

INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros
VMware
Ali Sajassi
Cisco Systems
John Drake
Juniper Networks
Jorge Rabadan
Nokia
Sam Aldrin
Google

Expires: September 7, 2019

March 6, 2019

EVPN control plane for Geneve
draft-boutros-bess-evpn-geneve-04.txt

Abstract

This document describes how Ethernet VPN (EVPN) control plane can be used with Network Virtualization Overlay over Layer 3 (NVO3) Generic Network Virtualization Encapsulation (Geneve) encapsulation for NVO3 solutions. EVPN control plane can also be used by a Network Virtualization Endpoints (NVEs) to express Geneve tunnel option TLV(s) supported in transmission and/or reception of Geneve encapsulated data packets.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	3
1.1	Terminology	3
2	GENEVE extensions	4
2.1	Ethernet option TLV	4
3	BGP Extensions	6
3.1	Geneve Tunnel Option Types sub-TLV	6
4	Operation	7
5	Security Considerations	8
6	IANA Considerations	8
7	Acknowledgements	9
8	References	9
8.1	Normative References	9
8.2	Informative References	10
	Authors' Addresses	10

1 Introduction

The Network Virtualization over Layer 3 (NVO3) solutions for network virtualization in data center (DC) environment are based on an IP-based underlay. An NVO3 solution provides layer 2 and/or layer 3 overlay services for virtual networks enabling multi-tenancy and workload mobility. The NVO3 working group have been working on different dataplane encapsulations. The Generic Network Virtualization Encapsulation [GENEVE] have been recently recommended to be the proposed standard for network virtualization overlay encapsulation.

This document describes how the EVPN control plane can signal Geneve encapsulation type in the BGP Tunnel Encapsulation Extended Community defined in [TUNNEL-ENCAP]. In addition, this document defines how to communicate the Geneve tunnel option types in a new BGP Tunnel Encapsulation Attribute sub-TLV. The Geneve tunnel options are encapsulated as TLVs after the Geneve base header in the Geneve packet as described in [GENEVE].

[DT-ENCAP] recommends that a control plane determines how Network Virtualization Edge devices (NVEs) use the GENEVE option TLVs when sending/receiving packets. In particular, the control plane negotiates the subset of option TLVs supported, their order and the total number of option TLVs allowed in the packets. This negotiation capability allows, for example, interoperability with hardware-based NVEs that can process fewer options than software-based NVEs.

This EVPN control plane extension will allow a Network Virtualization Edge (NVE) to express what Geneve option TLV types it is capable to receive or to send over the Geneve tunnel to its peers.

In the datapath, a transmitting NVE MUST NOT encapsulate a packet destined to another NVE with any option TLV(s) the receiving NVE is not capable of processing.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Most of the terminology used in this documents comes from [RFC7432] and [NVO3-FRWK].

NVO3: Network Virtualization Overlay over Layer 3

GENEVE: Generic Network Virtualization Encapsulation.

NVE: Network Virtualization Edge.

VNI: Virtual Network Identifier.

MAC: Media Access Control.

OAM: Operations, Administration and Maintenance.

PE: Provide Edge Node.

CE: Customer Edge device e.g., host or router or switch.

EVPN: Ethernet VPN.

EVI: An EVPN instance spanning the Provider Edge (PE) devices participating in that EVPN.

MAC-VRF: A Virtual Routing and Forwarding table for Media Access Control (MAC) addresses on a PE.

2. GENEVE extensions

This document adds some extensions to the [GENEVE] encapsulation that are relevant to the operation of EVPN.

2.1 Ethernet option TLV

[EVPN-OVERLAY] describes when an ingress NVE uses ingress replication to flood unknown unicast traffic to the egress NVEs, the ingress NVE needs to indicate to the egress NVE that the Encapsulated packet is a BUM traffic type. This is required to avoid transient packet duplication in all-active multi-homing scenarios. For GENVE encapsulation we need a bit to for this purpose.

[RFC8317] uses MPLS label for leaf indication of BUM traffic originated from a leaf AC in an ingress NVE so that the egress NVEs can filter BUM traffic toward their leaf ACs. For GENVE encapsulation we need a bit for this purpose.

Although the default mechanism for split-horizon filtering of BUM traffic on an Ethernet segment for IP-based encapsulations such as VxLAN, GPE, NVGRE, and GENVE, is local-bias as defined in section 8.3.1 of [EVPN-OVERLAY], there can be an incentive to leverage the same split-horizon filtering mechanism of [RFC7432] that uses a 20-bit MPLS label so that a) the a single filtering mechanism is used for all encapsulation types and b) the same PE can participate in a mix of MPLS and IP encapsulations. For this purpose a 20-bit label

field MAY be defined for GENVE encapsulation. The support for this label is optional.

If an NVE wants to use local-bias procedure, then it sends the new option TLV without ESI-label (e.g., length=4):

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Option Class=Ethernet      |Type=0      |B|L|R| Len=0x1 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

If an NVE wants to use ESI-label, then it sends the new option TLV with ESI-label (e.g., length=8)

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Option Class=Ethernet      |Typ=EVPN-OPTION|B|L|R| Len=0x2 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      Rsvd      |      Source-ID      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Where:

- Option Class is set to Ethernet (new Option Class requested to IANA)
- Type is set to EVPN-OPTION (new type requested to IANA) and C bit must be set.
- B bit is set to 1 for BUM traffic.
- L bit is set to 1 for Leaf-Indication.
- Source-ID is a 24-bit value that encodes the ESI-label value signaled on the EVPN Autodiscovery per-ES routes, as described in [RFC7432] for multi-homing and [RFC8317] for leaf-to-leaf BUM filtering. The ESI-label value is encoded in the high-order 20 bits of the Source-ESI field.

The egress NVEs that make use of ESIs in the data path (because they have a local multi-homed ES or support [RFC8317]) SHOULD advertise their Ethernet A-D per-ES routes along with the Geneve tunnel sub-TLV and in addition to the ESI-label Extended Community. The ingress NVE can then use the Ethernet option-TLV when sending GENEVE packets based on the [RFC7432] and [RFC8317] procedures. The egress NVE will use the Source-ID field in the received packets to make filtering decisions.

Note that [EVPN-OVERLAY] modifies the [RFC7432] split-horizon procedures for NVO3 tunnels using the "local-bias" procedure. "Local-

bias" relies on tunnel IP source address checks (instead of ESI-labels) to determine whether a packet can be forwarded to a local ES.

While "local-bias" MUST be supported along with GENEVE encapsulation, the use of the Ethernet option-TLV is RECOMMENDED to follow the same procedures used by EVPN MPLS.

An ingress NVE using ingress replication to flood BUM traffic MUST send B=1 in all the GENEVE packets that encapsulate BUM frames. An egress NVE SHOULD determine whether a received packet encapsulates a BUM frame based on the B bit. The use of the B bit is only relevant to GENEVE packets with Protocol Type 0x6558 (Bridged Ethernet).

3. BGP Extensions

As per [EVPN-OVERLAY] the BGP Encapsulation extended community defined in [TUNNEL-ENCAP] is included with all EVPN routes advertised by an egress NVE.

This document specifies a new BGP Tunnel Encapsulation Type for Geneve and a new Geneve tunnel option types sub-TLV as described below.

3.1 Geneve Tunnel Option Types sub-TLV

The Geneve tunnel option types is a new BGP Tunnel Encapsulation Attribute Sub-TLV.

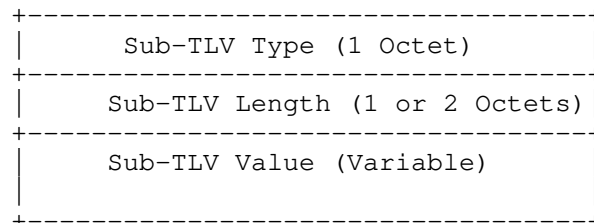


Figure 1: Geneve tunnel option types sub-TLV

The Sub-TLV Type field contains a value in the range from 192-252. To be allocated by IANA.

Sub-TLV value MUST match exactly the first 4-octets of the option TLV format. For instance, if we need to signal support for two option TLVs:

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
Option Class										Type										R R R Length																			
Option Class										Type										R R R Length																			

Where, an NVE receiving the above sub-TLV, will send GENEVE packets to the originator NVE with only the option TLVs the receiver NVE is capable of receiving, and following the same order. Also the high order bit in the type, is the critical bit, MUST be set accordingly.

The above sub-TLV(s) MAY be included with only Ethernet A-D per-ES routes.

4. Operation

The following figure shows an example of an NVO3 deployment with EVPN.

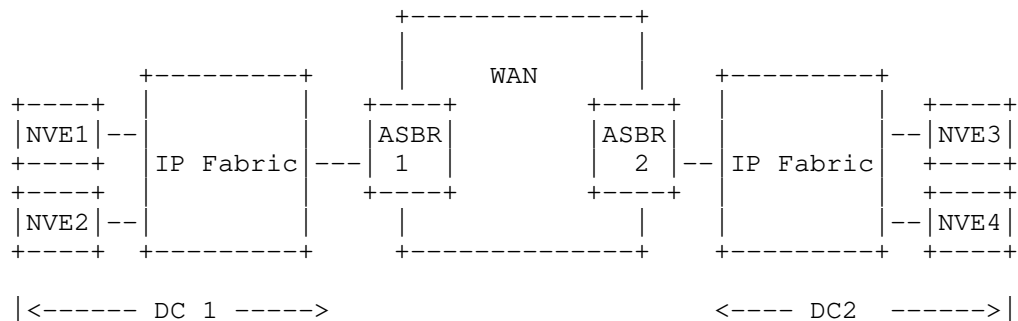


Figure 2: Data Center Interconnect with ASBR

iBGP sessions are established between NVE1, NVE2, ASBR1, possibly via a BGP route-reflector. Similarly, iBGP sessions are established between NVE3, NVE4, ASBR2.

eBGP sessions are established among ASBR1 and ASBR2.

All NVEs and ASBRs are enabled for the EVPN SAFI and exchange EVPN routes. For inter-AS option B, the ASBRs re-advertise these routes with NEXT_HOP attribute set to their IP addresses as per [RFC4271].

NVE1 sets the BGP Encapsulation extended community defined in all EVPN routes advertised. NVE1 sets the BGP Tunnel Encapsulation Attribute Tunnel Type to Geneve tunnel encapsulation, and sets the Tunnel Encapsulation Attribute Tunnel sub-TLV for the Geneve tunnel option types with all the Geneve option types it can transmit and receive.

All other NVE(s) learn what Geneve option types are supported by NVE1 through the EVPN control plane. In the datapath, NVE2, NVE3 and NVE4 only encapsulate overlay packets with the Geneve option TLV(s) that NVE1 is capable of receiving.

A PE advertises the BGP Encapsulation extended community defined in [RFC5512] if it supports any of the encapsulations defined in [EVPN-OVERLAY]. A PE advertises the BGP Tunnel Encapsulation Attribute defined in [TUNNEL-ENCAP] if it supports Geneve encapsulation.

5. Security Considerations

The mechanisms in this document use EVPN control plane as defined in [RFC7432]. Security considerations described in [RFC7432] are equally applicable.

This document uses IP-based tunnel technologies to support data plane transport. Security considerations described in [RFC7432] and in [EVPN-OVERLAY] are equally applicable.

6. IANA Considerations

IANA is requested to allocate the following:

BGP Tunnel Encapsulation Attribute
Tunnel Type:

XX Geneve Encapsulation

BGP Tunnel Encapsulation Attribute Sub-TLVs a Code point from the range of 192-252 for Geneve tunnel option types sub-TLV.

IANA is requested to assign a new option class from the "Geneve Option Class" registry for the Ethernet option TLV.

Option Class	Description
--------------	-------------

XXXX-----
Ethernet option

7. Acknowledgements

The authors wish to thank T. Sridhar, for his input, feedback, and helpful suggestions.

8. References

8.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<http://www.rfc-editor.org/info/rfc7432>>.

[RFC8317] Sajassi, et al. "Ethernet-Tree (E-Tree) Support in Ethernet VPN (EVPN) and Provider Backbone Bridging EVPN (PBB-EVPN)", RFC 8317, January 2018, <<http://www.rfc-editor.org/info/rfc8317>>.

[RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006, <<http://www.rfc-editor.org/info/rfc4271>>.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.

[GENEVE] Gross, et al. "Geneve: Generic Network Virtualization Encapsulation", draft-ietf-nvo3-geneve-05, work in progress, September, 2017.

[DT-ENCAP] Boutros, et al. "NVO3 Encapsulation Considerations", draft-ietf-nvo3-encap-01, work in progress, October, 2017.

[TUNNEL-ENCAP] Rosen et al., "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-07, work in progress, July, 2017.

[EVPN-OVERLAY] Sajassi-Drake et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-10.txt, work in progress, December, 2017

8.2 Informative References

[NVO3-FRWK] Lasserre et al., "Framework for DC Network Virtualization", RFC 7365, October 2014.

Authors' Addresses

Sami Boutros
VMware, Inc.
Email: boutross@vmware.com

Ali Sajassi
Cisco
Email: sajassi@cisco.com

John Drake
Juniper Networks
Email: jdrake@juniper.net

Jorge Rabadan
Nokia
Email: jorge.rabadan@nokia.com

Sam Aldrin
Google
Email: aldrin.ietf@gmail.com

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 9, 2020

G. Dawra, Ed.
LinkedIn
C. Filsfils
P. Brissette
S. Agrawal
Cisco Systems
J. Leddy
Comcast
D. Voyer
D. Bernier
Bell Canada
D. Steinberg
Steinberg Consulting
R. Raszuk
Bloomberg LP
B. Decraene
Orange
S. Matsushima
SoftBank
S. Zhuang
Huawei Technologies
J. Rabadan
Nokia
July 8, 2019

SRv6 BGP based Overlay services
draft-dawra-bess-srv6-services-02

Abstract

This draft defines procedures and messages for SRv6-based BGP services including L3VPN, EVPN and Internet services. It builds on RFC4364 "BGP/MPLS IP Virtual Private Networks (VPNs)" and RFC7432 "BGP MPLS-Based Ethernet VPN".

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 9, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. SRv6 Services TLVs	4
2.1. SRv6 Service Sub-TLVs	5
2.1.1. SRv6 SID Information Sub-TLV	6
2.1.2. SRv6 Service Data Sub-Sub-TLVs	7
3. BGP based L3 service over SRv6	9
3.1. IPv4 VPN Over SRv6 Core	10
3.2. IPv6 VPN Over SRv6 Core	10
3.3. Global IPv4 over SRv6 Core	11
3.4. Global IPv6 over SRv6 Core	11
4. BGP based Ethernet VPN (EVPN) over SRv6	12
4.1. Ethernet Auto-discovery route over SRv6 Core	12
4.1.1. Per-ES A-D route	13
4.1.2. Per-EVI A-D route	13

4.2.	MAC/IP Advertisement route over SRv6 Core	14
4.3.	Inclusive Multicast Ethernet Tag Route over SRv6 Core . .	16
4.4.	Ethernet Segment route over SRv6 Core	17
4.5.	IP prefix route over SRv6 Core	17
4.6.	EVPN multicast routes (Route Types 6, 7, 8) over SRv6 core	18
5.	Encoding SRv6 SID information	18
6.	Implementation Status	19
7.	Error Handling	20
8.	IANA Considerations	21
8.1.	BGP Prefix-SID TLV Types registry	21
8.2.	SRv6 Service Sub-TLV Types registry	22
8.3.	SRv6 Service Data Sub-Sub-TLV Types registry	22
9.	Security Considerations	22
10.	Conclusions	23
11.	References	23
11.1.	Normative References	23
11.2.	Informative References	24
Appendix A.	Contributors	26
	Authors' Addresses	26

1. Introduction

SRv6 refers to Segment Routing instantiated on the IPv6 dataplane [I-D.ietf-spring-srv6-network-programming] [I-D.ietf-6man-segment-routing-header].

SRv6 based BGP services refers to the L3 and L2 overlay services with BGP as control plane and SRv6 as dataplane.

SRv6 SID refers to a SRv6 Segment Identifier as defined in [I-D.ietf-spring-srv6-network-programming].

SRv6 Service SID refers to an SRv6 SID associated with one of the service specific behavior on the advertising Provider Edge (PE) router, such as (but not limited to), END.DT (Table lookup in a VRF) or END.DX (cross-connect to a nexthop) behaviors in the case of L3VPN service as defined in [I-D.ietf-spring-srv6-network-programming].

To provide SRv6 service with best-effort connectivity, the egress PE signals an SRv6 Service SID with the BGP overlay service route. The ingress PE encapsulates the payload in an outer IPv6 header where the destination address is the SRv6 Service SID provided by the egress PE. The underlay between the PEs only need to support plain IPv6 forwarding [RFC8200].

To provide SRv6 service in conjunction with an underlay SLA from the ingress PE to the egress PE, the egress PE colors the overlay service

route with a Color extended community [I-D.ietf-idr-segment-routing-te-policy]. The ingress PE encapsulates the payload packet in an outer IPv6 header with an SRH that contains the segment list of SR policy associated with the related SLA followed by the SRv6 Service SID associated with the route. The underlay nodes whose SRv6 SID's are part of the SRH MUST support SRv6 data plane.

BGP is used to advertise the reachability of prefixes of a particular service from an egress PE to ingress PE nodes.

This document describes how existing BGP messages between PEs may carry SRv6 Service SIDs as a means to interconnect PEs and form VPNs.

2. SRv6 Services TLVs

This document extends the BGP Prefix-SID attribute [I-D.ietf-idr-bgp-prefix-sid] to carry SRv6 SIDs and associated information.

The SRv6 Service TLVs are defined as two new TLVs of the BGP Prefix-SID Attribute to achieve signaling of SRv6 SIDs for L3 and L2 services.

- o SRv6 L3 Service TLV: This TLV encodes Service SID information for SRv6 based L3 services. It corresponds to the equivalent functionality provided by an MPLS Label when received with a Layer 3 service route. Some behaviors which MAY be encoded, but not limited to, are End.DX4, End.DT4, End.DX6, End.DT6, etc.
- o SRv6 L2 Service TLV: This TLV encodes Service SID information for SRv6 based L2 services. It corresponds to the equivalent functionality provided by an MPLS Label for EVPN Route-Types as defined in [RFC7432]. Some behaviors which MAY be encoded, but not limited to, are End.DX2, End.DX2V, End.DT2U, End.DT2M etc.

When an egress PE is enabled for BGP Services over SRv6 data-plane, it MUST signal one or more SRv6 Service SIDs enclosed in SRv6 Service TLV(s) within the BGP Prefix-SID Attribute attached to MP-BGP NLRIs defined in [RFC4760] [RFC4659] [RFC5549] [RFC7432] [RFC4364] where applicable as described in section 3 and 4.

The following depicts the SRv6 Service TLVs encoded in the BGP Prefix-SID Attribute:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
|  TLV Type   |          TLV Length          |  RESERVED   |
+-----+-----+-----+-----+-----+-----+-----+-----+
//  SRv6 Service Sub-TLVs                                     //
+-----+-----+-----+-----+-----+-----+-----+-----+

```

- o TLV Type (1 octet): This field is assigned values from the IANA registry "BGP Prefix-SID TLV Types". It is set to [TBD1] (to be assigned by IANA) for SRv6 L3 Service TLV. It is set to [TBD2] (to be assigned by IANA) for SRv6 L2 Service TLV.
- o TLV Length (2 octets): Specifies the total length of the TLV Value.
- o RESERVED (1 octet): This field is reserved; it SHOULD be set to 0 by the sender and MUST be ignored by the receiver.
- o SRv6 Service Sub-TLVs (variable): This field contains SRv6 Service related information and is encoded as an unordered list of Sub-TLVs whose format is described below.

2.1. SRv6 Service Sub-TLVs

The format of a single SRv6 Service Sub-TLV is depicted below:

```

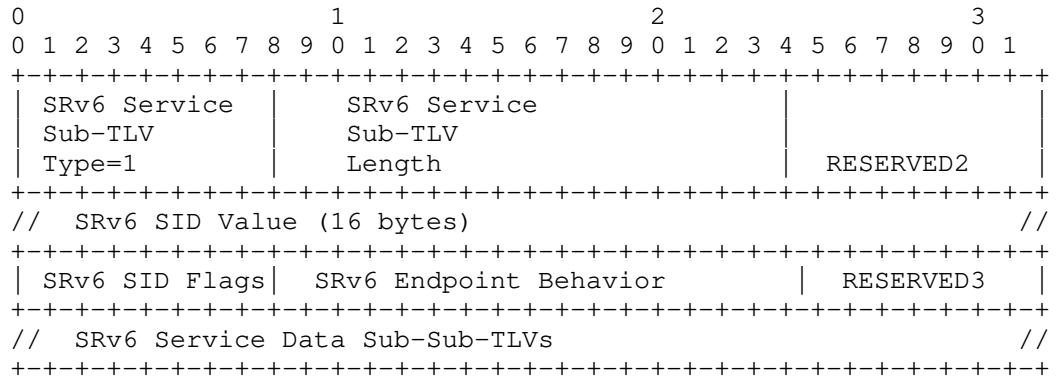
      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| SRv6 Service |          SRv6 Service          | SRv6 Service //
| Sub-TLV      |          Sub-TLV          | Sub-TLV      //
| Type         |          Length         | value        //
+-----+-----+-----+-----+-----+-----+-----+-----+

```

- o SRv6 Service Sub-TLV Type (1 octet): Identifies the type of SRv6 service information. It is assigned values from the IANA Registry "SRv6 Service Sub-TLV Types".
- o SRv6 Service Sub-TLV Length (2 octets): Specifies the total length of the Sub-TLV Value field.
- o SRv6 Service Sub-TLV Value (variable): Contains data specific to the Sub-TLV Type. In addition to fixed length data, this may also optionally contain other properties of the SRv6 Service encoded as a set of SRv6 Service Data Sub-Sub-TLVs whose format is described in another sub-section below.

2.1.1.1. SRv6 SID Information Sub-TLV

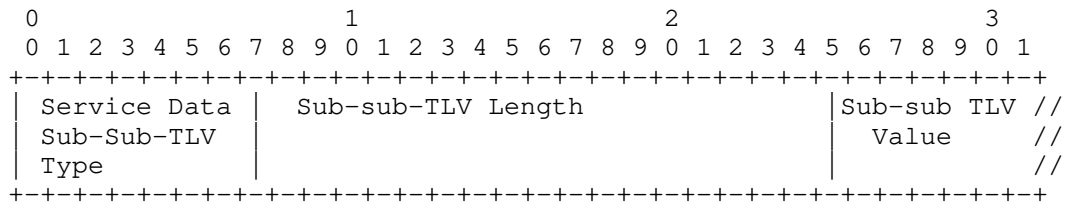
SRv6 Service Sub-TLV Type 1 is assigned for SRv6 SID Information Sub-TLV. This Sub-TLV contains a single SRv6 SID along with its properties. Its encoding is depicted below:



- o SRv6 Service Sub-TLV Type (1 octet): This field is set to 1 to represent SRv6 SID Information Sub-TLV.
- o SRv6 Service Sub-TLV Length (2 octets): This field contains the total length of the Value field of the Sub-TLV.
- o RESERVED2 (1 octet): SHOULD be set to 0 by the sender and MUST be ignored by the receiver.
- o SRv6 SID Value (16 octets): Encodes an SRv6 SID as defined in [I-D.ietf-spring-srv6-network-programming]
- o SRv6 SID Flags (1 octet): Encodes SRv6 SID Flags - none are currently defined.
- o SRv6 Endpoint Behavior (2 octets): Encodes SRv6 Endpoint behavior defined in [I-D.ietf-spring-srv6-network-programming]. This field SHOULD be set to the value 0xFFFF indicating opaque behavior unless the router wants to signal the actual behavior.
- o RESERVED3 (1 octet): SHOULD be set to 0 by the sender and MUST be ignored by the receiver.
- o SRv6 Service Data Sub-Sub-TLV Value (variable): This field contains optional properties of the SRv6 SID. It is encoded as a set of SRv6 Service Data Sub-Sub-TLVs.

2.1.2. SRv6 Service Data Sub-Sub-TLVs

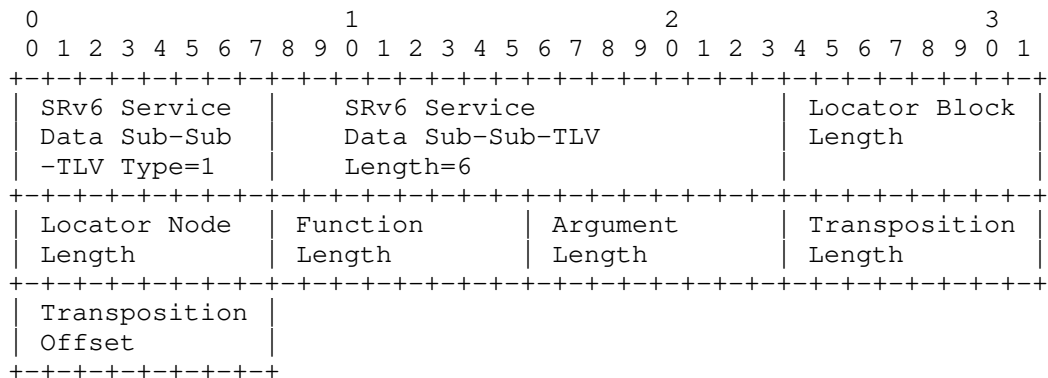
The format of the SRv6 Service Data Sub-Sub-TLV is depicted below:



- o SRv6 Service Data Sub-Sub-TLV Type (1 octet): Identifies the type of Sub-Sub-TLV. It is assigned values from the IANA Registry "SRv6 Service Data Sub-Sub-TLVs".
- o SRv6 Service Data Sub-Sub-TLV Length (2 octets): Specifies the total length of the Sub-Sub-TLV Value field.
- o SRv6 Service Data Sub-Sub-TLV Value (variable): Contains data specific to the Sub-Sub-TLV Type.

2.1.2.1. SRv6 SID Structure Sub-Sub-TLV

SRv6 Service Data Sub-Sub-TLV Type 1 is assigned for SRv6 SID structure Sub-Sub-TLV. SRv6 SID Structure Sub-Sub-TLV is used to advertise the lengths of each individual parts of the SRv6 SID as defined in [I-D.ietf-spring-srv6-network-programming]. It is carried as Sub-Sub-TLV in SRv6 SID Information Sub-TLV



- o SRv6 Service Data Sub-Sub-TLV Type (1 octet): This field is set to 1 to represent SRv6 SID Structure Sub-Sub-TLV.

- o SRv6 Service Data Sub-Sub-TLV Length (2 octets): This field contains the total length of 6 bytes.
- o Locator Block Length(1 octet): Contains length of SRv6 SID locator Block in bits.
- o Locator Node Length(1 octet): Contains length of SRv6 SID locator Node in bits.
- o Function Length(1 octet): Contains length of SRv6 SID Function in bits.
- o Arguments Length(1 octet): Contains length of SRv6 SID arguments in bits.
- o Transposition Length(1 octet): Size in bits for the part of SID that has been transposed (or shifted) into a label field
- o Transposition Offset(1 octet): The offset position in bits for the part of SID that has been transposed (or shifted) into a label field.

Section 5 describes mechanisms for signaling of the SRv6 Service SID by transposing a variable part of the SRv6 SID value (function and/or the argument parts) and carrying them in existing label fields to achieve more efficient packing of those service prefix NLRIs in BGP update messages. The SRv6 SID Structure Sub-Sub-TLV MUST be included with the appropriate length fields when the SRv6 Service SID is signaled in split parts to enable the receiver to put together the SID accurately.

Transposition Offset indicates the bit position and Transposition Length indicates the number of bits that are being taken out of the SRv6 SID value and put into high order bits of label field. The bits that have been shifted out MUST be set to 0 in the SID value.

Transposition Length of 0 indicates nothing is transposed and that the entire SRv6 SID value is encoded in the SID Information sub-TLV. In this case, the Transposition Offset MUST be set to 0.

Since size of label field is 24 bits, only that many bits can be transposed from the SRv6 SID value into it.

The SRv6 SID Structure Sub-Sub-TLV is optional and MAY be included when the entire SRv6 Service SID value is encoded in the SID Information Sub-TLV.

Arguments MAY be generally applicable for SIDs of only specific behaviors (e.g. End.DT2M) and therefore the argument length MUST be set to 0 for SIDs where the argument is not applicable.

3. BGP based L3 service over SRv6

BGP egress nodes (egress PEs) advertise a set of reachable prefixes. Standard BGP update propagation schemes[RFC4271], which may make use of route reflectors [RFC4456], are used to propagate these prefixes. BGP ingress nodes (ingress PEs) receive these advertisements and may add the prefix to the RIB in an appropriate VRF.

Egress PEs which supports SRv6 based L3 services advertises overlay service prefixes along with a Service SID enclosed in a SRv6 L3 Service TLV within the BGP Prefix-SID Attribute. This TLV serves two purposes - first, it indicates that the egress PE is reachable via an SRv6 underlay and the BGP ingress PE receiving this route MUST choose to perform IPv6 encapsulation and optionally insert an SRH when required; second, it indicates the value of the Service SID to be used in the encapsulation.

The Service SID thus signaled only has local significance at the egress PE, where it may be allocated or configured on a per-CE or per-VRF basis. In practice, the SID may encode a cross-connect to a specific Address Family table (END.DT) or next-hop/interface (END.DX) as defined in [I-D.ietf-spring-srv6-network-programming].

The SRv6 Service SID SHOULD be routable within the AS of the egress PE and serves the dual purpose of providing reachability between ingress PE and egress PE while also encoding the endpoint behavior.

At an ingress PE, BGP installs the received prefix in the correct RIB table, recursing via an SR Policy leveraging the received SRv6 Service SID.

Assuming best-effort connectivity to the egress PE, the SR policy has a path with a SID list made up of a single SID - the SRv6 Service SID received with the related BGP route update.

However, when the received route is colored with an extended color community 'C' and Next-Hop 'N', and the ingress PE has a valid SRv6 Policy (C, N) associated with SID list <S1,S2, S3> [I-D.filsfils-spring-segment-routing-policy], then the effective SR Policy is <S1, S2, S3, SRv6-Service-SID>.

Multiple VPN routes MAY resolve recursively via the same SR Policy.

3.1. IPv4 VPN Over SRv6 Core

IPv4 VPN Over IPv6 Core is defined in [RFC5549]. The MP_REACH_NLRI is encoded as follows for an SRv6 Core:

- o AFI = 1
- o SAFI = 128
- o Length of Next Hop Network Address = 16 (or 32)
- o Network Address of Next Hop = IPv6 address of the egress PE
- o NLRI = IPv4-VPN routes
- o Label = It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.

SRv6 Service SID is encoded as part of the SRv6 L3 Service TLV. The behavior of the SRv6 SID is entirely up to the originator of the advertisement. In practice, the behavior SHOULD be End.DX4 or End.DT4.

3.2. IPv6 VPN Over SRv6 Core

IPv6 VPN over IPv6 Core is defined in [RFC4659]. The MP_REACH_NLRI is encoded as follows for an SRv6 Core:

- o AFI = 2
- o SAFI = 128
- o Length of Next Hop Network Address = 24 (or 48)
- o Network Address of Next Hop = 8 octets of RD set to 0 followed by IPv6 address of the egress PE
- o NLRI = IPv6-VPN routes
- o Label = It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.

SRv6 Service SID is encoded as part of the SRv6 L3 Service TLV. The behavior of the SRv6 SID is entirely up to the originator of the advertisement. In practice, the behavior SHOULD be End.DX6 or End.DT6.

3.3. Global IPv4 over SRv6 Core

IPv4 over IPv6 Core is defined in [RFC5549]. The MP_REACH_NLRI is encoded with:

- o AFI = 1
- o SAFI = 1
- o Length of Next Hop Network Address = 16 (or 32)
- o Network Address of Next Hop = IPv6 address of Next Hop
- o NLRI = IPv4 routes

SRv6 Service SID is encoded as part of the SRv6 L3 Service TLV. The behavior of the SRv6 SID is entirely up to the originator of the advertisement. In practice, the behavior SHOULD be End.DX4 or End.DT4.

3.4. Global IPv6 over SRv6 Core

The MP_REACH_NLRI is encoded with:

- o AFI = 2
- o SAFI = 1
- o Length of Next Hop Network Address = 16 (or 32)
- o Network Address of Next Hop = IPv6 address of Next Hop
- o NLRI = IPv6 routes

SRv6 Service SID is encoded as part of the SRv6 L3 Service TLV. The behavior of the SRv6 SID is entirely up to the originator of the advertisement. In practice, the behavior SHOULD be End.DX4 or End.DT6.

Also, by utilizing the SRv6 L3 Service TLV to encode the Global SID, a BGP free core is possible by encapsulating all BGP traffic from edge to edge over SRv6 dataplane.

4. BGP based Ethernet VPN (EVPN) over SRv6

Ethernet VPN(EVPN), as defined in [RFC7432] provides an extendable method of building an EVPN overlay. It primarily focuses on MPLS based EVPNs but calls out the extensibility to IP based EVPN overlays. [RFC7432] defines 4 Route Types which carry prefixes and MPLS Label fields; the Label fields have specific use for MPLS encapsulation of EVPN traffic. Route Type 5 carrying MPLS label information (and thus encapsulation information) for EVPN is defined in [I-D.ietf-bess-evpn-prefix-advertisement]. Route Types 6, 7 and 8 are defined in [I-D.ietf-bess-evpn-igmp-mld-proxy].

- o Ethernet Auto-discovery Route (Route Type 1)
- o MAC/IP Advertisement Route (Route Type 2)
- o Inclusive Multicast Ethernet Tag Route (Route Type 3)
- o Ethernet Segment route (Route Type 4)
- o IP prefix route (Route Type 5)
- o Selective Multicast Ethernet Tag route (Route Type 6)
- o IGMP join sync route (Route Type 7)
- o IGMP leave sync route (Route Type 8)

To support SRv6 based EVPN overlays, one or more SRv6 Service SIDs are advertised with Route Type 1,2,3 and 5. The SRv6 Service SID(s) per Route Type are advertised in SRv6 L3/L2 Service TLVs within the BGP Prefix-SID Attribute. Signaling of SRv6 Service SID(s) serves two purposes - first, it indicates that the BGP egress device is reachable via an SRv6 underlay and the BGP ingress device receiving this route MUST choose to perform IPv6 encapsulation and optionally insert an SRH when required; second, it indicates the value of the Service SID(s) to be used in the encapsulation.

4.1. Ethernet Auto-discovery route over SRv6 Core

Ethernet Auto-Discovery (A-D) routes are Route Type 1 defined in [RFC7432] and may be used to achieve split horizon filtering, fast convergence and aliasing. EVPN Route Type 1 is also used in EVPN-VPWS as well as in EVPN flexible cross-connect; mainly used to advertise point-to-point services ID.

Multi-homed PEs MAY advertise an Ethernet Auto-Discovery route per Ethernet segment along with the ESI Label extended community defined

in [RFC7432]. PEs may identify other PEs connected to the same Ethernet segment after the EVPN Route Type 4 ES route exchange. All the multi-homed and remote PEs that are part of same EVI may import the Auto-Discovery route.

EVPN Route Type 1 is encoded as follows for SRv6 Core:

