exit
        bgp-evpn
        evi 1
        ip-route-advertisement domain-id 65000:1
        vxlan bgp 1 vxlan-instance 1
        no shutdown
        exit
    exit
    stp
        shutdown
    exit

A:DGW2# configure service vprn 20
A:DGW2>config>service>vprn# info
-----
        interface "sbd-1" create
            vpls "sbd-1"
            evpn-tunnel
        exit
    exit
    segment-routing-v6 1 create
        locator "LOC-1"
        function
        end-dt46
        exit
    exit
    exit
    bgp-ipvpn
        segment-routing-v6
            route-distinguisher 192.0.2.2:20
            srv6-instance 1 default-locator "LOC-1"
            source-address 2001:db8::2
            vrf-target target:64500:20
            domain-id 65000:2
            no shutdown
        exit
    exit
    no shutdown
*A:DGW2# configure service vpls "sbd-1"
*A:DGW2>config>service>vpls# info
-----
        allow-ip-int-bind
        exit
        vxlan instance 1 vni 1 create
        exit
        bgp
        exit
        bgp-evpn
        evi 1
        ip-route-advertisement domain-id 65000:1
        vxlan bgp 1 vxlan-instance 1
        no shutdown
        exit
    exit

```

```

    stp
      shutdown
    exit

```

The following considerations apply to the example configuration shown in [Figure 127: Use of D-PATH for Layer 3 DC gateway redundancy](#).

- Imported VPN-IP SRv6 routes are readvertised as EVPN-IFF VXLAN routes with a prepended D-PATH domain 65000:2:128.
- Imported EVPN-IFF VXLAN routes are readvertised as VPN-IP SRv6 routes with a prepended D-PATH domain 65000:1:70.

If PE1 sends an EVPN-IFF route 10.0.0.0/24 that is imported by both DGW1 and DGW2, then, when DGW1 and DGW2 receive each other's routes, they identify the D-PATH attribute and compare the list of domains with the locally configured domains in the VPRN. Since the domain matches one of the local domains, the route is not installed in the VPRN route table and it is flagged as a looped route (the **show router bgp routes detail** or **hunt** commands show **DPath Loop VRFs: 20**). In this way loops are prevented.

#### 5.4.21 Routing policies for BGP EVPN routes

Routing policies match on specific fields when importing or exporting EVPN routes. These matching fields (excluding route-table EVPN IP-prefix routes, unless explicitly mentioned), are:

- communities (*comm-val*), extended communities (*ext-comm*), and large communities (*large-comm*)
- well-known communities (*well-known-comm*); **no-export** | **no-export-subconfed** | **no-advertise**
- family EVPN
- protocol BGP-VPN (this term also matches VPN-IPv4 and VPN-IPv6 routes)
- prefix lists for type 2 routes when they contain an IP address, and for type 5 routes
- route tags that can be passed by EVPN to BGP from:
  - **service>epipe/vpls>bgp-evpn>mpls/vxlan>default-route-tag** (this route-tag can be matched on export only)
  - **service>vpls>proxy-arp/nd>evpn-route-tag** (this route tag can be matched on export only)
  - route-table route tags when exporting EVPN IP-prefix routes
- EVPN type
- BGP attributes that are applicable to EVPN routes (such as AS-path, local-preference, next-hop)

Additionally, the route tags can be used on export policies to match EVPN routes that belong to a service and BGP instance, routes that are created by the proxy-ARP/ND application, or IP-prefix routes that are added to the route-table with a route tag.

EVPN can pass only one route tag to BGP to achieve matching on export policies. In case of a conflict, the configured **default-route-tag** tag value has the least priority of the three potential tags added by EVPN.

For example, if VPLS 10 is configured with **proxy-arp>evpn-route-tag 20** and **bgp-evpn>mpls>default-route-tag 10**, all MAC/IP routes that are generated by the proxy-ARP application use route tag 20. Export policies can then use "from tag 20" to match all those routes. In this case, inclusive Multicast routes are matched by using "from tag 10".

### 5.4.21.1 Routing policies for BGP EVPN IP prefixes

BGP routing policies are supported for IP prefixes imported or exported through BGP-EVPN in R-VPLS services (EVPN-IFF routes) or VPRN services (EVPN-IFL routes).

When applying routing policies to control the distribution of prefixes between EVPN-IFF and IP-VPN (or EVPN-IFL), the user must consider that these owners are completely separate as far as BGP is concerned and when prefixes are imported in the VPRN routing table, the BGP attributes are lost to the other owner, unless the **iff-attribute-uniform-propagation** command is configured on the router.

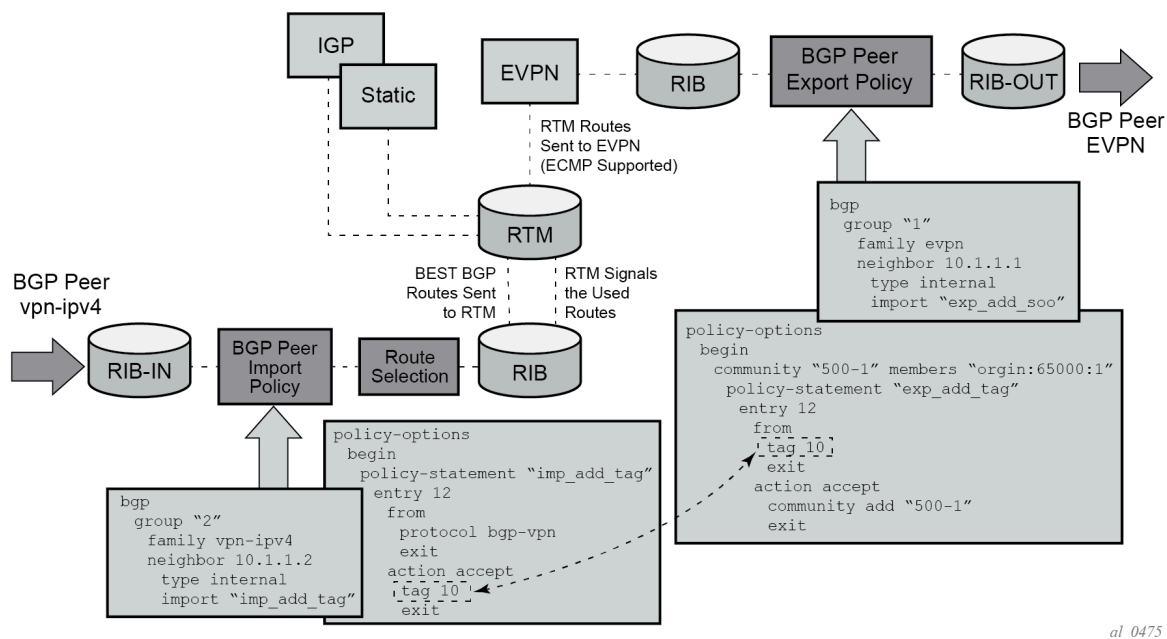
If the **iff-attribute-uniform-propagation** command is disabled, the use of route tags allows the controlled distribution of prefixes across the two families.

The following figure shows an example of how VPN-IPv4 routes are imported into the RTM, 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 128: IP-VPN import and EVPN export BGP workflow



al\_0475

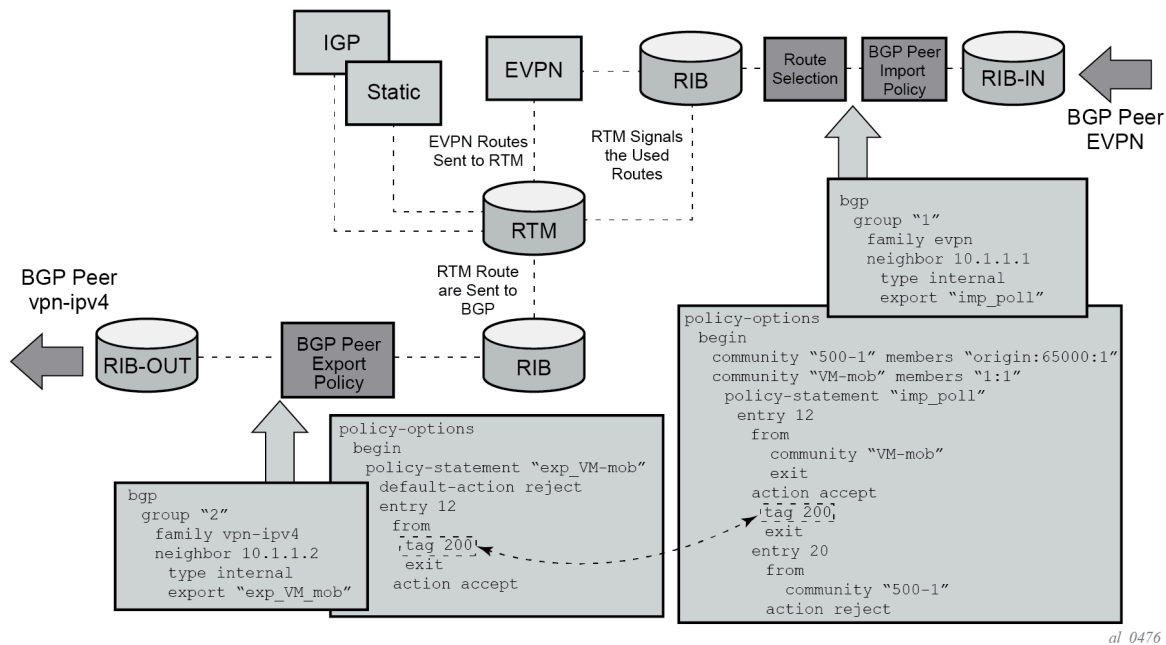
Policy tags can be used to match EVPN IP prefixes that were learned not only from BGP VPN-IPv4 but also from other routing protocols. The tag range supported for each protocol is different:

```

<tag> : accepts in decimal or hex
        [0x1..0xFFFFFFFF]H (for OSPF and IS-IS)
        [0x1..0xFFFF]H (for RIP)
        [0x1..0xFF]H (for BGP)
  
```

The following figure shows an example of the reverse workflow: routes imported from EVPN and exported from RTM to BGP VPN-IPv4.

Figure 129: EVPN import and I-VPN export BGP workflow



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

The following is a summary of the policy behavior for EVPN-IFF IP-prefixes when **iff-attribute-uniform-propagation** is disabled.

- For EVPN-IFF routes received and imported in RTM, policy entries (peer or VSI-import) match on communities or any of the following fields, and can add tags (as action):
  - communities, extended-communities or large communities
  - well-known communities
  - family EVPN
  - protocol bgp-vpn
  - prefix-lists
  - EVPN route type
  - BGP attributes (as-path, local-preference, next-hop)
- For exporting RTM to EVPN-IFF prefix routes, policy entries only match on tags, and based on this matching, add communities, accept, or reject. This applies to the peer level or on the VSI export level. Policy entries can also add tags for static routes, RIP, OSPF, IS-IS, BGP, and ARP-ND routes, which can then be matched on the BGP peer export policy, or on the VSI export policy for EVPN-IFF routes.

The following applies if the **iff-attribute-uniform-propagation** command is enabled.

For exporting RTM to EVPN-IFF prefix routes, in addition to matching on tags, matching path attributes on EVPN-IFF routes is supported in the following:

- vrf-export (when exporting the prefixes in VPN-IP or EVPN IFL or IP routes)
- vsi-export policies (when exporting the prefixes in EVPN-IFF routes)



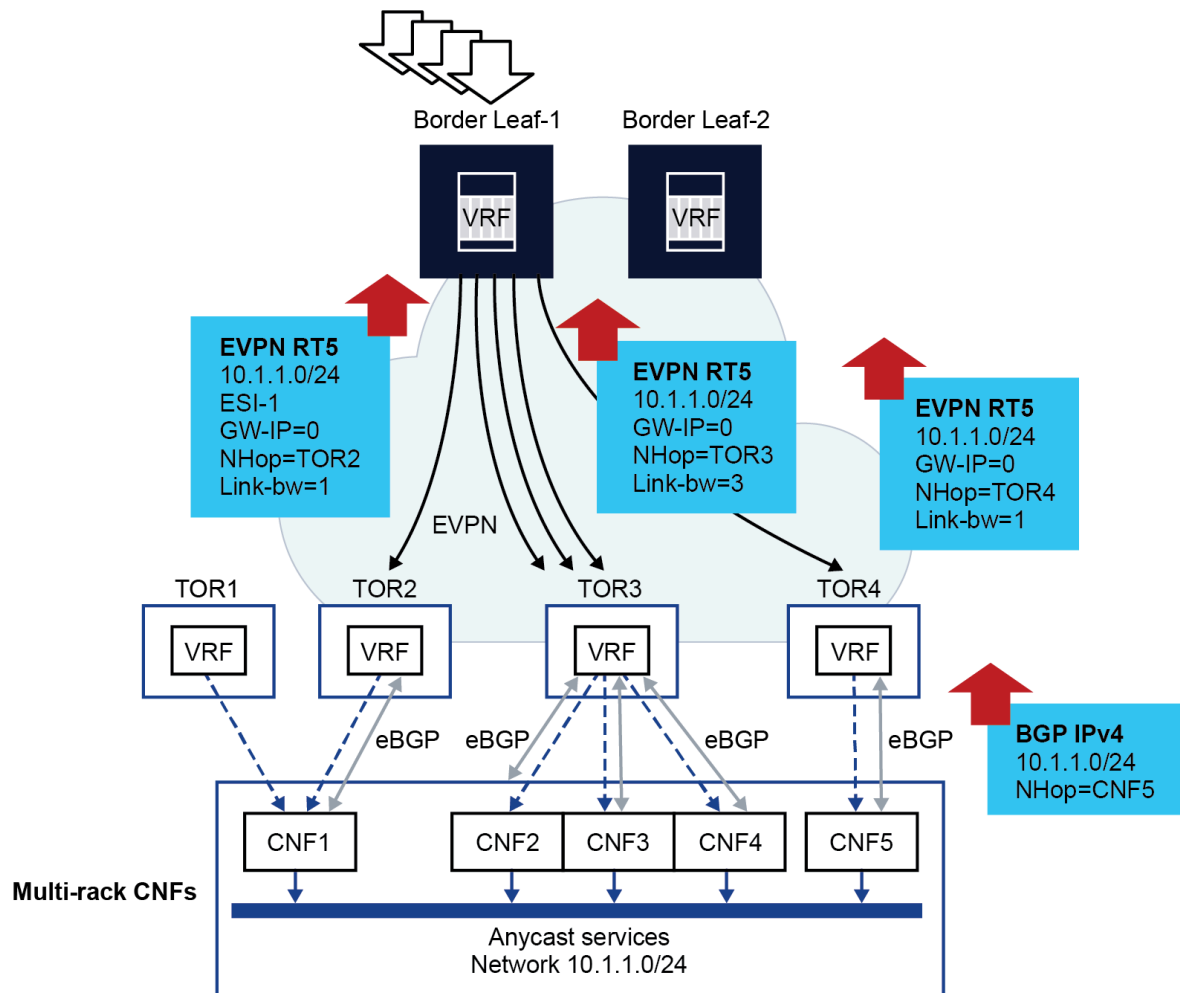
- for non-BGP route-owners (RIP, OSPF, IS-IS, static, ARP-ND), there are no changes and the only match criterion in vsi-export for EVPN-IFF routes is tags

#### 5.4.22 EVPN Weighted ECMP for IP prefix routes

SR OS supports Weighted ECMP for EVPN IP Prefix routes (IPv4 and IPv6), in the EVPN Interface-less (EVPN-IFL) and EVPN Interface-ful (EVPN-IFF) models.

Based on *draft-ietf-bess-evpn-unequal-lb*, the EVPN Link Bandwidth extended community is used in the IP Prefix routes to indicate a weight that the receiver PE must consider when load balancing traffic to multiple EVPN, CE, or both next hops. The supported weight in the extended community is of type Generalized weight and encodes the count of CEs that advertised prefix N to a PE in a BGP PE-CE route. The following figure shows the use of EVPN Weighted ECMP.

Figure 130: Weighted ECMP for IP Prefix routes use case



sw1323

In the preceding figure, some multi-rack Container Network Functions (CNFs) are connected to a few TORs in the EVPN network. Each CNF advertises the same anycast service network 10.1.1.0/24 using a single PE-CE BGP session. Without Weighted ECMP, the TOR2, TOR3 and TOR4 would re-advertise

the prefix in an EVPN IP-Prefix route and flows to 10.1.1.0/24 from the Border Leaf-1 would be equally distributed among TOR2, TOR3 and TOR4. However, the needed load balancing distribution is based on the count of CNFs that are attached to each TOR. That is, out of five flows to 10.1.1.0/24, three should be directed to TOR3 (because it has three CNFs attached), one to TOR4 and one to either TOR2 or TOR1 (since CNF1 is dual-homed to both).

Weighted ECMP achieves the needed unequal load balancing based on the CNF count on each TOR. In the [Figure 130: Weighted ECMP for IP Prefix routes use case](#) example, if Weighted ECMP is enabled, the TORs add a weight encoded in the EVPN IP Prefix route, where the weight matches the count of CNFs that each TOR has locally. The Border Leaf creates an ECMP set for prefix 10.1.1.0/24 where the weights are considered when distributing the load to the prefix.

The procedures associated with EVPN Weighted ECMP for IP Prefix routes can be divided into advertising and receiving procedures:

- Use the following commands to configure the advertising procedures for EVPN IFL.

```
configure service vprn bgp-evpn mpls evpn-link-bandwidth advertise
configure service vprn bgp-evpn segment-routing-v6 evpn-link-bandwidth advertise
```

Use the following command to configure the advertising procedures for EVPN IFF.

```
configure service vpls bgp-evpn ip-route-link-bandwidth advertise
```

The **advertise** command triggers the advertisement of the EVPN Link Bandwidth extended community with a weight that matches the CE count advertised by the route. The dynamic weight can, optionally, be overridden by configuring the **advertise weight** value.

- Use the following commands to configure the receiving procedures for EVPN-IFL.

```
configure service vprn bgp-evpn mpls evpn-link-bandwidth weighted-ecmp
configure service vprn bgp-evpn segment-routing-v6 evpn-link-bandwidth weighted-ecmp
```

Use the following command to configure the receiving procedures for EVPN-IFF.

```
configure service vpls bgp-evpn ip-route-link-bandwidth weighted-ecmp
```

When the **weighted-ecmp** command is enabled, the receiving PE installs IP Prefix routes in the VPRN route-table associated with a normalized weight that is derived from the signaled weight.

- For EVPN-IFL, for weighted ECMP across EVPN next hops and CE next hops, the following commands must be configured.

```
configure service vprn bgp group evpn-link-bandwidth add-to-received-bgp
configure service vprn bgp eibgp-loadbalance
```

- For EVPN-IFF, Weighted ECMP can only be applied to EVPN next hops and not to the **eibgp-loadbalance** command.

### Example: EVPN-IFL MPLS service configuration

The following example shows the configuration of the EVPN Weighted ECMP feature for EVPN IFL routes with MPLS transport. A similar example could have been added for EVPN IFL routes with SRv6 transport.

Suppose PE2, PE4, and PE5 are attached to the same EVPN-IFL service on vprn 2000. PE4 is connected to two CEs (CE-41 and CE-42) and PE5 to one CE (CE-51). The three CEs advertise the

same prefix 192.168.1.0/24 using PE-CE BGP and the goal is for PE2 to distribute to PE4 twice as many flows (to 192.168.1.0/24) as for PE5.

The configuration of PE4 and PE5 follows:

```
*A:PE-4# configure service vprn 2000
*A:PE-4>config>service>vprn# info
-----
      ecmp 10
      autonomous-system 64500
      interface "to-CE41" create
        address 10.41.0.1/24
        sap pxc-3.a:401 create
        exit
      exit
      interface "to-CE42" create
        address 10.42.0.1/24
        sap pxc-3.a:402 create
        exit
      exit
      bgp-evpn
        mpls
          auto-bind-tunnel
          resolution any
        exit
        evi 2000
        evpn-link-bandwidth
        advertise
        weighted-ecmp
        exit
        route-distinguisher 192.0.2.4:2000
        vrf-target target:64500:2000
        no shutdown
      exit
    exit
    bgp
      multi-path
        ipv4 10
      exit
      eibgp-loadbalance
      router-id 4.4.4.4
      rapid-withdrawal
      group "pe-ce"
        family ipv4 ipv6
        neighbor 10.41.0.2
        peer-as 64541
        evpn-link-bandwidth
        add-to-received-bgp 1
        exit
      exit
      neighbor 10.42.0.2
      peer-as 64542
      evpn-link-bandwidth
      add-to-received-bgp 1
      exit
    exit
  exit
  no shutdown
exit
no shutdown

A:PE-5# configure service vprn 2000
A:PE-5>config>service>vprn# info
```

```

-----
autonomous-system 64500
interface "to-CE51" create
    address 10.51.0.1/24
    sap pxc-3.a:501 create
    exit
exit
bgp-evpn
    mpls
        auto-bind-tunnel
            resolution any
        exit
        evi 2000
        evpn-link-bandwidth
        advertise
        weighted-ecmp
        exit
        route-distinguisher 192.0.2.5:2000
        vrf-target target:64500:2000
        no shutdown
    exit
exit
bgp
    multi-path
        ipv4 10
    exit
    eibgp-loadbalance
    router-id 5.5.5.5
    rapid-withdrawal
    group "pe-ce"
        family ipv4 ipv6
        neighbor 10.51.0.2
        peer-as 64551
        evpn-link-bandwidth
        add-to-received-bgp 1
        exit
    exit
exit
no shutdown
exit
no shutdown

```

The configuration on PE2 follows:

```

*A:PE-2# configure service vprn 2000
*A:PE-2>config>service>vprn# info
-----
ecmp 10
interface "to-PE" create
    address 20.10.0.1/24
    sap pxc-3.a:2000 create
    exit
exit
bgp-evpn
    mpls
        auto-bind-tunnel
            resolution any
        exit
        evi 2000
        evpn-link-bandwidth
        advertise
        weighted-ecmp
        exit

```

```

        route-distinguisher 192.0.2.2:2000
        vrf-target target:64500:2000
        no shutdown
    exit
exit
no shutdown

```

### Example: PE4 and PE5 IP Prefix route advertisement

As a result of the preceding configuration, PE4 (next-hop 2001:db8::4) and PE5 (next-hop 2001:db8::5) advertise the IP Prefix route from the CEs with weights 2 and 1 respectively:

```

*A:PE-2# show router bgp routes evpn ip-prefix prefix 192.168.1.0/24 community
target:64500:2000 hunt
=====
BGP Router ID:192.0.2.2      AS:64500      Local AS:64500
=====
Legend -
Status codes : u - used, s - suppressed, h - history, d - decayed, * - valid
               l - leaked, x - stale, > - best, b - backup, p - purge
Origin codes  : i - IGP, e - EGP, ? - incomplete
=====
BGP EVPN IP-Prefix Routes
=====
-----
RIB In Entries
-----
Network      : n/a
Nexthop      : 2001:db8::4
Path Id      : None
From         : 2001:db8::4
Res. Nexthop : fe80::b446:ffff:fe00:142
Local Pref.  : 100
Aggregator AS : None
Atomic Aggr. : Not Atomic
AIGP Metric  : None
Connector    : None
Community    : target:64500:2000 evpn-bandwidth:1:2
               bgp-tunnel-encap:MPLS
Cluster      : No Cluster Members
Originator Id : None
Flags        : Used Valid Best IGP
Route Source : Internal
AS-Path      : 64541
EVPN type    : IP-PREFIX
ESI          : ESI-0
Tag          : 0
Gateway Address: 00:00:00:00:00:00
Prefix       : 192.168.1.0/24
Route Dist.  : 192.0.2.4:2000
MPLS Label   : LABEL 524283
Route Tag    : 0
Neighbor-AS  : 64541
Orig Validation: N/A
Source Class : 0
Add Paths Send : Default
Last Modified : 01h19m43s

Network      : n/a
Nexthop      : 2001:db8::5
Path Id      : None
From         : 2001:db8::5

```

```

Res. Nexthop      : fe80::b449:1ff:fe01:1f
Local Pref.       : 100
Aggregator AS     : None
Atomic Aggr.      : Not Atomic
AIGP Metric       : None
Connector         : None
Community         : target:64500:2000 evpn-bandwidth:1:1
                  bgp-tunnel-encap:MPLS
Cluster           : No Cluster Members
Originator Id     : None
Flags             : Used Valid Best IGP
Route Source      : Internal
AS-Path           : 64551
EVPN type         : IP-PREFIX
ESI               : ESI-0
Tag               : 0
Gateway Address   : 00:00:00:00:00:00
Prefix            : 192.168.1.0/24
Route Dist.       : 192.0.2.5:2000
MPLS Label        : LABEL 524285
Route Tag         : 0
Neighbor-AS       : 64551
Orig Validation   : N/A
Source Class      : 0
Add Paths Send    : Default
Last Modified     : 00h08m45s
Interface Name    : int-PE-2-PE-5
Aggregator        : None
MED               : None
IGP Cost          : 10
Peer Router Id    : 192.0.2.5

```

```

-----
RIB Out Entries
-----

```

```

Routes : 2
=====

```

### Example: PE2 prefix installation

The **show router id route-table extensive** command performed on PE2, shows that PE2 installs the prefix with weights 2 and 1 respectively for PE4 and PE5:

```
*A:PE-2# show router 2000 route-table 192.168.1.0/24 extensive
```

```

=====
Route Table (Service: 2000)
=====

```

```

Dest Prefix      : 192.168.1.0/24
Protocol         : EVPN-IFL
Age              : 01h22m47s
Preference       : 170
Indirect Next-Hop : 2001:db8::4
Label            : 524283
QoS              : Priority=n/c, FC=n/c
Source-Class     : 0
Dest-Class       : 0
ECMP-Weight    : 2
Resolving Next-Hop : 2001:db8::4 (LDP tunnel)
Metric           : 10
ECMP-Weight      : N/A
Indirect Next-Hop : 2001:db8::5
Label            : 524285
QoS              : Priority=n/c, FC=n/c
Source-Class     : 0
Dest-Class       : 0
ECMP-Weight    : 1

```

```

Resolving Next-Hop : 2001:db8::5 (LDP tunnel)
Metric             : 10
ECMP-Weight        : N/A
-----
No. of Destinations: 1
=====

*A:PE-2# show router 2000 fib 1 192.168.1.0/24 extensive

=====
FIB Display (Service: 2000)
=====
Dest Prefix        : 192.168.1.0/24
Protocol           : EVPN-IFL
Installed          : Y
Indirect Next-Hop  : 2001:db8::4
Label              : 524283
QoS                : Priority=n/c, FC=n/c
Source-Class       : 0
Dest-Class         : 0
ECMP-Weight       : 2
Resolving Next-Hop : 2001:db8::4 (LDP tunnel)
ECMP-Weight        : 1
Indirect Next-Hop  : 2001:db8::5
Label              : 524285
QoS                : Priority=n/c, FC=n/c
Source-Class       : 0
Dest-Class         : 0
ECMP-Weight       : 1
Resolving Next-Hop : 2001:db8::5 (LDP tunnel)
ECMP-Weight        : 1
=====
Total Entries : 1
=====

```

### Example: EVPN-IFL handling

In case of EVPN-IFL, Weighted ECMP is also supported for EIBGP load balancing among EVPN and CE next hops. For example, PE4 installs the same prefix with an EVPN-IFL next hop and two CE next hops, and each one with its normalized weight:

```

*A:PE-4# /show router 2000 route-table 192.168.1.0/24 extensive

=====
Route Table (Service: 2000)
=====
Dest Prefix        : 192.168.1.0/24
Protocol           : BGP
Age                : 00h02m27s
Preference         : 170
Indirect Next-Hop  : 10.41.0.2
QoS                : Priority=n/c, FC=n/c
Source-Class       : 0
Dest-Class         : 0
ECMP-Weight       : 1
Resolving Next-Hop : 10.41.0.2
Interface          : to-CE41
Metric             : 0
ECMP-Weight        : N/A
Indirect Next-Hop  : 10.42.0.2
QoS                : Priority=n/c, FC=n/c
Source-Class       : 0

```

```

Dest-Class      : 0
ECMP-Weight    : 1
Resolving Next-Hop : 10.42.0.2
Interface       : to-CE42
Metric          : 0
ECMP-Weight     : N/A
Indirect Next-Hop : 2001:db8::5
Label           : 524285
QoS             : Priority=n/c, FC=n/c
Source-Class    : 0
Dest-Class      : 0
ECMP-Weight    : 1
Resolving Next-Hop : 2001:db8::5 (LDP tunnel)
Metric          : 10
ECMP-Weight     : N/A
-----
No. of Destinations: 1
=====

```

### 5.4.23 EVPN Sticky ECMP for IP prefix routes

SR OS supports sticky ECMP for EVPN-IFL and EVPN-IFF IP prefix routes. Non-sticky ECMP, or just ECMP, for a specific IP prefix with n number of next hops requires the router to rehash the flows when one of the next hops is removed or added. This may impact flows that are now sent to a different next hop.

Sticky ECMP refers to the property of the router to minimize the impact in case of a change in the number of next hops for a specific IP prefix. For example, suppose an EVPN IP prefix P has an associated ECMP set of four next hops. In this case, the following actions occur when sticky ECMP is enabled:

- Upon withdrawal of one of the next hops, only the affected flows are redistributed into the remaining three next hops, as equally as possible.
- Upon addition of the fifth next hop, the router minimizes the impact on existing flows.

The implementation of sticky ECMP is based on software. The router emulates the behavior by repeating each ECMP next hop of the sticky route a number of times (according to the next-hop normalized weight) in different hash buckets, to create a fill pattern of size N for the incoming flows. In general, the closer the number of next hops gets to the maximum number of ECMP paths, the worse the distribution algorithm works. For detailed information about the general implementation of sticky ECMP in SR OS, see "BGP support for sticky ECMP" in the *7705 SAR Gen 2 Unicast Routing Protocols Guide*.

An IP prefix is made sticky configuring the **sticky-ecmp** policy action on an import policy (at the peer or VPRN level). Sticky ECMP for EVPN IP-prefix routes is supported in combination with other ECMP features such as EVPN unequal ECMP or IP aliasing.

### 5.4.24 EVPN VLAN-aware bundle mode for BGP-EVPN VPLS or R-VPLS services

SR OS supports VLAN-aware bundle mode for BGP-EVPN VPLS or R-VPLS services, and is compliant with RFC 7432. A Broadcast Domain (BD) in RFC 7432 is mapped to a VPLS service in SR OS. Multiple BDs (VPLS services) can be grouped together under the same VLAN-aware bundle, where each BD is assigned a different Ethernet Tag ID.

Use the following command to associate a VPLS service with a bundle name.

```
configure service vpls bgp-evpn vlan-aware-bundle
```



Use the following commands to indicate the Ethernet Tag ID allocated for the VPLS service within the bundle:

- **MD-CLI**

```
configure service vpls bgp-evpn routes vlan-aware-bundle-eth-tag
```

- **classic CLI**

```
configure service vpls bgp-evpn vlan-aware-bundle eth-tag
```

When the **vlan-aware-bundle-eth-tag** command is set to a non-zero value, the EVPN service routes (types 1, 2 and 3) advertised for the VPLS service are advertised with this value into the Ethernet Tag ID field of the routes. On reception of EVPN routes with non-zero Ethernet Tag ID, BGP imports the routes based on the import route target as usual. However, the system checks the received Ethernet Tag ID field and only processes routes whose Ethernet Tag ID match the local **vlan-aware-bundle-eth-tag** value. In addition, use commands in the following context to display details of the VPLS services in a given bundle.

```
show service vlan-aware-bundle
```

The following example shows a configuration for VLAN-aware bundle, "bundle-1". This bundle is composed of two services with Ethernet Tag IDs 120 and 121, respectively.

### Example: MD-CLI

```
[ex:/configure service]
A:admin@node-2# info
  vpls "vpls-120-bundle-1" {
    admin-state enable
    service-id 120
    customer "1"
    routed-vpls {
    }
    bgp 1 {
    }
    bgp-evpn {
      evi 120
      vlan-aware-bundle "bundle-1"
      routes {
        vlan-aware-bundle-eth-tag 120
      }
    }
    mpls 1 {
      admin-state enable
      ingress-replication-bum-label true
      auto-bind-tunnel {
        resolution any
      }
    }
  }
  sap pxc-10.a:120 {
  }
}
vpls "vpls-121-bundle-1" {
  admin-state enable
  service-id 121
  customer "1"
  segment-routing-v6 1 {
    locator "LOC-1" {
      function {
        end-dt2u {

```

```

    }
    end-dt2m {
    }
  }
}
bgp 1 {
  route-target {
    export "target:64500:120"
    import "target:64500:120"
  }
}
bgp-evpn {
  evi 121
  vlan-aware-bundle "bundle-1"
  routes {
    vlan-aware-bundle-eth-tag 121
  }
  segment-routing-v6 1 {
    admin-state enable
    source-address 2001:db8::2
    srv6 {
      instance 1
      default-locator "LOC-1"
    }
  }
}
}
}

```

### Example: classic CLI

```

A:node-2>config>service# info
-----
vpls 120 name "vpls-120-bundle-1" customer 1 create
  allow-ip-int-bind
  exit
  bgp
  exit
  bgp-evpn
    vlan-aware-bundle "bundle-1" eth-tag 120
    evi 120
    mpls bgp 1
      ingress-replication-bum-label
      auto-bind-tunnel
      resolution any
    exit
    no shutdown
  exit
exit
stp
  shutdown
exit
sap pxc-10.a:120 create
  no shutdown
exit
no shutdown
exit
vpls 121 name "vpls-121-bundle-1" customer 1 create
  segment-routing-v6 1 create
    locator "LOC-1"
    function
    end-dt2u
    end-dt2m
  
```

```

        exit
    exit
exit
bgp
    route-target export target:64500:120 import target:64500:120
exit
bgp-evpn
    vlan-aware-bundle "bundle-1" eth-tag 121
    evi 121
    segment-routing-v6 bgp 1 srv6-instance 1 default-locator "LOC-1" create
        source-address 2001:db8::2
        no shutdown
    exit
exit
stp
    shutdown
exit
no shutdown
exit

```

Use the following command to display VLAN-aware bundle information.