```

+-----+
| RD (8 octets) |
+-----+
| Ethernet Segment Identifier (10 octets) |
+-----+
| Ethernet Tag ID (4 octets) |
+-----+
| MPLS label (3 octets) |
+-----+

```

4.1.1. Per-ES A-D route

- o BGP next-hop: IPv6 address of an egress PE
- o Ethernet Tag ID: set to MAX-ET per [RFC7432] section 8.2.1
- o MPLS Label: always set to zero per [RFC7432] section 8.2.1
- o ESI label extended community ESI label field: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Argument is encoded in the SID value (refer Section 5 for details). Otherwise it carries the Argument part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.

A Service SID enclosed in a SRv6 L2 Service TLV within the BGP Prefix-SID attribute is advertised along with the A-D route. The behavior of the Service SID thus signaled is entirely up to the originator of the advertisement. The Service SID is used to signal Arg.FE2 SID argument for applicable End.DT2M SIDs.

4.1.2. Per-EVI A-D route

- o BGP next-hop: IPv6 address of an egress PE
- o Ethernet Tag ID: non-zero for VLAN-aware bundling service, EVPN VPWS and FXC
- o MPLS Label: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for

details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.

A Service SID enclosed in a SRv6 L2 Service TLV within the BGP Prefix-SID attribute is advertised along with the A-D route. The behavior of the Service SID thus signaled is entirely up to the originator of the advertisement. In practice, the behavior would SHOULD be END.DX2, END.DX2V or END.DT2U.

4.2. MAC/IP Advertisement route over SRv6 Core

EVPN Route Type 2 is used to advertise unicast traffic MAC+IP address reachability through MP-BGP to all other PEs in a given EVPN instance.

EVPN Route Type 2 is encoded as follows for SRv6 Core:

RD (8 octets)
Ethernet Segment Identifier (10 octets)
Ethernet Tag ID (4 octets)
MAC Address Length (1 octet)
MAC Address (6 octets)
IP Address Length (1 octet)
IP Address (0, 4, or 16 octets)
MPLS Label1 (3 octets)
MPLS Label2 (0 or 3 octets)

- o BGP next-hop: IPv6 address of an egress PE
- o MPLS Label1: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.
- o MPLS Label2: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for details).

details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.

Service SIDs enclosed in SRv6 L2 Service TLV and optionally in SRv6 L3 Service TLV within the BGP SID attribute is advertised along with the MAC/IP Advertisement route.

Described below are different types of Route Type 2 advertisements.

- o MAC/IP Advertisement route with MAC Only
 - * BGP next-hop: IPv6 address of egress PE
 - * MPLS Label1: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.
- o A Service SID enclosed in a SRv6 L2 Service TLV within the BGP Prefix-SID attribute is advertised along with the route. The behavior of the Service SID thus signaled is entirely up to the originator of the advertisement. In practice, the behavior SHOULD be END.DX2 or END.DT2U.
- o MAC/IP Advertisement route with MAC+IP
 - * BGP next-hop: IPv6 address of egress PE
 - * MPLS Label1: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.
 - * MPLS Label2: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.
- o An L2 Service SID enclosed in a SRv6 L2 Service TLV within the BGP Prefix-SID attribute is advertised along with the route. In addition, an L3 Service SID enclosed in a SRv6 L3 Service TLV within the BGP SID attribute MAY also be advertised along with the route. The behavior of the Service SID(s) thus signaled is entirely up to the originator of the advertisement. In practice,

the behavior SHOULD be END.DX2 or END.DT2U for the L2 Service SID, and END.DT6/4 or END.DX6/4 for the L3 Service SID.

4.3. Inclusive Multicast Ethernet Tag Route over SRv6 Core

EVPN Route Type 3 is used to advertise multicast traffic reachability information through MP-BGP to all other PEs in a given EVPN instance.

EVPN Route Type 3 is encoded as follows for SRv6 core:

RD (8 octets)
Ethernet Tag ID (4 octets)
IP Address Length (1 octet)
Originating Router's IP Address (4 or 16 octets)

- o BGP next-hop: IPv6 address of egress PE

PMSI Tunnel Attribute [RFC6514] MAY contain MPLS Implicit NULL label and Tunnel Type would be similar to that defined in EVPN Route Type 6 i.e. Ingress replication route.

The format of PMSI Tunnel Attribute is encoded as follows for SRv6 Core:

Flag (1 octet)
Tunnel Type (1 octet)
MPLS label (3 octet)
Tunnel Identifier (variable)

- o Flag: zero value defined per [RFC7432]
- o Tunnel Type: defined per [RFC6514]
- o MPLS label: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for

details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.

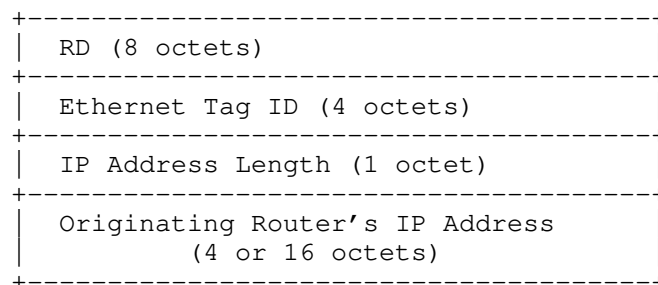
- o Tunnel Identifier: IP address of egress PE

A Service SID enclosed in a SRv6 L2 Service TLV within the BGP Prefix-SID attribute is advertised along with the route. The behavior of the Service SID thus signaled, is entirely up to the originator of the advertisement. In practice, the behavior of the SRv6 SID is as follows:

- o END.DX2 or END.DT2M behavior
- o The ESI Filtering argument (Arg.FE2) of the Service SID carried along with EVPN Route Type 1 route SHOULD be merged together with the applicable End.DT2M SID of Type 3 route advertised by remote PE by doing a bitwise logical-OR operation to create a single SID on the ingress PE for Split-horizon and other filtering mechanisms. Details of filtering mechanisms are described in [RFC7432].

4.4. Ethernet Segment route over SRv6 Core

An Ethernet Segment route i.e. EVPN Route Type 4 is encoded as follows for SRv6 core:



- o BGP next-hop: IPv6 address of egress PE

SRv6 Service TLVs within BGP SID attribute are not advertised along with this route. The processing of the route has not changed - it remains as described in [RFC7432].

4.5. IP prefix route over SRv6 Core

EVPN Route Type 5 is used to advertise IP address reachability through MP-BGP to all other PEs in a given EVPN instance. IP address may include host IP prefix or any specific subnet.

EVPN Route Type 5 is encoded as follows for SRv6 core:

RD (8 octets)
Ethernet Segment Identifier (10 octets)
Ethernet Tag ID (4 octets)
IP Prefix Length (1 octet)
IP Prefix (4 or 16 octets)
GW IP Address (4 or 16 octets)
MPLS Label (3 octets)

- o BGP next-hop: IPv6 address of egress PE
- o MPLS Label: It is set to Implicit NULL when the SID Structure Sub-Sub-TLV is not present or when it is present and indicates that the Function is encoded in the SID value (refer Section 5 for details). Otherwise it carries the Function part of SRv6 SID when indicated as such by the SID Structure Sub-Sub-TLV.

SRv6 Service SID is encoded as part of the SRv6 L3 Service TLV. The function of the SRv6 SID is entirely up to the originator of the advertisement. In practice, the behavior may SHOULD be End.DT4/6 or End.DX4/6.

4.6. EVPN multicast routes (Route Types 6, 7, 8) over SRv6 core

These routes do not require the advertisement of SRv6 Service TLVs along with them. Similar to EVPN Route Type 4, the BGP Nexthop is equal to the IPv6 address of egress PE. More details may be added in future revisions of this document.

5. Encoding SRv6 SID information

The SRv6 Service SID(s) for a BGP Service Prefix are carried in the SRv6 Services TLVs of the BGP Prefix-SID Attribute.

For certain types of BGP Services like L3VPN where a per-VRF SID allocation is used (i.e. End.DT4 or End.DT6 behaviors), the same SID is shared across multiple NLRIs thus providing efficient packing. However, for certain other types of BGP Services like EVPN VPWS where a per-PW SID allocation is required (i.e. End.DX2 behavior), each

NLRI would have its own unique SID there by resulting in inefficient packing.

To achieve efficient packing, this document allows flexibility in the advertisement of the SRv6 Service SID either as a whole in the SRv6 Services TLVs or the encoding of only the common parts of the SRv6 SID (e.g. Locator parts) in the SRv6 Services TLVs and encoding the variable (e.g. Function and Argument parts) in the existing label fields specific to that service encoding. The SRv6 SID Structure Sub-Sub-TLV describes the sizes of the parts of the SRv6 SID. It also indicate offset of variable part and its length in SRv6 SID value.

As an example, for the EVPN VPWS service prefix described in section 4.1.2, the function part of the SRv6 SID is encoded in the MPLS Label field of the NLRI and the SID value in the SRv6 Services TLV carries only the locator parts with the SRv6 SID Structure Sub-Sub-TLV included. The SRv6 SID Structure sub-sub-TLV defines the lengths of locator block, locator node and function parts (arguments are not applicable for the End.DX2 behavior). Transposition Offset indicates the bit position and Transposition Length indicates the number of bits that are being taken out of the SID and put into label field.

In yet another example, for the EVPN Per-ES A-D route described in section 4.1.1, only the argument of the SID needs to be signaled. This argument part of the SRv6 SID MAY be Transposed in the ESI Label field of the ESI Label Extended Community and the SID value in the SRv6 Services TLV is set to 0 with the SRv6 SID Structure Sub-Sub-TLV included. The SRv6 SID Structure sub-sub-TLV defines the lengths of locator block, locator node, function and argument parts. The offset and length of argument part SID value moved to label field is set in Transposition offset and length of SID structure TLV. The receiving router is then able to put together the entire SRv6 Service SID (e.g. for the End.DT2M behavior) placing the label value received in the ESI Label field of the Per-ES A-D route into the correct transposition offset and length in the SRv6 SID with the End.DT2M behavior received for a EVPN Route Type 3 value.

6. Implementation Status

The [I-D.matsushima-spring-srv6-deployment-status] describes the current deployment and implementation status of SRv6 which also includes the BGP services over SRv6 as specified in this document.

7. Error Handling

In case of any errors encountered while processing SRv6 Service TLVs, the details of the error SHOULD be logged for further analysis.

If multiple instances of SRv6 L3 Service TLV is encountered, all but the first instance MUST be ignored.

If multiple instances of SRv6 L2 Service TLV is encountered, all but the first instance MUST be ignored.

An SRv6 Service TLV is considered malformed in the following cases:

- o the TLV Length is less than 1
- o the TLV Length is inconsistent with the length of BGP SID attribute
- o atleast one of the constituent Sub-TLVs is malformed

An SRv6 Service Sub-TLV is considered malformed in the following cases:

- o the Sub-TLV Length is inconsistent with the length of the enclosing SRv6 Service TLV

An SRv6 SID Information Sub-TLV is considered malformed in the following cases:

- * the Sub-TLV Length is less than 21
- * the Sub-TLV Length is inconsistent with the length of the enclosing SRv6 Service TLV
- * atleast one of the constituent Sub-Sub-TLVs is malformed

An SRv6 Service Data Sub-sub-TLV is considered malformed in the following cases:

- o the Sub-Sub-TLV Length is inconsistent with the length of the enclosing SRv6 service Sub-TLV

Any TLV or Sub-TLV or Sub-Sub-TLV is not considered malformed because its Type is unrecognized.

Any TLV or Sub-TLV or Sub-Sub-TLV is not considered malformed because of failing any semantic validation of its Value field.

The BGP Prefix-SID attribute is considered malformed if it contains atleast one constituent SRv6 Service TLV that is malformed. In such cases, the attribute MUST be discarded [RFC7606] and not propagated further. Note that if a path whose BGP Prefix-SID attribute is discarded in this manner is selected as the best path to be installed in the RIB, traffic forwarding for the corresponding prefix may be affected. Implementations MAY choose to make such paths less preferable or even ineligible during the selection of best path for the corresponding prefix.

SRv6 SID value in SRv6 Service Sub-TLV is invalid when SID Structure Sub-Sub-TLV is present and transposition length is greater than 24. Path pointing to such Prefix-SID Attribute should be ineligible during the selection of best path for the corresponding prefix.

A BGP speaker receiving a path containing BGP Prefix-SID Attribute with one or more SRv6 Service TLVs observes the following rules when advertising the received path to other peers:

- o if the nexthop is unchanged during advertisement, the SRv6 Service TLVs, including any unrecognized Types of Sub-TLV and Sub-Sub-TLV, SHOULD be propagated further. In addition, all Reserved fields in the TLV or Sub-TLV or Sub-Sub-TLV MUST be propagated unchanged.
- o if the nexthop is changed during advertisement, any unrecognized Sub-TLVs and Sub-Sub-TLVs MUST NOT be propagated.
- o if the nexthop is changed during advertisement, the TLVs, Sub-TLVs and Sub-Sub-TLVs SHOULD be re-originated if appropriate, and not merely propagated unchanged. The interpretation of the meaning of re-origination versus propagation is a matter of local implementation.

8. IANA Considerations

8.1. BGP Prefix-SID TLV Types registry

This document defines two new TLV Types of the BGP Prefix-SID attribute. IANA is requested to assign Type values in the registry "BGP Prefix-SID TLV Types" as follows:

Value	Type	Reference

[TBD1]	SRv6 L3 Service TLV	<this document>
[TBD2]	SRv6 L2 Service TLV	<this document>

IANA is also requested to reserve the following Type value. This was used in some implementations of previous versions of this draft.

Value	Type	Reference

4	Reserved	<this document>

8.2. SRv6 Service Sub-TLV Types registry

IANA is requested to create and maintain a new registry called "SRv6 Service Sub-TLV Types". The allocation policy for this registry is:

0 : Reserved
1-127 : IETF Review
128-254 : First Come First Served
255 : Reserved

The following Sub-TLV Types are defined in this document:

Value	Type	Reference

1	SRv6 SID Information Sub-TLV	<this document>

8.3. SRv6 Service Data Sub-Sub-TLV Types registry

IANA is requested to create and maintain a new registry called "SRv6 Service Data Sub-Sub-TLV Types". The allocation policy for this registry is:

0 : Reserved
1-127 : IETF Review
128-254 : First Come First Served
255 : Reserved

The following Sub-Sub-TLV Types are defined in this document:

Value	Type	Reference

1	SRv6 SID Structure Sub-Sub-TLV	<this document>

9. Security Considerations

This document introduces no new security considerations beyond those already specified in [RFC4271] and [RFC8277].

10. Conclusions

This document proposes extensions to the BGP to allow advertising certain attributes and functionalities related to SRv6.

11. References

11.1. Normative References

- [I-D.filsfils-spring-segment-routing-policy]
Filsfils, C., Sivabalan, S., Hegde, S.,
daniel.voyer@bell.ca, d., Lin, S., bogdanov@google.com,
b., Krol, P., Horneffer, M., Steinberg, D., Decraene, B.,
Litkowski, S., Mattes, P., Ali, Z., Talaulikar, K., Liste,
J., Clad, F., and K. Raza, "Segment Routing Policy
Architecture", draft-filsfils-spring-segment-routing-
policy-06 (work in progress), May 2018.
- [I-D.ietf-6man-segment-routing-header]
Filsfils, C., Dukes, D., Previdi, S., Leddy, J.,
Matsushima, S., and d. daniel.voyer@bell.ca, "IPv6 Segment
Routing Header (SRH)", draft-ietf-6man-segment-routing-
header-21 (work in progress), June 2019.
- [I-D.ietf-spring-srv6-network-programming]
Filsfils, C., Camarillo, P., Leddy, J.,
daniel.voyer@bell.ca, d., Matsushima, S., and Z. Li, "SRv6
Network Programming", draft-ietf-spring-srv6-network-
programming-01 (work in progress), July 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119,
DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4456] Bates, T., Chen, E., and R. Chandra, "BGP Route
Reflection: An Alternative to Full Mesh Internal BGP
(IBGP)", RFC 4456, DOI 10.17487/RFC4456, April 2006,
<<https://www.rfc-editor.org/info/rfc4456>>.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP
Encodings and Procedures for Multicast in MPLS/BGP IP
VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012,
<<https://www.rfc-editor.org/info/rfc6514>>.

- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.
- [RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <<https://www.rfc-editor.org/info/rfc7606>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017, <<https://www.rfc-editor.org/info/rfc8277>>.

11.2. Informative References

- [I-D.ietf-bess-evpn-igmp-mld-proxy]
Sajassi, A., Thoria, S., Patel, K., Yeung, D., Drake, J., and W. Lin, "IGMP and MLD Proxy for EVPN", draft-ietf-bess-evpn-igmp-mld-proxy-03 (work in progress), June 2019.
- [I-D.ietf-bess-evpn-prefix-advertisement]
Rabadan, J., Henderickx, W., Drake, J., Lin, W., and A. Sajassi, "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-11 (work in progress), May 2018.
- [I-D.ietf-idr-bgp-prefix-sid]
Previdi, S., Filsfils, C., Lindem, A., Sreekantiah, A., and H. Gredler, "Segment Routing Prefix SID extensions for BGP", draft-ietf-idr-bgp-prefix-sid-27 (work in progress), June 2018.
- [I-D.ietf-idr-segment-routing-te-policy]
Previdi, S., Filsfils, C., Mattes, P., Rosen, E., Jain, D., and S. Lin, "Advertising Segment Routing Policies in BGP", draft-ietf-idr-segment-routing-te-policy-07 (work in progress), July 2019.

- [I-D.ietf-isis-segment-routing-extensions]
Previdi, S., Ginsberg, L., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", draft-ietf-isis-segment-routing-extensions-25 (work in progress), May 2019.
- [I-D.matsushima-spring-srv6-deployment-status]
Matsushima, S., Filsfils, C., Ali, Z., and Z. Li, "SRv6 Implementation and Deployment Status", draft-matsushima-spring-srv6-deployment-status-01 (work in progress), May 2019.
- [RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", RFC 3032, DOI 10.17487/RFC3032, January 2001, <<https://www.rfc-editor.org/info/rfc3032>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.
- [RFC4659] De Clercq, J., Ooms, D., Carugi, M., and F. Le Faucheur, "BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN", RFC 4659, DOI 10.17487/RFC4659, September 2006, <<https://www.rfc-editor.org/info/rfc4659>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, DOI 10.17487/RFC5549, May 2009, <<https://www.rfc-editor.org/info/rfc5549>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.

Appendix A. Contributors

Ali Sajassi
Cisco

Email: sajassi@cisco.com

Bart Peirens
Proximus
Belgium

Email: bart.peirens@proximus.com

Darren Dukes
Cisco

Email: ddukes@cisco.com

Pablo Camarilo
Cisco

Email: pcamaril@cisco.com

Shyam Sethuram
Cisco

Email: shsethur@cisco.com

Zafar Ali
Cisco

Email: zali@cisco.com

Ketan Talaulikar
Cisco

Email: ketant@cisco.com

Authors' Addresses

Gaurav Dawra (editor)
LinkedIn
USA

Email: gdawra.ietf@gmail.com

Clarence Filsfils
Cisco Systems
Belgium

Email: cfilsfil@cisco.com

Patrice Brissette
Cisco Systems
Canada

Email: pbrisset@cisco.com

Swadesh Agrawal
Cisco Systems
USA

Email: swaagraw@cisco.com

Jonh Leddy
Comcast
USA

Daniel Voyer
Bell Canada
Canada

Email: daniel.voyer@bell.ca

Daniel Bernier
Bell Canada
Canada

Email: daniel.bernier@bell.ca

Dirk Steinberg
Steinberg Consulting
Germany

Email: dws@steinberg.net

Robert Raszuk
Bloomberg LP
USA

Email: robert@raszuk.net

Bruno Decraene
Orange
France

Email: bruno.decraene@orange.com

Satoru Matsushima
SoftBank
1-9-1, Higashi-Shimbashi, Minato-Ku
Japan 105-7322

Email: satoru.matsushima@g.softbank.co.jp

Shunwan Zhuang
Huawei Technologies
China

Email: zhuangshunwan@huawei.com

Jorge Rabadan
Nokia
USA

Email: jorge.rabadan@nokia.com

Network Working Group
Internet Draft
Intended status: Informational
Expires: Dec 2019

L. Dunbar
J. Guichard
Huawei
Ali Sajassi
Cisco
J. Drake
Juniper
Ayan Barnerjee
D. Carrel
Cisco

July 8, 2019

BGP Usage for SDWAN Overlay Networks
draft-dunbar-bess-bgp-sdwan-usage-01

Abstract

The document describes three distinct SDWAN scenarios and discusses the applicability of BGP for each of those scenarios. The goal of the document is to make it easier for future SDWAN control plane protocols discussion.

SDWAN edge nodes are commonly interconnected by multiple underlay networks that are owned and managed by different network providers. A BGP-based control plane is chosen for handling large number of SDWAN edge nodes with little manual intervention.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79. This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that

other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on January 8, 2009.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	3
2. Conventions used in this document.....	4
3. Use Case Scenario Description and Requirements.....	5
3.1. Requirements.....	6
3.1.1. Client Service Requirement.....	6
3.1.2. SDWAN Node Provisioning.....	6
3.2. Scenarios #1: Homogeneous WAN.....	8
3.3. Scenario #2: SDWAN WAN ports to VPN's PEs and to Internet.....	9

3.4. Scenario #3: SDWAN WAN ports to MPLS VPN and the Internet	12
4. Provisioning Model.....	14
4.1. Client Service Provisioning Model.....	14
4.2. WAN Ports Provisioning Model.....	14
4.2.1. Why BGP as Control Plane for SDWAN WAN Ports Registration?.....	15
5. SDWAN Traffic Forwarding Walk Through.....	16
5.1. SDWAN Network Startup Procedures.....	16
5.2. Packet Walk-Through for Scenario #1.....	16
5.3. Packet Walk-Through for Scenario #2.....	17
5.3.1. SDWAN node WAN Ports Properties Registration.....	19
5.3.2. Controller Facilitated IPsec SA & NAT management....	19
5.3.3. BGP Based SDWAN client routes.....	21
5.4. Packet Walk-Through for Scenario #3.....	22
6. Manageability Considerations.....	23
7. Security Considerations.....	23
8. IANA Considerations.....	23
9. References.....	23
9.1. Normative References.....	23
9.2. Informative References.....	24
10. Acknowledgments.....	25

1. Introduction

An "SDWAN" network consists of many segments of parallel paths over different underlay networks, some of which are private networks over which traffic can traverse without encryption, others require encryption over untrusted public networks.

[Net2Cloud-Problem] describes the network related problems that enterprises face today in transitioning their IT infrastructure to support a digital economy, such as the need to connect enterprises' branch offices to dynamic workloads in different Cloud DCs, or aggregating multiple paths provided by different service providers to achieve better performance.

Even though SDWAN has been positioned as a flexible way to reach dynamic workloads in third party data centers over multiple underlay networks, scaling becomes a major issue when there are hundreds or thousands of nodes to be interconnected by the SDWAN overlay paths.

BGP is widely used by underlay networks. This document describes using BGP to enhance the scaling properties of SDWAN overlay networks.

2. Conventions used in this document

Cloud DC: Third party data centers that host applications and workloads owned by different organizations or tenants.

Controller: Used interchangeably with SDWAN controller to manage SDWAN overlay path creation/deletion and monitor the path conditions between sites.

CPE: Customer Premise Equipment

CPE-Based VPN: Virtual Private Secure network formed among CPEs. This is to differentiate from more commonly used PE-based VPNs [RFC 4364].

Homogeneous SDWAN: A type of SDWAN network in which all traffic to/from the SDWAN edge nodes has to be encrypted regardless of underlay networks. For lack of better terminology, we call this Homogeneous SDWAN throughout this document.

ISP: Internet Service Provider

NSP: Network Service Provider. NSP usually provides more advanced network services, such as MPLS VPN, private leased lines, or managed Secure WAN connections, many times within a private trusted domain, whereas an ISP usually provides plain internet services over public untrusted domains.

PE: Provider Edge

SDWAN End-point: a port (logical or physical) of a SDWAN edge node.

SDWAN: Software Defined Wide Area Network. In this document, "SDWAN" refers to the solutions of pooling WAN bandwidth from multiple underlay networks to get better WAN bandwidth management, visibility & control. When the underlay networks are private, traffic can traverse without additional encryption; when the underlay networks are public, such as the Internet, some traffic may need to be encrypted when traversing through (depending on user provided policies).

SDWAN IPsec SA: IPsec Security Association between two SDWAN ports or nodes.

SDWAN over Hybrid Networks: SDWAN over Hybrid Networks typically have edge nodes utilizing bandwidth resources from multiple service providers. In Hybrid SDWAN network, packets over private networks can go natively without encryption and are encrypted over the untrusted network, such as the public Internet.

WAN Port: A Port or Interface facing an ISP or Network Service Provider (NSP), with address (usually public routable address) allocated by the ISP or the NSP.

C-PE: SDWAN Edge node, which can be CPE for customer managed SDWAN, or PE that is for provider managed SDWAN services).

ZTP: Zero Touch Provisioning

3. Use Case Scenario Description and Requirements

SDWAN networks can have different topologies and have different traffic patterns. To make it easier for the focused discussion in subsequent drafts on SDWAN control plane and data plane, this section describes several SDWAN scenarios that may have different need or impact to their corresponding control planes & data planes.

3.1. Requirements

3.1.1. Client Service Requirement

Client interface of SDWAN nodes can be IP or Ethernet based.

For Ethernet based client interfaces, SDWAN edge should support VLAN-based service interfaces (EVI100), VLAN bundle service interfaces (EVI200), or VLAN-Aware bundling service interfaces. EVPN service requirements are applicable to the Client traffic, as described in the Section 3.1 of RFC8388.

For IP based client interfaces, L3VPN service requirements are applicable.

3.1.2. SDWAN Node Provisioning

Unlike traditional EVPN or L3VPN where PEs are deployed for long term, SDWAN edge nodes (virtual or physical) deployment at a specific location can be ephemeral. Therefore, Zero Touch Provisioning (ZTP) is a common requirement for SDWAN. ZTP for SDWAN can include many areas, but from network connectivity perspective, ZTP should include the following:

- Upon power up, an SDWAN node can reach a central SDWAN Controller (which can be burned or preconfigured in the device) via a TLS or SSL secure channel.

- The Central SDWAN Controller can designate a Local Network Controller in the proximity of the SDWAN node; the Local Network Controller and the SDWAN nodes might be connected by third party untrusted network. The Local controller does all the following 4 tasks:

- 1) ZTP
- 2) Auto-discovery of Network
- 3) (Auto)-Provisioning for IPsec SAs (initial provisioning part)
- 4) Signaling of tenant's routes/info

BGP is well suited for (4), using Route Reflector (RR) [RFC4456] to propagate network information among SDWAN edge nodes. The SDWAN

node can establish a secure connection (TLS, SSL, etc) to the Local Network Controller (RR).

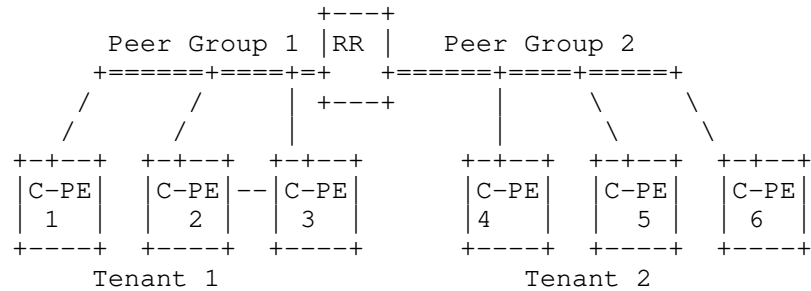


Figure 1: Peer Groups managed by Local Controller

The SDWAN nodes (a.k.a. C-PEs throughout this document) belonging to the same Tenant can be far apart and can be connected by third party untrusted networks. Therefore, it is not appropriate for a SDWAN edge node (C-PE) to advertise its SDWAN Port properties to its neighbors. Each C-PE propagates its SDWAN Port attributes via the secure channel (TLS, SSL, etc.) established with the Local Controller.

C-PE-1 should include the following aspects in addition to managing client routes:

- Register the SDWAN node's WAN port <-> local address mapping to its Local Controller. The Local Controller propagates the information to C-PE2 & C-PE3.
- Exchange IPsec property (capability such as the supported encryption algorithms, etc.) and ports NAT property (e.g. private addresses or dynamically assigned IP addresses) with the Local Controller.
- C-PE2 and C-PE3 can establish IPsec SA with the C-PE1 after receiving the information from the Local Controller.
- Then distribute the routes attached to the C-PE to its authorized peers.

Tenant separation is achieved by the Local Controller creating different Tenant based Peer Groups.

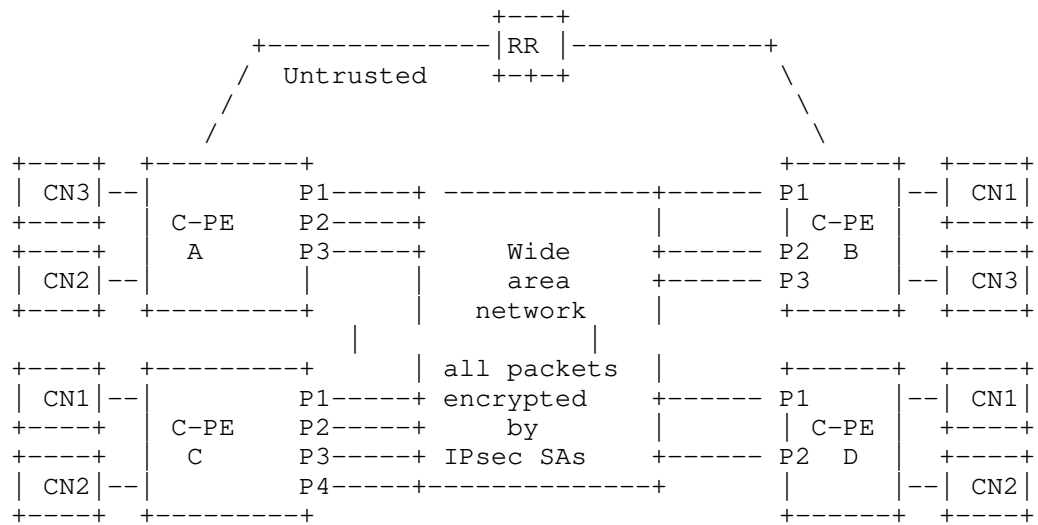
3.2. Scenarios #1: Homogeneous WAN

This is referring to a type of SDWAN network with edge nodes encrypting all traffic over WAN to other edge nodes, regardless of whether the underlay is private or public. For lack of better terminology, we call this Homogeneous SDWAN throughout this document.

Some typical scenarios for the use of a Homogeneous SDWAN network are as follows:

- A small branch office connecting to its HQ offices via the Internet. All sensitive traffic to/from this small branch office has to be encrypted, which is usually achieved using IPsec SAs.
- A store in a shopping mall may need to securely connect to its applications in one or more Cloud DCs via the Internet. A common way of achieving this is to establish IPsec SAs to the Cloud DC gateway to carry the sensitive data to/from the store.

As described in [SECURE-EVPN], the granularity of the IPsec SAs for Homogeneous SDWAN can be per site, per subnet, per tenant, or per address. Once the IPsec SA is established for a specific subnet/tenant/site, all traffic to/from the subnets/tenants/site are encrypted.



CN: Client Networks, which is same as Tenant Networks used by NVo3

Figure 1: Homogeneous SDWAN

One of the key properties of homogeneous SDWAN is that the SDWAN Local Network Controller (RR) is connected to C-PEs via untrusted public network, therefore, requiring secure connection between RR and C-PEs (TLS, DTLS, etc.).

Homogeneous SDWAN has some similarity to commonly deployed IPsec VPN, albeit the IPsec VPN is usually point-to-point among a small number of endpoints and with heavy manual configuration for IPsec between end-points, whereas an SDWAN network can have a large number of end-points with an SDWAN controller to manage requiring zero touch provisioning upon powering up.

Existing Private VPNs (e.g. MPLS based) can use homogeneous SDWAN to extend over public network to remote sites to which the VPN operator does not own or lease infrastructural connectivity, as described in [SECURE-EVPN] and [SECURE-L3VPN]

3.3. Scenario #2: SDWAN WAN ports to VPN's PEs and to Internet

In this scenario, SDWAN edge nodes (a.k.a. C-PEs) have some WAN ports connected to PEs of Private VPNs over which packets can be forwarded natively without encryption, and some WAN ports connected to the Internet over which sensitive traffic have to be encrypted (usually by IPsec SA).

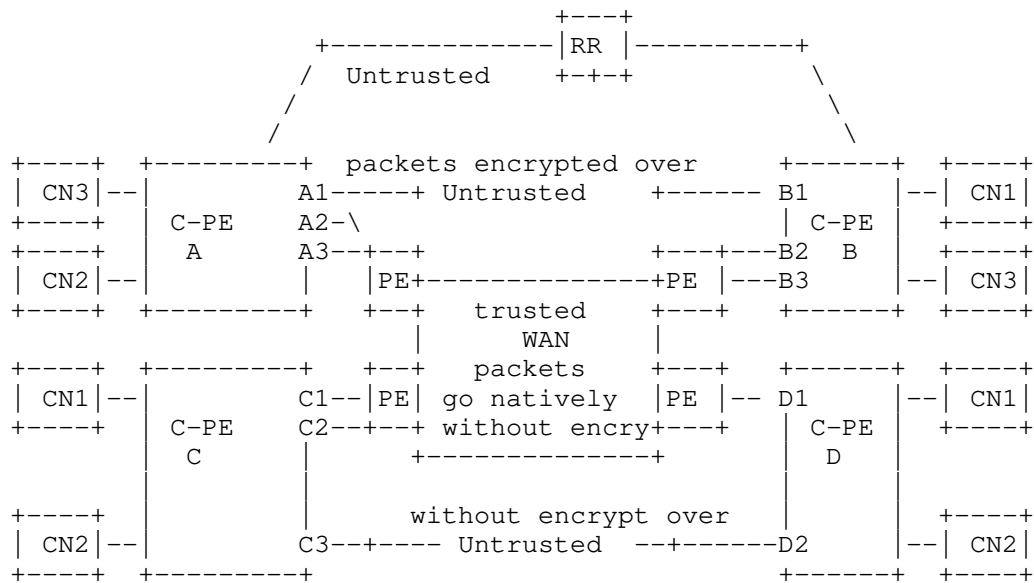
In this scenario, the SDWAN edge nodes' egress WAN ports are all IP/Ethernet based, either egress to PEs of the VPNs or to the Internet. Even if the VPN is a MPLS network, the VPN's PEs have IP/Ethernet connections to the SDWAN edge (C-PEs). Throughout this document, this scenario is also called CPE based SDWAN over Hybrid Networks.

Even though IPsec SA can secure the packets traversing the Internet, it does not offer the premium SLA commonly offered by Private VPNs, especially over long distance. Clients need to have policies to specify criteria for flows only traversing private VPNs or traversing either as long as encrypted when over the Internet. For example, client can have those policies for the flows:

1. A policy or criteria for sending the flows over a private network without encryption (for better performance),
2. A policy or criteria for sending the flows over any networks as long as the packets of the flows are encrypted when traversing untrusted networks, or
3. A policy of not needing encryption at all.

If a flow traversing multiple segments, such as A<->B<->C<->D, has either Policy 2 or 3 above, the flow can traverse different underlays in different segments, such as over Private network underlay between A<->B without encryption, or over the public internet between B<->C in an IPsec SA.

As shown in the figure below, C-PE-1 has two different types of interfaces (A1 to Internet and A2 & A3 to VPN). The C-PEs' loopback addresses and addresses attached to C-PEs may or may not be visible to the ISPs/NSPs. The addresses for the WAN ports can have addresses allocated by the service providers or dynamically assigned (e.g. by DHCP). One WAN port shown in the figure below (e.g. A1, A2, A3 etc.) is a logical representation of potential multiple physical ports on the C-PEs.



CN: Client Network

Figure 2: Hybrid SDWAN

Some key characteristics of a Hybrid SDWAN overlay network are as follows:

- one C-PE may be connected to different ISPs/NSPs, with some of its WAN ports addresses being assigned by the ISPs/NSPs.
- The WAN ports connected to PE of trusted private networks (e.g. MPLS VPN) hand off IP/Ethernet packets, just like today's CPE that do not handle MPLS packets and do not participate in the underlay VPN networks' control plane. Traffic can flow natively without encryption when be forwarded out through those WAN ports for better performance.
- The WAN ports connected to untrusted networks, e.g. the Internet, requires sensitive traffic to be encrypted, i.e. encrypted by IPsec SA.

- An SDWAN local Network Controller (RR) is connected to C-PEs via the untrusted public network, therefore, requiring secure connection between RR and C-PEs via TLS, DTLS, etc.
- The SDWAN nodes' [loopback] addresses might not be routable nor visible in the underlay ISP/NSP networks. Routes & services attached to SDWAN edges at the SDWAN overlay layer are in different address spaces than the underlay networks.
- There could be multiple SDWAN devices sharing a common property, such as a geographic location. Some applications over SDWAN may need to traverse specific geographic locations for various reasons, such as to comply regulatory rules, to utilize specific value added services, or others.
- The underlay path selection between sites can be a local section. Some policies allow one service from CPE1 -> CPE2 -> CPE3 using one ISP/NSP underlay in the first segment (CPE1 -> CPE2), and using a different ISP/NSP in the second segment (CPE2-> CPE3).
- Services may not be congruent, i.e. the packets from A-> B may traverse one underlay network, and the packets from B -> A may traverse a different underlay.
- Different services, routes, or VLANs attached to SDWAN nodes can be aggregated over one underlay path; same service/routes/VLAN can spread over multiple SDWAN underlays at different times depending on the policies specified for the service. For example, one tenant's packets to HQ need to be encrypted when sent over the Internet or have to be sent over private networks, while the same tenant's packets to Facebook can be sent over the Internet without encryption.

3.4. Scenario #3: SDWAN WAN ports to MPLS VPN and the Internet

This scenario refers to existing VPN (e.g. MPLS based VPN, such as EVPN or IPVPN) adding extra ports facing untrusted public networks allowing PEs to offload some low priority traffic to those ports facing public networks when the VPN MPLS paths are congested. Throughout this document, this scenario is also called Internet Offload for Private VPN, or PE based SDWAN.

In this scenario, it is important that the packets offloaded to untrusted public network be encrypted. In this scenario, there is a secure BGP connection between RR & PEs.

PE based SDWAN can be used by VPN service providers to temporarily increase bandwidth between sites when they are not sure if the demand will sustain for long period of time or as a temporary solution before the permanent infrastructure is built or leased.

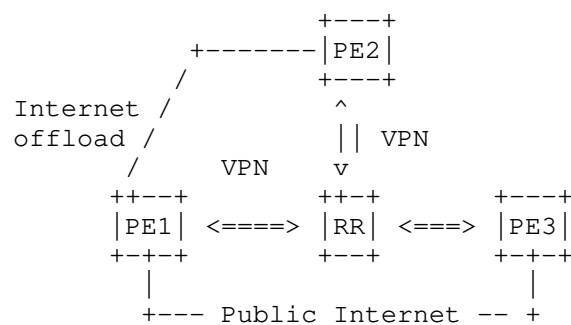


Figure 3: Additional Internet paths added to the VPN

Here are some key properties for PE based SDWAN:

- For MPLS based VPN, PEs continue having MPLS encapsulation handoff to existing paths.
- The BGP RR is connected to PEs in the same way as VPN, i.e. via the trusted network.
- For the added Internet ports, PEs have IP packets handoff, i.e. sending and receiving IP data frames. Internally, PEs can have the option to encapsulate the MPLS payload in IP, as specified by RFC4023.
- The ports facing public internet might get IP addresses assigned by ISPs, which may not be in the same address domain as PEs'.

- Ports facing public internet are not as secure as the ports facing private infrastructure. There could be spoofing, or DDOS attacks to the ports facing public internet. Extra consideration must be given when injecting the new routes from public network into VRFs.
- Even though packets are encrypted over public internet, the performance SLA is not guaranteed over public internet. Therefore, clients may have policies only allowing some flows to be offloaded to internet path.

4. Provisioning Model

4.1. Client Service Provisioning Model

The provisioning tasks described in Section 4 of RFC8388 are the same for the SDWAN client traffic. When client traffic are multi-homed to two (or more) C-PEs, the Non-Service-Specific parameters need to be provisioned per the Section 4.1.1 of RFC8388.

Since most SDWAN nodes are ephemeral and have small number of IP subnets or VLANs attached to the client ports of the SDWAN nodes, it is recommended to have default and simplified Service-specific parameters for each client port, remotely managed by the SDWAN Network Controller (i.e. the RR) via the secure channel (TLS/DTLS) between the controller and the C-PEs.

More details are to be added.

4.2. WAN Ports Provisioning Model

Since the deployment of PEs to MPLS VPN are for relatively long term, the common provisioning procedure for PE's WAN ports is via CLI.

A SDWAN node deployment can be ephemeral and its location can be in remote locations, manual provisioning for its WAN ports is not acceptable. In addition, a SDWAN WAN port's IP address can be dynamically assigned or using private addresses. Therefore, it is necessary to have a separate control protocol; something like NHRP did for ATM, for a SDWAN node to register its WAN property to its controller dynamically.

Unlike a PE to MPLS based VPN where its WAN ports are homogeneously facing MPLS private network and all traffic are egressed in MPLS data frames through its WAN ports, the WAN ports of a SDWAN node can be connected to a PE of VPN, MPLS private network directly, the public Internet, or the various combinations of all.

For Scenario #1 above, the WAN ports can face public internet or VPN.

For Scenario #2 above, WAN ports are either configured as connecting to PEs of VPN where traffic can be sent as IP/Ethernet without encryption, or configured as connecting to public Internet.

For Scenario #3 above, the WAN ports are either configured as VPN egress ports (hand off MPLS data frames), or as connecting to the public internet that requires MPLS in IP in IPsec encapsulation.

4.2.1. Why BGP as Control Plane for SDWAN WAN Ports Registration?

For a small sized SDWAN network, traditional hub & spoke model using NHRP or DSVPN/DMVPN with a hub node (or controller) managing SDWAN node WAN ports mapping (e.g. local & public addresses and tunnel identifiers mapping) can work reasonably well. However, for a large SDWAN network, say more than 100 nodes with different types of topologies, the traditional approach becomes very messy, complex and error prone.

Here are some of the compelling reasons of using BGP instead of extending NHRP/DSVPN/DMVPN. (Same as the reasons quoted by LSVR on why using BGP):

- BGP already widely deployed as sole protocol (see RFC 7938)
- Robust and simple implementation
- Wide acceptance - minimal learning
- Reliable transport
- Guaranteed in-order delivery
- Incremental updates
-
- Incremental updates upon session restart

- No flooding and selective filtering
- RR already has the capability to apply policies to communications among peers.

5. SDWAN Traffic Forwarding Walk Through

BGP based EVPN control plane are still applicable to routes attached to the client ports of SDWAN nodes. Section 5 of RFC8388 describes the BGP EVPN NLRI Usage for various routes of client traffic. The procedures described in the Section 6 of RFC8388 are same for the SDWAN client traffic.

The only additional consideration for SDWAN is to control how traffic egress the SDWAN edge node to various WAN ports.

5.1. SDWAN Network Startup Procedures

A SDWAN network can add or delete SDWAN edge nodes on regular basis depending on user requests.

- For Scenario #1: a SDWAN edge node in a shopping mall or Cloud DC can be added or removed on demand. The Zero Touch Provisioning described in 3.1.2 are required for the node startup.
- For Scenario #2: this can be Data Centers or enterprises upgrading their CPEs to add extra bandwidth via public internet in addition to VPN services that they already purchased. Before the node powers up or upgraded, there should be links connected to the PEs of a provider VPNs.
- For Scenario #3, the Internet facing WAN ports are added to (or removed from) existing VPN PEs.

5.2. Packet Walk-Through for Scenario #1

Upon power up, a SDWAN node can learn client routes from the Client facing ports, in the same way as EVPN described in RFC8388. Controller facilitates the IPsec SA establishment and rekey management as described in [SECURE-EVPN]. Controller manages how client's routes are associated with individual IPsec SA.

[SECURE-L3VPN] describes how to extend the RFC4364 VPN to allow some PEs being connected to other PEs via public networks. [SECURE-L3VPN] introduces the concept of Red Interface & Black Interface on those

PEs, with RED interfaces face clients' routes within the VPN and the Black Interfaces being WAN ports over which only IPsec-protected packets to the Internet or other backbone network are sent so that eliminating the need for MPLS transport in the backbone.

[SECURE-L3VPN] assumes PEs terminate MPLS packets, and use MPLS over IPsec when sending over the Black Interfaces.

[SECURE-EVPN] describes a solution where BGP point-to-multipoint signaling is leveraged as control plane for SDWAN Scenario #1. It utilizes the BGP RR to facilitate the key and policy exchange among PE devices to create private pair-wise IPsec Security Associations without IKEv2 point-to-point signaling or any other direct peer-to-peer session establishment messages.

When C-PEs do not support MPLS, the approaches described by RFC8365 can be used, with addition of IPsec encrypting the IP packets when sending packets over the Black Interfaces.

5.3. Packet Walk-Through for Scenario #2

In this scenario, C-PEs have some WAN ports connected to the public internet and some WAN ports connected to (i.e. directly connected to) PEs of trusted VPN. The C-PEs in Scenario #2 are almost like CPEs to MPLS VPN that have the IP/Ethernet data frames egress to the PEs of the VPN, except the packets need encryption if egress to the WAN ports facing public Internet.

Users specify the policy or criteria on which flows can only egress WAN ports facing trusted VPN without encryption, which can egress the WAN ports facing the public Internet with encryption, or which can egress WAN ports facing the public Internet without encryption.

The Control Plane should not learn routes from the Public Network facing WAN ports. Should strictly follow the policies specified by the users. The internet facing WAN ports can face potential DDoS attacks, additional anti-DDoS mechanism has to be implemented on WAN ports facing those public networks.

The Scenario #2 SDWAN Control Plane has three distinct functional components:

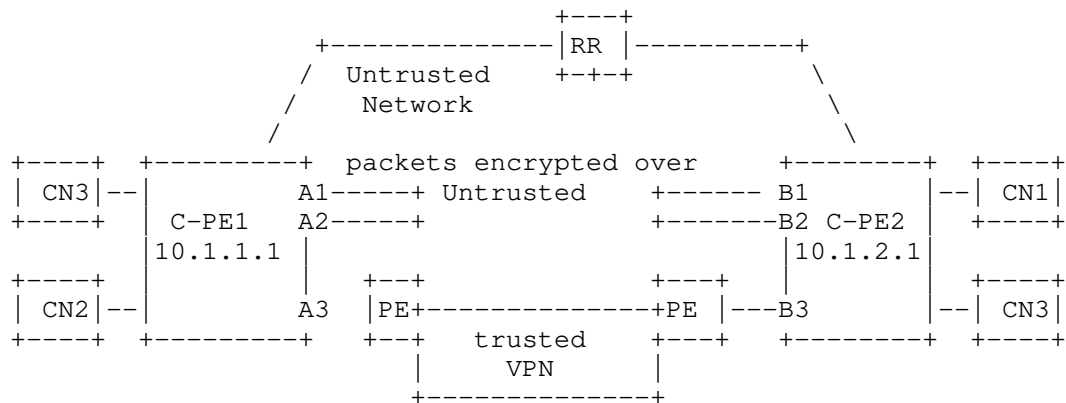


Figure 5: SDWAN Scenario #2

- SDWAN node's WAN ports property registration to the SDWAN Network Controller (BGP RR).
 - o This is used to inform the SDWAN controller of all the underlay networks to which the C-PE is connected.
 - o RR is responsible for propagating the C-PE WAN ports properties to authorized peers.
- Controller Facilitated IPsec SA management and NAT information distribution
 - o Used by the SDWAN controller to facilitate or manage the IPsec configurations and peer authentications for all IPsec SAs terminated at the SDWAN nodes.
 - o When WAN ports have private addresses, need exchange between SDWAN edges and the RR about the type of NAT, and mapping of the private addresses/ports <-> public addresses/ports.
- Attached routes distribution via BGP RR, which can be EVPN, IPVPN or others.
 - o This is for the overlay layer's route distribution, so that a C-PE can establish the overlay routing table that identifies the next hop for reaching a specific route/service attached to remote nodes. [SECURE-EVPN] describes EVPN and other options.

5.3.1. SDWAN node WAN Ports Properties Registration

In Figure 6, A1/A2/A3/B1/B2/B3 WAN ports can be from different network providers.

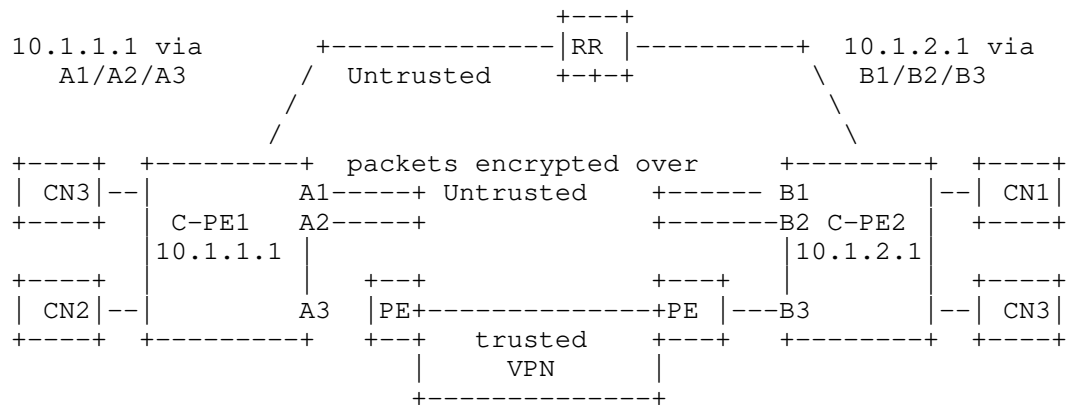


Figure 6: SDWAN Scenario #2 WAN Ports Registration

Each SDWAN edge(C-PE) needs to register its WAN ports properties along with its Loopback addresses to the SDWAN Network Controller (RR). The policies that govern the communications among peers are managed and controlled by the SDWAN Controller. Individual SDWAN edge relies on its SDWAN Controller to determine which peers can establish connections. The SDWAN controller is responsible for propagating the mapping information to the authorized peers. If C-PE-1 is not authorized to communicate with C-PE-n, C-PE-1's WAN port<->Loopback address mapping will not be propagated to C-PE-n.

A C-PE's Loopback addresses & attached routes may not be visible to some ISPs/NSPs to which the CPE's WAN port is connected.

5.3.2. Controller Facilitated IPsec SA & NAT management

One IPsec SA between two end points is straightforward. However, for a network with many IPsec SAs among many end points, the configuration and IPsec Key management for the entire network can be complex.

For a 1,000-node network, each node is responsible for maintaining and managing 999 keys to all their peers, which could potentially result in 1,000,000 key exchanges to authenticate among all nodes. In addition, when an edge node has multiple tenants attached, the edge node may need to establish multiple tunnels for tenants. For example, for a network with N nodes, a node A has 5 tenants app attached to it, then the node A has to maintain $5*(N-1)$ number of keys if each tenant needs to communicate with all other nodes.

In addition, all the IPsec keys have to be refreshed periodically, which adds more complexity. Therefore, simplification facilitated by an SDWAN controller is necessary for large-scale SDWAN deployment.

When the SDWAN IPsec SAs are fine-grained, such as per client address, per client's VLAN, the number of IPsec SAs & Keys to be managed can go much higher, leading to more IPsec management complexity. It is better to aggregate multiple flows into one IPsec SA.

SDWAN edge nodes can rely on the SDWAN controller to facilitate the pair-wise IPsec key establishment and refreshment [RFC7296] and maintain the Security Policy Database (SPD) [RFC4301].

- In the Figure 5 SDWAN Scenario #2 above, if C-PE1 & C-PE2 each has two ports facing two different ISPs networks, and their loopback addresses are not visible to the ISPs, i.e. the C-PE1 & C-PE2 are using a provider assigned IP addresses for A1/A2/B1/B2; you are going to need minimum four IPsec SAs between C-PE1 & C-PE2.
- When C-PEs loopback addresses are visible to ISPs/NSPs, i.e. the C-PEs' private source and destination IPs are part of a prefix exported to the ISP(s) in each site, it is possible to have one IPsec SA between C-PE1 & C-PE2.

The IP addresses of SDWAN WAN port can be dynamic (e.g. assigned by DHCP) or private IP. Some SDWAN nodes are identified by "System-ID" or Loopback addresses that are only locally significant. In some SDWAN environments, "System-ID + PortID" are used to uniquely identify a SDWAN WAN port. Sometimes, a SDWAN tunnel end-point can be associated with "private IP" + "public IP" (if NAT is used.)

When CPE WAN ports are private addresses, an additional sub-TLV has to be added to the [Tunnel-Encap] to describe the additional

information about the NAT property of SDWAN nodes' WAN ports. A SDWAN node can inquire STUN (Session Traversal of UDP through Network Address Translation [RFC 3489]) Server to get the NAT property, the public IP address and the Public Port number to pass to the authorized peers via the SDWAN Controller.

5.3.3. BGP Based SDWAN client routes

The client routes attached to SDWAN client ports have to be distributed to all SDWAN edge nodes, just like BGP/MPLS IP VPN [RFC4364], so that all SDWAN edges can establish the overlay routing table that identifies the remote SDWAN edges to reach a specific route/service. When C-PEs do not handle MPLS, RFC8365 can be used for packets over WAN ports, albeit applying IPsec SA encryption when sent over the WAN ports facing the public networks.

Using the terminologies described by [SECURE-L3VPN], the RED interface are the clients' ports and the ports facing private networks (e.g. connected to the PEs of MPLS VPN). Black Interfaces are ports facing public networks. The behavior described in [SECURE-L3VPN] applies to this scenario too, the C-PEs cannot mix the routes learned from the Black Interfaces with the Routes from RED Interfaces.

To minimize the burden on SDWAN edge nodes (especially low powered virtual SDWAN edges), some SDWAN network can let SDWAN controller take care of authenticating communications among SDWAN edge nodes instead of pushing down policies to SDWAN edge nodes. SDWAN Edge nodes might get clients routes from SDWAN controller instead of learning from clients ports.

The Hybrid SDWAN control plane for distributing clients' routes is more similar to overlay using EVPN [RFC8365], albeit the packets sent over the internet facing ports have to be encrypted by IPsec SA.

[Tunnel-Encap] can be used to associate client routes with specific tunnels:

- C-PE1 can advertise the following properties to others C-PEs via RR:
 - Encapsulation capability of the Ports to VPN PE
 - Encapsulation capability of the Ports to the Internet:
GRE-IPsec, or MPLS over GRE over IPsec

- with prior established IPsec SA
 - NAT information if ports are private addresses
- The Remote Endpoint sub-TLV is NOT appropriate because
 - The network to which a SDWAN port is connected might have an identifier that is more than the AS number. The SDWAN controller might use its own specific identifier for the network.
 - Suggest using an SDWAN overlay specific Transport-Network-ID to represents the connected networks.

The underlay network selections to next hop C-PE can be a local decision. Different services, routes, or VLANs can be aggregated to one underlay network between two C-PEs; the same service/routes/VLAN can spread over multiple SDWAN underlay networks at the next segment.

5.4. Packet Walk-Through for Scenario #3

The behavior described in [SECURE-L3VPN] applies to this scenario, except C-PEs not only have RED interfaces facing clients with within the VPN but also have RED interface facing MPLS backbone, with additional BLACK interfaces facing the untrusted public networks. The C-PEs cannot mix the routes learned from the Black Interfaces with the Routes from RED Interfaces. The routes learned from core-facing RED interfaces are for underlay and cannot be mixed with the routes learned over access-facing RED interfaces that are for overlay. Furthermore, the routes learned over core-facing interfaces (both RED and BLACK) can be shared in the same GLOBAL route table.

There may be some added risks of the packets from the ports facing the Internet. Therefore, special consideration has to be given to the routes from WAN ports facing the Internet. RFC4364 describes using an RD to create different routes for reaching same system. A similar approach can be considered to force packets received from the Internet facing ports to go through special security functions before being sent over to the VPN backbone WAN ports.

6. Manageability Considerations

SDWAN overlay networks utilize the SDWAN controller to facilitate route distribution, central configurations, and others. To minimize the burden on SDWAN edge nodes, SDWAN Edge nodes might not need to learn the routes from clients.

7. Security Considerations

Having WAN ports facing the public Internet introduces the following security risks:

- 1) Potential DDoS attack to the C-PEs with ports facing internet.
- 2) Potential risk of provider VPN network being injected with illegal traffic coming from the public Internet WAN ports on the C-PEs.

8. IANA Considerations

None

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4364] E. rosen, Y. Rekhter, "BGP/MPLS IP Virtual Private networks (VPNs)", Feb 2006.
- [RFC7296] C. Kaufman, et al, "Internet Key Exchange Protocol Version 2 (IKEv2)", Oct 2014.
- [RFC7432] A. Sajassi, et al, "BGP MPLS-Based Ethernet VPN", Feb 2015.
- [RFC8365] A. Sajassi, et al, "A network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", March 2018.

9.2. Informative References

- [RFC8192] S. Hares, et al, "Interface to Network Security Functions (I2NSF) Problem Statement and Use Cases", July 2017
- [RFC5521] P. Mohapatra, E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", April 2009.
- [BGP-SDWAN-Port] L. Dunbar, H. Wang, W. Hao, "BGP Extension for SDWAN Overlay Networks", draft-dunbar-idr-bgp-sdwan-overlay-ext-03, work-in-progress, Nov 2018.
- [Net2Cloud-Gap] L. Dunbar, A. Malis, C. Jacquenet, "Gap Analysis of Interconnecting Underlay with Cloud Overlay", draft-dm-net2cloud-gap-analysis-02, work in progress, Oct. 2018.
- [VPN-over-Internet] E. Rosen, "Provide Secure Layer L3VPNs over Public Infrastructure", draft-rosen-bess-secure-l3vpn-00, work-in-progress, July 2018
- [DMVPN] Dynamic Multi-point VPN:
<https://www.cisco.com/c/en/us/products/security/dynamic-multipoint-vpn-dmvpn/index.html>
- [DSVPN] Dynamic Smart VPN:
<http://forum.huawei.com/enterprise/en/thread-390771-1-1.html>
- [SECURE-EVPN] A. Sajassi, et al, "Secure EVPN", draft-sajassi-bess-secure-evpn-01, Work-in-progress, March 2019.
- [SECURE-L3VPN] E. Rosen, R. Bonica, "Secure Layer L3VPN over Public Infrastructure", draft-rosen-bess-secure-l3vpn-00, Work-in-progress, June 2018.
- [ITU-T-X1036] ITU-T Recommendation X.1036, "Framework for creation, storage, distribution and enforcement of policies for network security", Nov 2007.

[Net2Cloud-Problem] L. Dunbar and A. Malis, "Seamless Interconnect Underlay to Cloud Overlay Problem Statement", draft-dm-net2cloud-problem-statement-02, June 2018

[Net2Cloud-gap] L. Dunbar, A. Malis, and C. Jacquenet, "Gap Analysis of Interconnecting Underlay with Cloud Overlay", draft-dm-net2cloud-gap-analysis-02, work-in-progress, Aug 2018.

[Tunnel-Encap] E. Rosen, et al "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-10, Aug 2018.

10. Acknowledgments

Acknowledgements to Jim Guichard, John Scudder, Darren Dukes, Andy Malis and Donald Eastlake for their review and contributions.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Linda Dunbar
Huawei
Email: ldunbar@futurewei.com

James Guichard
Huawei
Email: james.n.guichard@futurewei.com

Ali Sajassi
Cisco
Email: sajassi@cisco.com

John Drake
Juniper
Email: jdrake@juniper.net

INTERNET-DRAFT
Intended status: Proposed Standard

V. Govindan
M. Mudigonda
A. Sajassi
Cisco Systems
G. Mirsky
ZTE
D. Eastlake
Futurewei Technologies
July 6, 2019

Expires: January 5, 2020

Fault Management for EVPN networks
draft-gsm-bess-evpn-bfd-03

Abstract

This document specifies proactive, in-band network OAM mechanisms to detect loss of continuity and miss-connection faults that affect unicast and multi-destination paths (used by Broadcast, Unknown Unicast and Multicast traffic) in an Ethernet VPN (EVPN) network. The mechanisms specified in the draft are based on the widely adopted Bidirectional Forwarding Detection (BFD) protocol.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Distribution of this document is unlimited. Comments should be sent to the authors or the BESS working group mailing list: bess@ietf.org.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>. The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Table of Contents

1. Introduction.....	3
1.1 Terminology.....	3
2. Scope of this Document.....	5
3. Motivation for Running BFD at the EVPN Network Layer....	6
4. Fault Detection for Unicast Traffic.....	7
5. Fault Detection for BUM Traffic.....	8
5.1 Ingress Replication.....	8
5.2 P2MP Tunnels (Label Switched Multicast).....	8
6. BFD Packet Encapsulation.....	9
6.1 MPLS Encapsulation.....	9
6.1.1 Unicast.....	9
6.1.2 Ingress Replication.....	10
6.1.3 LSM (Label Switched Multicast, P2MP).....	11
6.2 VXLAN Encapsulation.....	11
6.2.1 Unicast.....	11
6.2.2 Ingress Replication.....	13
6.2.3 LSM (Label Switched Multicast, P2MP).....	13
7. BGP Distribution of BFD Discriminators.....	14
8. Scalability Considerations.....	14
9. IANA Considerations.....	15
9.1 Pseudowire Associated Channel Type.....	15
9.2 MAC Address.....	15
10. Security Considerations.....	15
Acknowledgement.....	15
Normative References.....	16
Informative References.....	18
Authors' Addresses.....	19

1. Introduction

[ietf-bess-evpn-oam-req-frmwk] outlines the OAM requirements of Ethernet VPN networks (EVPN [RFC7432]). This document specifies mechanisms for proactive fault detection at the network (overlay) layer of EVPN. The mechanisms proposed in the draft use the widely adopted Bidirectional Forwarding Detection (BFD [RFC5880]) protocol.

EVPN fault detection mechanisms need to consider unicast traffic separately from Broadcast, Unknown Unicast, and Multicast (BUM) traffic since they map to different Forwarding Equivalency Classes (FECs) in EVPN. Hence this document proposes different fault detection mechanisms to suit each type, for unicast traffic using BFD [RFC5880] and for BUM traffic using BFD or [RFC8563] depending on whether an MP2P or P2MP tunnel is being used.

Packet loss and packet delay measurement are out of scope for this document.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

The following acronyms are used in this document.

BFD - Bidirectional Forwarding Detection [RFC5880]

BUM - Broadcast, Unknown Unicast, and Multicast

CC - Continuity Check

CV - Connectivity Verification

EVI - EVPN Instance

EVPN - Ethernet VPN [RFC7432]

FEC - Forwarding Equivalency Class

GAL - Generic Associated Channel Label [RFC5586]

LSM - Label Switched Multicast (P2MP)

LSP - Label Switched Path

MP2P - Multi-Point to Point

OAM - Operations Administration, and Maintenance

P2MP - Point to Multi-Point (LSM)

PE - Provider Edge

VXLAN - Virtual eXtensible Local Area Network (VXLAN) [RFC7348]

2. Scope of this Document

This document specifies BFD based mechanisms for proactive fault detection for EVPN both as specified in [RFC7432] and also for EVPN using VXLAN encapsulation [ietf-vxlan-bfd]. It covers the following:

- o Unicast traffic.
- o BUM traffic using Multi-point-to-Point (MP2P) tunnels (ingress replication).
- o BUM traffic using Point-to-Multipoint (P2MP) tunnels (Label Switched Multicast (LSM)).
- o MPLS and VXLAN encapsulation.

This document does not discuss BFD mechanisms for:

- o EVPN variants like PBB-EVPN [RFC7623]. It is intended to address this in future versions.
- o Integrated Routing and Bridging (IRB) solution based on EVPN [ietf-bess-evpn-inter-subnet-forwarding]. It is intended to address this in future versions.
- o EVPN using other encapsulations such as NVGRE or MPLS over GRE [RFC8365].
- o BUM traffic using MP2MP tunnels.

This specification specifies procedures for BFD asynchronous mode. BFD demand mode is outside the scope of this specification except as it is used in [RFC8563]. The use of the Echo function is outside the scope of this specification.

3. Motivation for Running BFD at the EVPN Network Layer

The choice of running BFD at the network layer of the OAM model for EVPN [ietf-bess-evpn-oam-req-frmwk] was made after considering the following:

- o In addition to detecting link failures in the EVPN network, BFD sessions at the network layer can be used to monitor the successful setup of MP2P and P2MP EVPN tunnels transporting Unicast and BUM traffic such as label programming. The scope of reachability detection covers the ingress and the egress EVPN PE nodes and the network connecting them.
- o Monitoring a representative set of path(s) or a particular path among the multiple paths available between two EVPN PE nodes could be done by exercising entropy mechanisms such as entropy labels, when they are used, or VXLAN source ports. However, paths that cannot be realized by entropy variations cannot be monitored. Fault monitoring requirements outlined by [ietf-bess-evpn-oam-req-frmwk] are addressed by the mechanisms proposed by this draft.

BFD testing between EVPN PE nodes does not guarantee that the EVPN service is functioning. (This can be monitored at the service level, that is CE to CE.) For example, an egress EVPN-PE could understand EVPN labeling received but could switch data to an incorrect interface. However, BFD testing in the EVPN Network Layer does provide additional confidence that data transported using those tunnels will reach the expected egress node. When BFD testing in the EVPN overlay fails, that can be used as an indication of a Loss-of-Connectivity defect in the EVPN underlay that would cause EVPN service failure.

4. Fault Detection for Unicast Traffic

The mechanisms specified in BFD for MPLS LSPs [RFC5884] [RFC7726] are applied to test the handling of unicast EVPN traffic. The discriminators required for de-multiplexing the BFD sessions are advertised through BGP as specified in Section 7. This is needed for MPLS since the label stack does not contain enough information to disambiguate the sender of the packet.

The usage of MPLS entropy labels or various VXLAN source ports takes care of the requirement to monitor various paths of the multi-path server layer network [RFC6790]. Each unique realizable path between the participating PE routers MAY be monitored separately when such entropy is used. At least one path of multi-path connectivity between two PE routers MUST be tracked with BFD, but in that case the granularity of fault-detection will be coarser. To support unicast OAM, each PE node MUST allocate a BFD discriminator to be used for BFD messages to that PE and MUST advertise this discriminator with BGP as specified in Section 7. Once the BFD session for the EVPN label is UP, the ends of the BFD session MUST NOT change the local discriminator values of the BFD Control packets they generate, unless they first bring down the session as specified in [RFC5884].

5. Fault Detection for BUM Traffic

Section 5.1 below discusses fault detection for MP2P tunnels using ingress replication and Section 5.2 discusses fault detection for P2MP tunnels.

5.1 Ingress Replication

Ingress replication uses separate MP2P tunnels for transporting BUM traffic from the ingress PE (head) to a set of one or more egress PEs (tails). The fault detection mechanism specified by this document takes advantage of the fact that the head makes a unique copy for each tail.

Another key aspect to be considered in EVPN is the advertisement of the inclusive multicast route. The BUM traffic flows from a head node to a particular tail only after the head receives the inclusive multicast route. This contains the BUM EVPN label (downstream allocated) corresponding to the MP2P tunnel for MPLS encapsulation and contains the IP address of the PE originating the inclusive multicast route for use in VXLAN encapsulation.

There MAY exist multiple BFD sessions between a head PE and an individual tail due to (1) the usage of MPLS entropy labels [RFC6790] or VXLAN source ports for an inclusive multicast FEC and (2) due to multiple MP2P tunnels indicated by different tail labels or IP addresses for MPLS or VXLAN. The BFD discriminator to be used is distributed by BGP as specified in Section 7. Once the BFD session for the EVPN label is UP, the BFD systems terminating the BFD session MUST NOT change the local discriminator values of the BFD Control packets they generate, unless they first bring down the session as specified in [RFC5884].

5.2 P2MP Tunnels (Label Switched Multicast)

Fault detection for BUM traffic distributed using a P2MP tunnel uses active tail multipoint BFD [RFC8563] in one of the three scenarios providing head notification (see Section 5.2 of [RFC8563]).

For MPLS encapsulation of the head to tails BFD, Label Switched Multicast is used. For VXLAN encapsulation, BFD is delivered to the tails through underlay multicast using an outer multicast IP address.

6. BFD Packet Encapsulation

The sections below describe the MPLS and VXLAN encapsulations of BFD for EVPN OAM use.

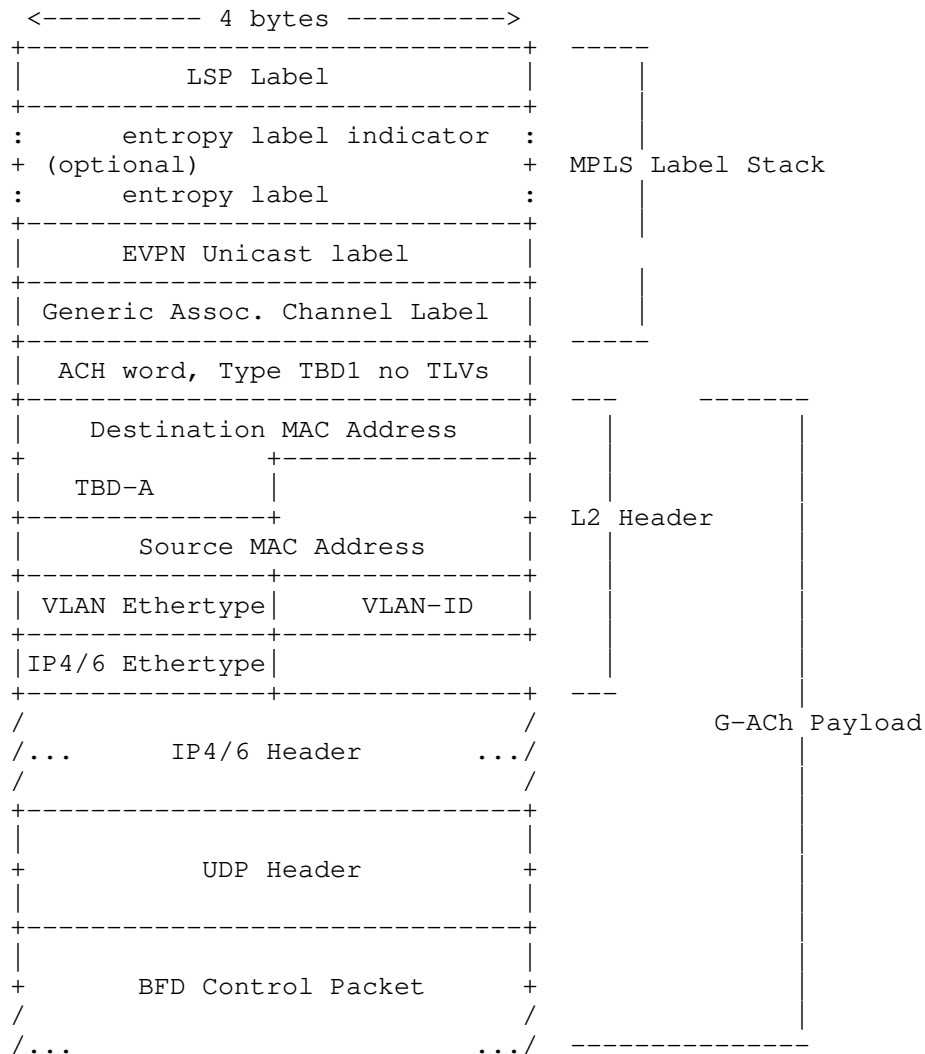
6.1 MPLS Encapsulation

This section describes use of the Generic Associated Channel Label (GAL) for BFD encapsulation in MPLS based EVPN OAM.

6.1.1 Unicast

The packet initially contains the following labels: LSP label (transport), the optional entropy label, and the EVPN Unicast label. The G-ACh type is set to TBD1. The G-ACh payload of the packet MUST contain the destination L2 header (in overlay space) followed by the IP header that encapsulates the BFD packet. The MAC address of the inner packet is used to validate the <EVI, MAC> in the receiving node.

- The destination MAC MUST be the dedicated MAC TBD-A (see Section 9) or the MAC address of the destination PE.
- The destination IP address MUST be in the 127.0.0.0/8 range for IPv4 or in the 0:0:0:0:0:FFFF:7F00:0/104 range for IPv6.
- The destination IP port MUST be 3784 [RFC5881].
- The source IP port MUST be in the range 49152 through 65535.
- The discriminator values for BFD are obtained through BGP as specified in Section 7 or are exchanged out-of-band or through some other means outside the scope of this document.



6.1.2 Ingress Replication

The packet initially contains the following labels: LSP label (transport), the optional entropy label, the BUM label, and the split horizon label [RFC7432] (where applicable). The G-ACh type is set to TBD1. The G-ACh payload of the packet is as described in Section 6.1.1.

6.1.3 LSM (Label Switched Multicast, P2MP)

The encapsulation is the same as in Section 6.1.2 for ingress replication except that the transport label identifies the P2MP tunnel, in effect the set of tail PEs, rather than identifying a single destination PE at the end of an MP2P tunnel.

6.2 VXLAN Encapsulation

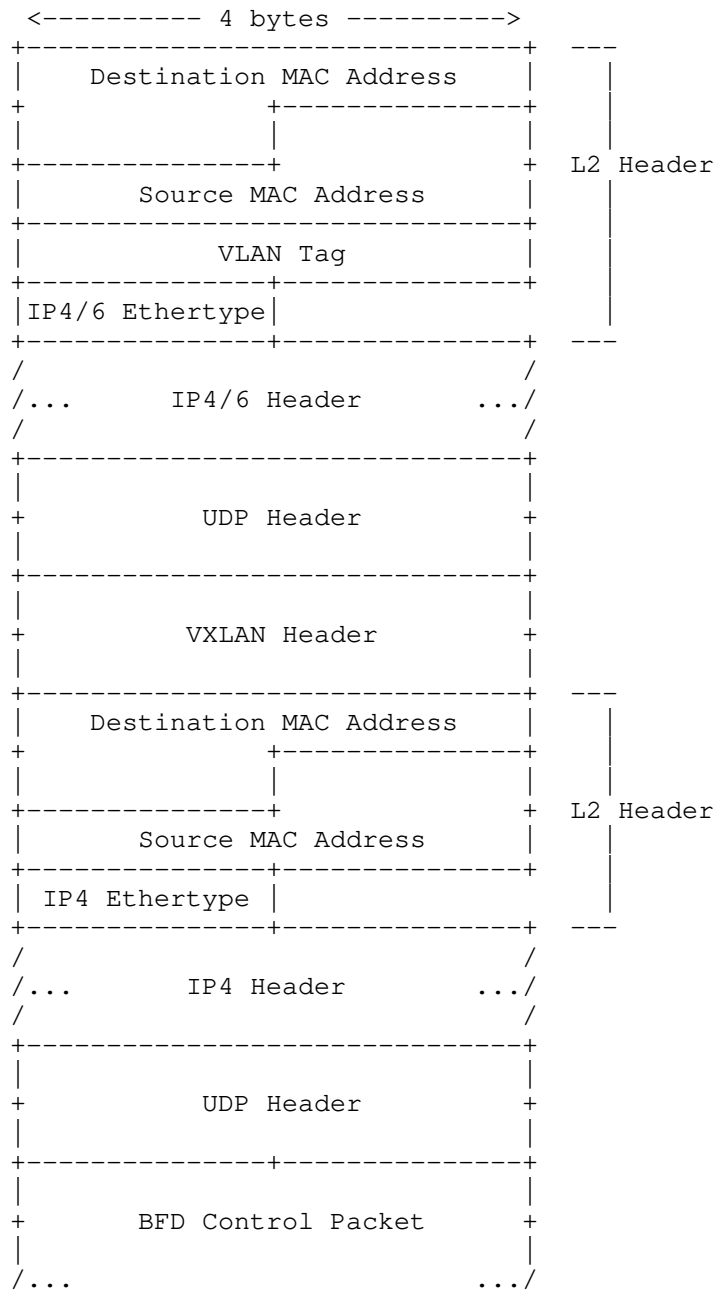
This section describes the use of the VXLAN [RFC7348] for BFD encapsulation in VXLAN based EVPN OAM. This specification conforms to [ietf-bfd-vxlan].

6.2.1 Unicast

The outer and inner IP headers have a unicast source IP address of the BFD message source and a destination IP address of the BFD message destination

The destination UDP port MUST be 3784 [RFC5881]. The source port MUST be in the range 49152 through 65535. If the BFD source has multiple IP addresses, entropy MAY be further obtained by using any of those addresses assuming the source is prepared for responses directed to the IP address used.

The Your BFD discriminator is the value distributed for this unicast OAM purpose by the destination using BGP as specified in Section 7 or is exchanged out-of-band or through some other means outside the scope of this document.



6.2.2 Ingress Replication

The BFD packet construction is as given in Section 6.2.1 except as follows:

- (1) The destination IP address used by the BFD message source is that advertised by the destination PE in its Inclusive Multicast EVPN route for the MP2P tunnel in question; and
- (2) The Your BFD discriminator used is the one advertised by the BFD destination using BGP as specified in Section 7 for the MP2P tunnel in question or is exchanged out-of-band or through some other means outside the scope of this document.

6.2.3 LSM (Label Switched Multicast, P2MP)

The VXLAN encapsulation for the head-to-tails BFD packets uses the multicast destination IP corresponding to the VXLAN VNI.

The destination port MUST be 3784. For entropy purposes, the source port can vary but MUST be in the range 49152 through 65535 [RFC5881]. If the head PE has multiple IP addresses, entropy MAY be further obtained by using any of those addresses.

The Your BFD discriminator is the value distributed for this unicast OAM purpose by the BFD message using BGP as specified in Section 7 or is exchanged out-of-band or through some other means outside the scope of this document.

7. BGP Distribution of BFD Discriminators

BGP is used to distribute BFD discriminators for use in EVPN OAM as follows using the BGP-BFD Attribute as specified in [ietf-bess-mvpn-fast-failover]. This attribute is included with appropriate EVPN routes as follows:

Unicast: MAC/IP Advertisement Route [RFC7432].

MP2P Tunnel: Inclusive Multicast Ethernet Tag Route [RFC7432].

P2MP: TBD

[Need more text on BFD sessions reacting to the new advertisement and withdrawal of the BGP-BFD Attribute.]

8. Scalability Considerations

The mechanisms proposed by this draft could affect the packet load on the network and its elements especially when supporting configurations involving a large number of EVIs. The option of slowing down or speeding up BFD timer values can be used by an administrator or a network management entity to maintain the overhead incurred due to fault monitoring at an acceptable level.

9. IANA Considerations

The following IANA Actions are requested.

9.1 Pseudowire Associated Channel Type

IANA is requested to assign a channel type from the "Pseudowire Associated Channel Types" registry in [RFC4385] as follows.

Value	Description	Reference
-----	-----	-----
TBD1	BFD-EVPN OAM	[this document]

9.2 MAC Address

IANA is requested to assign a multicast MAC address under the IANA OUI [0x01005E900004 suggested] as follows:

Address	Usage	Reference
-----	-----	-----
TBD-A	EVPN OAM	[this document]

10. Security Considerations

Security considerations discussed in [RFC5880], [RFC5883], and [RFC8029] apply.

MPLS security considerations [RFC5920] apply to BFD Control packets encapsulated in a MPLS label stack. When BPD Control packets are routed, the authentication considerations discussed in [RFC5883] should be followed.

VXLAN BFD security considerations in [ietf-vxlan-bfd] apply to BFD packets encapsulate in VXLAN.

Acknowledgement

The authors wish to thank the following for their comments and suggestions:

Mach Chen

Normative References

- [ietf-bess-evpn-inter-subnet-forwarding] Sajassi, A., Salam, S., Thoria, S., Rekhter, Y., Drake, J., Yong, L., and L. Dunbar, "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-08, work in progress, March 2019.
- [ietf-bess-mvpn-fast-failover] Morin, T., Kebler, R., Mirsky, G., "Multicast VPN fast upstream failover", draft-ietf-bess-mvpn-fast-failover-05 (work in progress), February 2019.
- [ietf-bfd-vxlan] Pallagatti, S., Paragiri, S., Govindan, V., Mudigonda, M., G. Mirsky, "BFD for VXLAN", draft-ietf-bfd-vxlan-07 (work in progress), May 2019.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<http://www.rfc-editor.org/info/rfc4385>>.
- [RFC5586] Bocci, M., Ed., Vigoureux, M., Ed., and S. Bryant, Ed., "MPLS Generic Associated Channel", RFC 5586, DOI 10.17487/RFC5586, June 2009, <<https://www.rfc-editor.org/info/rfc5586>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<http://www.rfc-editor.org/info/rfc5880>>.
- [RFC5881] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for IPv4 and IPv6 (Single Hop)", RFC 5881, DOI 10.17487/RFC5881, June 2010, <<https://www.rfc-editor.org/info/rfc5881>>.
- [RFC5883] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD) for Multihop Paths", RFC 5883, DOI 10.17487/RFC5883, June 2010, <<https://www.rfc-editor.org/info/rfc5883>>.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <<https://www.rfc-editor.org/info/rfc5884>>.

- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<http://www.rfc-editor.org/info/rfc6790>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<http://www.rfc-editor.org/info/rfc7432>>.
- [RFC7623] Sajassi, A., Ed., Salam, S., Bitar, N., Isaac, A., and W. Henderickx, "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, DOI 10.17487/RFC7623, September 2015, <<http://www.rfc-editor.org/info/rfc7623>>.
- [RFC7726] Govindan, V., Rajaraman, K., Mirsky, G., Akiya, N., and S. Aldrin, "Clarifying Procedures for Establishing BFD Sessions for MPLS Label Switched Paths (LSPs)", RFC 7726, DOI 10.17487/RFC7726, January 2016, <<https://www.rfc-editor.org/info/rfc7726>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <<https://www.rfc-editor.org/info/rfc8365>>.
- [RFC8563] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) Multipoint Active Tails", RFC 8563, DOI 10.17487/RFC8563, April 2019, <<https://www.rfc-editor.org/info/rfc8563>>.

Informative References

- [ietf-bess-evpn-oam-req-frmwk] Salam, S., Sajassi, A., Aldrin, S., J. Drake, and D. Eastlake, "EVPN Operations, Administration and Maintenance Requirements and Framework", draft-ietf-bess-evpn-oam-req-frmwk-00, work in progress, February 2019.
- [RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <<https://www.rfc-editor.org/info/rfc5920>>.

Authors' Addresses

Vengada Prasad Govindan
Cisco Systems

Email: venggovi@cisco.com

Mudigonda Mallik
Cisco Systems

Email: mmudigon@cisco.com

Ali Sajassi
Cisco Systems
170 West Tasman Drive
San Jose, CA 95134, USA

Email: sajassi@cisco.com

Gregory Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Donald Eastlake, 3rd
Huawei Technologies
1424 Pro Shop Court
Davenport, FL 33896 USA

Phone: +1-508-333-2270
Email: d3e3e3@gmail.com

Copyright, Disclaimer, and Additional IPR Provisions

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: 8 September 2022

P. Brissette, Ed.
A. Sajassi
LA. Burdet
Cisco
J. Drake
Juniper
J. Rabadan
Nokia
7 March 2022

Fast Recovery for EVPN Designated Forwarder Election
draft-ietf-bess-evpn-fast-df-recovery-05

Abstract

Ethernet Virtual Private Network (EVPN) solution provides Designated Forwarder election procedures for multihomed Ethernet Segments. These procedures have been enhanced further by applying Highest Random Weight (HRW) Algorithm for Designated Forwarder election in order to avoid unnecessary DF status changes upon a failure. This draft improves these procedures by providing a fast Designated Forwarder (DF) election upon recovery of the failed link or node associated with the multihomed Ethernet Segment. The solution is independent of number of EVIs associated with that Ethernet Segment and it is performed via a simple signaling between the recovered PE and each of the other PEs in the multihoming group.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119] and RFC 8174 [RFC8174].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 8 September 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Terminology	3
2. Challenges with Existing Solution	3
3. DF Election Synchronization Solution	4
3.1. Advantages	5
3.2. BGP Encoding	6
3.3. Synchronization Scenarios	7
3.4. Backwards Compatibility	8
4. Security Considerations	8
5. IANA Considerations	9
6. Normative References	9
Appendix A. Contributors	10
Appendix B. Acknowledgements	10
Authors' Addresses	10

1. Introduction

Ethernet Virtual Private Network (EVPN) solution [RFC7432] is becoming pervasive in data center (DC) applications for Network Virtualization Overlay (NVO) and DC interconnect (DCI) services, and in service provider (SP) applications for next generation virtual private LAN services.

The EVPN specification [RFC7432] describes DF election procedures for multihomed Ethernet Segments. These procedures are enhanced further in [RFC8584] by applying Highest Random Weight Algorithm for DF election in order to avoid DF status change unnecessarily upon a link or node failure associated with the multihomed Ethernet Segment. This draft makes further improvement to DF election procedures in [RFC8584] by providing an option for a fast DF election upon recovery of the failed link or node associated with the multihomed Ethernet Segment. This DF election is achieved independent of number of EVIs associated with that Ethernet Segment and it is performed via a simple signaling between the recovered PE and each of the other PEs in the multihomed group. The solution is based on simple one-way signaling mechanism.

1.1. Terminology

Provider Edge (PE): A device that sits in the boundary of Provider and Customer networks and performs encap/decap of data from L2 to L3 and vice-versa.

Designated Forwarder (DF): A PE that is currently forwarding (encapsulating/decapsulating) traffic for a given VLAN in and out of a site.

2. Challenges with Existing Solution

In EVPN technology, multiple PE devices have the ability to encap and decap data belonging to the same VLAN. In certain situations, this may cause L2 duplicates and even loops if there is a momentary overlap of forwarding roles between two or more PE devices, leading to broadcast storms.

EVPN [RFC7432] currently uses timer based synchronization among PE devices in redundancy group that can result in duplications (and even loops) because of multiple DFs if the timer is too short or blackholing if the timer is too long.

Using split-horizon filtering (Section 8.3 of [RFC7432]) can prevent loops (but not duplicates), however if there are overlapping DFs in two different sites at the same time for the same VLAN, the site identifier will be different upon re-entry of the packet and hence the split-horizon check will fail, leading to L2 loops.

The updated DF procedures in [RFC8584] use the well known HRW (Highest Random Weight) algorithm to avoid reshuffling of VLANs among PE devices in the redundancy group upon failure/recovery. This reduces the impact to VLANs not assigned to the failed/recovered ports and eliminates loops or duplicates at failure/recovery events.

However, upon PE insertion or port bring-up (recovery event), HRW also cannot help as a transfer of DF role to the newly inserted device/port must occur while the old DF is still active.

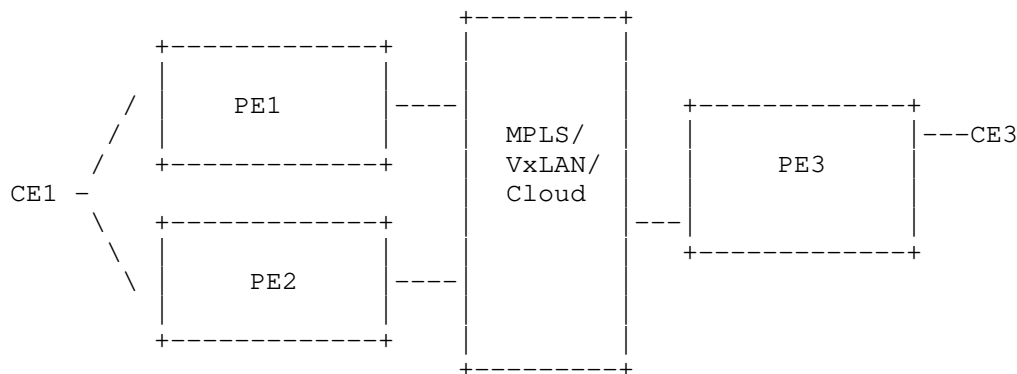


Figure 1: CE1 multihomed to PE1 and PE2.

In the Figure 1, when PE2 is inserted or booted up, PE1 will transfer DF role of some VLANs to PE2 to achieve load balancing. However, because there is no handshake mechanism between PE1 and PE2, duplication of DF roles for a given VLAN is possible. Duplication of DF roles may eventually lead to duplication of traffic as well as L2 loops.

Current EVPN specification [RFC7432] and [RFC8584] relies on a timer-based approach for transferring the DF role to the newly inserted device. This can cause the following issues:

- * Loops/Duplicates if the timer value is too short
- * Prolonged Traffic Blackholing if the timer value is too long

3. DF Election Synchronization Solution

The solution relies on the concept of common clock alignment between partner PEs participating to a common Ethernet Segment. The main idea is to have all peering PEs of that Ethernet Segment perform DF election, and apply their resulting carving state, at a same well-known time.

The DF Election procedure, as described in [RFC7432] and as optionally signalled in [RFC8584], is applied. All PEs attached to a given Ethernet Segment are clock-synchronized; using a networking protocol for clock synchronization (e.g. NTP, PTP, etc.). Newly

inserted device PE or during failure recovery of a PE, that PE communicates the current time to peering partners plus the remaining peering timer time left. This constitutes an "end time" or "absolute time" as seen from local PE. That absolute time is called "Service Carving Time" (SCT).

A new BGP Extended Community is advertised along with Ethernet Segment route (RT-4) to communicate to other partners the Service Carving Time.

Upon reception of that new BGP Extended Community, partner PEs know exactly its carving time. The notion of skew is introduced to eliminate any potential duplicate traffic or loops. They add a skew (default = -10ms) to the Service Carving Time to enforce this. The previously inserted PE(s) must carve first, followed shortly(skew) by the newly insterted PE.

To summarize, all peering PEs carve almost simultaneously at the time announced by newly added/recovered PE. The newly inserted PE initiates the SCT, and carves immediately on peering timer expiry. The previously inserted PE(s) receiving Ethernet Segment route (RT-4) with a SCT BGP extended community, carve shortly before Service Carving Time.

3.1. Advantages

There are multiples advantages of using the approach. Here is a non-exhaustive list:

- * A simple uni-directional signaling is all that is needed
- * Backwards-compatible: PEs supporting only older [RFC7432] shall simply discard unrecognized new "Service Carving Timestamp" BGP Extended Community
- * Multiple DF Election algorithms can be supported:
 - [RFC7432] default ordered list ordinal algorithm (Modulo),
 - [RFC8584] highest-random weight, etc.
- * Independent of BGP transmission delay regarding Ethernet Segment route (RT-4)
- * Agnostic of the time synchronization mechanism used (e.g. NTP, PTP, etc.)

3.2. BGP Encoding

A new BGP extended community needs to be defined to communicate the Service Carving Timestamp for each Ethernet Segment.

A new transitive extended community where the Type field is 0x06, and the Sub-Type is 0x0F is advertised along with Ethernet Segment route. The expected Service Carving Time is encoded as a 8-octet value as follows:

1																2																3																
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																	
Type = 0x06																Sub-Type(0x0F)																Timestamp Seconds																~
~ Timestamp Seconds																Timestamp Fractional Seconds																																

The timestamp exchanged uses the NTP epoch of January 1, 1900 [RFC5905]. The 64-bit timestamp of the NTP protocol consists of a 32-bit part for seconds and a 32-bit part for fractional second:

- * Timestamp Seconds: 32-bit NTP seconds are encoded in this field.
- * Timestamp Fractional Seconds: 16 bits of the NTP fractional seconds are encoded in this field. The use of a 16-bit fractional seconds yields adequate precision of 15 microseconds (2^{-16} s).

This document introduces a new flag called "T" (for Time Synchronization) to the bitmap field of the DF Election Extended Community defined in [RFC8584].

1																2																3																							
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1																								
Type = 0x06																Sub-Type(0x06)																RSV		DF Alg		A		T		~															
~ Bitmap																Reserved = 0																																							

- * Bit 3: Time Synchronization (corresponds to Bit 27 of the DF Election Extended Community). When set to 1, it indicates the desire to use Time Synchronization capability with the rest of the PEs in the Ethernet Segment.

This capability is used in conjunction with the agreed upon DF Type (DF Election Type). For example if all the PEs in the Ethernet Segment indicated that they have Time Synchronization capability and they want the DF type to be HRW, then HRW algorithm is used in conjunction with this capability.

3.3. Synchronization Scenarios

Let's take Figure 1 as an example where initially PE2 had failed and PE1 had taken over. This example shows the problem with the DF-Election mechanism in [RFC7432].

Based on Section 8.5 of [RFC7432], using the default 3 second peering timer:

1. Initial state: PE1 is in steady-state, PE2 is recovering
2. PE2 recovers at (absolute) time $t=99$
3. PE2 advertises RT-4 (sent at $t=100$) to partner PE1
4. PE2 starts a 3 second peering timer
5. PE1 carves immediately on RT-4 reception, i.e. $t=100$ + minimal BGP propagation delay
6. PE2 carves at time $t=103$

[RFC7432] aims of favouring traffic black hole over duplicate traffic. With above procedure, traffic black holing will occur as part of each PE recovery sequence since PE1 has transitioned some VLANs to Non-Designated-Forwarder (NDF) immediately upon reception. The peering timer value (default = 3 seconds) has a direct effect on the duration of the blackholing. A shorter (esp. zero) peering timer may, however, result in duplicate traffic or traffic loops.

Based on the Service Carving Time (SCT) approach:

1. Initial state: PE1 is in steady-state, PE2 is recovering
2. PE2 recovers at (absolute) time $t=99$
3. PE2 advertises RT-4 (sent at $t=100$) with target SCT value $t=103$ to partner PE1
4. PE2 starts 3 second peering timer
5. Both PE1 and PE2 carve at (absolute) time $t=103$

In fact, PE1 should carve slightly before PE2 (skew). The previously inserted PE2 that is recovering performs both transitions DF to NDF and NDF to DF per VLANs at the peering timer expiry. Since the goal is to prevent duplicates, the original PE1, which received the SCT will apply:

- * DF to NDF transition at $t = \text{SCT} - \text{skew}$ where both PEs are NDF for 'skew' amount of time
- * NDF to DF transition at $t = \text{SCT}$

It is this split-behaviour which ensures good transition of DF role with contained amount of loss.

Using SCT approach, the negative effect of the peering timer is mitigated. Furthermore, the BGP Ethernet Segment route (RT-4) transmission delay (from PE2 to PE1) becomes a non-issue. The use of SCT approach remedies the problem associated with the peering timer: the 3 second timer window is shortened to the order of milliseconds.

3.4. Backwards Compatibility

Per redundancy group, for the DF election procedures to be globally convergent and unanimous, it is necessary that all the participating PEs agree on the DF Election algorithm to be used. It is, however, possible that some PEs continue to use the existing modulo-based DF election and do not rely on the new SCT BGP extended community. PEs running a baseline DF election mechanism will simply discard the new SCT BGP extended community as unrecognized.

A PE can indicate its willingness to support clock-synched carving by signaling the new 'T' DF Election Capability as well as including the new Service Carving Time BGP extended community along with the Ethernet Segment Route (Type-4). In the case where one or more PEs attached to the Ethernet Segment do not signal $T=1$, all PEs in the Ethernet Segment SHALL revert back to the [RFC7432] timer approach. This is especially important in the context of the VLAN shuffling with more than 2 PEs.

4. Security Considerations

The mechanisms in this document use EVPN control plane as defined in [RFC7432]. Security considerations described in [RFC7432] are equally applicable. This document uses MPLS and IP-based tunnel technologies to support data plane transport. Security considerations described in [RFC7432] and in [RFC8365] are equally applicable.

5. IANA Considerations

This document solicits the allocation of the following sub-type in the "EVPN Extended Community Sub-Types" registry setup by [RFC7153]:

0x0F	Service Carving Timestamp	This document
------	---------------------------	---------------

This document solicits the allocation of the following values in the "DF Election Capabilities" registry setup by [RFC8584]:

Bit ----	Name -----	Reference -----
3	Time Synchronization	This document

6. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, DOI 10.17487/RFC5905, June 2010, <<https://www.rfc-editor.org/info/rfc5905>>.
- [RFC7153] Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", RFC 7153, DOI 10.17487/RFC7153, March 2014, <<https://www.rfc-editor.org/info/rfc7153>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8365] Sajassi, A., Ed., Drake, J., Ed., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution Using Ethernet VPN (EVPN)", RFC 8365, DOI 10.17487/RFC8365, March 2018, <<https://www.rfc-editor.org/info/rfc8365>>.

[RFC8584] Rabadan, J., Ed., Mohanty, S., Ed., Sajassi, A., Drake, J., Nagaraj, K., and S. Sathappan, "Framework for Ethernet VPN Designated Forwarder Election Extensibility", RFC 8584, DOI 10.17487/RFC8584, April 2019, <<https://www.rfc-editor.org/info/rfc8584>>.

Appendix A. Contributors

In addition to the authors listed on the front page, the following co-authors have also contributed substantially to this document:

Gaurav Badoni
Cisco

Email: gbadoni@cisco.com

Dhananjaya Rao
Cisco

Email: dhrao@cisco.com

Appendix B. Acknowledgements

Authors would like to acknowledge helpful comments and contributions of Satya Mohanty and Bharath Vasudevan. Also thank you to Anoop Ghanwani for his thorough review with valuable comments and corrections.

Authors' Addresses

Patrice Brissette (editor)
Cisco
Email: pbrisset@cisco.com

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Luc Andre Burdet
Cisco
Email: lburdet@cisco.com

John Drake
Juniper
Email: jdrake@juniper.net

Jorge Rabadan

Nokia

Email: jorge.rabadan@nokia.com

BESS Working Group
INTERNET-DRAFT

N. Malhotra, Ed.
Arrcus

Intended Status: Proposed Standard

A. Sajassi
A. Pattekar
Cisco

A. Lingala
AT&T

J. Rabadan
Nokia

J. Drake
Juniper Networks

Expires: Dec 21, 2019

June 19, 2019

Extended Mobility Procedures for EVPN-IRB
draft-ietf-bess-evpn-irb-extended-mobility-01

Abstract

Procedure to handle host mobility in a layer 2 Network with EVPN control plane is defined as part of RFC 7432. EVPN has since evolved to find wider applicability across various IRB use cases that include distributing both MAC and IP reachability via a common EVPN control plane. MAC Mobility procedures defined in RFC 7432 are extensible to IRB use cases if a fixed 1:1 mapping between VM IP and MAC is assumed across VM moves. Generic mobility support for IP and MAC that allows these bindings to change across moves is required to support a broader set of EVPN IRB use cases, and requires further consideration. EVPN all-active multi-homing further introduces scenarios that require additional consideration from mobility perspective. This document enumerates a set of design considerations applicable to mobility across these EVPN IRB use cases and defines generic sequence number assignment procedures to address these IRB use cases.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as

Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	4
1.1	Terminology	5
2.	Optional MAC only RT-2	6
3.	Mobility Use Cases	6
3.1	Host MAC+IP Move	6
3.2	Host IP Move to new MAC	6
3.2.1	VM Reload	7
3.2.2	MAC Sharing	7
3.2.3	Problem	7
3.3	Host MAC move to new IP	8
3.3.1	Problem	8
4.	EVPN All Active multi-homed ES	11
5.	Design Considerations	12
6.	Solution Components	13
6.1	Sequence Number Inheritance	13

6.2	MAC Sharing	14
6.3	Multi-homing Mobility Synchronization	15
7.	Requirements for Sequence Number Assignment	15
7.1	LOCAL MAC-IP learning	15
7.2	LOCAL MAC learning	16
7.3	Remote MAC OR MAC-IP Update	16
7.4	REMOTE (SYNC) MAC update	16
7.5	REMOTE (SYNC) MAC-IP update	17
7.6	Inter-op	17
7.7	MAC Sharing Race Condition	18
8.	Routed Overlay	18
9.	Duplicate Host Detection	19
9.1	Scenario A	19
9.2	Scenario B	20
9.2.1	Duplicate IP Detection Procedure for Scenario B	20
9.3	Scenario C	21
9.4	Duplicate Host Recovery	21
9.4.1	Route Un-freezing Configuration	21
9.4.2	Route Clearing Configuration	22
10.	Security Considerations	22
11.	IANA Considerations	23
12.	References	23
12.1	Normative References	23
12.2	Informative References	23
13.	Acknowledgements	23
	Authors' Addresses	23
	Appendix A	24

1 Introduction

EVPN-IRB enables capability to advertise both MAC and IP routes via a single MAC+IP RT-2 advertisement. MAC is imported into local bridge MAC table and enables L2 bridged traffic across the network overlay. IP is imported into the local ARP table in an asymmetric IRB design OR imported into the IP routing table in a symmetric IRB design, and enables routed traffic across the layer 2 network overlay. Please refer to [EVPN-IRB] for more background on EVPN IRB forwarding modes.

To support EVPN mobility procedure, a single sequence number mobility attribute is advertised with the combined MAC+IP route. A single sequence number advertised with the combined MAC+IP route to resolve both MAC and IP reachability implicitly assumes a 1:1 fixed mapping between IP and MAC. While a fixed 1:1 mapping between IP and MAC is a common use case that could be addressed via existing MAC mobility procedure, additional IRB scenarios need to be considered, that don't necessarily adhere to this assumption. Following IRB mobility scenarios are considered:

- o VM move results in VM IP and MAC moving together
- o VM move results in VM IP moving to a new MAC association
- o VM move results in VM MAC moving to a new IP association

While existing MAC mobility procedure can be leveraged for MAC+IP move in the first scenario, subsequent scenarios result in a new MAC-IP association. As a result, a single sequence number assigned independently per-[MAC, IP] is not sufficient to determine most recent reachability for both MAC and IP, unless the sequence number assignment algorithm is designed to allow for changing MAC-IP bindings across moves.

Purpose of this draft is to define additional sequence number assignment and handling procedures to adequately address generic mobility support across EVPN-IRB overlay use cases that allow MAC-IP bindings to change across VM moves and can support mobility for both MAC and IP components carried in an EVPN RT-2 for these use cases.

In addition, for hosts on an ESI multi-homed to multiple GW devices, additional procedure is proposed to ensure synchronized sequence number assignments across the multi-homing devices.

Content presented in this draft is independent of data plane encapsulation used in the overlay being MPLS or NVO Tunnels. It is also largely independent of the EVPN IRB solution being based on symmetric OR asymmetric IRB design as defined in [EVPN-INTER-SUBNET].

In addition to symmetric and asymmetric IRB, mobility solution for a routed overlay, where traffic to an end host in the overlay is always IP routed using EVPN RT-5 is also presented in section 8.

To summarize, this draft covers mobility mobility for the following independent of the overlay encapsulation being MPLS or an NVO Tunnel:

- o Symmetric EVPN IRB overlay
- o Asymmetric EVPN IRB overlay
- o Routed EVPN overlay

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

- o EVPN-IRB: A BGP-EVPN distributed control plane based integrated routing and bridging fabric overlay discussed in [EVPN-IRB]
- o Underlay: IP or MPLS fabric core network that provides IP or MPLS routed reachability between EVPN PEs.
- o Overlay: VPN or service layer network consisting of EVPN PEs OR VPN provider-edge (PE) switch-router devices that runs on top of an underlay routed core.
- o EVPN PE: A PE switch-router in a data-center fabric that runs overlay BGP-EVPN control plane and connects to overlay CE host devices. An EVPN PE may also be the first-hop layer-3 gateway for CE/host devices. This document refers to EVPN PE as a logical function in a data-center fabric. This EVPN PE function may be physically hosted on a top-of-rack switching device (ToR) OR at layer(s) above the ToR in the Clos fabric. An EVPN PE is typically also an IP or MPLS tunnel end-point for overlay VPN flow
- o Symmetric EVPN-IRB: An overlay fabric first-hop routing architecture as defined in [EVPN-IRB], wherein, overlay host-to-host routed inter-subnet flows are routed at both ingress and egress EVPN PEs.
- o Asymmetric EVPN-IRB: An overlay fabric first-hop routing architecture as defined in [EVPN-IRB], wherein, overlay host-to-host routed inter-subnet flows are routed and bridged at ingress PE and bridged at egress PEs.
- o ARP: Address Resolution Protocol [RFC 826]. ARP references in this document are equally applicable to ND as well.
- o ND: IPv6 Neighbor Discovery Protocol [RFC 4861].
- o Ethernet-Segment: physical Ethernet or LAG port that connects an access device to an EVPN PE, as defined in [RFC 7432].

- o ESI: Ethernet Segment Identifier as defined in [RFC 7432].
- o LAG: Layer-2 link-aggregation, also known as layer-2 bundle port-channel, or bond interface.
- o EVPN all-active multi-homing: PE-CE all-active multi-homing achieved via a multi-homed layer-2 LAG interface on a CE with member links to multiple PEs and related EVPN procedures on the PEs.
- o RT-2: EVPN route type 2 carrying both MAC and IP reachability.
- o RT-5: EVPN route type 5 carrying IP prefix reachability.
- o MAC-IP: IP association for a MAC, referred to in this document may be IPv4, IPv6 or both.

2. Optional MAC only RT-2

In an EVPN IRB scenario, where a single MAC+IP RT-2 advertisement carries both IP and MAC routes, a MAC only RT-2 advertisement is redundant for host MACs that are advertised via MAC+IP RT-2. As a result, a MAC only RT-2 is an optional route that may not be advertised from or received at an EVPN PE. This is an important consideration for mobility scenarios discussed in subsequent sections.