```
# show service vlan-aware-bundle
```

### Output example

```

=====
VLAN Aware Bundle
=====
Bundle                Service Id Eth Tag   Evi
-----
bundle-1              120      120    120
                     121      121    121
-----
Number of entries: 2
-----
=====

VLAN Aware Bundle Summary
=====
MAC Entries           : 2
EVPN-MPLS Destinations : 2
EVPN-MPLS Ethernet Segment Destinations: 0
VXLAN Destinations    : 0
VXLAN Ethernet Segment Destinations : 0
SRv6 Destinations     : 2
SRv6 Ethernet segment Destinations : 0
=====

```

Use the following command to display VLAN-aware bundle forwarding database information.

```
# show service vlan-aware-bundle "bundle-1" fdb
```

### Output example

```

=====
Service Id: 120  Name: vpls-120-bundle-1
=====
Forwarding Database, Service 120

```

| ServId  | MAC                    | Source-Identifier | Type    | Last Change       |
|---|------------------------|-------------------|---------|-------------------|
|   | Transport:Tnl-Id       |                   | Age     |                   |
| 120   | 00:ca:fe:ca:fe:01      | mpls-1:           | EvpnS:P | 01/22/24 14:33:31 |
|   | ldp:65543              | 192.0.2.5:524270  |         |                   |
| -----   |                        |                   |         |                   |
| No. of MAC Entries: 1   |                        |                   |         |                   |
| -----   |                        |                   |         |                   |
| Legend:L=Learned O=0am P=Protected-MAC C=Conditional S=Static Lf=Leaf T=Trusted |                        |                   |         |                   |
| =====   |                        |                   |         |                   |
| Service Id: 121 Name: vpls-121-bundle-1   |                        |                   |         |                   |
| =====   |                        |                   |         |                   |
| Forwarding Database, Service 121  |                        |                   |         |                   |
| =====   |                        |                   |         |                   |
| ServId  | MAC                    | Source-Identifier | Type    | Last Change       |
|   | Transport:Tnl-Id       |                   | Age     |                   |
| 121   | 00:ca:fe:ca:fe:01      | srv6-1:           | EvpnS:P | 01/22/24 14:33:39 |
|   | cafe:1:0:5:7b1c:d000:: | 192.0.2.5         |         |                   |
| -----   |                        |                   |         |                   |
| No. of MAC Entries: 1   |                        |                   |         |                   |
| -----   |                        |                   |         |                   |
| Legend:L=Learned O=0am P=Protected-MAC C=Conditional S=Static Lf=Leaf T=Trusted |                        |                   |         |                   |
| =====   |                        |                   |         |                   |

5.5 Configuring an EVPN service with CLI

This section provides information to configure VPLS using the command line interface.

5.5.1 EVPN-MPLS configuration examples

This section provides EVPN-MPLS configuration examples.

5.5.1.1 EVPN all-active multihoming example

This section shows a configuration example for three 7705 SAR Gen 2 PEs, all the following assumptions are considered:

- PE-1 and PE-2 are multihomed to CE-12 that uses a LAG to get connected to the network. CE-12 is connected to LAG SAPs configured with all-active multihoming.
- PE-3 is a remote PE that performs aliasing for traffic destined for the CE-12

Example: Configuration output of VPLS-1 on PE-1 and PE-2

The following configuration example applies to a VPLS-1 on PE-1 and PE-2, as well as the corresponding **lag** commands.

```
A:PE1# configure lag 1
```

```

A:PE1>config>lag# info
-----
mode access
encap-type dot1q
port 1/1/2
lacp active administrative-key 1 system-id 00:00:00:00:69:72
no shutdown
-----
A:PE1>config>lag# /configure service system bgp-evpn
A:PE1>config>service>system>bgp-evpn# info
-----
route-distinguisher 192.0.2.69:0
exit
-----
A:PE1>config>service>system>bgp-evpn# /configure service vpls 1
A:PE1>config>service>vpls# info
-----
bgp
exit
bgp-evpn
  cfm-mac-advertisement
  evi 1
  vxlan
    shutdown
  exit
  mpls bgp 1
    ingress-replication-bum-label
    auto-bind-tunnel
    resolution any
  exit
  no shutdown
exit
exit
stp
  shutdown
exit
sap lag-1:1 create

exit
no shutdown
-----

A:PE2# configure lag 1
A:PE2>config>lag# info
-----
mode access
encap-type dot1q
port 1/1/3
lacp active administrative-key 1 system-id 00:00:00:00:69:72
no shutdown
-----
A:PE2>config>lag# configure service vpls 1
A:PE2>config>service>vpls# info
-----
bgp
exit
bgp-evpn
  cfm-mac-advertisement
  evi 1
  vxlan
    shutdown
  exit
  mpls bgp 1
    ingress-replication-bum-label

```

```

        auto-bind-tunnel
        resolution any
    exit
    no shutdown
exit
exit
stp
    shutdown
exit
sap lag-1:1 create
exit
no shutdown
-----

```

### Example: Configuration on the remote PE

The following example shows the configuration on the remote PE (for example, PE-3), which supports aliasing to PE-1 and PE-2. PE-3 only requires the VPLS-1 configuration and `ecmp>1` to perform aliasing.

```

*A:PE3>config>service>vpls# info
-----
    bgp
    exit
    bgp-evpn
        cfm-mac-advertisement
        evi 1
        mpls bgp 1
            ingress-replication-bum-label
            ecmp 4
            auto-bind-tunnel
            resolution any
        exit
        no shutdown
    exit
exit
stp
    shutdown
exit
sap 1/1/1:1 create
exit
spoke-sdp 4:13 create
    no shutdown
exit
no shutdown
-----

```

#### 5.5.1.2 EVPN single-active multihoming example

To use single-active multihoming on PE-1 and PE-2 instead of all-active multihoming:

- change the LAG configuration to **multi-homing single-active**

The CE-12 is now configured with two different LAGs; therefore the key, system ID, and system priority values must be different on PE-1 and PE-2

No changes are needed at the service level on any of the three PEs.

## Example

The following is an example configuration output showing the differences between single-active multihoming and all-active multihoming.

```
A:PE1# configure lag 1
A:PE1>config>lag# info
-----
        mode access
        encap-type dot1q
        port 1/1/2
        lacp active administrative-key 1 system-id 00:00:00:00:69:69
        no shutdown
-----
A:PE1>config>lag# /configure service system bgp-evpn
A:PE1>config>service>system>bgp-evpn# info
-----
        route-distinguisher 192.0.2.69:0
        exit
-----
A:PE2# configure lag 1
A:PE2>config>lag# info
-----
        mode access
        encap-type dot1q
        port 1/1/3
        lacp active administrative-key 1 system-id 00:00:00:00:72:72
        no shutdown
-----
A:PE2>config>lag# /configure service system bgp-evpn
A:PE2>config>service>system>bgp-evpn# info
-----
        route-distinguisher 192.0.2.72:0
        exit
-----
```

## 6 7705 SAR Gen 2 pseudowire ports

This chapter provides information about 7705 SAR Gen 2 pseudowire ports (PW ports), process overview, and implementation notes.

### 6.1 PW port list

Each port eligible to transmit traffic on a Flex PW port, must be added to a pw-port-list.

```
config>service>system> PW port list# port ?  
- no port <port-id> [<port-id>... (16 max)]  
- port <port-id> [<port-id>... (16 max)]
```

Only hybrid ports (**configure port port-id ethernet mode hybrid**) can be members of a PW port list.

A port used by Flex PW port can be shared with any other Layer 2 or Layer 3 service. For example, a Layer 3 interface using a regular SAP can be associated with a VPRN service, while the underlying port is also used by a Flex PW port. Another regular SAP from the same port can be associated with a VPLS or Epipe service at the same time.

Follow these rules when populating a PW port list:

- A port must be in hybrid mode before it is added to a PW port list.
- Before a port is removed from or added to a PW port list, all PW ports must be dissociated from the corresponding Epipe services (the PW ports must be unconfigured). This implies that all PW SAPs be deleted.
- Network interfaces (configured in Base routing context) can be configured only on ports that are in the PW port list.
- A port mode (access, network, or hybrid) cannot be changed while the port is in the PW port list.

From this, a user can consider adding all ports that are in hybrid mode to a PW port list from the beginning of the system configuration. This ensures that those ports can be used by Flex PW port at any later time, independently of their current use.

### 6.2 Failover times

Traffic loss during port switchover depends on the following factors:

- Routing convergence; this depends on the number of routes in the network and the deployed routing protocol.
- The time it takes to associate PW SAPs with a new port. This action is performed within the 7705 SAR Gen 2 and the timing depends on the number of PW SAPs that are being moved from the old port to the new port. Note that PW SAPs are not recreated, instead the existing PW-SAPs are re-mapped to a new port.



The egress queues on the new port must be recreated. However, this does not incur additional downtime because a spare egress queue is always present on a port (referred to as a failover queue) and is used while per PW SAP egress queues are being created.

Depending on the scale and network load, downtime during a switchover can range from the sub-second range to a several seconds.

## 6.3 QoS

Egress queues are attached to the port that is used by a Flex PW port to forward traffic (a Flex PW port is bound to one of the ports in the PW port list). In similar fashion, if an egress port scheduler is used, it is attached to the same port. However, the egress port schedulers must be associated by configuration with every port in the PW port list while egress queues are instantiated only on a single port. During a port switchover, egress queues are recreated on the new port and while this is occurring, the failover queue is used to forward traffic. Each port has a single egress failover queue that is used to forward traffic while SAP or subscriber queues are being recreated during transitioning events.

On the other hand, egress port scheduler must be configured by the user in advance on each port in the PW port list so that it can be ready to treat traffic immediately after its children queues are recreated on this port.

Policers are used on ingress and they do not need to be recreated during port switchover. Instead, they are re-mapped to a new port.

An example QoS configuration is provided below:

### 1. Egress port scheduler definition

```
port-scheduler-policy "flex" create
  max-rate 1000000
  group "test" create
  exit
  level 1 rate 100000
exit
```

### 2. Association between the egress port scheduler and ports

```
configure port 1/1/1
  ethernet
    mode hybrid
    encap-type qinq
    egress-scheduler-policy "flex"
  exit
no shutdown
configure port 1/1/2
  ethernet
    mode hybrid
    encap-type qinq
    egress-scheduler-policy "flex"
  exit
no shutdown
```

### 3. Association between subscriber queues or policers and the egress port scheduler