MAC only RT-2 may still be advertised for non-IP host MACs that are not advertised via MAC+IP RT-2.

3. Mobility Use Cases

This section describes the IRB mobility use cases considered in this document. Procedures to address them are covered later in section 6 and section 7.

- o Host move results in Host IP and MAC moving together
- o Host move results in Host IP moving to a new MAC association
- o Host move results in Host MAC moving to a new IP association

3.1 Host MAC+IP Move

This is the baseline case, wherein a host move results in both host MAC and IP moving together with no change in MAC-IP binding across a move. Existing MAC mobility defined in RFC 7432 may be leveraged to apply to corresponding MAC+IP route to support this mobility scenario.

3.2 Host IP Move to new MAC

This is the case, where a host move results in VM IP moving to a new

MAC binding.

3.2.1 VM Reload

A host reload or an orchestrated host move that results in host being re-spawned at a new location may result in host getting a new MAC assignment, while maintaining existing IP address. This results in a host IP move to a new MAC binding:

IP-a, MAC-a ---> IP-a, MAC-b

3.2.2 MAC Sharing

This takes into account scenarios, where multiple hosts, each with a unique IP, may share a common MAC binding, and a host move results in a new MAC binding for the host IP.

As an example, hosts running on a single physical server, each with a unique IP, may share the same physical server MAC. In yet another scenario, an L2 access network may be behind a firewall, such that all hosts IPs on the access network are learnt with a common firewall MAC. In all such "shared MAC" use cases, multiple local MAC-IP ARP entries may be learnt with the same MAC. A host IP move, in such scenarios (for e.g., to a new physical server), could result in new MAC association for the host IP.

3.2.3 Problem

In both of the above scenarios, a combined MAC+IP EVPN RT-2 advertised with a single sequence number attribute implicitly assumes a fixed IP to MAC mapping. A host IP move to a new MAC breaks this assumption and results in a new MAC+IP route. If this new MAC+IP route is independently assigned a new sequence number, the sequence number can no longer be used to determine most recent host IP reachability in a symmetric EVPN-IRB design OR the most recent IP to MAC binding in an asymmetric EVPN-IRB design.

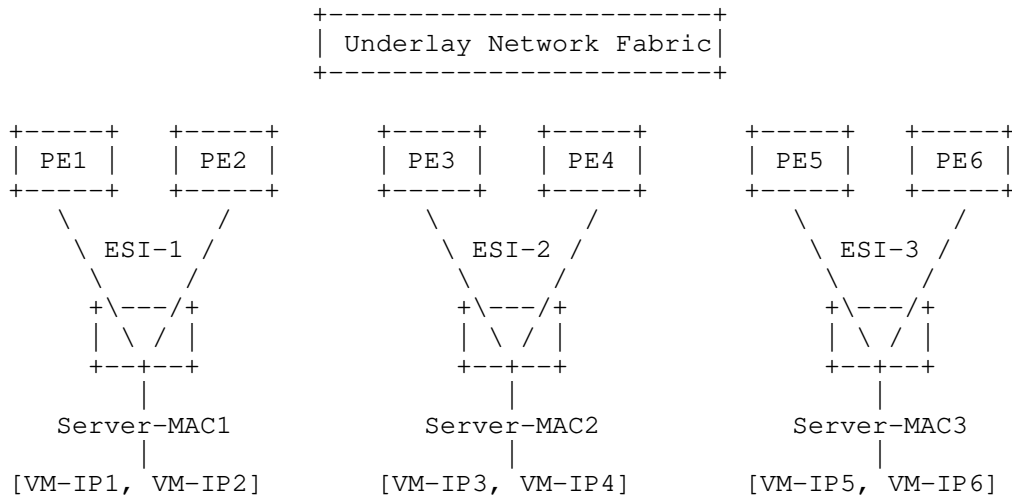


Figure 1

As an example, consider a topology shown in Figure 1, with host VMs sharing the physical server MAC. In steady state, [IP1, MAC1] route is learnt at [PE1, PE2] and advertised to remote PEs with a sequence number N. Now, VM-IP1 is moved to Server-MAC2. ARP or ND based local learning at [PE3, PE4] would now result in a new [IP1, MAC2] route being learnt. If route [IP1, MAC2] is learnt as a new MAC+IP route and assigned a new sequence number of say 0, mobility procedure for VM-IP1 will not trigger across the overlay network.

A sequence number assignment procedure needs to be defined to unambiguously determine the most recent IP reachability, IP to MAC binding, and MAC reachability for such a MAC sharing scenario.

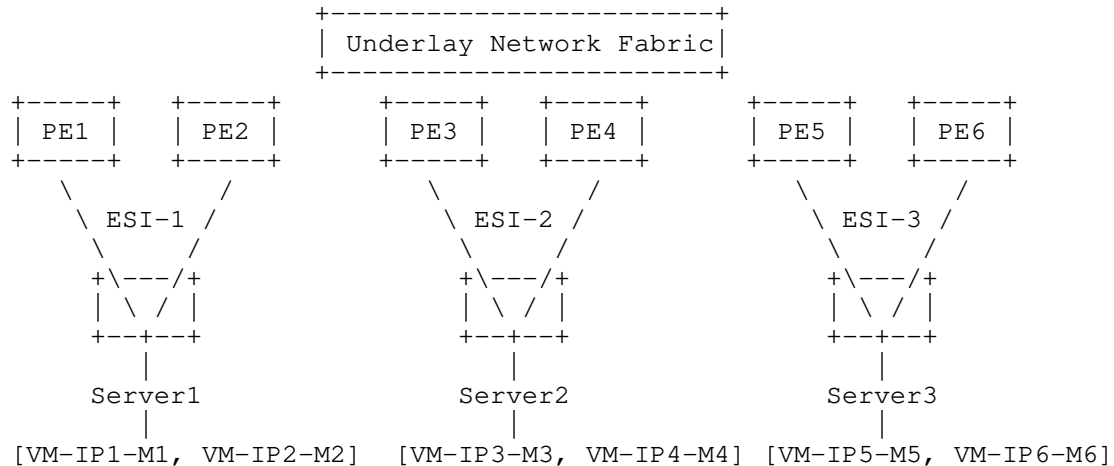
3.3 Host MAC move to new IP

This is a scenario where host move or re-provisioning behind a new gateway location may result in host getting a new IP address assigned, while keeping the same MAC.

3.3.1 Problem

Complication with this scenario is that MAC reachability could be carried via a combined MAC+IP route while a MAC only route may not be advertised at all. A single sequence number association with the MAC+IP route again implicitly assumes a fixed mapping between MAC and IP. A MAC move resulting in a new IP association for the host MAC breaks this assumption and results in a new MAC+IP route. If this new

MAC+IP route independently assumes a new sequence number, this mobility attribute can no longer be used to determine most recent host MAC reachability.



As an example, consider a host VM IP1-M1 that is learnt locally at [PE1, PE2] and advertised to remote hosts with a sequence number N. Consider a scenario where this VM with MAC M1 is re-provisioned at server 2, however, as part of this re-provisioning, assigned a different IP address say IP7. [IP7, M1] is learnt as a new route at [PE3, PE4] and advertised to remote PEs with a sequence number of 0. As a result, L3 reachability to IP7 would be established across the overlay, however, MAC mobility procedure for MAC1 will not trigger as a result of this MAC-IP route advertisement. If an optional MAC only route is also advertised, sequence number associated with the MAC only route would trigger MAC mobility as per [RFC7432]. However, in the absence of an additional MAC only route advertisement, a single sequence number advertised with a combined MAC+IP route would not be sufficient to update MAC reachability across the overlay.

A MAC-IP sequence number assignment procedure needs to be defined to unambiguously determine the most recent MAC reachability in such a scenario without a MAC only route being advertised.

Further, PE1/PE2, on learning new reachability for [IP7, M1] via PE3/PE4 MUST probe and delete any local IPs associated with MAC M1, such as [IP1, M1] in the above example.

Arguably, MAC mobility sequence number defined in [RFC7432], could be interpreted to apply only to the MAC part of MAC-IP route, and would

hence cover this scenario. It could hence be interpreted as a clarification to [RFC7432] and one of the considerations for a common sequence number assignment procedure across all MAC-IP mobility scenarios detailed in this document.

4. EVPN All Active multi-homed ES

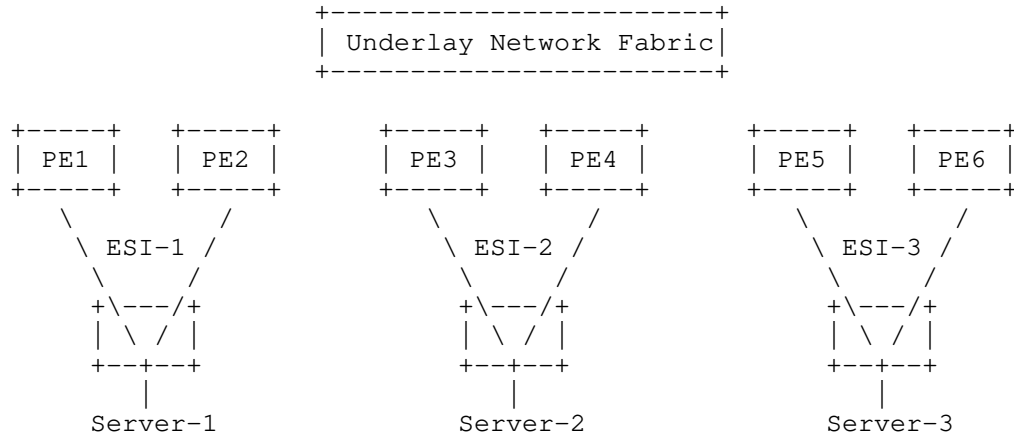


Figure 2

Consider an EVPN-IRB overlay network shown in Figure 2, with hosts multi-homed to two or more PE devices via an all-active multi-homed ES. MAC and ARP entries learnt on a local ES may also be synchronized across the multi-homing PE devices sharing this ES. This MAC and ARP SYNC enables local switching of intra and inter subnet ECMP traffic flows from remote hosts. In other words, local MAC and ARP entries on a given ES may be learnt via local learning and / or via sync from another PE device sharing the same ES.

For a host that is multi-homed to multiple PE devices via an all-active ES interface, local learning of host MAC and MAC-IP at each PE device is an independent asynchronous event, that is dependent on traffic flow and or ARP / ND response from the host hashing to a directly connected PE on the MC-LAG interface. As a result, sequence number mobility attribute value assigned to a locally learnt MAC or MAC-IP route at each device may not always be the same, depending on transient states on the device at the time of local learning.

As an example, consider a host VM that is deleted from ESI-2 and moved to ESI-1. It is possible for host to be learnt on say, PE1 following deletion of the remote route from [PE3, PE4], while being learnt on PE2 prior to deletion of remote route from [PE3, PE4]. If so, PE1 would process local host route learning as a new route and assign a sequence number of 0, while PE2 would process local host route learning as a remote to local move and assign a sequence number of N+1, N being the existing sequence number assigned at [PE3, PE4].

Inconsistent sequence numbers advertised from multi-homing devices introduces:

- o Ambiguity with respect to how the remote PEs should handle paths with same ESI and different sequence numbers. A remote PE may not program ECMP paths if it receives routes with different sequence numbers from a set of multi-homing PEs sharing the same ESI.
- o Breaks consistent route versioning across the network overlay that is needed for EVPN mobility procedures to work.

As an example, in this inconsistent state, PE2 would drop a remote route received for the same host with sequence number N (as its local sequence number is N+1), while PE1 would install it as the best route (as its local sequence number is 0).

There is need for a mechanism to ensure consistency of sequence numbers advertised from a set of multi-homing devices for EVPN mobility to work reliably.

In order to support mobility for multi-homed hosts using the sequence number mobility attribute, local MAC and MAC-IP routes learnt on a multi-homed ES MUST be advertised with the same sequence number by all PE devices that the ES is multi-homed to. There is need for a mechanism to ensure consistency of sequence numbers assigned across these PEs.

5. Design Considerations

To summarize, sequence number assignment scheme and implementation must take following considerations into account:

- o MAC+IP may be learnt on an ES multi-homed to multiple PE devices, hence requires sequence numbers to be synchronized across multi-homing PE devices.
- o MAC only RT-2 is optional in an IRB scenario and may not necessarily be advertised in addition to MAC+IP RT-2
- o Single MAC may be associated with multiple IPs, i.e., multiple host IPs may share a common MAC
- o Host IP move could result in host moving to a new MAC, resulting in a new IP to MAC association and a new MAC+IP route.
- o Host MAC move to a new location could result in host MAC being associated with a different IP address, resulting in a new MAC to

IP association and a new MAC+IP route

- o LOCAL MAC-IP learn via ARP would always accompanied by a LOCAL MAC learn event resulting from the ARP packet. MAC and MAC-IP learning, however, could happen in any order
- o Use cases discussed earlier that do not maintain a constant 1:1 MAC-IP mapping across moves could potentially be addressed by using separate sequence numbers associated with MAC and IP components of MAC+IP route. Maintaining two separate sequence numbers however adds significant overhead with respect to complexity, debugability, and backward compatibility. Hence, this document addresses these requirements via a single sequence number attribute.

6. Solution Components

This section goes over main components of the EVPN IRB mobility solution proposed in this draft. Later sections will go over exact sequence number assignment procedures resulting from concepts described in this section.

6.1 Sequence Number Inheritance

Main idea presented here is to view a LOCAL MAC-IP route as a child of the corresponding LOCAL MAC only route that inherits the sequence number attribute from the parent LOCAL MAC only route:

Mx-IPx -----> Mx (seq# = N)

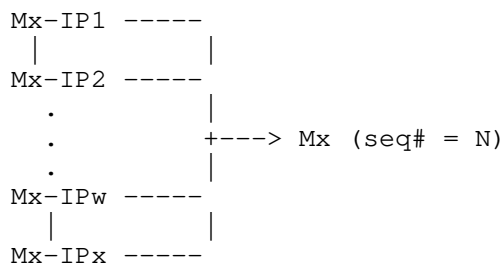
As a result, both parent MAC and child MAC-IP routes share one common sequence number associated with the parent MAC route. Doing so ensures that a single sequence number attribute carried in a combined MAC+IP route represents sequence number for both a MAC only route as well as a MAC+IP route, and hence makes the MAC only route truly optional. As a result, optional MAC only route with its own sequence number is not required to establish most recent reachability for a MAC in the overlay network. Specifically, this enables a MAC to assume a different IP address on a move, and still be able to establish most recent reachability to the MAC across the overlay network via mobility attribute associated with the MAC+IP route advertisement. As an example, when Mx moves to a new location, it would result in LOCAL Mx being assigned a higher sequence number at its new location as per RFC 7432. If this move results in Mx assuming a different IP address, IPz, LOCAL Mx+IPz route would inherit the new sequence number from Mx.

LOCAL MAC and LOCAL MAC-IP routes would typically be sourced from data plane learning and ARP learning respectively, and could get learnt in control plane in any order. Implementation could either replicate inherited sequence number in each MAC-IP entry OR maintain a single attribute in the parent MAC by creating a forward reference LOCAL MAC object for cases where a LOCAL MAC-IP is learnt before the LOCAL MAC.

Arguably, this inheritance may be assumed from RFC 7432, in which case, the above may be interpreted as a clarification with respect to interpretation of a MAC sequence number in a MAC-IP route.

6.2 MAC Sharing

Further, for the shared MAC scenario, this would result in multiple LOCAL MAC-IP siblings inheriting sequence number attribute from a common parent MAC route:



In such a case, a host-IP move to a different physical server would result in IP moving to a new MAC binding. A new MAC-IP route resulting from this move must now be advertised with a sequence number that is higher than the previous MAC-IP route for this IP, advertised from the prior location. As an example, consider a route Mx-IPx that is currently advertised with sequence number N from PE1. IPx moving to a new physical server behind PE2 results in IPx being associated with MAC Mz. A new local Mz-IPx route resulting from this move at PE2 must now be advertised with a sequence number higher than N. This is so that PE devices, including PE1, PE2, and other remote PE devices that are part of the overlay can clearly determine and program the most recent MAC binding and reachability for the IP. PE1, on receiving this new Mz-IPx route with sequence number say, N+1, for symmetric IRB case, would update IPx reachability via PE2 in forwarding, for asymmetric IRB case, would update IPx's ARP binding to Mz. In addition, PE1 would clear and withdraw the stale Mx-IPx route with the lower sequence number.

This also implies that sequence number associated with local MAC Mz

and all local MAC-IP children of Mz at PE2 must now be incremented to N+1, and re-advertised across the overlay. While this re-advertisement of all local MAC-IP children routes affected by the parent MAC route is an overhead, it avoids the need for two separate sequence number attributes to be maintained and advertised for IP and MAC components of MAC+IP RT-2. Implementation would need to be able to lookup MAC-IP routes for a given IP and update sequence number for its parent MAC and its MAC-IP children.

6.3 Multi-homing Mobility Synchronization

In order to support mobility for multi-homed hosts, local MAC and MAC-IP routes learnt on a shared ES MUST be advertised with the same sequence number by all PE devices that the ES is multi-homed to. This also applies to local MAC only routes. LOCAL MAC and MAC-IP may be learnt natively via data plane and ARP/ND respectively as well as via SYNC from another multi-homing PE to achieve local switching. Local and SYNC route learning can happen in any order. Local MAC-IP routes advertised by all multi-homing PE devices sharing the ES must carry the same sequence number, independent of the order in which they are learnt. This implies:

- o On local or sync MAC-IP route learning, sequence number for the local MAC-IP route MUST be compared and updated to the higher value.
- o On local or sync MAC route learning, sequence number for the local MAC route MUST be compared and updated to the higher value.

If an update to local MAC-IP sequence number is required as a result of above comparison with sync MAC-IP route, it would essentially amount to a sequence number update on the parent local MAC, resulting in inherited sequence number update on the MAC-IP route.

7. Requirements for Sequence Number Assignment

Following sections summarize sequence number assignment procedure needed on local and sync MAC and MAC-IP route learning events in order to accomplish the above.

7.1 LOCAL MAC-IP learning

A local Mx-IPx learning via ARP or ND should result in computation OR re-computation of parent MAC Mx's sequence number, following which the MAC-IP route Mx-IPx would simply inherit parent MAC's sequence number. Parent MAC Mx Sequence number should be computed as follows:

- o MUST be higher than any existing remote MAC route for Mx, as per RFC 7432.
- o MUST be at least equal to corresponding SYNC MAC sequence number if one is present.
- o If the IP is also associated with a different remote MAC "Mz", MUST be higher than "Mz" sequence number

Once new sequence number for MAC route Mx is computed as per above, all LOCAL MAC-IPs associated with MAC Mx MUST inherit the updated sequence number.

7.2 LOCAL MAC learning

Local MAC Mx Sequence number should be computed as follows:

- o MUST be higher than any existing remote MAC route for Mx, as per RFC 7432.
- o MUST be at least equal to corresponding SYNC MAC sequence number if one is present.
- o Once new sequence number for MAC route Mx is computed as per above, all LOCAL MAC-IPs associated with MAC Mx MUST inherit the updated sequence number.

Note that the local MAC sequence number might already be present if there was a local MAC-IP learnt prior to the local MAC, in which case the above may not result in any change in local MAC's sequence number.

7.3 Remote MAC OR MAC-IP Update

On receiving a remote MAC OR MAC-IP route update associated with a MAC Mx with a sequence number that is higher than or equal to sequence number assigned to a LOCAL route for MAC Mx:

- o PE MUST trigger probe and deletion procedure for all LOCAL IPs associated with MAC Mx
- o PE MUST trigger deletion procedure for LOCAL MAC route for Mx

7.4 REMOTE (SYNC) MAC update

Corresponding local MAC Mx (if present) sequence number should be re-computed as follows:

- o If the current sequence number is less than the received SYNC MAC sequence number, it MUST be increased to be equal to received SYNC MAC sequence number.
- o If a LOCAL MAC sequence number is updated as a result of the above, all LOCAL MAC-IPs associated with MAC Mx MUST inherit the updated sequence number.

7.5 REMOTE (SYNC) MAC-IP update

If this is a SYNCed MAC-IP on a local ES, it would also result in a derived SYNC MAC Mx route entry, as MAC only RT-2 advertisement is optional. Corresponding local MAC Mx (if present) sequence number should be re-computed as follows:

- o If the current sequence number is less than the received SYNC MAC sequence number, it MUST be increased to be equal to received SYNC MAC sequence number.
- o If a LOCAL MAC sequence number is updated as a result of the above, all LOCAL MAC-IPs associated with MAC Mx MUST inherit the updated sequence number.

7.6 Inter-op

In general, if all PE nodes in the overlay network follow the above sequence number assignment procedure, and the PE is advertising both MAC+IP and MAC routes, sequence number advertised with the MAC and MAC+IP routes with the same MAC would always be the same. However, an inter-op scenario with a different implementation could arise, where a PE implementation non-compliant with this document or with RFC 7432 assigns and advertises independent sequence numbers to MAC and MAC+IP routes. To handle this case, if different sequence numbers are received for remote MAC+IP and corresponding remote MAC routes from a remote PE, sequence number associated with the remote MAC route should be computed as:

- o Highest of the all received sequence numbers with remote MAC+IP and MAC routes with the same MAC.
- o MAC sequence number would be re-computed on a MAC or MAC+IP route withdraw as per above.

A MAC and / or IP move to the local PE would now result in the MAC (and hence all MAC-IP) sequence numbers incremented from the above computed remote MAC sequence number.

7.7 MAC Sharing Race Condition

In a MAC sharing use case described in section 6.2, a race condition is possible with simultaneous host moves between a pair of PEs. As an example, consider PE1 with local host IPs I1 and I2 sharing MAC M1, and PE2 with local host IPs I3 and I4 sharing MAC M2. A simultaneous move of I1 from PE1 to PE2 and of I3 from PE2 to PE1, such that I3 is learnt on PE1 before I1's local entry has been probed out on PE1 and/or I1 is learnt on PE2 before I3's local entry has been probed out on PE2 may trigger a race condition. This race condition together with MAC sequence number assignment rules defined in section 7.1 can cause new mac-ip routes [I1, M2] and [I3, M1] to bounce a couple of times with an incremented sequence number until stale entries [I1, M1] and [I3, M2] have been probed out from PE1 and PE2 respectively. An implementation MUST ensure proper probing procedures to remove stale ARP, ND, and local MAC entries, following a move, on learning remote routes as defined in section 7.3 (and as per [EVPN-IRB]) to minimize exposure to this race condition.

8. Routed Overlay

An additional use case is possible, such that traffic to an end host in the overlay is always IP routed. In a purely routed overlay such as this:

- o A host MAC is never advertised in EVPN overlay control plane
- o Host /32 or /128 IP reachability is distributed across the overlay via EVPN route type 5 (RT-5) along with a zero or non-zero ESI
- o An overlay IP subnet may still be stretched across the underlay fabric, however, intra-subnet traffic across the stretched overlay is never bridged
- o Both inter-subnet and intra-subnet traffic, in the overlay is IP routed at the EVPN PE.

Please refer to [RFC 7814] for more details.

Host mobility within the stretched subnet would still need to be supported for this use. In the absence of any host MAC routes, sequence number mobility EXT-COMM specified in [RFC7432], section 7.7 may be associated with a /32 OR /128 host IP prefix advertised via EVPN route type 5. MAC mobility procedures defined in RFC 7432 can now be applied as is to host IP prefixes:

- o On LOCAL learning of a host IP, on a new ESI, host IP MUST be

advertised with a sequence number attribute that is higher than what is currently advertised with the old ESI

- o on receiving a host IP route advertisement with a higher sequence number, a PE MUST trigger ARP/ND probe and deletion procedure on any LOCAL route for that IP with a lower sequence number. A PE would essentially move the forwarding entry to point to the remote route with a higher sequence number and send an ARP/ND PROBE for the local IP route. If the IP has indeed moved, PROBE would timeout and the local IP host route would be deleted.

Note that there is still only one sequence number associated with a host route at any time. For earlier use cases where a host MAC is advertised along with the host IP, a sequence number is only associated with a MAC. Only if the MAC is not advertised at all, as in this use case, is a sequence number associated with a host IP.

Note that this mobility procedure would not apply to "anycast IPv6" hosts advertised via NA messages with 0-bit=0. Please refer to [EVPN-PROXY-ARP].

9. Duplicate Host Detection

Duplicate host detection scenarios across EVPN IRB can be classified as follows:

- o Scenario A: where two hosts have the same MAC (host IPs may or may not be duplicate)
- o Scenario B: where two hosts have the same IP but different MACs
- o Scenario C: where two hosts have the same IP and host MAC is not advertised at all

Duplicate detection procedures for scenario B and C would not apply to "anycast IPv6" hosts advertised via NA messages with 0-bit=0. Please refer to [EVPN-PROXY-ARP].

9.1 Scenario A

For all use cases where duplicate hosts have the same MAC, MAC is detected as duplicate via duplicate MAC detection procedure described in RFC 7432. Corresponding MAC-IP routes with the same MAC do not require duplicate detection and MUST simply inherit the DUPLICATE property from the corresponding MAC route. In other words, if a MAC route is in DUPLICATE state, all corresponding MAC-IP routes MUST also be treated as DUPLICATE. Duplicate detection procedure need only be applied to MAC routes.

9.2 Scenario B

Due to misconfiguration, a situation may arise where hosts with different MACs are configured with the same IP. This scenario would not be detected by existing duplicate MAC detection procedure and would result in incorrect forwarding of routed traffic destined to this IP.

Such a situation, on LOCAL MAC-IP learning, would be detected as a move scenario via the following local MAC sequence number computation procedure described earlier in section 5.1:

- o If the IP is also associated with a different remote MAC "Mz", MUST be higher than "Mz" sequence number

Such a move that results in sequence number increment on local MAC because of a remote MAC-IP route associated with a different MAC MUST be counted as an "IP move" against the "IP" independent of MAC. Duplicate detection procedure described in RFC 7432 can now be applied to an "IP" entity independent of MAC. Once an IP is detected as DUPLICATE, corresponding MAC-IP route should be treated as DUPLICATE. Associated MAC routes and any other MAC-IP routes associated with this MAC should not be affected.

9.2.1 Duplicate IP Detection Procedure for Scenario B

Duplicate IP detection procedure for such a scenario is specified in [EVPN-PROXY-ARP]. What counts as an "IP move" in this scenario is further clarified as follows:

- o On learning a LOCAL MAC-IP route Mx-IPx, check if there is an existing REMOTE OR LOCAL route for IPx with a different MAC association, say, Mz-IPx. If so, count this as an "IP move" count for IPx, independent of the MAC
- o On learning a REMOTE MAC-IP route Mz-IPx, check if there is an existing LOCAL route for IPx with a different MAC association, say, Mx-IPx. If so, count this as an "IP move" count for IPx, independent of the MAC

A MAC-IP route SHOULD be treated as DUPLICATE if either of the following two conditions are met:

- o Corresponding MAC route is marked as DUPLICATE via existing duplicate detection procedure
- o Corresponding IP is marked as DUPLICATE via extended procedure described above

9.3 Scenario C

For a purely routed overlay scenario described in section 8, where only a host IP is advertised via EVPN RT-5, together with a sequence number mobility attribute, duplicate MAC detection procedures specified in RFC 7432 can be intuitively applied to IP only host routes for the purpose of duplicate IP detection.

- o On learning a LOCAL host IP route IPx, check if there is an existing REMOTE OR LOCAL route for IPx with a different ESI association. If so, count this as an "IP move" count for IPx.
- o On learning a REMOTE host IP route IPx, check if there is an existing LOCAL route for IPx with a different ESI association. If so, count this as an "IP move" count for IPx
- o With configurable parameters "N" and "M", If "N" IP moves are detected within "M" seconds for IPx, treat IPx as DUPLICATE

9.4 Duplicate Host Recovery

Once a MAC or IP is marked as DUPLICATE and FROZEN, corrective action must be taken to un-provision one of the duplicate MAC or IP. Un-provisioning a duplicate MAC or IP in this context refers to a corrective action taken on the host side. Once one of the duplicate MAC or IP is un-provisioned, normal operation would not resume until the duplicate MAC or IP ages out, following this correction, unless additional action is taken to speed up recovery.

This section lists possible additional corrective actions that could be taken to achieve faster recovery to normal operation.

9.4.1 Route Un-freezing Configuration

Unfreezing the DUPLICATE OR FROZEN MAC or IP via a CLI can be leveraged to recover from DUPLICATE and FROZEN state following corrective un-provisioning of the duplicate MAC or IP.

Unfreezing the frozen MAC or IP via a CLI at a PE should result in that MAC OR IP being advertised with a sequence number that is higher than the sequence number advertised from the other location of that MAC or IP.

Two possible corrective un-provisioning scenarios exist:

- o Scenario A: A duplicate MAC or IP may have been un-provisioned at the location where it was NOT marked as DUPLICATE and FROZEN

- o Scenario B: A duplicate MAC or IP may have been un-provisioned at the location where it was marked as DUPLICATE and FROZEN

Unfreezing the DUPLICATE and FROZEN MAC or IP, following the above corrective un-provisioning scenarios would result in recovery to steady state as follows:

- o Scenario A: If the duplicate MAC or IP was un-provisioned at the location where it was NOT marked as DUPLICATE, unfreezing the route at the FROZEN location will result in the route being advertised with a higher sequence number. This would in-turn result in automatic clearing of local route at the PE location, where the host was un-provisioned via ARP/ND PROBE and DELETE procedure specified earlier in section 8 and in [RFC 7432].
- o Scenario B: If the duplicate host is un-provisioned at the location where it was marked as DUPLICATE, unfreezing the route will trigger an advertisement with a higher sequence number to the other location. This would in-turn trigger re-learning of local route at the remote location, resulting in another advertisement with a higher sequence number from the remote location. Route at the local location would now be cleared on receiving this remote route advertisement, following the ARP/ND PROBE.

9.4.2 Route Clearing Configuration

In addition to the above, route clearing CLIs may also be leveraged to clear the local MAC or IP route, to be executed AFTER the duplicate host is un-provisioned:

- o clear mac CLI: A clear MAC CLI can be leveraged to clear a DUPLICATE MAC route, to recover from a duplicate MAC scenario
- o clear ARP/ND: A clear ARP/ND CLI may be leveraged to clear a DUPLICATE IP route to recover from a duplicate IP scenario

Note that the route unfreeze CLI may still need to be run if the route was un-provisioned and cleared from the NON-DUPLICATE / NON-FROZEN location. Given that unfreezing of the route via the un-freeze CLI would any ways result in auto-clearing of the route from the "un-provisioned" location, as explained in the prior section, need for a route clearing CLI for recovery from DUPLICATE / FROZEN state is truly optional.

10. Security Considerations

11. IANA Considerations

12. References

12.1 Normative References

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<http://www.rfc-editor.org/info/rfc7432>>.

[EVPN-PROXY-ARP] Rabadan et al., "Operational Aspects of Proxy-ARP/ND in EVPN Networks", draft-ietf-bess-evpn-proxy-arp-nd-06, work in progress, April 2019, <<https://tools.ietf.org/html/draft-ietf-bess-evpn-proxy-arp-nd-06>>.

[EVPN-IRB] Sajassi et al., "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-08, work in progress, March 2019, <<https://tools.ietf.org/html/draft-ietf-bess-evpn-inter-subnet-forwarding-08>>.

[RFC7814] Xu, X., Jacquenet, C., Raszuk, R., Boyes, T., Fee, B., "Virtual Subnet: A BGP/MPLS IP VPN-Based Subnet Extension Solution", RFC 7814, March 2016, <<https://tools.ietf.org/html/rfc7814>>.

12.2 Informative References

13. Acknowledgements

Authors would like to thank Vibov Bhan and Patrice Brisset for feedback the process of design and implementation of procedures defined in this document. Authors would like to thank Wen Lin for a detailed review and valuable comments related to MAC sharing race conditions.

Authors' Addresses

Neeraj Malhotra (Editor)
Arrcus
EMail: neeraj.ietf@gmail.com

Ali Sajassi
Cisco
EMail: sajassi@cisco.com

Aparna Pattekar
Cisco
Email: apjoshi@cisco.com

Jorge Rabadan
Nokia
Email: jorge.rabadan@nokia.com

Avinash Lingala
AT&T
Email: ar977m@att.com

John Drake
Juniper Networks
Email: jdrake@juniper.net

Appendix A

An alternative approach considered was to associate two independent sequence number attributes with MAC and IP components of a MAC-IP route. However, the approach of enabling IRB mobility procedures using a single sequence number associated with a MAC, as specified in this document was preferred for the following reasons:

- o Procedural overhead and complexity associated with maintaining two separate sequence numbers all the time, only to address scenarios with changing MAC-IP bindings is a big overhead for topologies where MAC-IP bindings never change.
- o Using a single sequence number associated with MAC is much simpler and adds no overhead for topologies where MAC-IP bindings never change.
- o Using a single sequence number associated with MAC is aligned with existing MAC mobility implementations. On other words, it is an easier implementation extension to existing MAC mobility procedure.

BESS Working Group
Internet-Draft
Intended Status: Proposed Standard

N. Malhotra, Ed.
Arrcus

A. Sajassi
S. Thoria
Cisco

J. Rabadan
Nokia

J. Drake
Juniper

A. Lingala
AT&T

Expires: Sept 26, 2019

March 25, 2019

Weighted Multi-Path Procedures for EVPN All-Active Multi-Homing
draft-ietf-bess-evpn-unequal-lb-01

Abstract

In an EVPN-IRB based network overlay, EVPN all-active multi-homing enables multi-homing for a CE device connected to two or more PEs via a LAG bundle, such that bridged and routed traffic from remote PEs can be equally load balanced (ECMPed) across the multi-homing PEs. This document defines extensions to EVPN procedures to optimally handle unequal access bandwidth distribution across a set of multi-homing PEs in order to:

- o provide greater flexibility, with respect to adding or removing individual PE-CE links within the access LAG
- o handle PE-CE LAG member link failures that can result in unequal PE-CE access bandwidth across a set of multi-homing PEs

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1	Introduction	4
1.1	PE CE Link Provisioning	5
1.2	PE CE Link Failures	6
1.3	Design Requirement	7
1.4	Terminology	7
2.	Solution Overview	8
3.	Weighted Unicast Traffic Load-balancing	8
3.1	LOCAL PE Behavior	8
3.1	Link Bandwidth Extended Community	8
3.2	REMOTE PE Behavior	9
4.	Weighted BUM Traffic Load-Sharing	10
4.1	The BW Capability in the DF Election Extended Community	10
4.2	BW Capability and Default DF Election algorithm	11
4.3	BW Capability and HRW DF Election algorithm (Type 1 and 4)	11
4.3.1	BW Increment	11
4.3.2	HRW Hash Computations with BW Increment	12

4.3.3 Cost-Benefit Tradeoff on Link Failures	13
4.4 BW Capability and Preference DF Election algorithm	14
5. Real-time Available Bandwidth	15
6. Routed EVPN Overlay	15
7. EVPN-IRB Multi-homing with non-EVPN routing	16
7. References	17
7.1 Normative References	17
7.2 Informative References	17
8. Acknowledgements	18
Authors' Addresses	18

1 Introduction

In an EVPN-IRB based network overlay, with a CE multi-homed via a EVPN all-active multi-homing, bridged and routed traffic from remote PEs can be equally load balanced (ECMPed) across the multi-homing PEs:

- o ECMP Load-balancing for bridged unicast traffic is enabled via aliasing and mass-withdraw procedures detailed in RFC 7432.
- o ECMP Load-balancing for routed unicast traffic is enabled via existing L3 ECMP mechanisms.
- o Load-sharing of bridged BUM traffic on local ports is enabled via EVPN DF election procedure detailed in RFC 7432

All of the above load-balancing and DF election procedures implicitly assume equal bandwidth distribution between the CE and the set of multi-homing PEs. Essentially, with this assumption of equal "access" bandwidth distribution across all PEs, ALL remote traffic is equally load balanced across the multi-homing PEs. This assumption of equal access bandwidth distribution can be restrictive with respect to adding / removing links in a multi-homed LAG interface and may also be easily broken on individual link failures. A solution to handle unequal access bandwidth distribution across a set of multi-homing EVPN PEs is proposed in this document. Primary motivation behind this proposal is to enable greater flexibility with respect to adding / removing member PE-CE links, as needed and to optimally handle PE-CE link failures.

1.1 PE CE Link Provisioning

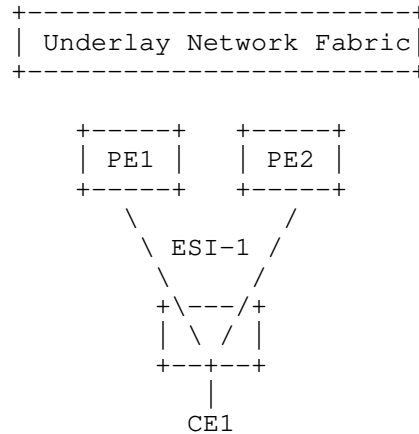


Figure 1

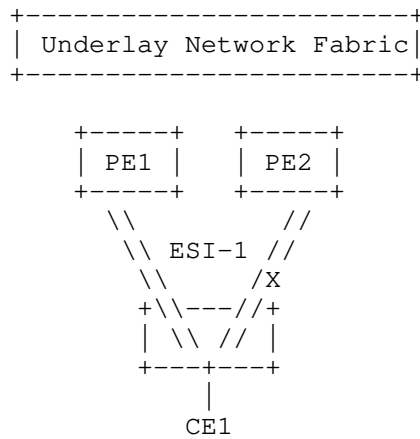
Consider a CE1 that is dual-homed to PE1 and PE2 via EVPN all-active multi-homing with single member links of equal bandwidth to each PE (aka, equal access bandwidth distribution across PE1 and PE2). If the provider wants to increase link bandwidth to CE1, it MUST add a link to both PE1 and PE2 in order to maintain equal access bandwidth distribution and inter-work with EVPN ECMP load-balancing. In other words, for a dual-homed CE, total number of CE links must be provisioned in multiples of 2 (2, 4, 6, and so on). For a triple-homed CE, number of CE links must be provisioned in multiples of three (3, 6, 9, and so on). To generalize, for a CE that is multi-homed to "n" PEs, number of PE-CE physical links provisioned must be an integral multiple of "n". This is restrictive in case of dual-homing and very quickly becomes prohibitive in case of multi-homing.

Instead, a provider may wish to increase PE-CE bandwidth OR number of links in ANY link increments. As an example, for CE1 dual-homed to PE1 and PE2 in all-active mode, provider may wish to add a third link to ONLY PE1 to increase total bandwidth for this CE by 50%, rather than being required to increase access bandwidth by 100% by adding a link to each of the two PEs. While existing EVPN based all-active load-balancing procedures do not necessarily preclude such asymmetric access bandwidth distribution among the PEs providing redundancy, it may result in unexpected traffic loss due to congestion in the access interface towards CE. This traffic loss is due to the fact that PE1 and PE2 will continue to attract equal amount of CE1 destined traffic from remote PEs, even when PE2 only has half the bandwidth to CE1 as PE1. This may lead to congestion and traffic loss on the PE2-CE1

link. If bandwidth distribution to CE1 across PE1 and PE2 is 2:1, traffic from remote hosts MUST also be load-balanced across PE1 and PE2 in 2:1 manner.

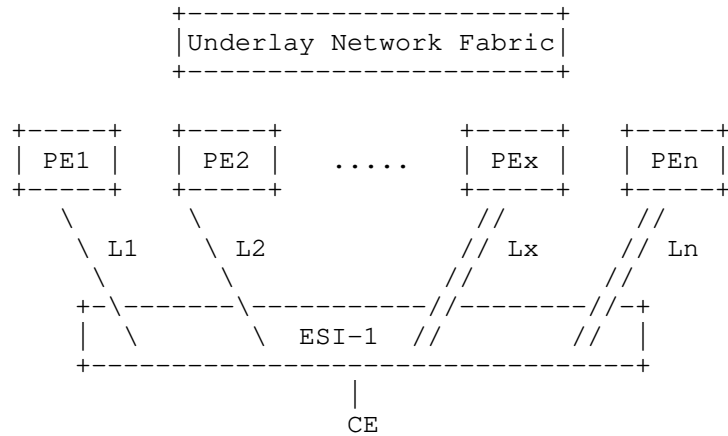
1.2 PE CE Link Failures

More importantly, unequal PE-CE bandwidth distribution described above may occur during regular operation following a link failure, even when PE-CE links were provisioned to provide equal bandwidth distribution across multi-homing PEs.



Consider a CE1 that is multi-homed to PE1 and PE2 via a link bundle with two member links to each PE. On a PE2-CE1 physical link failure, link bundle represented by an Ethernet Segment ESI-1 on PE2 stays up, however, it's bandwidth is cut in half. With existing ECMP procedures, both PE1 and PE2 will continue to attract equal amount of traffic from remote PEs, even when PE1 has double the bandwidth to CE1. If bandwidth distribution to CE1 across PE1 and PE2 is 2:1, traffic from remote hosts MUST also be load-balanced across PE1 and PE2 in 2:1 manner to avoid unexpected congestion and traffic loss on PE2-CE1 links within the LAG.

1.3 Design Requirement



To generalize, if total link bandwidth to a CE is distributed across "n" multi-homing PEs, with Lx being the number of links / bandwidth to PEx, traffic from remote PEs to this CE MUST be load-balanced unequally across [PE1, PE2,, PEn] such that, fraction of total unicast and BUM flows destined for CE that are serviced by PEx is:

$$Lx / [L1+L2+.....+Ln]$$

Solution proposed below includes extensions to EVPN procedures to achieve the above.

1.4 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

"LOCAL PE" in the context of an ESI refers to a provider edge switch OR router that physically hosts the ESI.

"REMOTE PE" in the context of an ESI refers to a provider edge switch OR router in an EVPN overlay, who's overlay reachability to the ESI is via the LOCAL PE.

2. Solution Overview

In order to achieve weighted load balancing for overlay unicast traffic, Ethernet A-D per-ES route (EVPN Route Type 1) is leveraged to signal the Ethernet Segment bandwidth to remote PEs. Using Ethernet A-D per-ES route to signal the Ethernet Segment bandwidth provides a mechanism to be able to react to changes in access bandwidth in a service and host independent manner. Remote PEs computing the MAC path-lists based on global and aliasing Ethernet A-D routes now have the ability to setup weighted load-balancing path-lists based on the ESI access bandwidth received from each PE that the ESI is multi-homed to. If Ethernet A-D per-ES route is also leveraged for IP path-list computation, as per [EVPN-IP-ALIASING], it also provides a method to do weighted load-balancing for IP routed traffic.

In order to achieve weighted load-balancing of overlay BUM traffic, EVPN ES route (Route Type 4) is leveraged to signal the ESI bandwidth to PEs within an ESI's redundancy group to influence per-service DF election. PEs in an ESI redundancy group now have the ability to do service carving in proportion to each PE's relative ESI bandwidth.

Procedures to accomplish this are described in greater detail next.

3. Weighted Unicast Traffic Load-balancing

3.1 LOCAL PE Behavior

A PE that is part of an Ethernet Segment's redundancy group would advertise a additional "link bandwidth" EXT-COMM attribute with Ethernet A-D per-ES route (EVPN Route Type 1), that represents total bandwidth of PE's physical links in an Ethernet Segment. BGP link bandwidth EXT-COMM defined in [BGP-LINK-BW] is re-used for this purpose.

3.1 Link Bandwidth Extended Community

Link bandwidth extended community described in [BGP-LINK-BW] for layer 3 VPNs is re-used here to signal local ES link bandwidth to remote PEs. link-bandwidth extended community is however defined in [BGP-LINK-BW] as optional non-transitive. In inter-AS scenarios, link-bandwidth may need to be signaled to an eBGP neighbor along with next-hop unchanged. It is work in progress with authors of [BGP-LINK-BW] to allow for this attribute to be used as transitive in inter-AS scenarios.

3.2 REMOTE PE Behavior

A receiving PE should use per-ES link bandwidth attribute received from each PE to compute a relative weight for each remote PE, per-ES, as shown below.

if,

$L(x,y)$: link bandwidth advertised by PE-x for ESI-y

$W(x,y)$: normalized weight assigned to PE-x for ESI-y

$H(y)$: Highest Common Factor (HCF) of $[L(1,y), L(2,y), \dots, L(n,y)]$

then, the normalized weight assigned to PE-x for ESI-y may be computed as follows:

$$W(x,y) = L(x,y) / H(y)$$

For a MAC+IP route (EVPN Route Type 2) received with ESI-y, receiving PE MUST compute MAC and IP forwarding path-list weighted by the above normalized weights.

As an example, for a CE dual-homed to PE-1, PE-2, PE-3 via 2, 1, and 1 GE physical links respectively, as part of a link bundle represented by ESI-10:

$$L(1, 10) = 2000 \text{ Mbps}$$

$$L(2, 10) = 1000 \text{ Mbps}$$

$$L(3, 10) = 1000 \text{ Mbps}$$

$$H(10) = 1000$$

Normalized weights assigned to each PE for ESI-10 are as follows:

$$W(1, 10) = 2000 / 1000 = 2.$$

$$W(2, 10) = 1000 / 1000 = 1.$$

$$W(3, 10) = 1000 / 1000 = 1.$$

For a remote MAC+IP host route received with ESI-10, forwarding load-balancing path-list must now be computed as: [PE-1, PE-1, PE-2, PE-3] instead of [PE-1, PE-2, PE-3]. This now results in load-balancing of all traffic destined for ESI-10 across the three multi-homing PEs in

proportion to ESI-10 bandwidth at each PE.

Above weighted path-list computation MUST only be done for an ESI, IF a link bandwidth attribute is received from ALL of the PE's advertising reachability to that ESI via Ethernet A-D per-ES Route Type 1. In the event that link bandwidth attribute is not received from one or more PEs, forwarding path-list would be computed using regular ECMP semantics.

4. Weighted BUM Traffic Load-Sharing

Optionally, load sharing of per-service DF role, weighted by individual PE's link-bandwidth share within a multi-homed ES may also be achieved.

In order to do that, a new DF Election Capability [EVPN-DF-ELECT-FRAMEWORK] called "BW" (Bandwidth Weighted DF Election) is defined. BW may be used along with some DF Election Types, as described in the following sections.

4.1 The BW Capability in the DF Election Extended Community

[EVPN-DF-ELECT-FRAMEWORK] defines a new extended community for PEs within a redundancy group to signal and agree on uniform DF Election Type and Capabilities for each ES. This document requests a bit in the DF Election extended community Bitmap:

Bit 28: BW (Bandwidth Weighted DF Election)

ES routes advertised with the BW bit set will indicate the desire of the advertising PE to consider the link-bandwidth in the DF Election algorithm defined by the value in the "DF Type".

As per [EVPN-DF-ELECT-FRAMEWORK], all the PEs in the ES MUST advertise the same Capabilities and DF Type, otherwise the PEs will fall back to Default [RFC7432] DF Election procedure.

The BW Capability MAY be advertised with the following DF Types:

- o Type 0: Default DF Election algorithm, as in [RFC7432]
- o Type 1: HRW algorithm, as in [EVPN-DF-ELECT-FRAMEWORK]
- o Type 2: Preference algorithm, as in [EVPN-DF-PREF]
- o Type 4: HRW per-multicast flow DF Election, as in [EVPN-PER-MCAST-FLOW-DF]

The following sections describe how the DF Election procedures are modified for the above DF Types when the BW Capability is used.

4.2 BW Capability and Default DF Election algorithm

When all the PEs in the Ethernet Segment (ES) agree to use the BW Capability with DF Type 0, the Default DF Election procedure is modified as follows:

- o Each PE advertises a "Link Bandwidth" EXT-COMM attribute along with the ES route to signal the PE-CE link bandwidth (LBW) for the ES.
- o A receiving PE MUST use the ES link bandwidth attribute received from each PE to compute a relative weight for each remote PE.
- o The DF Election procedure MUST now use this weighted list of PEs to compute the per-VLAN Designated Forwarder, such that the DF role is distributed in proportion to this normalized weight.

Considering the same example as in Section 3, the candidate PE list for DF election is:

[PE-1, PE-1, PE-2, PE-3].

The DF for a given VLAN-a on ES-10 is now computed as $(\text{VLAN-a} \% 4)$. This would result in the DF role being distributed across PE1, PE2, and PE3 in portion to each PE's normalized weight for ES-10.

4.3 BW Capability and HRW DF Election algorithm (Type 1 and 4)

[EVPN-DF-ELECT-FRAMEWORK] introduces Highest Random Weight (HRW) algorithm (DF Type 1) for DF election in order to solve potential DF election skew depending on Ethernet tag space distribution. [EVPN-PER-MCAST-FLOW-DF] further extends HRW algorithm for per-multicast flow based hash computations (DF Type 4). This section describes extensions to HRW Algorithm for EVPN DF Election specified in [EVPN-DF-ELECT-FRAMEWORK] and in [EVPN-PER-MCAST-FLOW-DF] in order to achieve DF election distribution that is weighted by link bandwidth.

4.3.1 BW Increment

A new variable called "bandwidth increment" is computed for each [PE, ES] advertising the ES link bandwidth attribute as follows:

In the context of an ES,

$L(i)$ = Link bandwidth advertised by PE(i) for this ES

$L(\text{min})$ = lowest link bandwidth advertised across all PEs for this ES

Bandwidth increment, " $b(i)$ " for a given PE(i) advertising a link

bandwidth of $L(i)$ is defined as an integer value computed as:

$$b(i) = L(i) / L(\min)$$

As an example,

with $PE(1) = 10$, $PE(2) = 10$, $PE(3) = 20$

bandwidth increment for each PE would be computed as:

$$b(1) = 1, b(2) = 1, b(3) = 2$$

with $PE(1) = 10$, $PE(2) = 10$, $PE(3) = 10$

bandwidth increment for each PE would be computed as:

$$b(1) = 1, b(2) = 1, b(3) = 1$$

Note that the bandwidth increment must always be an integer, including, in an unlikely scenario of a PE's link bandwidth not being an exact multiple of $L(\min)$. If it computes to a non-integer value (including as a result of link failure), it MUST be rounded down to an integer.

4.3.2 HRW Hash Computations with BW Increment

HRW algorithm as described in [EVPN-DF-ELECT-FRAMEWORK] and in [EVPN-PER-MCAST-FLOW-DF] compute a random hash value (referred to as affinity here) for each $PE(i)$, where, $(0 < i \leq N)$, $PE(i)$ is the PE at ordinal i , and $Address(i)$ is the IP address of PE at ordinal i .

For ' N ' PEs sharing an Ethernet segment, this results in ' N ' candidate hash computations. PE that has the highest hash value is selected as the DF.

Affinity computation for each $PE(i)$ is extended to be computed one per-bandwidth increment associated with $PE(i)$ instead of a single affinity computation per $PE(i)$.

$PE(i)$ with $b(i) = j$, results in j affinity computations:

$affinity(i, x)$, where $1 < x \leq j$

This essentially results in number of candidate HRW hash computations for each PE that is directly proportional to that PE's relative bandwidth within an ES and hence gives $PE(i)$ a probability of being DF in proportion to it's relative bandwidth within an ES.

As an example, consider an ES that is multi-homed to two PEs, PE1 and PE2, with equal bandwidth distribution across PE1 and PE2. This would result in a total of two candidate hash computations:

```
affinity(PE1, 1)
```

```
affinity(PE2, 1)
```

Now, consider a scenario with PE1's link bandwidth as 2x that of PE2. This would result in a total of three candidate hash computations to be used for DF election:

```
affinity(PE1, 1)
```

```
affinity(PE1, 2)
```

```
affinity(PE2, 1)
```

which would give PE1 2/3 probability of getting elected as a DF, in proportion to its relative bandwidth in the ES.

Depending on the chosen HRW hash function, affinity function MUST be extended to include bandwidth increment in the computation.

For e.g.,

affinity function specified in [EVPN-PER-MCAST-FLOW-DF] MAY be extended as follows to incorporate bandwidth increment j:

```
affinity(S,G,V, ESI, Address(i,j)) =
(1103515245.((1103515245.Address(i).j + 12345) XOR
D(S,G,V,ESI))+12345) (mod 2^31)
```

affinity or random function specified in [EVPN-DF-ELECT-FRAMEWORK] MAY be extended as follows to incorporate bandwidth increment j:

```
affinity(v, Es, Address(i,j)) = (1103515245((1103515245.Address(i).j
+ 12345) XOR D(v,Es))+12345) (mod 2^31)
```

4.3.3 Cost-Benefit Tradeoff on Link Failures

While incorporating link bandwidth into the DF election process provides optimal BUM traffic distribution across the ES links, it also implies that affinity values for a given PE are re-computed, and DF elections are re-adjusted on changes to that PE's bandwidth increment that might result from link failures or link additions. If the operator does not wish to have this level of churn in their DF

election, then they should not advertise the BW capability. Not advertising BW capability may result in less than optimal BUM traffic distribution while still retaining the ability to allow a remote ingress PE to do weighted ECMP for its unicast traffic to a set of multi-homed PEs, as described in section 3.2.

Same also applies to use of BW capability with service carving (DF Type 0), as specified in section 4.2.

4.4 BW Capability and Preference DF Election algorithm

This section applies to ES'es where all the PEs in the ES agree use the BW Capability with DF Type 2. The BW Capability modifies the Preference DF Election procedure [EVPN-DF-PREF], by adding the LBW value as a tie-breaker as follows:

- o Section 4.1, bullet (f) in [EVPN-DF-PREF] now considers the LBW value:
 - f) In case of equal Preference in two or more PEs in the ES, the tie-breakers will be the DP bit, the LBW value and the lowest IP PE in that order. For instance:
 - o If vES1 parameters were [Pref=500,DP=0,LBW=1000] in PE1 and [Pref=500,DP=1, LBW=2000] in PE2, PE2 would be elected due to the DP bit.
 - o If vES1 parameters were [Pref=500,DP=0,LBW=1000] in PE1 and [Pref=500,DP=0, LBW=2000] in PE2, PE2 would be elected due to a higher LBW, even if PE1's IP address is lower.
 - o The LBW exchanged value has no impact on the Non-Revertive option described in [EVPN-DF-PREF].

5. Real-time Available Bandwidth

PE-CE link bandwidth availability may sometimes vary in real-time disproportionately across PE-CE links within a multi-homed ESI due to various factors such as flow based hashing combined with fat flows and unbalanced hashing. Reacting to real-time available bandwidth is at this time outside the scope of this document. Procedures described in this document are strictly based on static link bandwidth parameter.

6. Routed EVPN Overlay

An additional use case is possible, such that traffic to an end host in the overlay is always IP routed. In a purely routed overlay such as this:

- o A host MAC is never advertised in EVPN overlay control plane o Host /32 or /128 IP reachability is distributed across the overlay via EVPN route type 5 (RT-5) along with a zero or non-zero ESI
- o An overlay IP subnet may still be stretched across the underlay fabric, however, intra-subnet traffic across the stretched overlay is never bridged
- o Both inter-subnet and intra-subnet traffic, in the overlay is IP routed at the EVPN GW.

Please refer to [RFC 7814] for more details.

Weighted multi-path procedure described in this document may be used together with procedures described in [EVPN-IP-ALIASING] for this use case. Ethernet A-D per-ES route advertised with Layer 3 VRF RTs would be used to signal ES link bandwidth attribute instead of the Ethernet A-D per-ES route with Layer 2 VRF RTs. All other procedures described earlier in this document would apply as is.

If [EVPN-IP-ALIASING] is not used for routed fast convergence, link bandwidth attribute may still be advertised with IP routes (RT-5) to achieve PE-CE link bandwidth based load-balancing as described in this document. In the absence of [EVPN-IP-ALIASING], re-balancing of traffic following changes in PE-CE link bandwidth will require all IP routes from that CE to be re-advertised in a prefix dependent manner.

7. EVPN-IRB Multi-homing with non-EVPN routing

EVPN-LAG based multi-homing on an IRB gateway may also be deployed together with non-EVPN routing, such as global routing or an L3VPN routing control plane. Key property that differentiates this set of use cases from EVPN IRB use cases discussed earlier is that EVPN control plane is used only to enable LAG interface based multi-homing and NOT as an overlay VPN control plane. EVPN control plane in this case enables:

- o DF election via EVPN RT-4 based procedures described in [RFC7432]
- o LOCAL MAC sync across multi-homing PEs via EVPN RT-2
- o LOCAL ARP and ND sync across multi-homing PEs via EVPN RT-2

Applicability of weighted ECMP procedures proposed in this document to these set of use cases will be addressed in subsequent revisions.

7. References

7.1 Normative References

- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<http://www.rfc-editor.org/info/rfc7432>>.
- [BGP-LINK-BW] Mohapatra, P., Fernando, R., "BGP Link Bandwidth Extended Community", January 2013, <<https://tools.ietf.org/html/draft-ietf-idr-link-bandwidth-06>>.
- [EVPN-IP-ALIASING] Sajassi, A., Badoni, G., "L3 Aliasing and Mass Withdrawal Support for EVPN", July 2017, <<https://tools.ietf.org/html/draft-sajassi-bess-evpn-ip-aliasing-00>>.
- [EVPN-DF-PREF] Rabadan, J., Sathappan, S., Przygienda, T., Lin, W., Drake, J., Sajassi, A., and S. Mohanty, "Preference-based EVPN DF Election", internet-draft ietf-bess-evpn-pref-df-01.txt, April 2018.
- [EVPN-PER-MCAST-FLOW-DF] Sajassi, et al., "Per multicast flow Designated Forwarder Election for EVPN", March 2018, <<https://tools.ietf.org/html/draft-sajassi-bess-evpn-per-mcast-flow-df-election-00>>.
- [EVPN-DF-ELECT-FRAMEWORK] Rabadan, Mohanty, et al., "Framework for EVPN Designated Forwarder Election Extensibility", March 2018, <<https://tools.ietf.org/html/draft-ietf-bess-evpn-df-election-framework-03>>.
- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", March 1997, <<https://tools.ietf.org/html/rfc2119>>.
- [RFC8174] B. Leiba, "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", May 2017, <<https://tools.ietf.org/html/rfc8174>>.

7.2 Informative References

8. Acknowledgements

Authors would like to thank Satya Mohanty for valuable review and inputs with respect to HRW algorithm refinements proposed in this document.

Authors' Addresses

Neeraj Malhotra, Editor.
Arrcus
Email: neeraj.ietf@gmail.com

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Jorge Rabadan
Nokia
Email: jorge.rabadan@nokia.com

John Drake
Juniper
EMail: jdrake@juniper.net

Avinash Lingala
AT&T
Email: ar977m@att.com

Samir Thoria
Cisco
Email: sthoria@cisco.com

BESS Working Group
Internet Draft
Intended Status: Proposed Standard
Expires: September 12, 2019

P. Brissette Ed.
Cisco System
H. Shah Ed.
Ciena Corporation
I. Chen Ed.
Jabil
I. Hussain Ed.
Infinera Corporation
K. Tiruveedhula Ed.
Juniper Networks
J. Rabadan Ed.
Nokia

March 11, 2019

Yang Data Model for EVPN
draft-ietf-bess-evpn-yang-07

Abstract

This document describes a YANG data model for Ethernet VPN services. The model is agnostic of the underlay. It apply to MPLS as well as to VxLAN encapsulation. The model is also agnostic of the services including E-LAN, E-LINE and E-TREE services. This document mainly focuses on EVPN and Ethernet-Segment instance framework.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Convention

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Table of Contents

1. Introduction	2
2. Specification of Requirements	3
3. EVPN YANG Model	4
3.1. Overview	4
3.2 Ethernet-Segment Model	4
3.3 EVPN Model	5
4. YANG Module	8
4.1 Ethernet Segment Yang Module	9
4.2 EVPN Yang Module	15
5. Security Considerations	26
6. IANA Considerations	26
7. References	26
7.1. Normative References	26
7.2. Informative References	27
Authors' Addresses	27

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is a network management protocol that defines mechanisms to manage network devices. YANG [RFC6020] is a modular language that represents data structures in an XML or JSON tree format, and is used as a data modeling language for the NETCONF.

This document introduces a YANG data model for Ethernet VPN services (EVPN) [RFC7432], Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN) [RFC7623] as well as other WG draft such as EVPN-VPWS, etc. The EVPN services runs over MPLS and VxLAN underlay.

The Yang data model in this document defines Ethernet VPN based services. The model leverages the definitions used in other IETF Yang draft such as L2VPN Yang.

The goal is to propose a data object model consisting of building blocks that can be assembled in different order to realize different EVPN-based services. The definition work is undertaken initially by a smaller working group with members representing various vendors and service providers. The EVPN basic framework consist of two modules: EVPN and Ethernet-Segment. These models are completely orthogonal. They usually work in pair but user can definitely use one or the other for its own need.

The data model is defined for following constructs that are used for managing the services:

- o Configuration
- o Operational State
- o Notifications

The document is organized to first define the data model for the configuration, operational state and notifications of EVPN and Ethernet-Segment.

The EVPN data object model defined in this document uses the instance centric approach whereby EVPN service attributes are specified for a given EVPN instance.

The Ethernet-Segment data object model defined in this document refer to a specific interface. That interface can be a physical interface, a bundle interface or virtual interface. The latter includes attachment-circuit and pseudowire. The purpose of creating a separate module is due to the fact that it can be used without having the need to have EVPN configured as layer 2/3 service. For example, an access node can be dual-homed to two service nodes servicing a VPLS or an IPVPN core. The access connectivity can be represented by an Ethernet-Segment where EVPN BGP DF election is performed over both service nodes.

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL

NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. EVPN YANG Model

3.1. Overview

Two top level module, Ethernet-Segment and EVPN, are defined. The Ethernet-Segment contains a list of interface to which any Ethernet-Segment attributes are configured/applied.

The EVPN module has two main containers: common and instance. The first one has common attributes to all VPNs where as the latter has attributes specific to an EVI (EVPN instance). This document state the scope of the EVPN object models definition. The following documents are within the scope. This is not an exhaustive list but a representation of documents that are covered for this work:

- o Reqs for EVPN:[RFC7209]
- o EVPN: [RFC7432]
- o PBB-EVPN: [RFC7623]
- o EVPN-VPWS: [RFC8214]
- o EVPN-ETREE: [RFC8317]
- o EVPN Overlay [RFC8365]

The integration with L2VPN instance Yang model is being done as part of the L2VPN Yang model.

Following documents will be covered at that time:

- o (PBB-)EVPN Seamless Integration with (PBB-)VPLS:
draft-ietf-bess-evpn-vpls-seamless-integ
- o EVPN Virtual Ethernet Segment:
draft-sajassi-bess-evpn-virtual-eth-segment
- o IP Prefix Advertisement in EVPN:
draft-ietf-bess-evpn-prefix-advertisement
- o VXLAN DCI Using EVPN:
draft-boutros-l2vpn-vxlan-evpn
- o Interconnect Solution for EVPN Overlay networks:
draft-ietf-bess-dci-evpn-overlay
- o Integrated Routing and Bridging in EVPN:
draft-ietf-bess-evpn-inter-subnet-forwarding

3.2 Ethernet-Segment Model

The Ethernet-Segment data model has a list of ES where each refer to an interface. All attributes are optional due to auto-sensing default mode where all values are auto-derive from the network connectivity.

module: ietf-ethernet-segment


```

+--rw ethernet-segments
  +--rw ethernet-segment* [name]
    +--rw name string
    +--ro service-type? string
    +--ro status? status-type
    +--rw (ac-or-pw)?
      | +--:(ac)
      | | +--rw ac* if:interface-ref
      | +--:(pw)
      | | +--rw pw* pw:pseudowire-ref
    +--ro interface-status? status-type
    +--rw ethernet-segment-identifier? ethernet-segment-identifier-ty
  +--rw (active-mode)
    | +--:(single-active)
    | | +--rw single-active-mode? empty
    | +--:(all-active)
    | | +--rw all-active-mode? empty
  +--rw pbb-parameters {ethernet-segment-pbb-params}?
  | +--rw backbone-src-mac? yang:mac-address
  +--rw bgp-parameters
    +--rw common
      +--rw rd-rt* [route-distinguisher]
        {ethernet-segment-bgp-params}?
      +--rw route-distinguisher
        rt-types:route-distinguisher
      +--rw vpn-targets
        rt-types:vpn-route-targets
  +--rw df-election
    +--rw df-election-method? df-election-method-type
    +--rw preference? uint16
    +--rw revertive? boolean
    +--rw election-wait-time? uint32
  +--rw ead-evi-route? boolean
  +--ro esi-label? string
  +--ro member*
    | +--ro ip-address? inet:ip-address
  +--ro df*
    +--ro service-identifier? uint32
    +--ro vlan? uint32
    +--ro ip-address? inet:ip-address

```

3.3 EVPN Model

The evpn-instances container contains a list of evpn-instance. Each entry of the evpn-instance represents a different Ethernet VPN and it is represented by a EVI. Again, mainly all attributes are optional for the same reason as for the Ethernet-Segment module.

```

module: ietf-evpn
  +--rw evpn
    +--rw common
      +--rw (replication-type)?
        +--:(ingress-replication)
          | +--rw ingress-replication?   boolean
        +--:(p2mp-replication)
          | +--rw p2mp-replication?      boolean
    +--rw evpn-instances
      +--rw evpn-instance* [name]
        +--rw name                               string
        +--rw evi?                               uint32
        +--rw pbb-parameters {evpn-pbb-params}?
          | +--rw source-bmac?   yang:mac-address
        +--rw bgp-parameters
          +--rw common
            +--rw rd-rt* [route-distinguisher]
                      {evpn-bgp-params}?
            +--rw route-distinguisher
              | rt-types:route-distinguisher
            +--rw vpn-targets
              | rt-types:vpn-route-targets
        +--rw arp-proxy?                          boolean
        +--rw arp-suppression?                     boolean
        +--rw nd-proxy?                           boolean
        +--rw nd-suppression?                      boolean
        +--rw underlay-multicast?                  boolean
        +--rw flood-unknown-unicast-supression?   boolean
        +--rw vpws-vlan-aware?                    boolean
        +--ro routes
          +--ro ethernet-auto-discovery-route*
            | +--ro rd-rt* [route-distinguisher]
            | | +--ro route-distinguisher
            | | | rt-types:route-distinguisher
            | | +--ro vpn-targets
            | | | rt-types:vpn-route-targets
            | +--ro ethernet-segment-identifier? es:ethernet-segment-i
dentifier-type
          +--ro ethernet-tag?                       uint32
          +--ro path*
            +--ro next-hop?   inet:ip-address
            +--ro label?      rt-types:mpls-label
            +--ro detail
              +--ro attributes
                | +--ro extended-community*   string
                +--ro bestpath?               empty
          +--ro mac-ip-advertisement-route*
            | +--ro rd-rt* [route-distinguisher]
            | | +--ro route-distinguisher

```

identfier-type	<pre> rt-types:route-distinguisher +--ro vpn-targets rt-types:vpn-route-targets +--ro ethernet-segment-identifier? es:ethernet-segment-i </pre>
	<pre> +--ro ethernet-tag? uint32 +--ro mac-address? yang:mac-address +--ro mac-address-length? uint8 +--ro ip-prefix? inet:ip-prefix +--ro path* +--ro next-hop? inet:ip-address +--ro label? rt-types:mpls-label +--ro label2? rt-types:mpls-label +--ro detail +--ro attributes +--ro extended-community* string +--ro bestpath? empty +--ro inclusive-multicast-ethernet-tag-route* +--ro rd-rt* [route-distinguisher] +--ro route-distinguisher rt-types:route-distinguisher +--ro vpn-targets rt-types:vpn-route-targets +--ro originator-ip-prefix? inet:ip-prefix +--ro path* +--ro next-hop? inet:ip-address +--ro label? rt-types:mpls-label +--ro detail +--ro attributes +--ro extended-community* string +--ro bestpath? empty +--ro ethernet-segment-route* +--ro rd-rt* [route-distinguisher] +--ro route-distinguisher rt-types:route-distinguisher +--ro vpn-targets rt-types:vpn-route-targets +--ro ethernet-segment-identifier? es:ethernet-segment-i </pre>
identfier-type	<pre> +--ro originator-ip-prefix? inet:ip-prefix +--ro path* +--ro next-hop? inet:ip-address +--ro detail +--ro attributes +--ro extended-community* string +--ro bestpath? empty +--ro ip-prefix-route* +--ro rd-rt* [route-distinguisher] +--ro route-distinguisher rt-types:route-distinguisher </pre>

```

    |
    |   +--ro vpn-targets
    |   |   rt-types:vpn-route-targets
    +--ro ethernet-segment-identifier?
    |   es:ethernet-segment-identifier-type
    +--ro ip-prefix?                       inet:ip-prefix
    +--ro path*
    |   +--ro next-hop?   inet:ip-address
    |   +--ro label?      rt-types:mpls-label
    |   +--ro detail
    |   |   +--ro attributes
    |   |   |   +--ro extended-community*   string
    |   |   +--ro bestpath?                 empty
    +--ro statistics
    |   +--ro tx-count?   yang:zero-based-counter32
    |   +--ro rx-count?   yang:zero-based-counter32
    |   +--ro detail
    |   |   +--ro broadcast-tx-count?
    |   |   |   yang:zero-based-counter32
    |   |   +--ro broadcast-rx-count?
    |   |   |   yang:zero-based-counter32
    |   |   +--ro multicast-tx-count?
    |   |   |   yang:zero-based-counter32
    |   |   +--ro multicast-rx-count?
    |   |   |   yang:zero-based-counter32
    |   |   +--ro unknown-unicast-tx-count?
    |   |   |   yang:zero-based-counter32
    |   |   +--ro unknown-unicast-rx-count?
    |   |   |   yang:zero-based-counter32
    augment /pw:pseudowires/pw:pseudowire/pw:pw-type:
    +--:(evpn-pw)
    |   +--rw evpn-pw
    |   |   +--rw remote-id?   uint32
    |   |   +--rw local-id?    uint32
    augment
    /ni:network-instances/ni:network-instance/ni:ni-type/l2vpn:l2vpn:
    |   +--rw evpn-instance?   evpn-instance-ref
    augment
    /ni:network-instances/ni:network-instance/ni:ni-type/l2vpn:l2vpn:
    |   +--rw vpls-contstraints

notifications:
    +---n evpn-state-change-notification
    |   +--ro evpn-instance?   evpn-instance-ref
    |   +--ro state?           identityref

```

4. YANG Module

The EVPN configuration container is logically divided into

following high level configuration areas:

4.1 Ethernet Segment Yang Module

```
<CODE BEGINS> file "ietf-ethernet-segment@2019-03-09.yang"
module iETF-ethernet-segment {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ethernet-segment";
  prefix "es";

  import iETF-yang-types {
    prefix "yang";
  }

  import iETF-inet-types {
    prefix "inet";
  }

  import iETF-routing-types {
    prefix "rt-types";
  }

  import iETF-interfaces {
    prefix "if";
  }

  import iETF-pseudowires {
    prefix "pw";
  }

  organization "ietf";
  contact "ietf";
  description "ethernet segment";

  revision "2019-03-09" {
    description " - Create an ethernet-segment type and change references " +
      " to ethernet-segment-identifier " +
      " - Updated Route-target lists to rt-types:vpn-route-targets
" +
      ";
    reference " ";
  }
  revision "2018-02-20" {
    description " - Change the type of attachment circuit to " +
      " if:interface-ref " +
      ";
    reference " ";
  }

  revision "2017-10-21" {
```

```
description " - Updated ethernet segment's AC/PW members to " +
"   accommodate more than one AC or more than one " +
"   PW " +
" - Added the new preference based DF election " +
"   method " +
" - Referenced pseudowires in the new " +
"   ietf-pseudowires.yang model " +
" - Moved model to NMDA style specified in " +
"   draft-dsdt-nmda-guidelines-01.txt " +
"";
reference   "";
}

revision "2017-03-08" {
  description " - Updated to use BGP parameters from " +
"   ietf-routing-types.yang instead of from " +
"   ietf-evpn.yang " +
" - Updated ethernet segment's AC/PW members to " +
"   accommodate more than one AC or more than one " +
"   PW " +
" - Added the new preference based DF election " +
"   method " +
"";
  reference   "";
}

revision "2016-07-08" {
  description " - Added the configuration option to enable or " +
"   disable per-EVI/EAD route " +
" - Added PBB parameter backbone-src-mac " +
" - Added operational state branch, initially " +
"   to match the configuration branch" +
"";
  reference   "";
}

revision "2016-06-23" {
  description "WG document adoption";
  reference   "";
}

revision "2015-10-15" {
  description "Initial revision";
  reference   "";
}

/* Features */
```

```
feature ethernet-segment-bgp-params {
  description "Ethernet segment's BGP parameters";
}

feature ethernet-segment-pbb-params {
  description "Ethernet segment's PBB parameters";
}

/* Typedefs */
typedef status-type {
  type enumeration {
    enum up {
      description "Status is up";
    }
    enum down {
      description "Status is down";
    }
  }
  description "status type";
}

typedef df-election-method-type {
  type enumeration {
    enum default {
      value 0;
      description "The default DF election method";
    }
    enum highest-random-weight {
      value 1;
      description "The highest random weight (HRW) method";
      reference "draft-mohanty-bess-evpn-df-election";
    }
    enum preference {
      value 2;
      description "The preference based method";
      reference "draft-rabadan-bess-evpn-pref-df";
    }
  }
  description "The DF election method type";
}

typedef ethernet-segment-identifier-type {
  type yang:hex-string {
    length "29";
  }
  description "10-octet Ethernet segment identifier (esi),
    ex: 00:5a:5a:5a:5a:5a:5a:5a:5a:5a";
}
```

```
/* EVPN Ethernet Segment YANG Model */

container ethernet-segments {
  description "ethernet-segment";
  list ethernet-segment {
    key "name";
    leaf name {
      type string;
      description "Name of the ethernet segment";
    }
    leaf service-type {
      type string;
      config false;
      description "service-type";
    }
    leaf status {
      type status-type;
      config false;
      description "Ethernet segment status";
    }
    choice ac-or-pw {
      description "ac-or-pw";
      case ac {
        leaf-list ac {
          type if:interface-ref;
          description "Name of attachment circuit";
        }
      }
      case pw {
        leaf-list pw {
          type pw:pseudowire-ref;
          description "Reference to a pseudowire";
        }
      }
    }
    leaf interface-status {
      type status-type;
      config false;
      description "interface status";
    }
    leaf ethernet-segment-identifier {
      type ethernet-segment-identifier-type;
      description "Ethernet segment identifier (esi)";
    }
    choice active-mode {
      mandatory true;
      description "Choice of active mode";
      case single-active {
```



```
        leaf single-active-mode {
            type empty;
            description "single-active-mode";
        }
    }
    case all-active {
        leaf all-active-mode {
            type empty;
            description "all-active-mode";
        }
    }
}
container pbb-parameters {
    if-feature ethernet-segment-pbb-params;
    description "PBB configuration";
    leaf backbone-src-mac {
        type yang:mac-address;
        description "backbone-src-mac, only if this is a PBB";
    }
}
container bgp-parameters {
    description "BGP parameters";
    container common {
        description "BGP parameters common to all pseudowires";
        list rd-rt {
            if-feature ethernet-segment-bgp-params;
            key "route-distinguisher";
            leaf route-distinguisher {
                type rt-types:route-distinguisher;
                description "Route distinguisher";
            }
            uses rt-types:vpn-route-targets;
            description "A list of route distinguishers and " +
                "corresponding VPN route targets";
        }
    }
}
container df-election {
    description "df-election";
    leaf df-election-method {
        type df-election-method-type;
        description "The DF election method";
    }
    leaf preference {
        when "../df-election-method = 'preference'" {
            description "The preference value is only applicable " +
                "to the preference based method";
        }
    }
}
```

```
        type uint16;
        description "The DF preference";
    }
    leaf revertive {
        when "../df-election-method = 'preference'" {
            description "The revertive value is only applicable " +
                "to the preference method";
        }
        type boolean;
        default true;
        description "The 'preempt' or 'revertive' behavior";
    }
    leaf election-wait-time {
        type uint32;
        description "election-wait-time";
    }
}
leaf ead-evi-route {
    type boolean;
    default false;
    description "Enable (true) or disable (false) ead-evi-route";
}
leaf esi-label {
    type rt-types:mpls-label;
    config false;
    description "esi-label";
}
list member {
    config false;
    leaf ip-address {
        type inet:ip-address;
        description "ip-address";
    }
    description "member of the ethernet segment";
}
list df {
    config false;
    leaf service-identifier {
        type uint32;
        description "service-identifier";
    }
    leaf vlan {
        type uint32;
        description "vlan";
    }
    leaf ip-address {
        type inet:ip-address;
        description "ip-address";
    }
}
```

```
    }
    description "df of an evpn instance's vlan";
  }
  description "An ethernet segment";
}
}
}
}
<CODE ENDS>
```

4.2 EVPN Yang Module

```
<CODE BEGINS> file "ietf-evpn@2019-03-09.yang"
module ietf-evpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-evpn";
  prefix "evpn";

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-routing-types {
    prefix "rt-types";
  }

  import ietf-network-instance {
    prefix "ni";
  }

  import ietf-l2vpn {
    prefix "l2vpn";
  }

  import ietf-pseudowires {
    prefix "pw";
  }

  import ietf-ethernet-segment {
    prefix "es";
  }

  organization "ietf";
  contact "ietf";
```

```
description    "evpn";

revision "2019-03-09" {
  description " - Incorporated ietf-ethernet-segment model and" +
    "    normalised ethernet-segment entries on routes " +
    " - Updated Route-target lists to rt-types:vpn-route-targets" +
  " +
    ";
  reference   " ";
}

revision "2018-02-20" {
  description " - Incorporated ietf-network-instance model" +
    "    on which ietf-l2vpn is now based " +
    ";
  reference   " ";
}

revision "2017-10-21" {
  description " - Modified the operational state augment " +
    " - Renamed evpn-instances-state to evpn-instances" +
    " - Added vpws-vlan-aware to an EVPN instance " +
    " - Added a new augment to L2VPN to add EPVN " +
    " - pseudowire for the case of EVPN VPWS " +
    " - Added state change notification " +
    ";
  reference   " ";
}

revision "2017-03-13" {
  description " - Added an augment to base L2VPN model to " +
    "    reference an EVPN instance " +
    " - Reused ietf-routing-types.yang " +
    "    vpn-route-targets grouping instead of " +
    "    defining it in this module " +
    ";
  reference   " ";
}

revision "2016-07-08" {
  description " - Added operational state" +
    " - Added a configuration knob to enable/disable " +
    "    underlay-multicast " +
    " - Added a configuration knob to enable/disable " +
    "    flooding of unknown unicast " +
    " - Added several configuration knobs " +
    "    to manage ARP and ND " +
    ";
  reference   " ";
}
```