```
configure qos sap-egress 2
  queue 1 create
    port-parent level 1
    rate 10000
  exit
  queue 2 create
    port-parent level 1
    rate 10000
  exit
  queue 3 create
    port-parent level 2
    rate 1000
  exit
```

#### 4. Applying queue policy to an object:

- Subscriber management

```
configure subscriber-mgmt sla-profile "sla-profile-1"
  egress
    qos 2
  exit
```

- PW SAP in a Layer 2 service

```
configure service epipe 10
  sap pw-1:1.2
  egress
    qos 2
  exit
```

- PW SAP in a Layer 3 service

```
configure service vprn 11
  interface 'flex-int'
    address 1.1.1.1/24
    sap pw-1:1.3
    egress
      qos 2
    exit
  exit
exit
```

## 6.4 PW port termination for various tunnel types

The MPLS-based spoke SDP and L2oGRE-based spoke SDP tunnel types are supported on a Flex PW port.

## 6.4.1 MPLS-based spoke SDP

An MPLS-based spoke SDP can be rerouted between the ports defined in the PW port list and still be mapped to the same PW port based on the service label. Ethernet payload within the spoke SDP can be extracted onto a PW SAP with minimal traffic loss during port switchover.

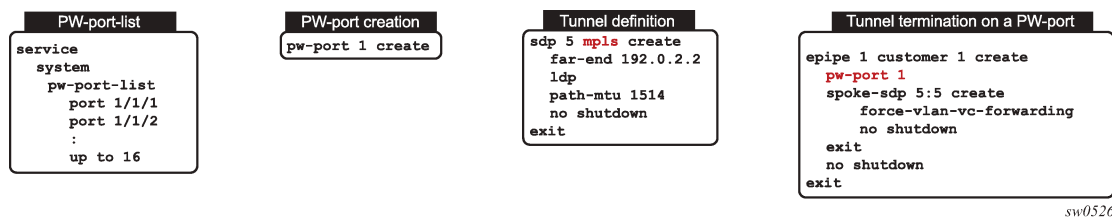
### 6.4.1.1 Provisioning

The termination of a MPLS-based spoke SDP on a Flex PW port follows the common provisioning framework:

1. Create a PW port list.
2. Add ports that are in hybrid mode to the PW port list.
3. Create a PW port.
4. Configure a tunnel.
5. Terminate a tunnel on a PW port via an Epipe service. A PW port must be configured within the Epipe before a spoke SDP is added to the same Epipe.

The steps for MPLS-based spoke SDP termination on a Flex PW port are displayed in [Figure 131: Provisioning MPLS-based spoke SDP termination on a flex PW port](#).

*Figure 131: Provisioning MPLS-based spoke SDP termination on a flex PW port*



6. When a Flex PW port is associated with a tunnel, a payload from the tunnel can be extracted using service delimiting tags (Ethernet VLANs to S-tags, C-tags in the inner Ethernet header) on a PW SAP on a Layer 2 or Layer 3 service. See [Figure 132: PW SAP configuration example](#)

Figure 132: PW SAP configuration example

**PW-SAP under L3 interface in VPRN**

```

service vprn 1 customer 1 create epipe 1 customer 1 create
interface example-if
address 192.168.1.1/24
  sap pw-1:1.2 create
    ingress
      filter ip 1000
    egress
      filter ip 2000

```

**PW-SAP under EPIPE**

```

service epipe 1 customer 1 create
  sap pw-1:* create
  spoke-sdp 1:1

```

**ESM – Capture PW-SAP**

```

service vpls 1 customer 1 create
  trigger-packet dhscp pppoe
  sap pw-1:.* capture create

```

**ESM – Static PW-SAP**

```

service vprn 1 customer 1 create
  interface subscriber-interface <name>
  address 192.168.1.1/24
  group-interface <name>
    sap pw-1:1.2 create

```

sw0527

### 6.4.1.2 Flex PW-port operational state for MPLS based spoke SDP

The operational state of the Flex PW port is driven by the ability of the Epipe service (that ties the PW port to the spoke SDP) to forward traffic. The following events renders the PW port non-operational and triggers propagation of the PW status bits toward the remote end:

The Epipe service is shut down. This raises the following flags on the local end:

- lacIngressFault
- lacEgressFault
- psnEgressFault

The corresponding PW status bits that are propagated to the remote end raises the counterpart flags on the remote end.

The PW port within the Epipe service is shutdown. This raises the following flags on the local end:

- lacIngressFault
- lacEgressFault

The corresponding PW status bits that are propagated to the remote end raises the counterpart flags on the remote end.

MTU mismatch. This raises the following flags on the local end:

- lacIngressFault
- lacEgressFault
- pwNotForwarding

The corresponding PW status bits that are propagated to the remote end raises the counterpart flags on the remote end.

In addition, PW port transitions into a non-operation state without propagating any PW status bits if the remote end cannot be reached.

The operation state of the Flex PW port state can be observed through the state of the underlying tunnel and the corresponding service via the following show command:

```
show pw-port 10 detail
=====
PW Port Information
=====
PW Port      : 10
Encap        : dot1q
IfIndex      : 1526726666
Description  : PW Port
Dot1Q Ethertype : 0x8100
Service Id   : 10239
Admin Status : up
Oper Status  : up
=====
```

### 6.4.1.3 Statistics

Statistics for the number of forwarded or dropped packets per octets per direction on a Flex PW port associated with a MPLS based spoke SDP are maintained per the spoke SDP. Octets field counts octets in customer frame (including customer's Ethernet header with VLAN tags).

The following command is used to display Flex PW port statistics along with the status of the spoke SDP associated with the PW port:

```
config>service>epipe# show pw-port 10 statistics
=====
Pw-Port 10
=====
Statistics      :
I. Fwd. Pkts.   : 110          I. Dro. Pkts.      : 0
I. Fwd. Octs.   : 23060        I. Dro. Octs.      : 0
E. Fwd. Pkts.   : 76           E. Fwd. Octets     : 16660