```
}

revision "2016-06-23" {
  description "WG document adoption";
  reference   "";
}

revision "2015-10-15" {
  description "Initial revision";
  reference   "";
}

feature evpn-bgp-params {
  description "EVPN's BGP parameters";
}

feature evpn-pbb-params {
  description "EVPN's PBB parameters";
}

/* Identities */

identity evpn-notification-state {
  description "The base identity on which EVPN notification " +
              "states are based";
}

identity MAC-duplication-detected {
  base "evpn-notification-state";
  description "MAC duplication is detected";
}

identity mass-withdraw-received {
  base "evpn-notification-state";
  description "Mass withdraw received";
}

identity static-MAC-move-detected {
  base "evpn-notification-state";
  description "Static MAC move is detected";
}

/* Typedefs */

typedef evpn-instance-ref {
  type leafref {
    path "/evpn/evpn-instances/evpn-instance/name";
  }
}
```

```
    description "A leafref type to an EVPN instance";
  }

/* Groupings */

grouping route-rd-rt-grp {
  description "A grouping for a route's route distinguishers " +
    "and route targets";
  list rd-rt {
    key "route-distinguisher";
    leaf route-distinguisher {
      type rt-types:route-distinguisher;
      description "Route distinguisher";
    }
    list vpn-target {
      key "route-target";
      leaf route-target {
        type rt-types:route-target;
        description "BGP route target";
      }
    }
    description "A list of route targets";
  }
  description "A list of route distinguishers and " +
    "corresponding VPN route targets";
}

grouping next-hop-label-grp {
  description "next-hop-label-grp";
  leaf next-hop {
    type inet:ip-address;
    description "next-hop";
  }
  leaf label {
    type rt-types:mpls-label;
    description "label";
  }
}

grouping next-hop-label2-grp {
  description "next-hop-label2-grp";
  leaf label2 {
    type rt-types:mpls-label;
    description "label2";
  }
}

grouping path-detail-grp {
```

```
description "path-detail-grp";
container detail {
  config false;
  description "path details";
  container attributes {
    leaf-list extended-community {
      type string;
      description "extended-community";
    }
    description "attributes";
  }
  leaf bestpath {
    type empty;
    description "Indicate this path is the best path";
  }
}
}

/* EVPN YANG Model */

container evpn {
  description "evpn";
  container common {
    description "common evpn attributes";
    choice replication-type {
      description "A choice of replication type";
      case ingress-replication {
        leaf ingress-replication {
          type boolean;
          description "ingress-replication";
        }
      }
      case p2mp-replication {
        leaf p2mp-replication {
          type boolean;
          description "p2mp-replication";
        }
      }
    }
  }
}

container evpn-instances {
  description "evpn-instances";
  list evpn-instance {
    key "name";
    description "An EVPN instance";
    leaf name {
      type string;
      description "Name of EVPN instance";
    }
  }
}
```

```
    }
    leaf evi {
        type uint32;
        description "evi";
    }
    container pbb-parameters {
        if-feature "evpn-pbb-params";
        description "PBB parameters";
        leaf source-bmac {
            type yang:hex-string;
            description "source-bmac";
        }
    }
    container bgp-parameters {
        description "BGP parameters";
        container common {
            description "BGP parameters common to all pseudowires";
            list rd-rt {
                if-feature evpn-bgp-params;
                key "route-distinguisher";
                leaf route-distinguisher {
                    type rt-types:route-distinguisher;
                    description "Route distinguisher";
                }
                uses rt-types:vpn-route-targets;
                description "A list of route distinguishers and " +
                    "corresponding VPN route targets";
            }
        }
    }
    leaf arp-proxy {
        type boolean;
        default false;
        description "Enable (TRUE) or disable (FALSE) ARP proxy";
    }
    leaf arp-suppression {
        type boolean;
        default false;
        description "Enable (TRUE) or disable (FALSE) " +
            "ARP suppression";
    }
    leaf nd-proxy {
        type boolean;
        default false;
        description "Enable (TRUE) or disable (FALSE) ND proxy";
    }
    leaf nd-suppression {
        type boolean;
```



```
    default false;
    description "Enable (TRUE) or disable (FALSE) " +
        "ND suppression";
}
leaf underlay-multicast {
    type boolean;
    default false;
    description "Enable (TRUE) or disable (FALSE) " +
        "underlay multicast";
}
leaf flood-unknown-unicast-supression {
    type boolean;
    default false;
    description "Enable (TRUE) or disable (FALSE) " +
        "flood unknown unicast suppression";
}
leaf vpws-vlan-aware {
    type boolean;
    default false;
    description "Enable (TRUE) or disable (FALSE) " +
        "VPWS VLAN aware";
}
container routes {
    config false;
    description "routes";
    list ethernet-auto-discovery-route {
        uses route-rd-rt-grp;
        leaf ethernet-segment-identifier {
            type es:ethernet-segment-identifier-type;
            description "Ethernet segment identifier (esi)";
        }
        leaf ethernet-tag {
            type uint32;
            description "An ethernet tag (etag) indentifying a " +
                "broadcast domain";
        }
        list path {
            uses next-hop-label-grp;
            uses path-detail-grp;
            description "path";
        }
        description "ethernet-auto-discovery-route";
    }
    list mac-ip-advertisement-route {
        uses route-rd-rt-grp;
        leaf ethernet-segment-identifier {
            type es:ethernet-segment-identifier-type;
            description "Ethernet segment identifier (esi)";
        }
    }
}
```

```
    }
    leaf ethernet-tag {
        type uint32;
        description "An ethernet tag (etag) indentifying a " +
            "broadcast domain";
    }
    leaf mac-address {
        type yang:mac-address;
        description "Route mac address";
    }
    leaf mac-address-length {
        type uint8 {
            range "0..48";
        }
        description "mac address length";
    }
    leaf ip-prefix {
        type inet:ip-prefix;
        description "ip-prefix";
    }
    list path {
        uses next-hop-label-grp;
        uses next-hop-label2-grp;
        uses path-detail-grp;
        description "path";
    }
    description "mac-ip-advertisement-route";
}
list inclusive-multicast-ethernet-tag-route {
    uses route-rd-rt-grp;
    leaf originator-ip-prefix {
        type inet:ip-prefix;
        description "originator-ip-prefix";
    }
    list path {
        uses next-hop-label-grp;
        uses path-detail-grp;
        description "path";
    }
    description "inclusive-multicast-ethernet-tag-route";
}
list ethernet-segment-route {
    uses route-rd-rt-grp;
    leaf ethernet-segment-identifier {
        type es:ethernet-segment-identifier-type;
        description "Ethernet segment identifier (esi)";
    }
    leaf originator-ip-prefix {
```

```
        type inet:ip-prefix;
        description "originator ip-prefix";
    }
    list path {
        leaf next-hop {
            type inet:ip-address;
            description "next-hop";
        }
        uses path-detail-grp;
        description "path";
    }
    description "ethernet-segment-route";
}
list ip-prefix-route {
    uses route-rd-rt-grp;
    leaf ethernet-segment-identifier {
        type es:ethernet-segment-identifier-type;
        description "Ethernet segment identifier (esi)";
    }
    leaf ip-prefix {
        type inet:ip-prefix;
        description "ip-prefix";
    }
    list path {
        uses next-hop-label-grp;
        uses path-detail-grp;
        description "path";
    }
    description "ip-prefix route";
}
}
container statistics {
    config false;
    description "Statistics";
    leaf tx-count {
        type yang:zero-based-counter32;
        description "transmission count";
    }
    leaf rx-count {
        type yang:zero-based-counter32;
        description "receive count";
    }
}
container detail {
    description "Detailed statistics";
    leaf broadcast-tx-count {
        type yang:zero-based-counter32;
        description "broadcast transmission count";
    }
}
```

```
    leaf broadcast-rx-count {
      type yang:zero-based-counter32;
      description "broadcast receive count";
    }
    leaf multicast-tx-count {
      type yang:zero-based-counter32;
      description "multicast transmission count";
    }
    leaf multicast-rx-count {
      type yang:zero-based-counter32;
      description "multicast receive count";
    }
    leaf unknown-unicast-tx-count {
      type yang:zero-based-counter32;
      description "unknown unicast transmission count";
    }
    leaf unknown-unicast-rx-count {
      type yang:zero-based-counter32;
      description "unknown-unicast receive count";
    }
  }
}
}
}
}

/* augments */

augment "/pw:pseudowires/pw:pseudowire/pw:pw-type" {
  description "Augment for an L2VPN instance to add EVPN VPWS " +
    "pseudowire";
  case evpn-pw {
    container evpn-pw {
      description "EVPN pseudowire";
      leaf remote-id {
        type uint32;
        description "Remote pseudowire ID";
      }
      leaf local-id {
        type uint32;
        description "Local pseudowire ID";
      }
    }
  }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
  "/l2vpn:l2vpn" {
```

```

    description "Augment for an L2VPN instance and EVPN association";
    leaf evpn-instance {
        type evpn-instance-ref;
        description "Reference to an EVPN instance";
    }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
    "/l2vpn:l2vpn" {
    when "l2vpn:type = 'l2vpn:vpls-instance-type'" {
        description "Constraints only for VPLS pseudowires";
    }
    description "Augment for VPLS instance";
    container vpls-contstraints {
        must "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:pw/l2vpn:name]" +
            "    /evpn-pw/local-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:primary-pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:primary-pw/l2vpn:name]" +
            "    /evpn-pw/local-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:backup-pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:backup-pw/l2vpn:name]" +
            "    /evpn-pw/local-id))" {
            description "A VPLS pseudowire must not be EVPN PW";
        }
        description "VPLS constraints";
    }
}

/* Notifications */

notification evpn-state-change-notification {
    description "EVPN state change notification";
}

```

```
    leaf evpn-instance {
      type evpn-instance-ref;
      description "Related EVPN instance";
    }
    leaf state {
      type identityref {
        base evpn-notification-state;
      }
      description "State change notification";
    }
  }
}
<CODE ENDS>
```

5. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

The security concerns listed above are, however, no different than faced by other routing protocols. Hence, this draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg]

6. IANA Considerations

None.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294,

DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.

7.2. Informative References

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC7209] Sajassi, A., Aggarwal, R., Uttaro, J., Bitar, N., Henderickx, W., and A. Isaac, "Requirements for Ethernet VPN (EVPN)", RFC 7209, DOI 10.17487/RFC7209, May 2014, <<https://www.rfc-editor.org/info/rfc7209>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.
- [RFC7623] Sajassi, A., Ed., Salam, S., Bitar, N., Isaac, A., and W. Henderickx, "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, DOI 10.17487/RFC7623, September 2015, <<https://www.rfc-editor.org/info/rfc7623>>.
- [RFC8214] Boutros, S., Sajassi, A., Salam, S., Drake, J., and J. Rabadan, "Virtual Private Wire Service Support in Ethernet VPN", RFC 8214, DOI 10.17487/RFC8214, August 2017, <<https://www.rfc-editor.org/info/rfc8214>>.

Authors' Addresses

Patrice Brissette
Cisco Systems, Inc.
EMail: pbrisset@cisco.com

Himanshu Shah
Ciena Corporation
EMail: hshah@ciena.com

Helen Chen
Jabil
EMail: Ing-Wher_Chen@jabil.com

Iftekar Hussain
Infinera Corporation
EMail: ihussain@infinera.com

Kishore Tiruveedhula
Juniper Networks
EMail: kishoret@juniper.net

Jorge Rabadan
Nokia
EMail: jorge.rabadan@nokia.com

Ali Sajassi
Cisco Systems, Inc.
EMail: sajassi@cisco.com

Zhenbin Li
Huawei Technologies
EMail: lizhenbin@huawei.com

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2020

H. Shah, Ed.
Ciena Corporation
P. Brissette, Ed.
Cisco Systems, Inc.
I. Chen, Ed.
The MITRE Corporation
I. Hussain, Ed.
Infinera Corporation
B. Wen, Ed.
Comcast
K. Tiruveedhula, Ed.
Juniper Networks
July 02, 2019

YANG Data Model for MPLS-based L2VPN
draft-ietf-bess-l2vpn-yang-10.txt

Abstract

This document describes a YANG data model for Layer 2 VPN (L2VPN) services over MPLS networks. These services include point-to-point Virtual Private Wire Service (VPWS) and multipoint Virtual Private LAN service (VPLS) that uses LDP and BGP signaled Pseudowires. It is expected that this model will be used by the management tools run by the network operators in order to manage and monitor the network resources that they use to deliver L2VPN services.

This document also describes the YANG data model for the Pseudowires. The independent definition of the Pseudowires facilitates its use in Ethernet Segment and EVPN data models defined in separate document.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 3, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. Specification of Requirements	4
3. L2VPN YANG Model	4
3.1. Overview	4
3.2. Latest addition	7
3.3. Open issues and next steps	8
3.4. Pseudowire Common	8
3.4.1. Pseudowire	8
3.4.2. pw-templates	8
3.5. L2VPN Common	8
3.5.1. redundancy-group-templates	8
3.6. L2VPN instance	9
3.6.1. common attributes	9
3.6.2. PW list	9
3.6.3. List of endpoints	9
3.6.4. point-to-point or multipoint service	10
3.6.5. multi-segment pseudowire	11
3.7. Operational State	11
3.8. Yang tree	11
4. YANG Module	14
5. Security Considerations	43
6. IANA Considerations	43
7. Acknowledgments	43
8. References	44
8.1. Normative References	44
8.2. Informative References	44
Appendix A. Example Configuration	47
Appendix B. Contributors	47
Authors' Addresses	48

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is a network management protocol that defines mechanisms to manage network devices. YANG [RFC7950] is a modular language that represents data structures in an XML or JSON tree format, and is used as a data modeling language for the NETCONF.

This document defines a YANG data model for MPLS based Layer 2 VPN services (L2VPN) [RFC4664] and includes switching between the local attachment circuits. The L2VPN model covers point-to-point VPWS and Multipoint VPLS services. These services use signaling of Pseudowires across MPLS networks using LDP [RFC8077][RFC4762] or BGP[RFC4761].

The data model covers Ethernet based Layer 2 services. The Ethernet Attachment Circuits are not defined. Instead, they are leveraged from other standards organizations such as IEEE802.1 and Metro Ethernet Forum (MEF).

Other Layer 2 services, such as ATM, Frame Relay, TDM, etc are included in the scope but will be covered as the future work items.

The objective of the model is to define building blocks that can easily be assembled in different order to realize different services.

The data model uses following constructs for configuration and management:

- o Configuration
- o Operational State
- o Executables (Actions)
- o Notifications

This document focuses on definition of configuration, state and notification objects.

The L2VPN data object model uses the instance centric approach. The L2VPN instance is recognized by network instance model. The network-instance container is defined in network instance model [I-D.ietf-netmod-ni-model].

Within this network instance, L2VPN container contains definitions of a set of common parameters, a list of PWs and a list of endpoints. A

special constraint is added for the VPWS configuration such that only two endpoints are allowed in the list of endpoints.

The Pseudowire data object model is defined independent of the L2VPN data object model to allow its inclusion in the Ethernet Segment and EVPN data objects.

The L2VPN data object model augments Psuedowire data object for its definition.

The document also includes Notifications used by the L2VPN object model

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. L2VPN YANG Model

3.1. Overview

The document defines configuration of one single container for L2VPN. Within the l2vpn container, common parameters and a list of endpoints are defined. For the point-to-point VPWS configuration, endpoint list is used with the constraint that limits the number of endpoints to be two. For the multipoint service, endpoint list is used. Each endpoint contains the common definition that is either an attachment circuit, a pseudowire or a redundancy group. The previous versions of this document represented VPWS service with definition of endpoint-a and endpoint-z while VPLS with a list of endpoints. This duplication is removed with simplified version whereby list of endpoints is used for both. When defining VPWS, the numnber of endpoints is constrained to two endpoints.

The l2vpn container also includes definition of common building blocks for redundancy-grp templates and pseudowire-templates.

The State objects have been consolidated with the configuration object as per the recommendations provided by the Guidelines for Yang Module Authors document.

The IETF working group has defined the VPWS and VPLS services that leverages the pseudowire technologies defined by the PWE3 working group. A large number of RFCs from these working groups cover this subject matter. Hence, it is prudent that this document state the scope of the MPLS L2VPN object model definitions.

The following documents are within the scope. This is not an exhaustive list but a representation of documents that are covered for this work:

- o Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3) [RFC3916]
- o Pseudo-wire Emulation Edge-to-Edge (PWE3) Architecture [RFC3985]
- o IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3) [RFC4446]
- o Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) [RFC8077]
- o Encapsulation Methods for Transport of Ethernet over MPLS Networks [RFC4448]
- o Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN [RFC4385]
- o Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3) [RFC5254]
- o An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge [RFC5659]
- o Segmented Pseudowire [RFC6073]
- o Framework for Layer 2 Virtual Private Networks [RFC4664]
- o Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks [RFC4665]
- o Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling [RFC4761]
- o Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling [RFC4762]
- o Attachment Individual Identifier (AII) Types for Aggregation [RFC5003]
- o Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs) [RFC6074]
- o Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network [RFC6391]

- o Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling [RFC6624]
- o Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging [RFC7041]
- o LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS) [RFC7361]
- o Using the generic associated channel label for Pseudowire in the MPLS Transport Profile [RFC6423]
- o Pseudowire status for static pseudowire [RFC6478]

The specifics of pseudowire over MPLS-TP LSPs is in scope. However, the initial effort addresses definitions of object models that are commonly deployed.

The IETF work in L2VPN and PWE3 working group relating to L2TP, OAM, multicast (e.g. p2mp, etree, etc) and access specific protocols such as G.8032, MSTP, etc is out-of-scope for this document.

The following is the high level view of the L2VPN data model.

```

PW // Container
    PW specific attributes

    PW template definition

template-ref Redundancy-Group // redundancy-group
    template
    attributes

Network Instance // container
    l2vpn // container

        common attributes

        BGP-parameters // container
            common attributes
            auto-discovery attributes
            signaling attributes

        // list of PWs being used
        PW // container
            template-ref PW
            attribute-override

        PBB-parameters // container
            pbb specific attributes

        VPWS-constraints // rule to limit number of endpoints to two

        // List of endpoints, where each member endpoint container is -
        PW // reference
        redundancy-grp // container
            AC // eventual reference to standard AC
            PW // reference

```

Figure 1

3.2. Latest addition

Pseudowire module is extended to include,

Multi-segment PW - a new attribute is added to pseudowire that identifies the pseudowire as a member of the multi-segment

pseudowire. Two pseudowire members in a VPWS, configures a multi-segment pseudowire at the switching PE.

Pseudowire load-balancing - The load-balancing behaviour for a pseudowire can be configured either using the FAT label that resides below the pseudowire label or Entropy label with Entropy label indicator above the pseudowire label. By default, the load-balancing is disabled.

FEC 129 related - AGI, SAI and TAI string configurations is added to facilitate FEC 129 based pseudowire configuration.

3.3. Open issues and next steps

This section provides updates on open issues and will be removed before publication. Authors believes the document has covered the topics within the scope of the document. However, there are items, such as PW Headend, VPLS IRB, etc that can be candidate for inclusion. The authors would like to progress the document to publication for general availability with current content and tackle the other topics in a follow up document.

3.4. Pseudowire Common

3.4.1. Pseudowire

Pseudowire definitions is moved to a separate container in order to allow Ethernet Segment and EVPN models can refer without having to pull down L2VPN container.

3.4.2. pw-templates

The pw-templates container contains a list of pw-template. Each pw-template defines a list of common pseudowire attributes such as PW MTU, control word support etc.

3.5. L2VPN Common

3.5.1. redundancy-group-templates

The redundancy-group-template contains a list of templates. Each template defines common attributes related to redundancy such as protection mode, reversion parameters, etc.

3.6. L2VPN instance

The network instance container defined in the network instance model [I-D.ietf-rtgwg-ni-model] identifies the L2VPN instance. One of the value defined by the ni-type used in the instance model refers to VSI (Virtual Switch Instance) to denote the L2VPN instance. The name attribute field is used as the key to refer to specific network instance. Network Instance of type VSI anchors L2VPN container with a list of endpoints which when limited to two entries represents point to point service (i.e. VPWS) while more than two endpoints represent multipoint service (i.e. VPLS). Within a service instance, a set of common attributes are defined, followed by a list of PWs and a list of endpoints.

3.6.1. common attributes

The common attributes apply to entire L2VPN instance. These attributes typically include attributes such as mac-aging-timer, BGP related parameters (if using BGP signaling), discovery-type, etc.

3.6.2. PW list

The PW list is the number of PWs that are being used for a given L2VPN instance. Each PW entry refers to PW template to inherit common attributes for the PW. The one or more attributes from the template can be overridden. It further extends definitions of more PW specific attributes such as use of control word, mac withdraw, what type of signaling (i.e. LDP or BGP), setting of the TTL, etc.

3.6.3. List of endpoints

The list of endpoints define the characteristics of the L2VPN service. In the case of VPWS, the list is limited to two entries while for VPLS, there could be many.

Each entry in the endpoint list, may hold AC, PW or redundancy-grp references. The core aspect of endpoint container is its flexible personality based on what user decides to include in it. It is future-proofed with possible extensions that can be included in the endpoint container such as Integrated Route Bridging (IRB), PW Headend, Virtual Switch Instance, etc.

The endpoint entry also includes the split-horizon attribute which defines the frame forwarding restrictions between the endpoints belonging to same split-horizon group. This construct permits multiple instances of split horizon groups with its own endpoint members. The frame forwarding restrictions does not apply between endpoints that belong to two different split horizon groups.

3.6.3.1. ac

Attachment Circuit (AC) resides within endpoint entry either as an independent entity or as a member of the redundancy group. AC is not defined in this document but references the definitions specified by other working groups and standard bodies.

3.6.3.2. pw

The Pseudo-wire resides within endpoint entry either as an independent entity or as a member of the redundancy group. The PW refers to one of the entry in the list of PWs defined with the L2VPN instance.

3.6.3.3. redundancy-grp choice

The redundancy-grp is a generic redundancy construct which can hold primary and backup members of AC and PWs. This flexibility permits combinations of -

- o primary and backup AC
- o primary and backup PW
- o primary AC and backup PW
- o primary PW and backup AC

The redundancy group also defines attributes of the type of redundancy, such as protection mode, reroute mode, reversion related parameters, etc.

3.6.4. point-to-point or multipoint service

The point-to-point service as defined for VPWS is represented by a list of endpoints and is limited to two entries by the VPWS constrain rules

The multipoint service as defined for VPLS is represented by a list of endpoints.

The list of endpoints with one entry is invalid.

The augmentation of ietf-l2vpn module is TBD. All IP addresses defined in this module are currently scoped under global VRF/table.

3.6.5. multi-segment pseudowire

The multi-segment pseudowire is expressed as configuration of two pseudowire segments at the switching PEs that provides end-to-end PW path between two terminating PEs consisting of multiple pseudowire segments.

The multi-segment pseudowire is configured at switching PE using two endpoints that consists of pseudowires of type "ms-pw-members". The VPWS service construct is used with "vpws constraint" that restricts the number of endpoints to two.

To verify consistency, a) verify that both endpoints are using ms-pw-member pseudowires and b) it is only used as for VPWS configuration at the switching PE.

3.7. Operational State

The operational state of L2VPN attributes has been consolidated with the configuration as per recommendations from the guidelines for the YANG author document.

3.8. Yang tree

```

module: ietf-pseudowires
  +--rw pseudowires
    +--rw pseudowire* [name]
      +--rw name                               string
      +--ro state?                             pseudowire-status-type
      +--rw template?                          pw-template-ref
      +--rw mtu?                                uint16
      +--rw mac-withdraw?                       boolean
      +--rw pw-loadbalance?                     enumeration
      +--rw ms-pw-member?                       boolean
      +--rw cw-negotiation?                     cw-negotiation-type
      +--rw tunnel-policy?                      string
      +--rw (pw-type)?
        +--:(configured-pw)
          +--rw peer-ip?                        inet:ip-address
          +--rw pw-id?                          uint32
          +--rw group-id?                       uint32
          +--rw icb?                            boolean
          +--rw transmit-label?                  rt-types:mpls-label
          +--rw receive-label?                   rt-types:mpls-label
          +--rw generalized?                     boolean
          +--rw agi?                             string
          +--rw saii?                           string

```

```

    |   |   +--rw taii?                string
    |   +---:(bgp-pw)
    |   |   +--rw remote-pe-id?        inet:ip-address
    |   +---:(bgp-ad-pw)
    |       +--rw remote-ve-id?        uint16
+--rw pw-templates
  +--rw pw-template* [name]
    +--rw name                string
    +--rw mtu?                uint16
    +--rw cw-negotiation?     cw-negotiation-type
    +--rw tunnel-policy?      string

module: ietf-l2vpn
+--rw l2vpn
  +--rw redundancy-group-templates
    +--rw redundancy-group-template* [name]
      +--rw name                string
      +--rw protection-mode?    enumeration
      +--rw reroute-mode?       enumeration
      +--rw dual-receive?       boolean
      +--rw revert?             boolean
      +--rw reroute-delay?      uint16
      +--rw revert-delay?       uint16

augment /ni:network-instances/ni:network-instance/ni:ni-type:
+--:(l2vpn)
  +--rw type?                  identityref
  +--rw mtu?                    uint16
  +--rw mac-aging-timer?       uint32
  +--rw service-type?          l2vpn-service-type
  +--rw discovery-type?        l2vpn-discovery-type
  +--rw signaling-type          l2vpn-signaling-type
  +--rw bgp-parameters
    |   +--rw vpn-id?           string
    |   +--rw rd-rt
    |       +--rw route-distinguisher? rt-types:route-distinguisher
    |       +--rw vpn-target* [route-target]
    |           +--rw route-target          rt-types:route-target
    |           +--rw route-target-type     rt-types:route-target-type
  +--rw bgp-signaling
    |   +--rw site-id?           uint16
    |   +--rw site-range?       uint16
  +--rw endpoint* [name]
    |   +--rw name                string
    |   +--rw (ac-or-pw-or-redundancy-grp)?
    |       |   +--:(ac)
    |       |   |   +--rw ac* [name]
    |       |       +--rw name        if:interface-ref

```

```

| | | | | +--ro state? operational-state-type
| | | | | +---:(pw)
| | | | | | +--rw pw* [name]
| | | | | | +--rw name pw:pseudowire-ref
| | | | | | +--ro state? -> /pw:pseudowires/pseudowire[pw:name=current (
| | | | | )/../../name]/state
| | | | | +---:(redundancy-grp)
| | | | | | +--rw (primary)
| | | | | | | +---:(primary-ac)
| | | | | | | | +--rw primary-ac
| | | | | | | | +--rw name? if:interface-ref
| | | | | | | | +--ro state? operational-state-type
| | | | | | | +---:(primary-pw)
| | | | | | | | +--rw primary-pw* [name]
| | | | | | | | +--rw name pw:pseudowire-ref
| | | | | | | | +--ro state? -> /pw:pseudowires/pseudowire[pw:name=cu
| | | | | rrent()/../../name]/state
| | | | | | +--rw (backup)?
| | | | | | | +---:(backup-ac)
| | | | | | | | +--rw backup-ac
| | | | | | | | +--rw name? if:interface-ref
| | | | | | | | +--ro state? operational-state-type
| | | | | | | +---:(backup-pw)
| | | | | | | | +--rw backup-pw* [name]
| | | | | | | | +--rw name pw:pseudowire-ref
| | | | | | | | +--ro state? -> /pw:pseudowires/pseudowire[pw:na
| | | | | me=current()/../../name]/state
| | | | | | +--rw precedence? uint32
| | | | | | +--rw template? redundancy-group-template-ref
| | | | | | +--rw protection-mode? enumeration
| | | | | | +--rw reroute-mode? enumeration
| | | | | | +--rw dual-receive? boolean
| | | | | | +--rw revert? boolean
| | | | | | +--rw reroute-delay? uint16
| | | | | | +--rw revert-delay? uint16
| | | | | | +--rw split-horizon-group? string
| | | | | +--rw vpws-constraints
| | | | | +--rw pbb-parameters
| | | | | | +--rw (component-type)?
| | | | | | | +---:(i-component)
| | | | | | | | +--rw i-sid? i-sid-type
| | | | | | | | +--rw backbone-src-mac? yang:mac-address
| | | | | | | +---:(b-component)
| | | | | | | | +--rw bind-b-component-name? l2vpn-instance-name-ref
| | | | | | | | +--ro bind-b-component-type? identityref
| | | | | augment /pw:pseudowires/pw:pseudowire:
| | | | | | +--rw vccv-ability? boolean
| | | | | | +--rw request-vlanid? uint16
| | | | | | +--rw vlan-tpid? string
| | | | | | +--rw ttl? uint8
| | | | | augment /pw:pseudowires/pw:pseudowire/pw:pw-type:

```

```

+--: (bgp-pw)
|   +--rw bgp-pw
|       +--rw remote-pe-id?    inet:ip-address
+--: (bgp-ad-pw)
|   +--rw bgp-ad-pw
|       +--rw remote-ve-id?    uint16

notifications:
+---n l2vpn-state-change-notification
|   +--ro l2vpn-instance-name?    l2vpn-instance-name-ref
|   +--ro l2vpn-instance-type?    -> /ni:network-instances/network-instance
[ni:name=current()/../l2vpn-instance-name]/l2vpn:type
|   +--ro endpoint?              -> /ni:network-instances/network-instance
[ni:name=current()/../l2vpn-instance-name]/l2vpn:endpoint/name
|   +--ro (ac-or-pw-or-redundancy-grp)?
|   |   +--: (ac)
|   |   |   +--ro ac?            -> /ni:network-instances/network-instance
[ni:name=current()/../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/../
./endpoint]/ac/name
|   |   +--: (pw)
|   |   |   +--ro pw?            -> /ni:network-instances/network-instance
[ni:name=current()/../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/../
./endpoint]/pw/name
|   |   +--: (redundancy-grp)
|   |   |   +--ro (primary)
|   |   |   |   +--: (primary-ac)
|   |   |   |   |   +--ro primary-ac?    -> /ni:network-instances/network-instance
[ni:name=current()/../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/../
./endpoint]/primary-ac/name
|   |   |   |   +--: (primary-pw)
|   |   |   |   |   +--ro primary-pw?    -> /ni:network-instances/network-instance
[ni:name=current()/../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/../
./endpoint]/primary-pw/name
|   |   |   +--ro (backup)?
|   |   |   |   +--: (backup-ac)
|   |   |   |   |   +--ro backup-ac?    -> /ni:network-instances/network-instance
[ni:name=current()/../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/../
./endpoint]/backup-ac/name
|   |   |   |   +--: (backup-pw)
|   |   |   |   |   +--ro backup-pw?    -> /ni:network-instances/network-instance
[ni:name=current()/../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/../
./endpoint]/backup-pw/name
|   |   +--ro state?            identityref

```

Figure 2

4. YANG Module

The L2VPN configuration container is logically divided into following high level config areas:

```

<CODE BEGINS> file "ietf-pseudowires@2018-10-17.yang"
module ietf-pseudowires {
  namespace "urn:ietf:params:xml:ns:yang:ietf-pseudowires";
  prefix "pw";

  import ietf-inet-types {
    prefix "inet";

```



```
}

import ietf-routing-types {
  prefix "rt-types";
}

organization "ietf";
contact "ietf";
description "Pseudowire YANG model";

revision "2018-10-17" {
  description "Second revision " +
    " - Added group-id and attachment identifiers " +
    "";
  reference "";
}

revision "2017-06-26" {
  description "Initial revision " +
    " - Created a new model for pseudowires, which used " +
    " to be defined within the L2VPN model " +
    "";
  reference "";
}

/* Typedefs */

typedef pseudowire-ref {
  type leafref {
    path "/pw:pseudowires/pw:pseudowire/pw:name";
  }
  description "A type that is a reference to a pseudowire";
}

typedef pw-template-ref {
  type leafref {
    path "/pw:pseudowires/pw:pw-templates/pw:pw-template/pw:name";
  }
  description "A type that is a reference to a pw-template";
}

typedef cw-negotiation-type {
  type enumeration {
    enum "non-preferred" {
      description "No preference for control-word";
    }
    enum "preferred" {
      description "Prefer to have control-word negotiation";
    }
  }
}
```



```
    }
  }
  description "control-word negotiation preference type";
}

typedef pseudowire-status-type {
  type bits {
    bit pseudowire-forwarding {
      position 0;
      description "Pseudowire is forwarding";
    }
    bit pseudowire-not-forwarding {
      position 1;
      description "Pseudowire is not forwarding";
    }
    bit local-attachment-circuit-receive-fault {
      position 2;
      description "Local attachment circuit (ingress) receive " +
        "fault";
    }
    bit local-attachment-circuit-transmit-fault {
      position 3;
      description "Local attachment circuit (egress) transmit " +
        "fault";
    }
    bit local-PSN-facing-PW-receive-fault {
      position 4;
      description "Local PSN-facing PW (ingress) receive fault";
    }
    bit local-PSN-facing-PW-transmit-fault {
      position 5;
      description "Local PSN-facing PW (egress) transmit fault";
    }
    bit PW-preferential-forwarding-status {
      position 6;
      description "Pseudowire preferential forwarding status";
    }
    bit PW-request-switchover-status {
      position 7;
      description "Pseudowire request switchover status";
    }
  }
  description
    "Pseudowire status type, as registered in the IANA " +
    "Pseudowire Status Code Registry";
}

/* Data */
```

```
container pseudowires {
  description "Configuration management of pseudowires";
  list pseudowire {
    key "name";
    description "A pseudowire";
    leaf name {
      type string;
      description "pseudowire name";
    }
    leaf state {
      type pseudowire-status-type;
      config false;
      description "pseudowire operation status";
      reference "RFC 4446 and IANA Pseudowire Status Codes " +
        "Registry";
    }
    leaf template {
      type pw-template-ref;
      description "pseudowire template";
    }
    leaf mtu {
      type uint16;
      description "PW MTU";
    }
    leaf mac-withdraw {
      type boolean;
      default false;
      description "Enable (true) or disable (false) MAC withdraw";
    }
    leaf pw-loadbalance {
      type enumeration {
        enum "disabled" {
          value 0;
          description "load-balancing disabled";
        }
        enum "fat-pw" {
          value 1;
          description "load-balance using FAT label below PW label";
        }
        enum "entropy" {
          value 2;
          description "load-balance using ELI/EL above PW label";
        }
      }
      description "PW load-balancing";
    }
    leaf ms-pw-member {
      type boolean;
    }
  }
}
```

```
    default false;
    description "Enable (true) or disable (false) not a member of MS-PW";
}
leaf cw-negotiation {
    type cw-negotiation-type;
    description "cw-negotiation";
}
leaf tunnel-policy {
    type string;
    description "tunnel policy name";
}
choice pw-type {
    description "A choice of pseudowire type";
    case configured-pw {
        leaf peer-ip {
            type inet:ip-address;
            description "peer IP address";
        }
        leaf pw-id {
            type uint32;
            description "pseudowire id";
        }
        leaf group-id {
            type uint32;
            description "group id";
        }
        leaf icb {
            type boolean;
            description "inter-chassis backup";
        }
        leaf transmit-label {
            type rt-types:mpls-label;
            description "transmit lable";
        }
        leaf receive-label {
            type rt-types:mpls-label;
            description "receive label";
        }
        leaf generalized {
            type boolean;
            description "generalized pseudowire id FEC element";
        }
        leaf agi {
            type string;
            description "attachment group identifier";
        }
        leaf saii {
            type string;
        }
    }
}
```

```
        description "source attachment individual identifier";
    }
    leaf taii {
        type string;
        description "target attachment individual identifier";
    }
}
case bgp-pw {
    leaf remote-pe-id {
        type inet:ip-address;
        description "remote pe id";
    }
}
case bgp-ad-pw {
    leaf remote-ve-id {
        type uint16;
        description "remote ve id";
    }
}
}
}
container pw-templates {
    description "pw-templates";
    list pw-template {
        key "name";
        description "pw-template";
        leaf name {
            type string;
            description "name";
        }
        leaf mtu {
            type uint16;
            description "pseudowire mtu";
        }
        leaf cw-negotiation {
            type cw-negotiation-type;
            default "preferred";
            description
                "control-word negotiation preference";
        }
        leaf tunnel-policy {
            type string;
            description "tunnel policy name";
        }
    }
}
}
```

```
<CODE ENDS>
<CODE BEGINS> file "ietf-l2vpn@2019-05-28.yang"
module ietf-l2vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn";
  prefix "l2vpn";

  import ietf-inet-types {
    prefix "inet";
  }

  import ietf-yang-types {
    prefix "yang";
  }

  import ietf-routing-types {
    prefix "rt-types";
  }

  import ietf-interfaces {
    prefix "if";
  }

  import ietf-network-instance {
    prefix "ni";
  }

  import ietf-pseudowires {
    prefix "pw";
  }

  organization "ietf";
  contact "ietf";
  description "l2vpn";

  revision "2019-05-28" {
    description "Nineth revision " +
      " - Used bgp parameters hierarchy common to L2VPN and EVPN " +
      "";
    reference "";
  }

  revision "2018-02-06" {
    description "Eighth revision " +
      " - Incorporated ietf-network-instance model " +
      " - change the type of attachment circuit to " +
      " if:interface-ref " +
      "";
    reference "";
  }
}
```

```
}

revision "2017-09-21" {
  description "Seventh revision " +
    " - Fixed yangdump errors " +
    "";
  reference  "";
}
revision "2017-06-26" {
  description "Sixth revision " +
    " - Removed unused module mpls " +
    " - Renamed l2vpn-instances-state to l2vpn-instances " +
    " - Added pseudowire status as defined in RFC4446 and " +
    "   IANA Pseudowire Status Codes Register " +
    " - Added notifications " +
    " - Moved PW definition out of L2VPN " +
    " - Moved model to NMDA style specified in " +
    "   draft-dsdt-nmda-guidelines-01.txt " +
    " - Renamed l2vpn-instances and l2vpn-instance to " +
    "   instances and instance to shorten xpaths " +
    "";
  reference  "";
}

revision "2017-03-06" {
  description "Sixth revision " +
    " - Removed the 'common' container and move pw-templates " +
    "   and redundancy-group-templates up a level " +
    " - Consolidated the endpoint configuration such that " +
    "   all L2VPN instances has a list of endpoint. For " +
    "   certain types of L2VPN instances such as VPWS where " +
    "   each L2VPN instance is limited to at most two " +
    "   endpoint, additional augment statements were included " +
    "   to add necessary constraints " +
    " - Removed discovery-type and signaling-type operational " +
    "   state from VPLS pseudowires, as these two parameters " +
    "   are configured as L2VPN parameters rather than " +
    "   pseudowire paramteres " +
    " - Renamed l2vpn-instances to l2vpn-instances-state " +
    "   in the operational state branch " +
    " - Removed BGP parameter groupings and reused " +
    "   ietf-routing-types.yang module instead " +
    "";
  reference  "";
}

revision "2016-10-24" {
  description "Fifth revision " +
```

```
    " - Edits based on Giles's comments " +
    " 5) Remove relative leafrefs in groupings, " +
    " and the resulting new groupings are: " +
    " (a) bgp-auto-discovery-parameters-grp " +
    " (b) bgp-signaling-parameters-grp " +
    " (c) endpoint-grp " +
    " 11) Merge VPLS and VPWS into one single list " +
    " and use augment statements to handle " +
    " differences between VPLS and VPWS " +
    " - Add a new grouping l2vpn-common-parameters-grp " +
    " to make VPLS and VPWS more consistent";
  reference "";
}

revision "2016-05-31" {
  description "Fourth revision " +
    " - Edits based on Giles's comments " +
    " 1) Change enumeration to identityref type for: " +
    " (a) l2vpn-service-type " +
    " (b) l2vpn-discovery-type " +
    " (c) l2vpn-signaling-type " +
    " bgp-rt-type, cw-negotiation, and " +
    " pbb-component remain enumerations " +
    " 2) Define i-sid-type for leaf 'i-sid' " +
    " (which is renamed from 'i-tag') " +
    " 3) Rename 'vpn-targets' to 'vpn-target' " +
    " 4) Import ietf-mpls.yang and reuse the " +
    " 'mpls-label' type defined in ietf-mpls.yang " +
    " transmit-label and receive-label " +
    " 8) Change endpoint list's key to name " +
    " 9) Changed MTU to type uint16 " +
    "";
  reference "";
}

revision "2016-03-07" {
  description "Third revision " +
    " - Changed the module name to ietf-l2vpn " +
    " - Merged EVPN into L2VPN " +
    " - Eliminated the definitions of attachment " +
    " circuit with the intention to reuse other " +
    " layer-2 definitions " +
    " - Added state branch";
  reference "";
}

revision "2015-10-08" {
  description "Second revision " +
```

```
        " - Added container vpls-instances " +
        " - Rearranged groupings and typedefs to be " +
        "   reused across vpls-instance and vpws-instances";
    reference "";
}

revision "2015-06-30" {
    description "Initial revision";
    reference "";
}

/* identities */

identity l2vpn-instance-type {
    description "Base identity from which identities of " +
               "l2vpn service instance types are derived";
}

identity vpws-instance-type {
    base l2vpn-instance-type;
    description "This identity represents VPWS instance type";
}

identity vpls-instance-type {
    base l2vpn-instance-type;
    description "This identity represents VPLS instance type";
}

identity link-discovery-protocol {
    description "Base identity from which identities describing " +
               "link discovery protocols are derived";
}

identity lacp {
    base "link-discovery-protocol";
    description "This identity represents LACP";
}

identity lldp {
    base "link-discovery-protocol";
    description "This identity represents LLDP";
}

identity bpdu {
    base "link-discovery-protocol";
    description "This identity represents BPDU";
}
```



```
identity cpd {
  base "link-discovery-protocol";
  description "This identity represents CPD";
}

identity udld {
  base "link-discovery-protocol";
  description "This identity represens UDLD";
}

identity l2vpn-service {
  description "Base identity from which identities describing " +
    "L2VPN services are derived";
}

identity Ethernet {
  base "l2vpn-service";
  description "This identity represents Ethernet service";
}

identity ATM {
  base "l2vpn-service";
  description "This identity represents Asynchronous Transfer " +
    "Mode service";
}

identity FR {
  base "l2vpn-service";
  description "This identity represent Frame-Relay service";
}

identity TDM {
  base "l2vpn-service";
  description "This identity represent Time Devision " +
    "Multiplexing service";
}

identity l2vpn-discovery {
  description "Base identity from which identities describing " +
    "L2VPN discovery protocols are derived";
}

identity manual-discovery {
  base "l2vpn-discovery";
  description "Manual configuration of l2vpn service";
}

identity bgp-auto-discovery {
  base "l2vpn-discovery";
```

```
    description "Border Gateway Protocol (BGP) auto-discovery of " +
        "l2vpn service";
}

identity ldp-discovery {
    base "l2vpn-discovery";
    description "Label Distribution Protocol (LDP) discovery of " +
        "l2vpn service";
}

identity mixed-discovery {
    base "l2vpn-discovery";
    description "Mixed discovery methods of l2vpn service";
}

identity l2vpn-signaling {
    description "Base identity from which identities describing " +
        "L2VPN signaling protocols are derived";
}

identity static-configuration {
    base "l2vpn-signaling";
    description "Static configuration of labels (no signaling)";
}

identity ldp-signaling {
    base "l2vpn-signaling";
    description "Label Distribution Protocol (LDP) signaling";
}

identity bgp-signaling {
    base "l2vpn-signaling";
    description "Border Gateway Protocol (BGP) signaling";
}

identity mixed-signaling {
    base "l2vpn-signaling";
    description "Mixed signaling methods";
}

identity l2vpn-notification-state {
    description "The base identity on which notification states " +
        "are based";
}

identity MAC-limit-reached {
    base "l2vpn-notification-state";
    description "MAC limit is reached";
}
```

```
}
identity MAC-limit-cleared {
    base "l2vpn-notification-state";
    description "MAC limit is cleared";
}

identity MTU-mismatched {
    base "l2vpn-notification-state";
    description "MAC is mismatched";
}

identity MTU-mismatched-cleared {
    base "l2vpn-notification-state";
    description "MAC is mismatch is cleared";
}

identity state-changed-to-up {
    base "l2vpn-notification-state";
    description "State is changed to UP";
}

identity state-changed-to-down {
    base "l2vpn-notification-state";
    description "State is changed to down";
}

identity MAC-move-limit-exceeded {
    base "l2vpn-notification-state";
    description "MAC move limit is exceeded";
}

identity MAC-move-limit-exceeded-cleared {
    base "l2vpn-notification-state";
    description "MAC move limit exceeded is cleared";
}

identity MAC-flap-detected {
    base "l2vpn-notification-state";
    description "MAC flap detected";
}

identity port-disabled-due-to-MAC-flap {
    base "l2vpn-notification-state";
    description "Port disabled due to MAC flap";
}

/* typedefs */
```

```
typedef l2vpn-service-type {
  type identityref {
    base "l2vpn-service";
  }
  description "L2VPN service type";
}

typedef l2vpn-discovery-type {
  type identityref {
    base "l2vpn-discovery";
  }
  description "L2VPN discovery type";
}

typedef l2vpn-signaling-type {
  type identityref {
    base "l2vpn-signaling";
  }
  description "L2VPN signaling type";
}

typedef link-discovery-protocol-type {
  type identityref {
    base "link-discovery-protocol";
  }
  description "This type is used to identify " +
    "link discovery protocol";
}

typedef pbb-component-type {
  type enumeration {
    enum "b-component" {
      description "Identifies as a b-component";
    }
    enum "i-component" {
      description "Identifies as an i-component";
    }
  }
  description "This type is used to identify " +
    "the type of PBB component";
}

typedef redundancy-group-template-ref {
  type leafref {
    path "/l2vpn:l2vpn/l2vpn:redundancy-group-templates" +
      "/l2vpn:redundancy-group-template/l2vpn:name";
  }
  description "redundancy-group-template-ref";
}
```

```
}
typedef l2vpn-instance-name-ref {
  type leafref {
    path "/ni:network-instances/ni:network-instance" +
        "/ni:name";
  }
  description "l2vpn-instance-name-ref";
}

typedef l2vpn-instance-type-ref {
  type leafref {
    path "/ni:network-instances/ni:network-instance" +
        "/l2vpn:type";
  }
  description "l2vpn-instance-type-ref";
}

typedef operational-state-type {
  type enumeration {
    enum 'up' {
      description "Operational state is up";
    }
    enum 'down' {
      description "Operational state is down";
    }
  }
  description "operational-state-type";
}

typedef i-sid-type {
  type uint32 {
    range "0..16777216";
  }
  description "I-SID type that is 24-bits. " +
    "This should be moved to ieee-types.yang at " +
    "http://www.ieee802.org/1/files/public/docs2015 " +
    "/new-mholness-ieee-types-yang-v01.yang";
}

/* groupings */

grouping pbb-parameters-grp {
  description "PBB parameters grouping";
  container pbb-parameters {
    description "pbb-parameters";
    choice component-type {
      description "PBB component type";
      case i-component {
```

```
    leaf i-sid {
        type i-sid-type;
        description "I-SID";
    }
    leaf backbone-src-mac {
        type yang:mac-address;
        description "backbone-src-mac";
    }
}
case b-component {
    leaf bind-b-component-name {
        type l2vpn-instance-name-ref;
        must "/ni:network-instances" +
            "/ni:network-instance[ni:name=current()]" +
            "/l2vpn:type = 'l2vpn:vpls-instance-type'" {
            description "A b-component must be an L2VPN instance " +
                "of type vpls-instance-type";
        }
        description "Reference to the associated b-component";
    }
    leaf bind-b-component-type {
        type identityref {
            base l2vpn-instance-type;
        }
        must ". = 'l2vpn:vpls-instance-type'" {
            description "The associated b-component must have " +
                "type vpls-instance-type";
        }
        config false;
        description "Type of the associated b-component";
    }
}
}
}

grouping pbb-parameters-state-grp {
    description "PBB parameters grouping";
    container pbb-parameters {
        description "pbb-parameters";
        choice component-type {
            description "PBB component type";
            case i-component {
                leaf i-sid {
                    type i-sid-type;
                    description "I-SID";
                }
                leaf backbone-src-mac {
```

```
        type yang:mac-address;
        description "backbone-src-mac";
    }
}
case b-component {
    leaf bind-b-component-name {
        type string;
        description "Name of the associated b-component";
    }
    leaf bind-b-component-type {
        type identityref {
            base l2vpn-instance-type;
        }
        must ". = 'l2vpn:vpls-instance-type'" {
            description "The associated b-component must have " +
                "type vpls-instance-type";
        }
        description "Type of the associated b-component";
    }
}
}
}

grouping l2vpn-common-parameters-grp {
    description "L2VPN common parameters";
    leaf type {
        type identityref {
            base l2vpn-instance-type;
        }
        description "Type of L2VPN service instance";
    }
    leaf mtu {
        type uint16;
        description "MTU of L2VPN service";
    }
    leaf mac-aging-timer {
        type uint32;
        description "mac-aging-timer, the duration after which" +
            "a MAC entry is considered aged out";
    }
    leaf service-type {
        type l2vpn-service-type;
        default Ethernet;
        description "L2VPN service type";
    }
    leaf discovery-type {
        type l2vpn-discovery-type;
    }
}
```

```
        default manual-discovery;
        description "L2VPN service discovery type";
    }
    leaf signaling-type {
        type l2vpn-signaling-type;
        mandatory true;
        description "L2VPN signaling type";
    }
}
grouping bgp-signaling-parameters-grp {
    description "BGP parameters for signaling";
    leaf site-id {
        type uint16;
        description "Site ID";
    }
    leaf site-range {
        type uint16;
        description "Site Range";
    }
}

grouping redundancy-group-properties-grp {
    description "redundancy-group-properties-grp";
    leaf protection-mode {
        type enumeration {
            enum "frr" {
                value 0;
                description "fast reroute";
            }
            enum "master-slave" {
                value 1;
                description "master-slave";
            }
            enum "independent" {
                value 2;
                description "independent";
            }
        }
        description "protection-mode";
    }
    leaf reroute-mode {
        type enumeration {
            enum "immediate" {
                value 0;
                description "immediate reroute";
            }
            enum "delayed" {
                value 1;
            }
        }
    }
}
```



```
        description "delayed reroute";
    }
    enum "never" {
        value 2;
        description "never reroute";
    }
}
description "reroute-mode";
}
leaf dual-receive {
    type boolean;
    description
        "allow extra traffic to be carried by backup";
}
leaf revert {
    type boolean;
    description "allow forwarding to revert to primary " +
        "after restoring primary";
}
leaf reroute-delay {
    when "../reroute-mode = 'delayed'" {
        description "Specify amount of time to " +
            "delay reroute only when " +
            "delayed route is configured";
    }
    type uint16;
    description "amount of time to delay reroute";
}
leaf revert-delay {
    when "../revert = 'true'" {
        description "Specify the amount of time to " +
            "wait to revert to primary " +
            "only if reversion is configured";
    }
    type uint16;
    description "amount of time to wait to revert to primary";
}
}

grouping endpoint-grp {
    description "A grouping that defines the structure of " +
        "an endpoint";
    choice ac-or-pw-or-redundancy-grp {
        description "A choice of attachment circuit or " +
            "pseudowire or redundancy group";
        case ac {
            description "Attachment circuit(s) as an endpoint";
        }
    }
}
```

```
    case pw {
      description "Pseudowire(s) as an endpoint";
    }
    case redundancy-grp {
      description "Redundancy group as an endpoint";
      choice primary {
        mandatory true;
        description "primary options";
        case primary-ac {
          description "primary-ac";
        }
        case primary-pw {
          description "primary-pw";
        }
      }
      choice backup {
        description "backup options";
        case backup-ac {
          description "backup-ac";
        }
        case backup-pw {
          description "backup-pw";
        }
      }
    }
  }
}

/* L2VPN YANG Model */

container l2vpn {
  description "L2VPN specific data";

  container redundancy-group-templates {
    description "redundancy group templates";
    list redundancy-group-template {
      key "name";
      description "redundancy-group-template";
      leaf name {
        type string;
        description "name";
      }
      uses redundancy-group-properties-grp;
    }
  }
}

/* augments */
```

```
augment "/ni:network-instances/ni:network-instance/ni:ni-type" {
  description
    "Augmentation for L2VPN instance";
  case l2vpn {
    description "An L2VPN service instance";
    uses l2vpn-common-parameters-grp;
    container bgp-parameters {
      when "../discovery-type = 'l2vpn:bgp-auto-discovery'" {
        description "Parameters used when discovery type is " +
          "bgp-auto-discovery";
      }
      description "BGP auto-discovery parameters";
      leaf vpn-id {
        type string;
        description "VPN ID";
      }
    }
    container rd-rt {
      leaf route-distinguisher {
        type rt-types:route-distinguisher;
        description "BGP route distinguisher";
      }
      uses rt-types:vpn-route-targets;
      description "Route distinguisher and " +
        "corresponding VPN route targets";
    }
  }
  container bgp-signaling {
    when "../signaling-type = 'l2vpn:bgp-signaling'" {
      description "Check signaling type: " +
        "Can only configure BGP signaling if " +
        "signaling type is BGP";
    }
    description "BGP signaling parameters";
    uses bgp-signaling-parameters-grp;
  }
  list endpoint {
    key "name";
    description "An endpoint";
    leaf name {
      type string;
      description "endpoint name";
    }
    uses endpoint-grp {
      augment "ac-or-pw-or-redundancy-grp/ac" {
        description "Augment for attachment circuit(s) " +
          "as an endpoint";
        list ac {
          key "name";
        }
      }
    }
  }
}
```

```
    leaf name {
      type if:interface-ref;
      description "Name of attachment circuit";
    }
    leaf state {
      type operational-state-type;
      config false;
      description "attachment circuit up/down state";
    }
    description "An L2VPN instance's " +
      "attachment circuit list";
  }
}
augment "ac-or-pw-or-redundancy-grp/pw" {
  description "Augment for pseudowire(s) as an endpoint";
  list pw {
    key "name";
    leaf name {
      type pw:pseudowire-ref;
      must "(../../../type = " +
        "'l2vpn:vpws-instance-type') or " +
        "(not(boolean(/pw:pseudowires" +
          "/pw:pseudowire[pw:name = current()]" +
          "/vccv-ability)) and " +
        "not(boolean(/pw:pseudowires" +
          "/pw:pseudowire[pw:name = current()]" +
          "/request-vlanid)) and " +
        "not(boolean(/pw:pseudowires" +
          "/pw:pseudowire[pw:name = current()]" +
          "/vlan-tpid)) and " +
        "not(boolean(/pw:pseudowires" +
          "/pw:pseudowire[pw:name = current()]" +
          "/ttl)))" {
        description "Only a VPWS PW has parameters " +
          "vccv-ability, request-vlanid, " +
          "vlan-tpid, and ttl";
      }
    }
    description "Pseudowire name";
  }
  leaf state {
    type leafref {
      path "/pw:pseudowires" +
        "/pw:pseudowire[pw:name=current()../../name]" +
        "/pw:state";
    }
    config false;
    description "Pseudowire state";
  }
}
```

```

        description "An L2VPN instance's pseudowire list";
    }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "primary/primary-ac" {
    description "Augment for primary-ac";
    container primary-ac {
        description "Primary AC";
        leaf name {
            type if:interface-ref;
            description "Name of attachment circuit";
        }
        leaf state {
            type operational-state-type;
            config false;
            description "attachment circuit up/down state";
        }
    }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "primary/primary-pw" {
    description "Augment for primary-pw";
    list primary-pw {
        key "name";
        leaf name {
            type pw:pseudowire-ref;
            must "(!../..../type = " +
                "'l2vpn:vpws-instance-type') or " +
                "(not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /vccv-ability)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /request-vlanid)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /vlan-tpid)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /ttl)))" {
                description "Only a VPWS PW has parameters " +
                    "vccv-ability, request-vlanid, " +
                    "vlan-tpid, and ttl";
            }
        }
        description "Pseudowire name";
    }
    leaf state {
        type leafref {

```

```

        path "/pw:pseudowires" +
            "/pw:pseudowire[pw:name=current()/../name]" +
            "/pw:state";
    }
    config false;
    description "Pseudowire state";
}
description "An L2VPN instance's pseudowire list";
}
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "backup/backup-ac" {
    description "Augment for backup-ac";
    container backup-ac {
        description "Backup AC";
        leaf name {
            type if:interface-ref;
            description "Name of attachment circuit";
        }
        leaf state {
            type operational-state-type;
            config false;
            description "attachment circuit up/down state";
        }
    }
}
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "backup/backup-pw" {
    description "Augment for backup-pw";
    list backup-pw {
        key "name";
        leaf name {
            type pw:pseudowire-ref;
            must "(../../../../../type = " +
                "'l2vpn:vpws-instance-type') or " +
                "(not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /vccv-ability)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /request-vlanid)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /vlan-tpid)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /ttl)))" {
            description "Only a VPWS PW has parameters " +

```

```
        "vccv-ability, request-vlanid, " +
        "vlan-tpid, and ttl";
    }
    description "Pseudowire name";
}
leaf state {
    type leafref {
        path "/pw:pseudowires" +
            "/pw:pseudowire[pw:name=current()/../name]" +
            "/pw:state";
    }
    config false;
    description "Pseudowire state";
}
description "A list of backup pseudowires";
}
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp" {
    description "Augment for redundancy group properties";
    leaf template {
        type redundancy-group-template-ref;
        description "Reference a redundancy group " +
            "properties template";
    }
    uses redundancy-group-properties-grp;
}
}
}
}

augment "/pw:pseudowires/pw:pseudowire" {
    description "Augment for pseudowire parameters for " +
        "VPWS pseudowires";
    leaf vccv-ability {
        type boolean;
        description "vccvability";
    }
    leaf request-vlanid {
        type uint16;
        description "request vlanid";
    }
    leaf vlan-tpid {
        type string;
        description "vlan tpid";
    }
    leaf ttl {
        type uint8;
    }
}
```

```
        description "time-to-live";
    }
}

augment "/pw:pseudowires/pw:pseudowire/pw:pw-type" {
    description "Additional pseudowire types";
    case bgp-pw {
        container bgp-pw {
            description "BGP pseudowire";
            leaf remote-pe-id {
                type inet:ip-address;
                description "remote pe id";
            }
        }
    }
    case bgp-ad-pw {
        container bgp-ad-pw {
            description "BGP auto-discovery pseudowire";
            leaf remote-ve-id {
                type uint16;
                description "remote ve id";
            }
        }
    }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
    "/l2vpn:l2vpn" {
    when "l2vpn:type = 'l2vpn:vpws-instance-type'" {
        description "Constraints only for VPWS pseudowires";
    }
    description "Augment for VPWS instance";
    container vpws-constraints {
        must "(count(..endpoint) <= 2) and " +
            "(count(..endpoint/pw) <= 1) and " +
            "(count(..endpoint/ac) <= 1) and " +
            "(count(..endpoint/primary-pw) <= 1) and " +
            "(count(..endpoint/backup-pw) <= 1) " {
            description "A VPWS L2VPN instance has at most 2 endpoints " +
                "and each endpoint has at most 1 pseudowire or " +
                "1 attachment circuit";
        }
        description "VPWS constraints";
    }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
    "/l2vpn:l2vpn" {
```



```
    when "l2vpn:type = 'l2vpn:vpls-instance-type'" {
        description "Parameters specifically for a VPLS instance";
    }
    description "Augment for parameters for a VPLS instance";
    uses pbb-parameters-grp;
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
    "/l2vpn:l2vpn/l2vpn:endpoint" {
    when "../l2vpn:type = 'l2vpn:vpls-instance-type'" {
        description "Endpoint parameter specifically for " +
            "a VPLS instance";
    }
    description "Augment for endpoint parameters for a VPLS instance";
    leaf split-horizon-group {
        type string;
        description "Identify a split horizon group";
    }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
    "/l2vpn:l2vpn/l2vpn:endpoint" +
    "/l2vpn:ac-or-pw-or-redundancy-grp" +
    "/l2vpn:redundancy-grp/l2vpn:backup" +
    "/l2vpn:backup-pw/l2vpn:backup-pw" {
    when "../..../l2vpn:type = 'l2vpn:vpls-instance-type'" {
        description "Backup pseudowire parameter specifically for " +
            "a VPLS instance";
    }
    description "Augment for backup pseudowire paramters for " +
        "a VPLS instance";
    leaf precedence {
        type uint32;
        description "precedence of the pseudowire";
    }
}

/* Notifications */

notification l2vpn-state-change-notification {
    description "L2VPN and constituents state change notification";
    leaf l2vpn-instance-name {
        type l2vpn-instance-name-ref;
        description "The L2VPN instance name";
    }
    leaf l2vpn-instance-type {
        type leafref {
            path "/ni:network-instances" +

```

```
        "/ni:network-instance" +
        "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:type";
    }
    description "The L2VPN instance type";
}
leaf endpoint {
    type leafref {
        path "/ni:network-instances" +
            "/ni:network-instance" +
            "[ni:name=current()/../l2vpn-instance-name]" +
            "/l2vpn:endpoint/l2vpn:name";
    }
    description "The endpoint";
}
uses endpoint-grp {
    augment "ac-or-pw-or-redundancy-grp/ac" {
        description "Augment for attachment circuit(s) " +
            "as an endpoint";
        leaf ac {
            type leafref {
                path "/ni:network-instances" +
                    "/ni:network-instance" +
                    "[ni:name=current()/../l2vpn-instance-name]" +
                    "/l2vpn:endpoint" +
                    "[l2vpn:name=current()/../endpoint]" +
                    "/l2vpn:ac/l2vpn:name";
            }
            description "Related attachment circuit";
        }
    }
    augment "ac-or-pw-or-redundancy-grp/pw" {
        description "Augment for pseudowire(s) as an endpoint";
        leaf pw {
            type leafref {
                path "/ni:network-instances" +
                    "/ni:network-instance" +
                    "[ni:name=current()/../l2vpn-instance-name]" +
                    "/l2vpn:endpoint[l2vpn:name=current()/../endpoint]" +
                    "/l2vpn:pw/l2vpn:name";
            }
            description "Related pseudowire";
        }
    }
    augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
        "primary/primary-ac" {
        description "Augment for primary-ac";
        leaf primary-ac {
```

```
    type leafref {
      path "/ni:network-instances" +
        "/ni:network-instance" +
          "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:endpoint" +
          "[l2vpn:name=current()/../endpoint]" +
        "/l2vpn:primary-ac/l2vpn:name";
    }
    description "Related primary attachment circuit";
  }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
  "primary/primary-pw" {
  description "Augment for primary-pw";
  leaf primary-pw {
    type leafref {
      path "/ni:network-instances" +
        "/ni:network-instance" +
          "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:endpoint" +
          "[l2vpn:name=current()/../endpoint]" +
        "/l2vpn:primary-pw/l2vpn:name";
    }
    description "Related primary pseudowire";
  }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
  "backup/backup-ac" {
  description "Augment for backup-ac";
  leaf backup-ac {
    type leafref {
      path "/ni:network-instances" +
        "/ni:network-instance" +
          "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:endpoint" +
          "[l2vpn:name=current()/../endpoint]" +
        "/l2vpn:backup-ac/l2vpn:name";
    }
    description "Related backup attachment circuit";
  }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
  "backup/backup-pw" {
  description "Augment for backup-pw";
  leaf backup-pw {
    type leafref {
      path "/ni:network-instances" +
        "/ni:network-instance" +
```

```
        "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:endpoint" +
        "[l2vpn:name=current()/../endpoint]" +
        "/l2vpn:backup-pw/l2vpn:name";
    }
    description "Related backup pseudowire";
}
}
}
leaf state {
    type identityref {
        base l2vpn-notification-state;
    }
    description "State change notification";
}
}
}
}
<CODE ENDS>
```

Figure 3

5. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

The security concerns listed above are, however, no different than faced by other routing protocols. Hence, this draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg]

6. IANA Considerations

None.

7. Acknowledgments

The authors would like to acknowledge Giles Heron and others for their useful comments.

MITRE has approved this document for Public Release, Distribution Unlimited, with Public Release Case Number 19-0683.

8. References

8.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

8.2. Informative References

- [RFC3916] Xiao, X., Ed., McPherson, D., Ed., and P. Pate, Ed., "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, DOI 10.17487/RFC3916, September 2004, <<https://www.rfc-editor.org/info/rfc3916>>.
- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, DOI 10.17487/RFC4446, April 2006, <<https://www.rfc-editor.org/info/rfc4446>>.
- [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <<https://www.rfc-editor.org/info/rfc4448>>.
- [RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <<https://www.rfc-editor.org/info/rfc4664>>.
- [RFC4665] Augustyn, W., Ed. and Y. Serbest, Ed., "Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks", RFC 4665, DOI 10.17487/RFC4665, September 2006, <<https://www.rfc-editor.org/info/rfc4665>>.

- [RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <<https://www.rfc-editor.org/info/rfc4761>>.
- [RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <<https://www.rfc-editor.org/info/rfc4762>>.
- [RFC5003] Metz, C., Martini, L., Balus, F., and J. Sugimoto, "Attachment Individual Identifier (AII) Types for Aggregation", RFC 5003, DOI 10.17487/RFC5003, September 2007, <<https://www.rfc-editor.org/info/rfc5003>>.
- [RFC5254] Bitar, N., Ed., Bocci, M., Ed., and L. Martini, Ed., "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, DOI 10.17487/RFC5254, October 2008, <<https://www.rfc-editor.org/info/rfc5254>>.
- [RFC5659] Bocci, M. and S. Bryant, "An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge", RFC 5659, DOI 10.17487/RFC5659, October 2009, <<https://www.rfc-editor.org/info/rfc5659>>.
- [RFC6073] Martini, L., Metz, C., Nadeau, T., Bocci, M., and M. Aissaoui, "Segmented Pseudowire", RFC 6073, DOI 10.17487/RFC6073, January 2011, <<https://www.rfc-editor.org/info/rfc6073>>.
- [RFC6074] Rosen, E., Davie, B., Radoaca, V., and W. Luo, "Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)", RFC 6074, DOI 10.17487/RFC6074, January 2011, <<https://www.rfc-editor.org/info/rfc6074>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.

- [RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <<https://www.rfc-editor.org/info/rfc6391>>.
- [RFC6423] Li, H., Martini, L., He, J., and F. Huang, "Using the Generic Associated Channel Label for Pseudowire in the MPLS Transport Profile (MPLS-TP)", RFC 6423, DOI 10.17487/RFC6423, November 2011, <<https://www.rfc-editor.org/info/rfc6423>>.
- [RFC6478] Martini, L., Swallow, G., Heron, G., and M. Bocci, "Pseudowire Status for Static Pseudowires", RFC 6478, DOI 10.17487/RFC6478, May 2012, <<https://www.rfc-editor.org/info/rfc6478>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC6624] Kompella, K., Kothari, B., and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, DOI 10.17487/RFC6624, May 2012, <<https://www.rfc-editor.org/info/rfc6624>>.
- [RFC7041] Balus, F., Ed., Sajassi, A., Ed., and N. Bitar, Ed., "Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging", RFC 7041, DOI 10.17487/RFC7041, November 2013, <<https://www.rfc-editor.org/info/rfc7041>>.
- [RFC7361] Dutta, P., Balus, F., Stokes, O., Calvignac, G., and D. Fedyk, "LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS)", RFC 7361, DOI 10.17487/RFC7361, September 2014, <<https://www.rfc-editor.org/info/rfc7361>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8077] Martini, L., Ed. and G. Heron, Ed., "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", STD 84, RFC 8077, DOI 10.17487/RFC8077, February 2017, <<https://www.rfc-editor.org/info/rfc8077>>.

Appendix A. Example Configuration

This section shows an example configuration using the YANG data model defined in the document.

Appendix B. Contributors

The editors gratefully acknowledge the following people for their contributions to this document.

Reshad Rahman
Cisco Systems, Inc.
Email: rrahman@cisco.com

Kamran Raza
Cisco Systems, Inc.
Email: skraza@cisco.com

Giles Heron
Cisco Systems, Inc.
Email: giheron@cisco.com

Tapraj Singh
Cisco Systems, Inc.
Email: tsingh@cisco.com

Zhenbin Li
Huawei Technologies
Email: lizhenbin@huawei.com

Zhuang Shunwan
Huawei Technologies
Email: Zhuangshunwan@huawei.com

Wang Haibo
Huawei Technologies
Email: rainsword.wang@huawei.com

Sajjad Ahmed
Ericsson
Email: sajjad.ahmed@ericsson.com

Matthew Bocci
Nokia
Email: matthew.bocci@nokia.com

Jorge Rabadan
Nokia

Email: jorge.rabadan@nokia.com

Jonathan Hardwick
Metaswitch
Email: jonathan.hardwick@metaswitch.com

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

Nick Delregno
Verizon
Email: nick.deregn@verizon.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com

Maria Joecylyn
Verizon
Email: joecylyn.malit@verizon.com

Figure 4

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Ing-When Chen
The MITRE Corporation

Email: ingwherchen@mitre.org

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Kishore Tiruveedhula
Juniper Networks

Email: kishoret@juniper.net

BESS
Internet-Draft
Updates: 6514 (if approved)
Intended status: Standards Track
Expires: November 25, 2021

Z. Zhang
L. Giuliano
Juniper Networks
May 24, 2021

MVPN and MSDP SA Interoperation
draft-ietf-bess-mvpn-msdp-sa-interoperation-08

Abstract

This document specifies the procedures for interoperation between Multicast Virtual Private Network (MVPN) Source Active routes and customer Multicast Source Discovery Protocol (MSDP) Source Active routes, which is useful for MVPN provider networks offering services to customers with an existing MSDP infrastructure. Without the procedures described in this document, VPN-specific MSDP sessions are required among the PEs that are customer MSDP peers. This document updates RFC6514.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 25, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Terminologies	2
2. Introduction	3
2.1. MVPN RPT-SPT Mode	4
3. Specification	4
4. Security Considerations	5
5. IANA Considerations	6
6. Acknowledgements	6
7. References	6
7.1. Normative References	6
7.2. Informative References	6
Authors' Addresses	7

1. Terminologies

Familiarity with MVPN [RFC6513] [RFC6514] and MSDP [RFC3618] protocols and procedures is assumed. Some terminologies are listed below for convenience.

- o ASM: Any source multicast.
- o SPT: Source-specific Shortest-path Tree.
- o RPT: Rendezvous Point Tree.
- o C-S: A multicast source address, identifying a multicast source located at a VPN customer site.
- o C-G: A multicast group address used by a VPN customer.
- o C-RP: A multicast Rendezvous Point for a VPN customer.

- o C-Multicast: Multicast for a VPN customer.
- o EC: Extended Community.
- o GTM: Global Table Multicast, i.e., multicast in the default or global routing table vs. VRF table.

2. Introduction

Section "14. Supporting PIM-SM without Inter-Site Shared C-Trees" of [RFC6514] specifies the procedures for MVPN PEs to discover (C-S,C-G) via MVPN Source Active A-D routes and then send Source Tree Join (C-S,C-G) C-multicast routes towards the ingress PEs, to establish SPTs for customer ASM flows for which they have downstream receivers. (C-*,C-G) C-multicast routes are not sent among the PEs so inter-site shared C-Trees are not used and the method is generally referred to as "spt-only" mode.

With this mode, the MVPN Source Active routes are functionally similar to MSDP Source-Active messages. For a VPN, one or more of the PEs, say PE1, either acts as a C-RP and learns of (C-S,C-G) via PIM Register messages, or has MSDP sessions with some MSDP peers and learn (C-S,C-G) via MSDP SA messages. In either case, PE1 will then originate MVPN SA routes for other PEs to learn the (C-S,C-G).

[RFC6514] only specifies that a PE receiving the MVPN SA routes, say PE2, will advertise Source Tree Join (C-S,C-G) C-multicast routes if it has corresponding (C-*,C-G) state learnt from its CE. PE2 may also have MSDP sessions for the VPN with other C-RPs at its site, but [RFC6514] does not specify that PE2 advertises MSDP SA messages to those MSDP peers for the (C-S,C-G) that it learns via MVPN SA routes. PE2 would need to have an MSDP session with PE1 (that advertised the MVPN SA messages) to learn the sources via MSDP SA messages, for it to advertise the MSDP SA to its local peers. To make things worse, unless blocked by policy control, PE2 would in turn advertise MVPN SA routes because of those MSDP SA messages that it receives from PE1, which are redundant and unnecessary. Also notice that the PE1-PE2 MSDP session is VPN-specific (i.e., only for a single VPN), while the BGP sessions over which the MVPN routes are advertised are not.

If a PE does advertise MSDP SA messages based on received MVPN SA routes, the VPN-specific MSDP sessions with other PEs are no longer needed. Additionally, this MVPN/MSDP SA interoperation has the following inherent benefits for a BGP based solution.

- o MSDP SA refreshes are replaced with BGP hard state.

- o Route Reflectors can be used instead of having peer-to-peer sessions.
- o VPN Extranet [RFC2764] mechanisms can be used to propagate (C-S,C-G) information across VPNs with flexible policy control.

While MSDP Source Active routes contain the source, group and RP addresses of a given multicast flow, MVPN Source Active routes only contain the source and group. MSDP requires the RP address information in order to perform MSDP peer-RPF. Therefore, this document describes how to convey the RP address information into the MVPN Source Active route using an Extended Community so this information can be shared with an existing MSDP infrastructure.

The procedures apply to Global Table Multicast (GTM) [RFC7716] as well.

2.1. MVPN RPT-SPT Mode

For comparison, another method of supporting customer ASM is generally referred to as "rpt-spt" mode. Section "13. Switching from a Shared C-Tree to a Source C-Tree" of [RFC6514] specifies the MVPN SA procedures for that mode, but those SA routes are a replacement for PIM-ASM assert and (s,g,rpt) prune mechanisms, not for source discovery purposes. MVPN/MSDP SA interoperation for the "rpt-spt" mode is outside the scope of this document. In the rest of the document, the "spt-only" mode is assumed.

3. Specification

The MVPN PEs that act as customer RPs or have one or more MSDP sessions in a VPN (or the global table in case of GTM) are treated as an MSDP mesh group for that VPN (or the global table). In the rest of the document, it is referred to as the PE mesh group. This PE mesh group MUST NOT include other MSDP speakers, and is integrated into the rest of MSDP infrastructure for the VPN (or the global table) following normal MSDP rules and practices.

When an MVPN PE advertises an MVPN SA route following procedures in [RFC6514] for the "spt-only" mode, it MUST attach an "MVPN SA RP-address Extended Community". This is a Transitive IPv4-Address-Specific Extended Community. The Local Administrative field is set to zero and the Global Administrative field is set to an RP address determined as the following:

- o If the (C-S,C-G) is learnt as result of PIM Register mechanism, the local RP address for the C-G is used.

- o If the (C-S,C-G) is learnt as result of incoming MSDP SA messages, the RP address in the selected MSDP SA message is used.

In addition to procedures in [RFC6514], an MVPN PE may be provisioned to generate MSDP SA messages from received MVPN SA routes, with or without local policy control. If a received MVPN SA route triggers an MSDP SA message, the MVPN SA route is treated as if a corresponding MSDP SA message was received from within the PE mesh group and normal MSDP procedure is followed (e.g. an MSDP SA message is advertised to other MSDP peers outside the PE mesh group). The (S,G) information comes from the (C-S,C-G) encoding in the MVPN SA NLRI and the RP address comes from the "MVPN SA RP-address EC" mentioned above. If the received MVPN SA route does not have the EC (this could be from a legacy PE that does not have the capability to attach the EC), the local RP address for the C-G is used. In that case, it is possible that the RP inserted into the MSDP SA message for the C-G is actually the MSDP peer to which the generated MSDP message is advertised, causing the peer to discard it due to RPF failure. To get around that problem the peer SHOULD use local policy to accept the MSDP SA message.

An MVPN PE MAY treat only the best MVPN SA route selected by the BGP route selection process (instead of all MVPN SA routes) for a given (C-S,C-G) as a received MSDP SA message (and advertise the corresponding MSDP message). In that case, if the selected best MVPN SA route does not have the "MVPN SA RP-address EC" but another route for the same (C-S, C-G) does, then the next best route with the EC SHOULD be chosen. As a result, when/if the best MVPN SA route with the EC changes, a new MSDP SA message is advertised if the RP address determined according to the newly selected MVPN SA route is different from before. The MSDP SA state associated with the previously advertised MSDP SA message with the older RP address will be timed out.

4. Security Considerations

RFC6514 specifies the procedure for a PE to generate an MVPN SA upon discovering a (C-S,C-G) flow (e.g. via a received MSDP SA message) in a VPN. This document extends this capability in the reverse direction - upon receiving an MVPN SA route in a VPN generate a corresponding MSDP SA and advertise it to MSDP peers in the same VPN. As such, the capabilities specified in this document introduce no additional security considerations beyond those already specified in RFC6514 and RFC3618. Moreover, the capabilities specified in this document actually eliminate the control message amplification that exists today where VPN-specific MSDP sessions are required among the PEs that are customer MSDP peers, which lead to redundant messages (MSDP SAs and MVPN SAs) being carried in parallel between PEs.

5. IANA Considerations

This document introduces a new Transitive IPv4 Address Specific Extended Community "MVPN SA RP-address Extended Community". IANA has registered subcode 0x20 in the Transitive IPv4-Address-Specific Extended Community Sub-Types registry for this EC.

6. Acknowledgements

The authors thank Eric Rosen and Vinod Kumar for their review, comments, questions and suggestions for this document. The authors also thank Yajun Liu for her review and comments.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3618] Fenner, B., Ed. and D. Meyer, Ed., "Multicast Source Discovery Protocol (MSDP)", RFC 3618, DOI 10.17487/RFC3618, October 2003, <<https://www.rfc-editor.org/info/rfc3618>>.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <<https://www.rfc-editor.org/info/rfc6514>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

7.2. Informative References

- [RFC2764] Gleeson, B., Lin, A., Heinanen, J., Armitage, G., and A. Malis, "A Framework for IP Based Virtual Private Networks", RFC 2764, DOI 10.17487/RFC2764, February 2000, <<https://www.rfc-editor.org/info/rfc2764>>.
- [RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <<https://www.rfc-editor.org/info/rfc6513>>.

[RFC7716] Zhang, J., Giuliano, L., Rosen, E., Ed., Subramanian, K.,
and D. Pacella, "Global Table Multicast with BGP Multicast
VPN (BGP-MVPN) Procedures", RFC 7716,
DOI 10.17487/RFC7716, December 2015,
<<https://www.rfc-editor.org/info/rfc7716>>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Lenny Giuliano
Juniper Networks

EMail: lenny@juniper.net

BESS Working Group
Internet Draft
Intended status: Standards Track
Expires: Aug 21, 2021

Y. Liu
China Mobile
F. Guo
Huawei
S. Litkowski
Cisco
X. Liu
Volta Networks
R. Kebler
M. Sivakumar
Juniper
February 21, 2021

Yang Data Model for Multicast in MPLS/BGP IP VPNs
draft-ietf-bess-mvpn-yang-05

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on August 21, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Abstract

This document defines a YANG data model that can be used to configure and manage multicast in MPLS/BGP IP VPNs.

Table of Contents

1. Introduction	2
1.1. Terminology	3
1.2. Tree Diagrams	3
1.3. Prefixes in Data Node Names	4
2. Design of Data Model.....	4
2.1. Scope of Model	4
2.2. Optional Capabilities	4
2.3. Position of Address Family in Hierarchy	5
3. Module Structure	5
4. MVPN YANG Modules	13
5. Security Considerations	36
6. IANA Considerations	38
7. References	39
7.1. Normative References	39
7.2. Informative References	40
8. Acknowledgments	40
Authors' Addresses	41

1. Introduction

YANG [RFC6020] [RFC7950] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. REST) and encoding other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interface, such as CLI and Programmatic APIs.

This document defines a YANG data model that can be used to configure and manage Multicast in MPLS/BGP IP VPN (MVPN). It includes Cisco systems' solution [RFC6037], BGP MVPN [RFC6513] [RFC6514] etc. This model will support the core MVPN protocols, as well as many other features mentioned in separate MVPN RFCs. In addition, Non-core features described in MVPN standards other than mentioned above RFC in separate documents.

1.1. Terminology

The terminology for describing YANG data models is found in [RFC6020] & [RFC7950].

The following abbreviations are used in this document and the defined model:

MVPN: Multicast Virtual Private Network [RFC6513].

PMSI: P-Multicast Service Interface [RFC6513].

PIM: Protocol Independent Multicast [RFC7761].

SM: Sparse Mode [RFC7761].

SSM: Source Specific Multicast [RFC4607].

BIDIR-PIM: Bidirectional Protocol Independent Multicast [RFC5015].

MLDP P2MP: Multipoint Label Distribution Protocol for Point to Multipoint [RFC6388].

MLDP MP2MP: Multipoint Label Distribution Protocol for Multipoint to Multipoint [RFC6388].

RSVP TE P2MP: Resource Reservation Protocol - Traffic Engineering for Point to Multipoint [RFC4875].

BIER: Bit Index Explicit Replication [RFC8279].

1.2. Tree Diagrams

Tree diagrams used in this document follow the notation defined in [RFC8340].

1.3. Prefixes in Data Node Names

In this document, names of data nodes, actions, and other data model objects are often used without a prefix, as long as it is clear from the context in which YANG module each name is defined. Otherwise, names are prefixed using the standard prefix associated with the corresponding YANG module, as shown in Table 1

Prefix	YANG module	Reference
ni	ietf-network-instance	[RFC8529]
l3vpn	ietf-bgp-l3vpn	[I-D.ietf-l3vpn-yang]
inet	ietf-inet-types	[RFC6991]
rt-types	ietf-routing-types	[RFC8294]
acl	ietf-access-control-list	[RFC8519]

Table 1: Prefixes and Corresponding YANG Modules

2. Design of Data Model

2.1. Scope of Model

The model covers Rosen MVPN [RFC6037], BGP MVPN [RFC6513] [RFC6514]. The configuration of MVPN features, and the operational state fields and RPC definitions are not all included in this document of the data model. This model can be extended, though the structure of what has been written may be taken as representative of the structure of the whole model.

This model does not cover other MVPN related protocols such as MVPN Extranet [RFC7900] or MVPN MLDP In-band signaling [RFC7246] etc., these will be specified in separate documents.

2.2. Optional Capabilities

This model is designed to represent the capabilities of MVPN devices with various specifications, including some with basic subsets of the MVPN protocols. The main design goals of this document are that any major now-existing implementation may be said to support the basic model, and that the configuration of all implementations meeting the specification is easy to express through some

combination of the features in the basic model and simple vendor augmentations.

On the other hand, operational state parameters are not so widely designated as features, as there are many cases where the defaulting of an operational state parameter would not cause any harm to the system, and it is much more likely that an implementation without native support for a piece of operational state would be able to derive a suitable value for a state variable that is not natively supported.

For the same reason, wide constant ranges (for example, timer maximum and minimum) will be used in the model. It is expected that vendors will augment the model with any specific restrictions that might be required. Vendors may also extend the features list with proprietary extensions.

2.3. Position of Address Family in Hierarchy

The current draft contains MVPN IPv4 and IPv6 as separate schema branches in the structure. The reason for this is to inherit l3vpn yang model structure and make it easier for implementations which may optionally choose to support specific address families. And the names of some objects may be different between the IPv4 and IPv6 address families.

3. Module Structure

The MVPN YANG model follows the Guidelines for YANG Module Authors (NMDA) [RFC8342]. The operational state data is combined with the associated configuration data in the same hierarchy [RFC8407]. The MVPN modules define for both IPv4 and IPv6 in a two-level hierarchy as listed below:

Instance level: Only including configuration data nodes now. MVPN configuration attributes for the entire routing instance, including route-target, I-PMSI tunnel and S-PMSI number, common timer etc.

PMSI tunnel level: MVPN configuration attributes applicable to the I-PMSI and per S-PMSI tunnel configuration attributes, including tunnel mode, tunnel specific parameters and threshold etc. MVPN PMSI tunnel operational state attributes applicable to the I-PMSI and per S-PMSI tunnel operational state attributes, including tunnel mode, tunnel role, tunnel specific parameters and referenced private source and group address etc.

Where fields are not genuinely essential to protocol operation, they are marked as optional. Some fields will be essential but have a default specified, so that they need not be configured explicitly.

This MVPN model augments `"/ni:network-instances/ni:network-instance/ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv4:"` for IPv4 MVPN service and `"/ni:network-instances/ni:network-instance/ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv6:"` for IPv6 MVPN service specified in [I-D.ietf-l3vpn-yang].