Grp Enc Stats   :
I. Dro. Inv. Spi. : 0           I. Dro. 0thEncPkts.: 0
E. Dro. Enc. Pkts. : 0
=====
```

## 6.4.2 L2oGRE-based spoke SDP

L2oGRE is supported for IPv4 and IPv6 transport with a termination IP address that must reside in the base router. Multiple L2oGRE tunnels can share the same termination IP address.

Each L2oGRE tunnel is represented by a unique pair of tunnel-end IP addresses. As the local endpoint address in 7705 SAR Gen 2 is usually shared between the tunnels, the tunnel far-end IP address becomes a differentiating field.

In 7705 SAR Gen 2, an L2oGRE tunnel is represented by an SDP, which is then mapped as a spoke-SDP to a Flex PW port. Although it is mandatory to configure a VC-ID, in spoke-SDP the VC-ID loses its meaning because of the nature of L2oGRE tunnel: no sub-tunnels based on an MLPS label can be multiplexed within the two L2oGRE endpoints.

### 6.4.2.1 Provisioning

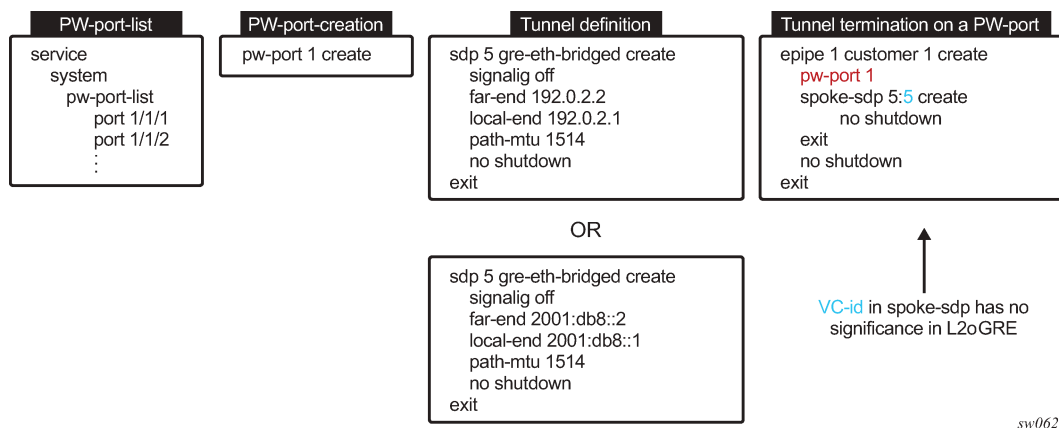
Perform the following common provisioning steps to terminate an L2oGRE tunnel on a Flex PW port:

1. Create a PW port list.
2. Add ports that are in hybrid mode to the PW port list.
3. Create a PW port.
4. Configure an L2oGRE tunnel using spoke-SDP.
5. Terminate a tunnel on a PW port using an Epipe service.

A PW port must be configured within the Epipe before a spoke-SDP is added to the same Epipe.

The steps for L2oGRE termination on a Flex PW port are displayed in [Figure 133: Provisioning L2oGRE spoke-SDP termination on a flex PW port](#).

Figure 133: Provisioning L2oGRE spoke-SDP termination on a flex PW port



After a Flex PW port is associated with a tunnel, a payload from the tunnel can be extracted using service delimiting tags (such as S-tags or C-tags in the inner Ethernet header) on a PW SAP in a Layer 2 or Layer 3 service.

### 6.4.2.2 Flex PW-port operational state for L2oGRE-based spoke SDP

The operational state of the Flex PW port is determined by the ability of the stitching service (that is, the Epipe that ties the PW port to the tunnel using L2oGRE spoke-SDP) to forward traffic. This relationship can cause the stitching service's operational status to transition to a down state in the following cases:

- the SDP far-end is not reachable
- the route-table entry is missing
- SDP is down
- the Epipe service is administratively or operationally down

### 6.4.2.3 Reassembly

Reassembly of L2oGRE over IPv4 transport is supported through a generic reassembly function that requires a vISA. Filters redirect fragmented traffic, as it enters the 7705 SAR Gen 2 node, to a vISA. After the traffic is reassembled in the vISA, it is re-inserted into the vFP where normal processing continues, as if the non-fragmented traffic had originally entered the node.

Perform the steps in [Table 16: Configuration steps for L2oGRE reassembly](#) to configure reassembly for L2oGRE.

Table 16: Configuration steps for L2oGRE reassembly

| Step   | Example CLI  | Comments  |
|--|--|---|
| 1. Create a NAT-group that contains MS-ISAs  | <pre>configure isa nat-group 1 mda 1/1</pre>   | The reassembly function is performed in a NAT group that contains a vISA.   |
| 2. Reference a reassembly-group that is used for traffic in the base routing context | <pre>configure router reassembly-group 1</pre>   | The reassembly-group that is used for traffic in the base routing context is identified. Upon reassembly, traffic is re-inserted in the same base routing context. The <b>reassembly-group id</b> corresponds to the <b>nat-group id</b> (in this case, the ID is 1). |
| 3. Identify and direct fragmented traffic to the reassembly function                 | <pre>configure filter ip-filter id default-action forward entry id create match protocol gre fragment true exit action reassemble exit</pre> | Fragmented GRE traffic is identified using a filter and is then redirected to the reassembly function. This filter must be applied to all ingress interfaces on which GRE traffic is expected to arrive.  |

## 7 Standards and protocol support

**Note:**

The information provided in this chapter is subject to change without notice and may not apply to all platforms.

Nokia assumes no responsibility for inaccuracies.

### 7.1 Bidirectional Forwarding Detection (BFD)

RFC 5880, *Bidirectional Forwarding Detection (BFD)*

RFC 5881, *Bidirectional Forwarding Detection (BFD) IPv4 and IPv6 (Single Hop)*

RFC 5882, *Generic Application of Bidirectional Forwarding Detection (BFD)*

### 7.2 Border Gateway Protocol (BGP)

draft-hares-idr-update-attr-low-bits-fix-01, *Update Attribute Flag Low Bits Clarification*

draft-ietf-idr-add-paths-guidelines-08, *Best Practices for Advertisement of Multiple Paths in IBGP*

draft-ietf-idr-best-external-03, *Advertisement of the best external route in BGP*

draft-ietf-idr-bgp-gr-notification-01, *Notification Message support for BGP Graceful Restart*

draft-ietf-idr-bgp-optimal-route-reflection-10, *BGP Optimal Route Reflection (BGP-ORR)*

draft-ietf-idr-error-handling-03, *Revised Error Handling for BGP UPDATE Messages*

draft-ietf-idr-link-bandwidth-03, *BGP Link Bandwidth Extended Community*

RFC 1772, *Application of the Border Gateway Protocol in the Internet*

RFC 1997, *BGP Communities Attribute*

RFC 2385, *Protection of BGP Sessions via the TCP MD5 Signature Option*

RFC 2439, *BGP Route Flap Damping*

RFC 2545, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*

RFC 2858, *Multiprotocol Extensions for BGP-4*

RFC 2918, *Route Refresh Capability for BGP-4*

RFC 4271, *A Border Gateway Protocol 4 (BGP-4)*

RFC 4360, *BGP Extended Communities Attribute*

RFC 4364, *BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4456, *BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)*

RFC 4486, *Subcodes for BGP Cease Notification Message*

RFC 4659, *BGP/MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN*



RFC 4684, *Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)*

RFC 4724, *Graceful Restart Mechanism for BGP – helper mode*

RFC 4760, *Multiprotocol Extensions for BGP-4*

RFC 4798, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*

RFC 5004, *Avoid BGP Best Path Transitions from One External to Another*

RFC 5065, *Autonomous System Confederations for BGP*

RFC 5291, *Outbound Route Filtering Capability for BGP-4*

RFC 5396, *Textual Representation of Autonomous System (AS) Numbers – asplain*

RFC 5492, *Capabilities Advertisement with BGP-4*

RFC 5668, *4-Octet AS Specific BGP Extended Community*

RFC 6286, *Autonomous-System-Wide Unique BGP Identifier for BGP-4*

RFC 6368, *Internal BGP as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 6793, *BGP Support for Four-Octet Autonomous System (AS) Number Space*

RFC 6810, *The Resource Public Key Infrastructure (RPKI) to Router Protocol*

RFC 6811, *Prefix Origin Validation*

RFC 6996, *Autonomous System (AS) Reservation for Private Use*

RFC 7311, *The Accumulated IGP Metric Attribute for BGP*

RFC 7606, *Revised Error Handling for BGP UPDATE Messages*

RFC 7607, *Codification of AS 0 Processing*

RFC 7752, *North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP*

RFC 7911, *Advertisement of Multiple Paths in BGP*

RFC 7999, *BLACKHOLE Community*

RFC 8092, *BGP Large Communities Attribute*

RFC 8097, *BGP Prefix Origin Validation State Extended Community*

RFC 8212, *Default External BGP (EBGP) Route Propagation Behavior without Policies*

RFC 8277, *Using BGP to Bind MPLS Labels to Address Prefixes*

RFC 9294, *Application-Specific Link Attributes Advertisement Using the Border Gateway Protocol - Link State (BGP LS)*

RFC 9494, *Long-Lived Graceful Restart for BGP*

## 7.3 Bridging and management

IEEE 802.1AB, *Station and Media Access Control Connectivity Discovery*

IEEE 802.1ad, *Provider Bridges*

IEEE 802.1AX, *Link Aggregation*

IEEE 802.1D, *MAC Bridges*  
IEEE 802.1p, *Traffic Class Expediting*  
IEEE 802.1Q, *Virtual LANs*  
IEEE 802.1s, *Multiple Spanning Trees*  
IEEE 802.1w, *Rapid Reconfiguration of Spanning Tree*

## 7.4 Certificate management

RFC 4210, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*  
RFC 4211, *Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)*  
RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*  
RFC 6712, *Internet X.509 Public Key Infrastructure -- HTTP Transfer for the Certificate Management Protocol (CMP)*  
RFC 7030, *Enrollment over Secure Transport*  
RFC 7468, *Textual Encodings of PKIX, PKCS, and CMS Structures*

## 7.5 Ethernet VPN (EVPN)

draft-ietf-bess-evpn-ipvpn-interworking-14, *EVPN Interworking with IPVPN*  
RFC 7432, *BGP MPLS-Based Ethernet VPN*  
RFC 8214, *Virtual Private Wire Service Support in Ethernet VPN*  
RFC 8365, *A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)*  
RFC 8560, *Seamless Integration of Ethernet VPN (EVPN) with Virtual Private LAN Service (VPLS) and Their Provider Backbone Bridge (PBB) Equivalents*  
RFC 9047, *Propagation of ARP/ND Flags in an Ethernet Virtual Private Network (EVPN)*  
RFC 9135, *Integrated Routing and Bridging in Ethernet VPN (EVPN)*  
RFC 9136, *IP Prefix Advertisement in Ethernet VPN (EVPN)*  
RFC 9161, *Operational Aspects of Proxy ARP/ND in Ethernet Virtual Private Networks*  
RFC 9251, *Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)*

## 7.6 gRPC Remote Procedure Calls (gRPC)

cert.proto version 0.1.0, *gNOI Certificate Management Service*  
file.proto version 0.1.0, *gNOI File Service*  
gnmi.proto version 0.8.0, *gNMI Service Specification*  
gnmi\_ext.proto, *gNMI Commit Confirmed Extension*

gnmi\_ext.proto, *gNMI Config Subscription Extension*  
gnmi\_ext.proto, *gNMI Depth Extension*  
system.proto version 1.0.0, *gNOI System Service*  
tunnel.proto version 0.2, *gRPC Tunnel Service*  
PROTOCOL-HTTP2, *gRPC over HTTP2*

## 7.7 Intermediate System to Intermediate System (IS-IS)

draft-ietf-isis-mi-02, *IS-IS Multi-Instance*  
draft-ietf-lsr-igp-ureach-prefix-announce-01, *IGP Unreachable Prefix Announcement – without U-Flag and UP-Flag*  
draft-kaplan-isis-ext-eth-02, *Extended Ethernet Frame Size Support*  
ISO/IEC 10589:2002 Second Edition, *Intermediate system to Intermediate system intra-domain routing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)*  
RFC 1195, *Use of OSI IS-IS for Routing in TCP/IP and Dual Environments*  
RFC 2973, *IS-IS Mesh Groups*  
RFC 3359, *Reserved Type, Length and Value (TLV) Codepoints in Intermediate System to Intermediate System*  
RFC 3719, *Recommendations for Interoperable Networks using Intermediate System to Intermediate System (IS-IS)*  
RFC 3787, *Recommendations for Interoperable IP Networks using Intermediate System to Intermediate System (IS-IS)*  
RFC 5120, *M-ISIS: Multi Topology (MT) Routing in IS-IS*  
RFC 5130, *A Policy Control Mechanism in IS-IS Using Administrative Tags*  
RFC 5301, *Dynamic Hostname Exchange Mechanism for IS-IS*  
RFC 5302, *Domain-wide Prefix Distribution with Two-Level IS-IS*  
RFC 5303, *Three-Way Handshake for IS-IS Point-to-Point Adjacencies*  
RFC 5304, *IS-IS Cryptographic Authentication*  
RFC 5305, *IS-IS Extensions for Traffic Engineering TE*  
RFC 5306, *Restart Signaling for IS-IS – helper mode*  
RFC 5308, *Routing IPv6 with IS-IS*  
RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*  
RFC 5310, *IS-IS Generic Cryptographic Authentication*  
RFC 6213, *IS-IS BFD-Enabled TLV*  
RFC 6232, *Purge Originator Identification TLV for IS-IS*  
RFC 6233, *IS-IS Registry Extension for Purges*  
RFC 7775, *IS-IS Route Preference for Extended IP and IPv6 Reachability*

RFC 7794, *IS-IS Prefix Attributes for Extended IPv4 and IPv6 Reachability* – sections 2.1 and 2.3  
RFC 7981, *IS-IS Extensions for Advertising Router Information*  
RFC 7987, *IS-IS Minimum Remaining Lifetime*  
RFC 8202, *IS-IS Multi-Instance* – single topology  
RFC 8570, *IS-IS Traffic Engineering (TE) Metric Extensions* – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE  
RFC 8919, *IS-IS Application-Specific Link Attributes*

## 7.8 Internet Protocol (IP) general

RFC 768, *User Datagram Protocol*  
RFC 793, *Transmission Control Protocol*  
RFC 854, *Telnet Protocol Specifications*  
RFC 1350, *The TFTP Protocol (revision 2)*  
RFC 2784, *Generic Routing Encapsulation (GRE)*  
RFC 3164, *The BSD syslog Protocol*  
RFC 4250, *The Secure Shell (SSH) Protocol Assigned Numbers*  
RFC 4251, *The Secure Shell (SSH) Protocol Architecture*  
RFC 4252, *The Secure Shell (SSH) Authentication Protocol* – publickey, password  
RFC 4253, *The Secure Shell (SSH) Transport Layer Protocol*  
RFC 4254, *The Secure Shell (SSH) Connection Protocol*  
RFC 4511, *Lightweight Directory Access Protocol (LDAP): The Protocol*  
RFC 4513, *Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms* – TLS  
RFC 4632, *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*  
RFC 5082, *The Generalized TTL Security Mechanism (GTSM)*  
RFC 5246, *The Transport Layer Security (TLS) Protocol Version 1.2* – TLS client, RSA public key  
RFC 5289, *TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)*  
RFC 5425, *Transport Layer Security (TLS) Transport Mapping for Syslog* – RFC 3164 with TLS  
RFC 5656, *Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer* – ECDSA  
RFC 5925, *The TCP Authentication Option*  
RFC 5926, *Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)*  
RFC 6398, *IP Router Alert Considerations and Usage* – MLD  
RFC 6528, *Defending against Sequence Number Attacks*  
RFC 8446, *The Transport Layer Security (TLS) Protocol Version 1.3*  
RFC 8907, *The Terminal Access Controller Access-Control System Plus (TACACS+) Protocol*

## 7.9 Internet Protocol (IP) multicast

RFC 1112, *Host Extensions for IP Multicasting*  
RFC 2236, *Internet Group Management Protocol, Version 2*  
RFC 2365, *Administratively Scoped IP Multicast*  
RFC 2375, *IPv6 Multicast Address Assignments*  
RFC 2710, *Multicast Listener Discovery (MLD) for IPv6*  
RFC 3376, *Internet Group Management Protocol, Version 3*  
RFC 3446, *Anycast Rendezvous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)*  
RFC 3590, *Source Address Selection for the Multicast Listener Discovery (MLD) Protocol*  
RFC 3618, *Multicast Source Discovery Protocol (MSDP)*  
RFC 3810, *Multicast Listener Discovery Version 2 (MLDv2) for IPv6*  
RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*  
RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*  
RFC 4610, *Anycast-RP Using Protocol Independent Multicast (PIM)*  
RFC 4611, *Multicast Source Discovery Protocol (MSDP) Deployment Scenarios*  
RFC 5059, *Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM)*  
RFC 5186, *Internet Group Management Protocol Version 3 (IGMPv3) / Multicast Listener Discovery Version 2 (MLDv2) and Multicast Routing Protocol Interaction*  
RFC 5384, *The Protocol Independent Multicast (PIM) Join Attribute Format*  
RFC 7761, *Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)*  
RFC 8487, *Mtrace Version 2: Traceroute Facility for IP Multicast*

## 7.10 Internet Protocol (IP) version 4

RFC 791, *Internet Protocol*  
RFC 792, *Internet Control Message Protocol*  
RFC 826, *An Ethernet Address Resolution Protocol*  
RFC 1034, *Domain Names - Concepts and Facilities*  
RFC 1035, *Domain Names - Implementation and Specification*  
RFC 1191, *Path MTU Discovery – router specification*  
RFC 1519, *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*  
RFC 1812, *Requirements for IPv4 Routers*  
RFC 1918, *Address Allocation for Private Internets*

RFC 2131, *Dynamic Host Configuration Protocol*; Relay only  
RFC 2132, *DHCP Options and BOOTP Vendor Extensions* – DHCP  
RFC 2401, *Security Architecture for Internet Protocol*  
RFC 3021, *Using 31-Bit Prefixes on IPv4 Point-to-Point Links*  
RFC 3046, *DHCP Relay Agent Information Option (Option 82)*  
RFC 3768, *Virtual Router Redundancy Protocol (VRRP)*  
RFC 4884, *Extended ICMP to Support Multi-Part Messages* – ICMPv4 and ICMPv6 Time Exceeded

## 7.11 Internet Protocol (IP) version 6

RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*  
RFC 2529, *Transmission of IPv6 over IPv4 Domains without Explicit Tunnels*  
RFC 3122, *Extensions to IPv6 Neighbor Discovery for Inverse Discovery Specification*  
RFC 3315, *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*  
RFC 3587, *IPv6 Global Unicast Address Format*  
RFC 3596, *DNS Extensions to Support IP version 6*  
RFC 3633, *IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6*  
RFC 3736, *Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6*  
RFC 3971, *SEcure Neighbor Discovery (SEND)*  
RFC 4007, *IPv6 Scoped Address Architecture*  
RFC 4191, *Default Router Preferences and More-Specific Routes* – Default Router Preference  
RFC 4193, *Unique Local IPv6 Unicast Addresses*  
RFC 4291, *Internet Protocol Version 6 (IPv6) Addressing Architecture*  
RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*  
RFC 4861, *Neighbor Discovery for IP version 6 (IPv6)*  
RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6*  
RFC 5722, *Handling of Overlapping IPv6 Fragments*  
RFC 5798, *Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6 – IPv6*  
RFC 5952, *A Recommendation for IPv6 Address Text Representation*  
RFC 6164, *Using 127-Bit IPv6 Prefixes on Inter-Router Links*  
RFC 8021, *Generation of IPv6 Atomic Fragments Considered Harmful*  
RFC 8200, *Internet Protocol, Version 6 (IPv6) Specification*



## 7.12 Internet Protocol Security (IPsec)

draft-ietf-ipsec-isakmp-mode-cfg-05, *The ISAKMP Configuration Method*  
draft-ietf-ipsec-isakmp-xauth-06, *Extended Authentication within ISAKMP/Oakley (XAUTH)*  
RFC 2401, *Security Architecture for the Internet Protocol*  
RFC 2403, *The Use of HMAC-MD5-96 within ESP and AH*  
RFC 2404, *The Use of HMAC-SHA-1-96 within ESP and AH*  
RFC 2405, *The ESP DES-CBC Cipher Algorithm With Explicit IV*  
RFC 2406, *IP Encapsulating Security Payload (ESP)*  
RFC 2407, *IPsec Domain of Interpretation for ISAKMP (IPsec DoI)*  
RFC 2408, *Internet Security Association and Key Management Protocol (ISAKMP)*  
RFC 2409, *The Internet Key Exchange (IKE)*  
RFC 2410, *The NULL Encryption Algorithm and Its Use With IPsec*  
RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*  
RFC 3526, *More Modular Exponential (MODP) Diffie-Hellman group for Internet Key Exchange (IKE)*  
RFC 3566, *The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec*  
RFC 3602, *The AES-CBC Cipher Algorithm and Its Use with IPsec*  
RFC 3706, *A Traffic-Based Method of Detecting Dead Internet Key Exchange (IKE) Peers*  
RFC 3947, *Negotiation of NAT-Traversal in the IKE*  
RFC 3948, *UDP Encapsulation of IPsec ESP Packets*  
RFC 4106, *The Use of Galois/Counter Mode (GCM) in IPsec ESP*  
RFC 4109, *Algorithms for Internet Key Exchange version 1 (IKEv1)*  
RFC 4301, *Security Architecture for the Internet Protocol*  
RFC 4303, *IP Encapsulating Security Payload*  
RFC 4307, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*  
RFC 4308, *Cryptographic Suites for IPsec*  
RFC 4434, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*  
RFC 4543, *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*  
RFC 4754, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*  
RFC 4835, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*  
RFC 4868, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*  
RFC 4945, *The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2 and PKIX*  
RFC 5019, *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*  
RFC 5282, *Using Authenticated Encryption Algorithms with the Encrypted Payload of the IKEv2 Protocol*

RFC 5903, *ECP Groups for IKE and IKEv2*  
RFC 5996, *Internet Key Exchange Protocol Version 2 (IKEv2)*  
RFC 5998, *An Extension for EAP-Only Authentication in IKEv2*  
RFC 6379, *Suite B Cryptographic Suites for IPsec*  
RFC 6380, *Suite B Profile for Internet Protocol Security (IPsec)*  
RFC 6960, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*  
RFC 7296, *Internet Key Exchange Protocol Version 2 (IKEv2)*  
RFC 7321, *Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)*  
RFC 7383, *Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation*  
RFC 7427, *Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)*  
RFC 8784, *Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security*

## 7.13 Label Distribution Protocol (LDP)

draft-pdutta-mpls-ldp-adj-capability-00, *LDP Adjacency Capabilities*  
draft-pdutta-mpls-ldp-v2-00, *LDP Version 2*  
RFC 3037, *LDP Applicability*  
RFC 3478, *Graceful Restart Mechanism for Label Distribution Protocol – helper mode*  
RFC 5036, *LDP Specification*  
RFC 5283, *LDP Extension for Inter-Area Label Switched Paths (LSPs)*  
RFC 5443, *LDP IGP Synchronization*  
RFC 5561, *LDP Capabilities*  
RFC 5919, *Signaling LDP Label Advertisement Completion*

## 7.14 Multiprotocol Label Switching (MPLS)

RFC 3031, *Multiprotocol Label Switching Architecture*  
RFC 3032, *MPLS Label Stack Encoding*  
RFC 3270, *Multi-Protocol Label Switching (MPLS) Support of Differentiated Services – E-LSP*  
RFC 3443, *Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks*  
RFC 5332, *MPLS Multicast Encapsulations*  
RFC 5884, *Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)*  
RFC 6424, *Mechanism for Performing Label Switched Path Ping (LSP Ping) over MPLS Tunnels*  
RFC 7308, *Extended Administrative Groups in MPLS Traffic Engineering (MPLS-TE)*



RFC 7746, *Label Switched Path (LSP) Self-Ping*

## 7.15 Network Address Translation (NAT)

RFC 4787, *Network Address Translation (NAT) Behavioral Requirements for Unicast UDP*

RFC 5382, *NAT Behavioral Requirements for TCP*

RFC 5508, *NAT Behavioral Requirements for ICMP*

## 7.16 Network Configuration Protocol (NETCONF)

RFC 5277, *NETCONF Event Notifications*

RFC 6020, *YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)*

RFC 6022, *YANG Module for NETCONF Monitoring*

RFC 6241, *Network Configuration Protocol (NETCONF)*

RFC 6242, *Using the NETCONF Protocol over Secure Shell (SSH)*

RFC 6243, *With-defaults Capability for NETCONF*

RFC 8071, *NETCONF Call Home and RESTCONF Call Home – NETCONF*

RFC 8342, *Network Management Datastore Architecture (NMDA) – Startup, Candidate, Running and Intended datastores*

RFC 8525, *YANG Library*

RFC 8526, *NETCONF Extensions to Support the Network Management Datastore Architecture – <get-data> operation*

## 7.17 Media Sanitization

NIST Special Publication 800-88 Revision 1, *Guidelines for Media Sanitization* – CF, MMC, SSD, SD, USB

## 7.18 Open Shortest Path First (OSPF)

RFC 1765, *OSPF Database Overflow*

RFC 2328, *OSPF Version 2*

RFC 3101, *The OSPF Not-So-Stubby Area (NSSA) Option*

RFC 3509, *Alternative Implementations of OSPF Area Border Routers*

RFC 3623, *Graceful OSPF Restart Graceful OSPF Restart – helper mode*

RFC 3630, *Traffic Engineering (TE) Extensions to OSPF Version 2*

RFC 4576, *Using a Link State Advertisement (LSA) Options Bit to Prevent Looping in BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 4577, *OSPF as the Provider/Customer Edge Protocol for BGP/MPLS IP Virtual Private Networks (VPNs)*

RFC 5185, *OSPF Multi-Area Adjacency*

RFC 5243, *OSPF Database Exchange Summary List Optimization*

RFC 5250, *The OSPF Opaque LSA Option*

RFC 5309, *Point-to-Point Operation over LAN in Link State Routing Protocols*

RFC 5642, *Dynamic Hostname Exchange Mechanism for OSPF*

RFC 6549, *OSPFv2 Multi-Instance Extensions*

RFC 6987, *OSPF Stub Router Advertisement*

RFC 7471, *OSPF Traffic Engineering (TE) Metric Extensions – Min/Max Unidirectional Link Delay metric for flex-algo, RSVP, SR-TE*

RFC 7684, *OSPFv2 Prefix/Link Attribute Advertisement*

RFC 7770, *Extensions to OSPF for Advertising Optional Router Capabilities*

RFC 8920, *OSPF Application-Specific Link Attributes*

## 7.19 Path Computation Element Protocol (PCEP)

draft-alvarez-pce-path-profiles-04, *PCE Path Profiles*

draft-ietf-pce-binding-label-sid-15, *Carrying Binding Label/Segment Identifier (SID) in PCE-based Networks*. – MPLS binding SIDs

RFC 5440, *Path Computation Element (PCE) Communication Protocol (PCEP)*

RFC 8231, *Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE*

RFC 8253, *PCEPS: Usage of TLS to Provide a Secure Transport for the Path Computation Element Communication Protocol (PCEP)*

RFC 8281, *PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model*

RFC 8408, *Conveying Path Setup Type in PCE Communication Protocol (PCEP) Messages*

RFC 8664, *Path Computation Element Communication Protocol (PCEP) Extensions for Segment Routing*

## 7.20 Pseudowire (PW)

draft-ietf-l2vpn-vpws-iw-oam-04, *OAM Procedures for VPWS Interworking*

MFA Forum 12.0.0, *Multiservice Interworking - Ethernet over MPLS*

MFA Forum 13.0.0, *Fault Management for Multiservice Interworking v1.0*

MFA Forum 16.0.0, *Multiservice Interworking - IP over MPLS*

RFC 3916, *Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*

RFC 3985, *Pseudo Wire Emulation Edge-to-Edge (PWE3)*

RFC 4385, *Pseudo Wire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*

RFC 4446, *IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)*  
RFC 4447, *Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*  
RFC 4448, *Encapsulation Methods for Transport of Ethernet over MPLS Networks*  
RFC 5085, *Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires*  
RFC 5659, *An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge*  
RFC 5885, *Bidirectional Forwarding Detection (BFD) for the Pseudowire Virtual Circuit Connectivity Verification (VCCV)*  
RFC 6073, *Segmented Pseudowire*  
RFC 6310, *Pseudowire (PW) Operations, Administration, and Maintenance (OAM) Message Mapping*  
RFC 6391, *Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network*  
RFC 6575, *Address Resolution Protocol (ARP) Mediation for IP Interworking of Layer 2 VPNs*  
RFC 6718, *Pseudowire Redundancy*  
RFC 6829, *Label Switched Path (LSP) Ping for Pseudowire Forwarding Equivalence Classes (FECs) Advertised over IPv6*  
RFC 6870, *Pseudowire Preferential Forwarding Status bit*  
RFC 7023, *MPLS and Ethernet Operations, Administration, and Maintenance (OAM) Interworking*  
RFC 7267, *Dynamic Placement of Multi-Segment Pseudowires*  
RFC 7392, *Explicit Path Routing for Dynamic Multi-Segment Pseudowires – ER-TLV and ER-HOP IPv4 Prefix*  
RFC 8395, *Extensions to BGP-Signaled Pseudowires to Support Flow-Aware Transport Labels*

## 7.21 Quality of Service (QoS)

RFC 2430, *A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)*  
RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*  
RFC 2597, *Assured Forwarding PHB Group*  
RFC 3140, *Per Hop Behavior Identification Codes*  
RFC 3246, *An Expedited Forwarding PHB (Per-Hop Behavior)*

## 7.22 Remote Authentication Dial In User Service (RADIUS)

draft-oscca-cfrg-sm3-02, *The SM3 Cryptographic Hash Function*  
RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*  
RFC 2866, *RADIUS Accounting*  
RFC 3162, *RADIUS and IPv6*  
RFC 6613, *RADIUS over TCP – with TLS*  
RFC 6614, *Transport Layer Security (TLS) Encryption for RADIUS*

RFC 6929, *Remote Authentication Dial-In User Service (RADIUS) Protocol Extensions*

## 7.23 Resource Reservation Protocol - Traffic Engineering (RSVP-TE)

RFC 2702, *Requirements for Traffic Engineering over MPLS*

RFC 2747, *RSVP Cryptographic Authentication*

RFC 2961, *RSVP Refresh Overhead Reduction Extensions*

RFC 3097, *RSVP Cryptographic Authentication -- Updated Message Type Value*

RFC 3209, *RSVP-TE: Extensions to RSVP for LSP Tunnels*

RFC 4124, *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering*

RFC 4125, *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4127, *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering*

RFC 4561, *Definition of a Record Route Object (RRO) Node-Id Sub-Object*

RFC 5817, *Graceful Shutdown in MPLS and Generalized MPLS Traffic Engineering Networks*

## 7.24 Routing Information Protocol (RIP)

RFC 1058, *Routing Information Protocol*

RFC 2080, *RIPng for IPv6*

RFC 2082, *RIP-2 MD5 Authentication*

RFC 2453, *RIP Version 2*

## 7.25 Segment Routing (SR)

RFC 8287, *Label Switched Path (LSP) Ping/Traceroute for Segment Routing (SR) IGP-Prefix and IGP-Adjacency Segment Identifiers (SIDs) with MPLS Data Planes*

RFC 8426, *Recommendations for RSVP-TE and Segment Routing (SR) Label Switched Path (LSP) Coexistence*

RFC 8476, *Signaling Maximum SID Depth (MSD) Using OSPF – node MSD*

RFC 8491, *Signaling Maximum SID Depth (MSD) Using IS-IS – node MSD*

RFC 8660, *Segment Routing with the MPLS Data Plane*

RFC 8661, *Segment Routing MPLS Interworking with LDP*

RFC 8665, *OSPF Extensions for Segment Routing*

RFC 8667, *IS-IS Extensions for Segment Routing*

RFC 8669, *Segment Routing Prefix Segment Identifier Extensions for BGP*

RFC 9256, *Segment Routing Policy Architecture*

RFC 9350, *IGP Flexible Algorithm*

## 7.26 Simple Network Management Protocol (SNMP)

draft-blumenthal-aes-usm-04, *The AES Cipher Algorithm in the SNMP's User-based Security Model – CFB128-AES-192 and CFB128-AES-256*

draft-ietf-isis-wg-mib-06, *Management Information Base for Intermediate System to Intermediate System (IS-IS)*

draft-ietf-mpls-ldp-mib-07, *Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP)*

draft-ietf-mpls-lsr-mib-06, *Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base Using SMIv2*

draft-ietf-mpls-te-mib-04, *Multiprotocol Label Switching (MPLS) Traffic Engineering Management Information Base*

draft-ietf-ospf-mib-update-08, *OSPF Version 2 Management Information Base*

draft-ietf-vrrp-unified-mib-06, *Definitions of Managed Objects for the VRRP over IPv4 and IPv6 – IPv6*

ESO-CONSORTIUM-MIB revision 200406230000Z, *esoConsortiumMIB*

IANA-ADDRESS-FAMILY-NUMBERS-MIB revision 200203140000Z, *ianaAddressFamilyNumbers*

IANAifType-MIB revision 200505270000Z, *ianaifType*

IANA-RTPROTO-MIB revision 200009260000Z, *ianaRtProtoMIB*

IEEE8021-CFM-MIB revision 200706100000Z, *ieee8021CfmMib*

IEEE8021-PAE-MIB revision 200101160000Z, *ieee8021paeMIB*

IEEE8023-LAG-MIB revision 200006270000Z, *lagMIB*

LLDP-MIB revision 200505060000Z, *lldpMIB*

RFC 1157, *A Simple Network Management Protocol (SNMP)*

RFC 1212, *Concise MIB Definitions*

RFC 1215, *A Convention for Defining Traps for use with the SNMP*

RFC 1724, *RIP Version 2 MIB Extension*

RFC 1901, *Introduction to Community-based SNMPv2*

RFC 2021, *Remote Network Monitoring Management Information Base Version 2 using SMIv2*

RFC 2206, *RSVP Management Information Base using SMIv2*

RFC 2578, *Structure of Management Information Version 2 (SMIv2)*

RFC 2579, *Textual Conventions for SMIv2*

RFC 2580, *Conformance Statements for SMIv2*

RFC 2787, *Definitions of Managed Objects for the Virtual Router Redundancy Protocol*

RFC 2819, *Remote Network Monitoring Management Information Base*

RFC 2856, *Textual Conventions for Additional High Capacity Data Types*

RFC 2863, *The Interfaces Group MIB*

RFC 2864, *The Inverted Stack Table Extension to the Interfaces Group MIB*

RFC 2933, *Internet Group Management Protocol MIB*

RFC 3014, *Notification Log MIB*

RFC 3273, *Remote Network Monitoring Management Information Base for High Capacity Networks*

RFC 3410, *Introduction and Applicability Statements for Internet Standard Management Framework*

RFC 3430, *Simple Network Management Protocol (SNMP) over Transmission Control Protocol (TCP) Transport Mapping*

RFC 3411, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*

RFC 3412, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*

RFC 3413, *Simple Network Management Protocol (SNMP) Applications*

RFC 3414, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*

RFC 3415, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*

RFC 3416, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*

RFC 3417, *Transport Mappings for the Simple Network Management Protocol (SNMP) – SNMP over UDP over IPv4*

RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*

RFC 3419, *Textual Conventions for Transport Addresses*

RFC 3434, *Remote Monitoring MIB Extensions for High Capacity Alarms*

RFC 3584, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*

RFC 3593, *Textual Conventions for MIB Modules Using Performance History Based on 15 Minute Intervals*

RFC 3635, *Definitions of Managed Objects for the Ethernet-like Interface Types*

RFC 3826, *The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model*

RFC 3877, *Alarm Management Information Base (MIB)*

RFC 4001, *Textual Conventions for Internet Network Addresses*

RFC 4022, *Management Information Base for the Transmission Control Protocol (TCP)*

RFC 4113, *Management Information Base for the User Datagram Protocol (UDP)*

RFC 4273, *Definitions of Managed Objects for BGP-4*

RFC 4292, *IP Forwarding Table MIB*

RFC 4293, *Management Information Base for the Internet Protocol (IP)*

RFC 4878, *Definitions and Managed Objects for Operations, Administration, and Maintenance (OAM) Functions on Ethernet-Like Interfaces*

RFC 7420, *Path Computation Element Communication Protocol (PCEP) Management Information Base (MIB) Module*

RFC 7630, *HMAC-SHA-2 Authentication Protocols in the User-based Security Model (USM) for SNMPv3*



## 7.27 Timing

RFC 3339, *Date and Time on the Internet: Timestamps*

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification*

RFC 8573, *Message Authentication Code for the Network Time Protocol*

## 7.28 Two-Way Active Measurement Protocol (TWAMP)

RFC 5357, *A Two-Way Active Measurement Protocol (TWAMP) – server, unauthenticated mode*

RFC 5938, *Individual Session Control Feature for the Two-Way Active Measurement Protocol (TWAMP)*

RFC 6038, *Two-Way Active Measurement Protocol (TWAMP) Reflect Octets and Symmetrical Size Features*

RFC 8545, *Well-Known Port Assignments for the One-Way Active Measurement Protocol (OWAMP) and the Two-Way Active Measurement Protocol (TWAMP) – TWAMP*

RFC 8762, *Simple Two-Way Active Measurement Protocol – unauthenticated*

RFC 8972, *Simple Two-Way Active Measurement Protocol Optional Extensions – unauthenticated*

RFC 9503, *Simple Two-Way Active Measurement Protocol (STAMP) Extensions for Segment Routing Networks – excluding Sections 3, 4.1.2 and 4.1.3*

## 7.29 Virtual Private LAN Service (VPLS)

RFC 4761, *Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling*

RFC 4762, *Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling*

RFC 5501, *Requirements for Multicast Support in Virtual Private LAN Services*

RFC 6074, *Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)*

RFC 7117, *Multicast in Virtual Private LAN Service (VPLS)*

## 7.30 Yet Another Next Generation (YANG)

RFC 6991, *Common YANG Data Types*

RFC 7950, *The YANG 1.1 Data Modeling Language*

RFC 7951, *JSON Encoding of Data Modeled with YANG*

# Customer document and product support



## Customer documentation

[Customer documentation welcome page](#)



## Technical support

[Product support portal](#)



## Documentation feedback

[Customer documentation feedback](#)