```
augment /ni:network-instances/ni:network-instance/ni:ni-type
  /l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv4:
  +--rw multicast
    +--rw signaling-mode?          enumeration
    +--rw auto-discovery-mode?     enumeration
    +--rw mvpn-type?               enumeration
    +--rw is-sender-site?          boolean {mvpn-sender}?
    +--rw rpt-spt-mode?            enumeration
    +--rw ecmp-load-balance-mode?
      | enumeration {mvpn-ecmp-load-balance}?
    +--rw mvpn-route-targets {mvpn-separate-rt}?
      | +--rw mvpn-route-target* [mvpn-rt-type mvpn-rt-value]
      |   +--rw mvpn-rt-type       enumeration
      |   +--rw mvpn-rt-value      string
    +--rw mvpn-ipmsi-tunnel-ipv4
      | +--rw tunnel-type?                p-tunnel
      | +--rw (ipmsi-tunnel-attribute)?
      |   | +--:(rsvp-te-p2mp)
      |   | | +--rw rsvp-te-p2mp-template?      string
      |   | +--:(mldp-p2mp)
      |   | +--:(pim-ssm)
      |   | | +--rw ssm-default-group-addr?
      |   | |   rt-types:ip-multicast-group-address
      |   | +--:(pim-sm)
      |   | | +--rw sm-default-group-addr?
      |   | |   rt-types:ip-multicast-group-address
      |   | +--:(bidir-pim)
      |   | | +--rw bidir-default-group-addr?
      |   | |   rt-types:ip-multicast-group-address
      |   | +--:(ingress-replication)
      |   | +--:(mldp-mp2mp)
      |   | +--:(bier)
      |   +--rw inclusive-sub-domain-id?      uint8
```

```

|         +---rw inclusive-bitstring-length?    uint16
+---ro (pmsi-tunnel-state-attribute)?
|         +---: (rsvp-te-p2mp)
|         |         +---ro p2mp-id?                uint16
|         |         +---ro tunnel-id?              uint16
|         |         +---ro extend-tunnel-id?       uint16
|         +---: (mldp-p2mp)
|         |         +---ro mldp-root-addr?         inet:ip-address
|         |         +---ro mldp-lsp-id?            string
|         +---: (pim-ssm)
|         |         +---ro ssm-group-addr?         rt-types:ip-multicast-group-address
|         +---: (pim-sm)
|         |         +---ro sm-group-addr?          rt-types:ip-multicast-group-address
|         +---: (bidir-pim)
|         |         +---ro bidir-group-addr?       rt-types:ip-multicast-group-address
|         +---: (ingress-replication)
|         +---: (mldp-mp2mp)
|         +---: (bier)
|         |         +---ro sub-domain-id?          uint8
|         |         +---ro bitstring-length?      uint16
|         |         +---ro bfir-id?               uint16
+---ro tunnel-role?                                enumeration
+---ro upstream-vpn-label?
|         rt-types:mpls-label {mvpn-aggregation-tunnel}?
+---ro mvpn-pmsi-ipv4-ref-sg-entries
|         +---ro mvpn-pmsi-ipv4-ref-sg-entries*
|         |         [ipv4-source-address ipv4-group-address]
|         +---ro ipv4-source-address              inet:ipv4-address
|         +---ro ipv4-group-address
|         |         rt-types:ipv4-multicast-group-address
+---rw mvpn-spmsi-tunnels-ipv4
+---rw switch-delay-time?                          uint8
+---rw switch-back-holddown-time?                  uint16
+---rw tunnel-limit?                              uint16
+---rw mvpn-spmsi-tunnel-ipv4* [tunnel-type]
|         +---rw tunnel-type                                p-tunnel
|         +---rw (spmsi-tunnel-attribute)?
|         |         +---: (rsvp-te-p2mp)
|         |         |         +---rw rsvp-te-p2mp-template?    string

```



```

+---:(p2mp-mldp)
+---:(pim-ssm)
|   +---rw ssm-group-pool-addr?
|       |
|       |   rt-types:ip-multicast-group-address
|   +---rw ssm-group-pool-masklength?      uint8
+---:(pim-sm)
|   +---rw sm-group-pool-addr?
|       |
|       |   rt-types:ip-multicast-group-address
|   +---rw sm-group-pool-masklength?      uint8
+---:(bidir-pim)
|   +---rw bidir-group-pool-addr?
|       |
|       |   rt-types:ip-multicast-group-address
|   +---rw bidir-group-pool-masklength?   uint8
+---:(ingress-replication)
+---:(mldp-mp2mp)
+---:(bier)
|   +---rw selective-sub-domain-id?        uint8
|   +---rw selective-bitstring-length?     uint16
+---rw switch-threshold?                  uint32
+---rw per-item-tunnel-limit?             uint16
+---rw switch-wildcard-mode?
|   enumeration {mvpn-switch-wildcard}?
+---rw explicit-tracking-mode?
|   enumeration {mvpn-explicit-tracking}?
+---rw (address-mask-or-acl)?
|   +---:(address-mask)
|       |   +---rw ipv4-group-addr?
|       |       |
|       |       |   rt-types:ipv4-multicast-group-address
|       |   +---rw ipv4-group-masklength?      uint8
|       |   +---rw ipv4-source-addr?
|       |       |
|       |       |   inet:ipv4-address
|       |   +---rw ipv4-source-masklength?     uint8
|   +---:(acl-name)
|       +---rw group-acl-ipv4?
|           -> /acl:acls/acl/name
+---ro (pmsi-tunnel-state-attribute)?
|   +---:(rsvp-te-p2mp)
|       |   +---ro p2mp-id?                      uint16
|       |   +---ro tunnel-id?                    uint16
|       |   +---ro extend-tunnel-id?             uint16
|   +---:(mldp-p2mp)

```

```

    | | +--ro mldp-root-addr?                inet:ip-address
    | | +--ro mldp-lsp-id?                  string
    | +--:(pim-ssm)
    | | +--ro ssm-group-addr?
    | | | rt-types:ip-multicast-group-address
    | +--:(pim-sm)
    | | +--ro sm-group-addr?
    | | | rt-types:ip-multicast-group-address
    | +--:(bidir-pim)
    | | +--ro bidir-group-addr?
    | | | rt-types:ip-multicast-group-address
    | +--:(ingress-replication)
    | +--:(mldp-mp2mp)
    | +--:(bier)
    | | +--ro sub-domain-id?                uint8
    | | +--ro bitstring-length?            uint16
    | | +--ro bfir-id?                     uint16
    +--ro tunnel-role?                      enumeration
    +--ro upstream-vpn-label?
    | | rt-types:mpls-label {mvpn-aggregation-tunnel}?
    +--ro mvpn-pmsi-ipv4-ref-sg-entries
    | +--ro mvpn-pmsi-ipv4-ref-sg-entries*
    | | [ipv4-source-address ipv4-group-address]
    | | +--ro ipv4-source-address            inet:ipv4-address
    | | +--ro ipv4-group-address
    | | | rt-types:ipv4-multicast-group-address
augment /ni:network-instances/ni:network-instance/ni:ni-type
  /l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv6:
  +--rw multicast
  | +--rw signaling-mode?                  enumeration
  | +--rw auto-discovery-mode?            enumeration
  | +--rw mvpn-type?                      enumeration
  | +--rw is-sender-site?                  boolean {mvpn-sender}?
  | +--rw rpt-spt-mode?                    enumeration
  | +--rw ecmp-load-balance-mode?
  | | enumeration {mvpn-ecmp-load-balance}?
  +--rw mvpn-route-targets {mvpn-separate-rt}?
  | +--rw mvpn-route-target* [mvpn-rt-type mvpn-rt-value]
  | | +--rw mvpn-rt-type                  enumeration
  | | +--rw mvpn-rt-value                 string
  +--rw mvpn-ipmsi-tunnel-ipv6

```

```

+--rw tunnel-type?                                p-tunnel
+--rw (ipmsi-tunnel-attribute)?
|   +--:(rsvp-te-p2mp)
|   |   +--rw rsvp-te-p2mp-template?              string
|   +--:(mldp-p2mp)
|   +--:(pim-ssm)
|   |   +--rw ssm-default-group-addr?
|   |   |   rt-types:ip-multicast-group-address
|   +--:(pim-sm)
|   |   +--rw sm-default-group-addr?
|   |   |   rt-types:ip-multicast-group-address
|   +--:(bidir-pim)
|   |   +--rw bidir-default-group-addr?
|   |   |   rt-types:ip-multicast-group-address
|   +--:(ingress-replication)
|   +--:(mldp-mp2mp)
|   +--:(bier)
|   |   +--rw inclusive-sub-domain-id?              uint8
|   |   +--rw inclusive-bitstring-length?          uint16
+--ro (pmsi-tunnel-state-attribute)?
|   +--:(rsvp-te-p2mp)
|   |   +--ro p2mp-id?                              uint16
|   |   +--ro tunnel-id?                            uint16
|   |   +--ro extend-tunnel-id?                     uint16
|   +--:(mldp-p2mp)
|   |   +--ro mldp-root-addr?                        inet:ip-address
|   |   +--ro mldp-lsp-id?                          string
|   +--:(pim-ssm)
|   |   +--ro ssm-group-addr?
|   |   |   rt-types:ip-multicast-group-address
|   +--:(pim-sm)
|   |   +--ro sm-group-addr?
|   |   |   rt-types:ip-multicast-group-address
|   +--:(bidir-pim)
|   |   +--ro bidir-group-addr?
|   |   |   rt-types:ip-multicast-group-address
|   +--:(ingress-replication)
|   +--:(mldp-mp2mp)
|   +--:(bier)
|   |   +--ro sub-domain-id?                        uint8
|   |   +--ro bitstring-length?                    uint16
|   |   +--ro bfir-id?                             uint16

```

```

+--ro tunnel-role?                               enumeration
+--ro upstream-vpn-label?
|   rt-types:mpls-label {mvpn-aggregation-tunnel}?
+--ro mvpn-pmsi-ipv6-ref-sg-entries
|   +--ro mvpn-pmsi-ipv6-ref-sg-entries*
|   |   [ipv6-source-address ipv6-group-address]
|   |   +--ro ipv6-source-address    inet:ipv6-address
|   |   +--ro ipv6-group-address
|   |       rt-types:ipv6-multicast-group-address
+--rw mvpn-spmsi-tunnels-ipv6
|   +--rw switch-delay-time?                uint8
|   +--rw switch-back-holddown-time?        uint16
|   +--rw tunnel-limit?                     uint16
|   +--rw mvpn-spmsi-tunnel-ipv6* [tunnel-type]
|   |   +--rw tunnel-type                                p-tunnel
|   |   +--rw (spmsi-tunnel-attribute)?
|   |   |   +--:(rsvp-te-p2mp)
|   |   |   |   +--rw rsvp-te-p2mp-template?            string
|   |   |   +--:(p2mp-mldp)
|   |   |   +--:(pim-ssm)
|   |   |   |   +--rw ssm-group-pool-addr?
|   |   |   |       rt-types:ip-multicast-group-address
|   |   |   |   +--rw ssm-group-pool-masklength?        uint8
|   |   |   +--:(pim-sm)
|   |   |   |   +--rw sm-group-pool-addr?
|   |   |   |       rt-types:ip-multicast-group-address
|   |   |   |   +--rw sm-group-pool-masklength?        uint8
|   |   |   +--:(bidir-pim)
|   |   |   |   +--rw bidir-group-pool-addr?
|   |   |   |       rt-types:ip-multicast-group-address
|   |   |   |   +--rw bidir-group-pool-masklength?      uint8
|   |   |   +--:(ingress-replication)
|   |   |   +--:(mldp-mp2mp)
|   |   |   +--:(bier)
|   |   |   |   +--rw selective-sub-domain-id?          uint8
|   |   |   |   +--rw selective-bitstring-length?      uint16
|   |   +--rw switch-threshold?                uint32
|   |   +--rw per-item-tunnel-limit?            uint16
|   |   +--rw switch-wildcard-mode?
|   |   |   enumeration {mvpn-switch-wildcard}?
|   +--rw explicit-tracking-mode?
|   |   enumeration {mvpn-explicit-tracking}?

```

```

+--rw (address-mask-or-acl)?
+--:(address-mask)
+--rw ipv6-group-addr?
|   rt-types:ipv6-multicast-group-address
+--rw ipv6-groupmasklength?      uint8
+--rw ipv6-source-addr?
|   inet:ipv6-address
+--rw ipv6-source-masklength?    uint8
+--:(acl-name)
+--rw group-acl-ipv6?
|   -> /acl:acls/acl/name
+--ro (pmsi-tunnel-state-attribute)?
+--:(rsvp-te-p2mp)
+--ro p2mp-id?                    uint16
+--ro tunnel-id?                  uint16
+--ro extend-tunnel-id?          uint16
+--:(mldp-p2mp)
+--ro mldp-root-addr?            inet:ip-address
+--ro mldp-lsp-id?              string
+--:(pim-ssm)
+--ro ssm-group-addr?
|   rt-types:ip-multicast-group-address
+--:(pim-sm)
+--ro sm-group-addr?
|   rt-types:ip-multicast-group-address
+--:(bidir-pim)
+--ro bidir-group-addr?
|   rt-types:ip-multicast-group-address
+--:(ingress-replication)
+--:(mldp-mp2mp)
+--:(bier)
+--ro sub-domain-id?            uint8
+--ro bitstring-length?        uint16
+--ro bfir-id?                  uint16
+--ro tunnel-role?              enumeration
+--ro upstream-vpn-label?
|   rt-types:mpls-label {mvpn-aggregation-tunnel}?
+--ro mvpn-pmsi-ipv6-ref-sg-entries
+--ro mvpn-pmsi-ipv6-ref-sg-entries*
|   [ipv6-source-address ipv6-group-address]
+--ro ipv6-source-address      inet:ipv6-address

```

```
    +---ro ipv6-group-address
        rt-types:ipv6-multicast-group-address
```

4. MVPN YANG Modules

```
<CODE BEGINS> file ietf-mvpn@2019-12-02.yang
module ietf-mvpn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-mvpn";
  prefix mvpn;

  import ietf-network-instance {
    prefix ni;
  }

  import ietf-bgp-l3vpn {
    prefix l3vpn;
  }

  import ietf-inet-types {
    prefix inet;
  }

  import ietf-routing-types {
    prefix rt-types;
  }

  import ietf-access-control-list {
    prefix acl;
  }

  organization
    "IETF BESS(BGP Enabled Services) Working Group";
  contact
    "
      Yisong Liu
      <mailto:liuyisong.ietf@gmail.com>
      Stephane Litkowski
      <mailto:slitkows@cisco.com>
      Feng Guo
      <mailto:guofeng@huawei.com>
      Xufeng Liu
```

```
<mailto:xufeng.liu.ietf@gmail.com>
Robert Kebler
<mailto:rkebler@juniper.net>
Mahesh Sivakumar
<mailto:sivakumar.mahesh@gmail.com>";
description
  "This YANG module defines the generic configuration
  and operational state data for mvpn, which is common across
  all of the vendor implementations of the protocol. It is
  intended that the module will be extended by vendors to
  define vendor-specific mvpn parameters.";

revision 2019-12-02 {
  description
    "Update the contact information of co-authors.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2019-03-05 {
  description
    "Add bier as a type of P-Tunnel and Errata.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2018-11-08 {
  description
    "Update for leaf type and reference.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2018-05-10 {
  description
    "Update for Model structure and errata.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2017-09-15 {
  description
    "Update for NMDA version and errata.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
```

```
revision 2017-07-03 {
  description
    "Update S-PMSI configuration and errata.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
revision 2016-10-28 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: A YANG Data Model for MVPN";
}
/* Features */
feature mvpn-sender {
  description
    "Support configuration to specify the current PE as the
    sender PE";
}
feature mvpn-separate-rt {
  description
    "Support route-targets configuration of MVPN when they are
    different from the route-targets of unicast L3VPN.";
}
feature mvpn-switch-wildcard {
  description
    "Support configuration to use wildcard mode when multicast
    packets switch from I-PMSI to S-PMSI.";
}
feature mvpn-explicit-tracking {
  description
    "Support configuration to use explicit tracking for leaf PEs
    when multicast packets forward by I-PMSI or S-PMSI.";
}
feature mvpn-aggregation-tunnel {
  description
    "Support more than one VPN multicast service to use the same
    p-tunnel.";
}
feature mvpn-ecmp-load-balance {
  description
    "Support multicast entries in the private network to be
```



```
distributed on the ECMP path of bier in the public
network.";
}
```

```
typedef p-tunnel {
  type enumeration {
    enum no-tunnel-present {
      value 0;
      description "No tunnel information present";
    }
    enum rsvp-te-p2mp {
      value 1;
      description "RSVP TE P2MP tunnel";
    }
    enum mldp-p2mp {
      value 2;
      description "MLDP P2MP tunnel";
    }
    enum pim-ssm {
      value 3;
      description "PIM SSM tree in public net";
    }
    enum pim-sm {
      value 4;
      description "PIM SM tree in public net";
    }
    enum bidir-pim {
      value 5;
      description "BIDIR-PIM tree in public net";
    }
    enum ingress-replication {
      value 6;
      description "Ingress Replication p2p tunnel.";
    }
    enum mldp-mp2mp {
      value 7;
      description "MLDP MP2MP tunnel";
    }
    enum bier {
      value 11;
      description "bier underlay";
    }
  }
}
```

```
    }
    description "Provider tunnel type definition.";
  }

  grouping mvpn-instance-config {
    description "Mvpn basic configuration per instance.";

    leaf signaling-mode {
      type enumeration {
        enum none {
          value 0;
          description "No signaling";
        }
        enum bgp {
          value 1;
          description "bgp signaling";
        }
        enum pim {
          value 2;
          description "pim signaling";
        }
        enum mldp-in-band {
          value 3;
          description "mldp in-band signaling";
        }
      }
      default "none";
      description "Signaling mode for C-multicast route.";
    }
    leaf auto-discovery-mode {
      type enumeration {
        enum none {
          value 0;
          description "no auto-discovery signaling";
        }
        enum pim {
          value 1;
          description "auto-discovery by PIM signaling";
        }
        enum bgp {
          value 2;
          description "auto-discovery by BGP signaling";
        }
      }
    }
  }
}
```

```
    }
  }
  default "none";
  description "Auto discovery mode of MVPN PE members.";
}
leaf mvpn-type {
  type enumeration {
    enum rosen-mvpn {
      value 0;
      description "Rosen mvpn mode referenced RFC6037";
    }
    enum ng-mvpn {
      value 1;
      description
        "BGP/MPLS mvpn mode referenced RFC6513&RFC6514";
    }
  }
  default "ng-mvpn";
  description
    "Mvpn type, which can be rosen mvpn mode or ng mvpn mode.";
}
leaf is-sender-site {
  if-feature mvpn-sender;
  type boolean;
  default false;
  description "Configure the current PE as a sender PE.";
}
leaf rpt-spt-mode {
  type enumeration {
    enum spt-only {
      value 0;
      description
        "Only spt entries can cross the public net.";
    }
    enum rpt-spt {
      value 1;
      description
        "Both rpt and spt entries can corss the public net.";
    }
  }
  description
    "ASM mode in multicast private network for crossing
```

```
        public net.";
    }
    leaf ecmp-load-balance-mode {
        if-feature mvpn-ecmp-load-balance;
        type enumeration {
            enum none {
                value 0;
                description
                    "No load balancing for multicast entries.";
            }
            enum source {
                value 1;
                description
                    "Load balancing based on multicast source address.";
            }
            enum group {
                value 2;
                description
                    "Load balancing based on multicast group address.";
            }
            enum source-group {
                value 3;
                description
                    "Load balancing based on multicast source and group
                    address.";
            }
        }
        description
            "Distribution mode of multicast entries in the private
            network on the ECMP path of bier in the public network.";
    }
}/* mvpn-instance-config */

grouping mvpn-rt {
    description
        "May be different from l3vpn unicast route-targets.";
    container mvpn-route-targets {
        if-feature mvpn-separate-rt;
        description "Multicast vpn route-targets";
        list mvpn-route-target {
            key "mvpn-rt-type mvpn-rt-value" ;
            description

```

```
    "List of multicast route-targets" ;
leaf mvpn-rt-type {
  type enumeration {
    enum export-extcommunity {
      value 0;
      description "export-extcommunity";
    }
    enum import-extcommunity {
      value 1;
      description "import-extcommunity";
    }
  }
  description
    "rt types are as follows:
    export-extcommunity: specifies the value of
    the extended community attribute of the
    route from an outbound interface to the
    destination vpn.
    import-extcommunity: receives routes that
    carry the specified extended community
    attribute";
}
leaf mvpn-rt-value {
  type string {
    length "3..21";
  }
  description
    "the available mvpn target formats are as
    follows:
    - 16-bit as number:32-bit user-defined
    number, for example, 1:3. an as number
    ranges from 0 to 65535, and a user-defined
    number ranges from 0 to 4294967295. The as
    number and user-defined number cannot be
    both 0s. That is, a vpn target cannot be 0:0.
    - 32-bit ip address:16-bit user-defined
    number, for example, 192.168.122.15:1.
    The ip address ranges from 0.0.0.0 to
    255.255.255.255, and the user-defined
    number ranges from 0 to 65535.";
}
}
```

```
    }  
  }  
  
  grouping mvpn-ipmsi-tunnel-config {  
    description  
      "Configuration of default mdt for rosen mvpn  
      and I-PMSI for ng mvpn";  
    leaf tunnel-type {  
      type p-tunnel;  
      description "I-PMSI tunnel type.";  
    }  
    choice ipmsi-tunnel-attribute {  
      description "I-PMSI tunnel attributes configuration";  
      case rsvp-te-p2mp {  
        description "RSVP TE P2MP tunnel";  
        leaf rsvp-te-p2mp-template {  
          type string {  
            length "1..31";  
          }  
          description "RSVP TE P2MP tunnel template";  
        }  
      }  
      case mldp-p2mp {  
        description "MLDP P2MP tunnel";  
      }  
      case pim-ssm {  
        description "PIM SSM tree in the public net";  
        leaf ssm-default-group-addr {  
          type rt-types:ip-multicast-group-address;  
          description  
            "Default mdt or I-PMSI group address for SSM mode.";  
        }  
      }  
      case pim-sm {  
        description "PIM SM tree in the public net";  
        leaf sm-default-group-addr {  
          type rt-types:ip-multicast-group-address;  
          description  
            "Default mdt or I-PMSI group address for SM mode.";  
        }  
      }  
    }  
    case bidir-pim {
```

```
        description "BIDIR PIM tree in the public net";
        leaf bidir-default-group-addr {
            type rt-types:ip-multicast-group-address;
            description
                "Default mdt or I-PMSI group address for BIDIR mode.";
        }
    }
    case ingress-replication {
        description "Ingress replication p2p tunnel";
    }
    case mldp-mp2mp {
        description "MLDP MP2MP tunnel";
    }
    case bier {
        description "bier underlay";
        leaf inclusive-sub-domain-id {
            type uint8;
            description "Subdomain ID of bier.";
        }
        leaf inclusive-bitstring-length {
            type uint16 {
                range "64|128|256|512|1024|2048|4096";
            }
            description "BitString length of bier underlay.";
        }
    }
}
}
}
/* mvpn-ipmsi-tunnel-config */

grouping mvpn-spmsi-tunnel-per-item-config {
    description "S-PMSI tunnel basic configuration";
    leaf tunnel-type {
        type p-tunnel;
        description "S-PMSI tunnel type.";
    }
    choice spmsi-tunnel-attribute {
        description "S-PMSI tunnel attributes configuration";
        case rsvp-te-p2mp {
            description "RSVP TE P2MP tunnel";
            leaf rsvp-te-p2mp-template {
                type string {
                    length "1..31";
                }
            }
        }
    }
}
```

```
    }
    description "RSVP TE P2MP tunnel template";
  }
}
case p2mp-mldp {
  description "MLDP P2MP tunnel";
}
case pim-ssm {
  description "PIM SSM tree in the public net";
  leaf ssm-group-pool-addr {
    type rt-types:ip-multicast-group-address;
    description
      "Group pool address for data mdt or s-pmsi in SSM
mode";
  }
  leaf ssm-group-pool-masklength {
    type uint8 {
      range "8..128";
    }
    description
      "Group pool mask length for data mdt or s-pmsi in
      SSM mode";
  }
}
case pim-sm {
  description "PIM SM tree in the public net";
  leaf sm-group-pool-addr {
    type rt-types:ip-multicast-group-address;
    description
      "Group pool address for data mdt or s-pmsi in SM mode";
  }
  leaf sm-group-pool-masklength {
    type uint8 {
      range "8..128";
    }
    description
      "Group pool mask length for data mdt or s-pmsi in
      SM mode";
  }
}
case bidir-pim {
  description "BIDIR PIM tree in the public net";
```



```
    leaf bidir-group-pool-addr {
      type rt-types:ip-multicast-group-address;
      description
        "Group pool address for data mdt or s-pmsi in
        BIDIR mode";
    }
    leaf bidir-group-pool-masklength {
      type uint8 {
        range "8..128";
      }
      description
        "Group pool mask length for data mdt or s-pmsi in
        BIDIR mode";
    }
  }
  case ingress-replication {
    description "Ingress replication p2p tunnel";
  }
  case mldp-mp2mp {
    description "MLDP MP2MP tunnel";
  }
  case bier {
    description "bier underlay";
    leaf selective-sub-domain-id {
      type uint8;
      description "Subdomain ID of bier.";
    }
    leaf selective-bitstring-length {
      type uint16 {
        range "64|128|256|512|1024|2048|4096";
      }
      description "BitString length of bier underlay.";
    }
  }
}
leaf switch-threshold {
  type uint32 {
    range "0..4194304";
  }
  units kbps;
  default 0;
  description
```

```
    "Multicast packet rate threshold for
    triggering the switching from the
    I-PMSI to the S-PMSI. The value is
    an integer ranging from 0 to 4194304, in
    kbps. The default value is 0.";
  }
  leaf per-item-tunnel-limit {
    type uint16 {
      range "1..1024";
    }
    description
      "Maximum number of S-PMSI tunnels allowed
      per S-PMSI configuration item per mvpn instance.";
  }
  leaf switch-wildcard-mode {
    if-feature mvpn-switch-wildcard;
    type enumeration {
      enum source-group {
        value 0;
        description
          "Wildcard neither for source or group address.";
      }
      enum star-star {
        value 1;
        description
          "Wildcard for both source and group address.";
      }
      enum star-group {
        value 2;
        description
          "Wildcard only for source address.";
      }
      enum source-star {
        value 3;
        description
          "Wildcard only for group address.";
      }
    }
    description
      "I-PMSI switching to S-PMSI mode for private net
      wildcard mode, which including (*,*), (*,G), (S,*),
      (S,G) four modes.";
```

```
    }
    leaf explicit-tracking-mode {
      if-feature mvpn-explicit-tracking;
      type enumeration {
        enum no-leaf-info-required {
          value 0;
          description "No need to track leaf information.";
        }
        enum leaf-info-required {
          value 1;
          description "Need to track leaf information.";
        }
        enum leaf-info-required-per-flow {
          value 2;
          description
            "Need to track leaf information based on
             per multicast flow.";
        }
      }
      description "Tracking mode for leaf information.";
    }
  }/* mvpn-spmsi-tunnel-per-item-config */

grouping mvpn-spmsi-tunnel-common-config {
  description
    "Data mdt for rosen mvpn or S-PMSI for ng mvpn configuration
     attributes for both IPv4 and IPv6 private network";
  leaf switch-delay-time {
    type uint8 {
      range "3..60";
    }
    units seconds;
    default 5;
    description
      "Delay for switching from the I-PMSI to
       the S-PMSI. The value is an integer
       ranging from 3 to 60, in seconds. ";
  }
  leaf switch-back-holddown-time {
    type uint16 {
      range "0..512";
    }
  }
}
```

```
        units seconds;
        default 60;
        description
            "Delay for switching back from the S-PMSI
             to the I-PMSI. The value is an integer
             ranging from 0 to 512, in seconds. ";
    }
    leaf tunnel-limit {
        type uint16 {
            range "1..8192";
        }
        description
            "Maximum number of s-pmsi tunnels allowed
             per mvpn instance.";
    }
}/* mvpn-spmsi-tunnel-common-config */

grouping mvpn-pmsi-state {
    description "PMSI tunnel operational state information";

    choice pmsi-tunnel-state-attribute {
        config false;
        description
            "PMSI tunnel operational state information for each type";
        case rsvp-te-p2mp {
            description "RSVP TE P2MP tunnel";
            leaf p2mp-id {
                type uint16 {
                    range "0..65535";
                }
                description "P2MP ID of the RSVP TE P2MP tunnel";
            }
            leaf tunnel-id {
                type uint16 {
                    range "1..65535";
                }
                description "Tunnel ID of the RSVP TE P2MP tunnel";
            }
            leaf extend-tunnel-id {
                type uint16 {
                    range "1..65535";
                }
            }
        }
    }
}
```

```
        description
            "Extended tunnel ID of the RSVP TE P2MP Tunnel";
    }
}
case mldp-p2mp {
    description "MLDP P2MP tunnel";
    leaf mldp-root-addr {
        type inet:ip-address;
        description "IP address of the root of a MLDP P2MP lsp.";
    }
    leaf mldp-lsp-id {
        type string {
            length "1..256";
        }
        description "MLDP P2MP lsp ID.";
    }
}
case pim-ssm {
    description "PIM SSM tree in the public net";
    leaf ssm-group-addr {
        type rt-types:ip-multicast-group-address;
        description "Group address for pim ssm";
    }
}
case pim-sm {
    description "PIM SM tree in the public net";
    leaf sm-group-addr {
        type rt-types:ip-multicast-group-address;
        description "Group address for pim sm";
    }
}
case bidir-pim {
    description "BIDIR PIM tree in the public net";
    leaf bidir-group-addr {
        type rt-types:ip-multicast-group-address;
        description "Group address for bidir-pim";
    }
}
case ingress-replication {
    description "Ingress replication p2p tunnel";
}
case mldp-mp2mp {
```

```
        description "MLDP MP2MP tunnel";
    }
    case bier {
        description "bier underlay";
        leaf sub-domain-id {
            type uint8;
            description "Subdomain ID of bier.";
        }
        leaf bitstring-length {
            type uint16 {
                range "64|128|256|512|1024|2048|4096";
            }
            description "BitString length of bier underlay.";
        }
        leaf bfir-id {
            type uint16;
            description "ID of BIER sender PE of MVPN.";
        }
    }
}
leaf tunnel-role {
    type enumeration {
        enum none {
            value 0;
            description "none";
        }
        enum root {
            value 1;
            description "root";
        }
        enum leaf {
            value 2;
            description "leaf";
        }
        enum root-and-leaf {
            value 3;
            description "root-and-leaf";
        }
    }
    config false;
    description "Role of a node for a p-tunnel.";
}
```

```
    leaf upstream-vpn-label {
      if-feature mvpn-aggregation-tunnel;
      type rt-types:mpls-label;
      config false;
      description
        "VPN context label for the multicast data of the VPN instance
         in an aggregation P-tunnel.";
    }
  }/* mvpn-pmsi-state */

  grouping mvpn-pmsi-ipv4-entry {
    description
      "Multicast entries in ipv4 mvpn referenced the pmsi tunnel";
    container mvpn-pmsi-ipv4-ref-sg-entries {
      config false;
      description
        "Multicast entries in ipv4 mvpn referenced the pmsi
tunnel";
      list mvpn-pmsi-ipv4-ref-sg-entries {
        key "ipv4-source-address ipv4-group-address";
        description
          "IPv4 source and group address of private network entry";
        leaf ipv4-source-address {
          type inet:ipv4-address;
          description
            "IPv4 source address of private network entry
             in I-PMSI or S-PMSI.";
        }
        leaf ipv4-group-address {
          type rt-types:ipv4-multicast-group-address;
          description
            "IPv4 group address of private network entry
             in I-PMSI or S-PMSI.";
        }
      }
    }
  }/* mvpn-pmsi-ipv4-entry */

  grouping mvpn-pmsi-ipv6-entry {
    description
      "Multicast entries in ipv6 mvpn referenced the pmsi tunnel";
```

```
    container mvpn-pmsi-ipv6-ref-sg-entries {
        config false;
        description
            "Multicast entries in ipv6 mvpn referenced the pmsi
tunnel";
        list mvpn-pmsi-ipv6-ref-sg-entries {
            key "ipv6-source-address ipv6-group-address";
            description
                "IPv6 source and group address of private network entry";
            leaf ipv6-source-address {
                type inet:ipv6-address;
                description
                    "IPv6 source address of private network entry
                    in I-PMSI or S-PMSI.";
            }
            leaf ipv6-group-address {
                type rt-types:ipv6-multicast-group-address;
                description
                    "IPv6 group address of private network entry
                    in I-PMSI or S-PMSI.";
            }
        }
    }
}
}/* mvpn-pmsi-ipv6-entry */

grouping mvpn-ipmsi-tunnel-info-ipv4 {
    description
        "Default mdt or I-PMSI configuration and
        operational state information";
    container mvpn-ipmsi-tunnel-ipv4 {
        description
            "Default mdt or I-PMSI configuration and
            operational state information";
        uses mvpn-ipmsi-tunnel-config;
        uses mvpn-pmsi-state;
        uses mvpn-pmsi-ipv4-entry;
    }
}

grouping mvpn-ipmsi-tunnel-info-ipv6 {
    description
        "Default mdt or I-PMSI configuration and
```



```
        operational state information";
    container mvpn-ipmsi-tunnel-ipv6 {
        description
            "Default mdt or I-PMSI configuration and
            operational state information";
        uses mvpn-ipmsi-tunnel-config;
        uses mvpn-pmsi-state;
        uses mvpn-pmsi-ipv6-entry;
    }
}

grouping mvpn-spmsi-tunnel-info-ipv4 {
    description
        "Data mdt for rosen mvpn or S-PMSI for ng mvpn in
        IPv4 private network";

    container mvpn-spmsi-tunnels-ipv4 {
        description
            "S-PMSI tunnel configuration and
            operational state information.";
        uses mvpn-spmsi-tunnel-common-config;

        list mvpn-spmsi-tunnel-ipv4 {
            key "tunnel-type";
            description
                "S-PMSI tunnel attributes configuration and
                operational state information.";

            uses mvpn-spmsi-tunnel-per-item-config;
            choice address-mask-or-acl {
                description
                    "Type of definition of private network
                    multicast address range";
                case address-mask {
                    description "Use the type of address and mask";
                    leaf ipv4-group-addr {
                        type rt-types:ipv4-multicast-group-address;
                        description
                            "Start address of the IPv4 group
                            address range in private net. ";
                    }
                    leaf ipv4-group-masklength {
```

```
    type uint8 {
      range "4..32";
    }
    description
      "Group mask length for the IPv4
      group address range in private net.";
  }
  leaf ipv4-source-addr {
    type inet:ipv4-address;
    description
      "Start address of the IPv4 source
      address range in private net.";
  }
  leaf ipv4-source-masklength {
    type uint8 {
      range "0..32";
    }
    description
      "Source mask length for the IPv4
      source address range in private net.";
  }
}
case acl-name {
  description "Use the type of acl";
  leaf group-acl-ipv4 {
    type leafref {
      path "/acl:acls/acl:acl/acl:name";
    }
    description
      "Specify the (s, g) entry on which the
      S-PMSI tunnel takes effect.
      The value is an integer ranging from 3000
      to 3999 or a string of 32 case-sensitive
      characters. If no value is specified, the
      switch-group address pool takes effect on
      all (s, g).";
  }
}
}
uses mvpn-pmsi-state;
uses mvpn-pmsi-ipv4-entry;
}/* list mvpn-spmsi-tunnel-ipv4 */
```

```
    }/* container mvpn-spmsi-tunnels-ipv4 */
  }/* grouping mvpn-spmsi-tunnel-info-ipv4 */
  grouping mvpn-spmsi-tunnel-info-ipv6 {
    description
      "Data mdt for rosen mvpn or S-PMSI for ng mvpn in
      IPv6 private network";

    container mvpn-spmsi-tunnels-ipv6 {
      description
        "S-PMSI tunnel configuration and
        operational state information.";
      uses mvpn-spmsi-tunnel-common-config;

      list mvpn-spmsi-tunnel-ipv6 {
        key "tunnel-type";
        description
          "S-PMSI tunnel attributes configuration and
          operational state information.";
        uses mvpn-spmsi-tunnel-per-item-config;

        choice address-mask-or-acl {
          description
            "Type of definition of private network
            multicast address range";
          case address-mask {
            description "Use the type of address and mask";

            leaf ipv6-group-addr {
              type rt-types:ipv6-multicast-group-address;
              description
                "Start address of the IPv6 group
                address range in private net. ";
            }
            leaf ipv6-groupmasklength {
              type uint8 {
                range "8..128";
              }
              description
                "Group mask length for the IPv6
                group address range in private net.";
            }
          }
          leaf ipv6-source-addr {
```

```
    type inet:ipv6-address;
    description
      "Start address of the IPv6 source
       address range in private net.";
  }
  leaf ipv6-source-masklength {
    type uint8 {
      range "0..128";
    }
    description
      "Source mask length for the IPv6
       source address range in private net.";
  }
}
case acl-name {
  description "Use the type of acl";
  leaf group-acl-ipv6 {
    type leafref {
      path "/acl:acls/acl:acl/acl:name";
    }
    description
      "Specify the (s, g) entry on which the
       S-PMSI tunnel takes effect.
       The value is an integer ranging from 3000
       to 3999 or a string of 32 case-sensitive
       characters. If no value is specified, the
       switch-group address pool takes effect on
       all (s, g).";
  }
}
}
uses mvpn-pmsi-state;
uses mvpn-pmsi-ipv6-entry;
}/* list mvpn-spmsi-tunnel-ipv6 */
}/* container mvpn-spmsi-tunnels-ipv6 */
}/* grouping mvpn-spmsi-tunnel-info-ipv6 */

augment "/ni:network-instances/ni:network-instance/ni:ni-type/"
  +"l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv4" {
  description
    "Augment l3vpn ipv4 container for per multicast VRF
     configuration and operational state.";
```

```
    container multicast {
      description
        "Configuration and operational state of multicast IPv4 vpn
        specific parameters";
      uses mvpn-instance-config;
      uses mvpn-rtts;
      uses mvpn-ipmsi-tunnel-info-ipv4;
      uses mvpn-spmsi-tunnel-info-ipv4;
    }
  }

  augment "/ni:network-instances/ni:network-instance/ni:ni-type/"
    +"l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv6" {
    description
      "Augment l3vpn ipv6 container for per multicast VRF
      configuration and operational state.";
    container multicast {
      description
        "Configuration and operational state of multicast IPv6 vpn
        specific parameters";
      uses mvpn-instance-config;
      uses mvpn-rtts;
      uses mvpn-ipmsi-tunnel-info-ipv6;
      uses mvpn-spmsi-tunnel-info-ipv6;
    }
  }
}
<CODE ENDS>
```

5. Security Considerations

The YANG module specified in this document defines a schema for data that is designed to be accessed via network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest NETCONF layer is the secure transport layer, and the mandatory-to-implement secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the mandatory-to-implement secure transport is TLS [RFC8446].

The Network Configuration Access Control Model (NACM) [RFC8341] provides the means to restrict access for particular NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

Under /ni:network-instances/ni:network-instance/ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv4:,

multicast:

This subtree specifies the configuration for the IPv4 MVPN attributes at the instance level on a MVPN instance. Modifying the configuration can cause IPv4 MVPN PMSI tunnels to be deleted or reconstructed on the MVPN instance.

multicast:mvpn-ipmsi-tunnel-ipv4

This subtree specifies the configuration for the IPv4 MVPN I-PMSI tunnel attributes at the PMSI tunnel level on a MVPN instance. Modifying the configuration can cause IPv4 MVPN I-PMSI tunnel to be deleted or reconstructed on the MVPN instance.

multicast:mvpn-spmsi-tunnels-ipv4

This subtree specifies the configuration for the IPv4 MVPN S-PMSI attributes at the PMSI tunnel level on a MVPN instance. Modifying the configuration can cause IPv4 MVPN S-PMSI tunnels to be deleted or reconstructed on the MVPN instance.

Under /ni:network-instances/ni:network-instance/ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv6:,

multicast:

This subtree specifies the configuration for the IPv6 MVPN attributes at the instance level on a MVPN instance. Modifying the configuration can cause IPv6 MVPN PMSI tunnels to be deleted or reconstructed on the MVPN instance.

multicast:mvpn-ipmsi-tunnel-ipv6

This subtree specifies the configuration for the IPv6 MVPN I-PMSI tunnel attributes at the PMSI tunnel level on a MVPN instance. Modifying the configuration can cause IPv6 MVPN I-PMSI tunnel to be deleted or reconstructed on the MVPN instance.

multicast:mvpn-spmsi-tunnels-ipv6

This subtree specifies the configuration for the IPv6 MVPN S-PMSI attributes at the PMSI tunnel level on a MVPN instance. Modifying the configuration can cause IPv6 MVPN S-PMSI tunnels to be deleted or reconstructed on the MVPN instance.

Unauthorized access to any data node of these subtrees can adversely affect the PMSI tunnels of the MVPN instances on the local device. This may lead to network malfunctions, delivery of packets to inappropriate destinations, and other problems.

Some of the readable data nodes in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control read access (e.g., via get, get-config, or notification) to these data nodes. These are the subtrees and data nodes and their sensitivity/vulnerability:

/ni:network-instances/ni:network-instance/ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv4/multicast

/ni:network-instances/ni:network-instance/ni:ni-type/l3vpn:l3vpn/l3vpn:l3vpn/l3vpn:ipv6/multicast

Unauthorized access to any data node of the above subtree can disclose the operational state information of MVPN on this device.

6. IANA Considerations

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

URI: urn:ietf:params:xml:ns:yang:ietf-mvpn

Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

Name: ietf-mvpn

Namespace: urn:ietf:params:xml:ns:yang:ietf-mvpn

Prefix: mvpn

Reference: RFCXXX

7. References

7.1. Normative References

- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010
- [RFC6037] Rosen, E., Cai, Y., and IJ. Wijnands, "Cisco Systems' Solution for Multicast in BGP/MPLS IP VPNs", RFC 6037, October 2010.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011
- [RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, July 2013
- [RFC7246] IJ. Wijnands, P. Hitchen, N. Leymann, W. Henderickx, A. Gulko and J. Tantsura, " Multipoint Label Distribution Protocol In-Band Signaling in a Virtual Routing and Forwarding (VRF) Table Context ", RFC 7246, June 2014.
- [RFC7900] Y. Rekhter, E. Rosen, R. Aggarwal, Arkatan, Y. Cai and T. Morin, " Extranet Multicast in BGP/IP MPLS VPNs ", RFC 7900, June 2016.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, November 2017
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294, December 2017

- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, March 2018
- [RFC8519] M. Jethanandani, S. Agarwal, L. Huang and D. Blair, "Yang Data Model for Network Access Control Lists (ACL) ", RFC8519, March 2019
- [RFC8529] L. Berger, C. Hopps, A. Lindem, D. Bogdanovic and X. Liu, "YANG Data Model for Network Instances", RFC8529, March 2019.
- [I-D.ietf-l3vpn-yang] D. Jain, K. Patel, P. Brissette, Z. Li, S. Zhuang, X. Liu, J. Haas, S. Esale and B. Wen, "Yang Data Model for BGP/MPLS L3 VPNs", draft-ietf-bess-l3vpn-yang-04(expired), October 2018.

7.2. Informative References

- [RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, March 2018
- [RFC8407] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC8407, October 2018

8. Acknowledgments

The authors would like to thank the following for their valuable contributions of this document:

TBD

Authors' Addresses

Yisong Liu
China Mobile
China
Email: liuyisong@chinamobile.com

Feng Guo
Huawei Technologies
China
Email: guofeng@huawei.com

Stephane Litkowski
Cisco Systems

Email: slitkows.ietf@gmail.com

Xufeng Liu
Volta Networks

Email: xufeng.liu.ietf@gmail.com

Robert Kebler
Juniper Networks
USA
Email: rkebler@juniper.net

Mahesh Sivakumar
Juniper Networks
USA
Email: sivakumar.mahesh@gmail.com

BESS Workgroup
Internet-Draft
Intended status: Standards Track
Expires: June 23, 2019

P. Jain, Ed.
S. Salam
A. Sajassi
Cisco Systems, Inc.
S. Boutros
VmWare, Inc.
G. Mirsky
ZTE Corporation.
December 20, 2018

LSP-Ping Mechanisms for EVPN and PBB-EVPN
draft-jain-bess-evpn-lsp-ping-08

Abstract

LSP-Ping is a widely deployed Operation, Administration, and Maintenance (OAM) mechanism in MPLS networks. This document describes mechanisms for detecting data-plane failures using LSP Ping in MPLS based EVPN and PBB-EVPN networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on June 23, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Specification of Requirements	3
3. Terminology	3
4. Proposed Target FEC Stack Sub-TLVs	3
4.1. EVPN MAC Sub-TLV	4
4.2. EVPN Inclusive Multicast Sub-TLV	4
4.3. EVPN Auto-Discovery Sub-TLV	5
4.4. EVPN IP Prefix Sub-TLV	6
5. Encapsulation of OAM Ping Packets	7
6. Operations	7
6.1. Unicast Data-plane connectivity checks	7
6.2. Inclusive Multicast Data-plane Connectivity Checks	8
6.2.1. Ingress Replication	9
6.2.2. Using P2MP P-tree	10
6.2.3. Controlling Echo Responses when using P2MP P-tree	11
6.3. EVPN Aliasing Data-plane connectivity check	11
6.4. EVPN IP Prefix (RT-5) Data-plane connectivity check	11
7. Security Considerations	12
8. IANA Considerations	12
8.1. Sub-TLV Type	12
8.2. Proposed new Return Codes	12
9. Acknowledgments	12
10. References	13
10.1. Normative References	13
10.2. Informative References	13
Authors' Addresses	14

1. Introduction

[RFC7432] describes MPLS based Ethernet VPN (EVPN) technology. An EVPN comprises CE(s) connected to PE(s). The PEs provide layer 2 EVPN among the CE(s) over the MPLS core infrastructure. In EVPN networks, PEs advertise the MAC addresses learned from the locally connected CE(s), along with MPLS Label, to remote PE(s) in the control plane using multi-protocol BGP. EVPN enables multi-homing of CE(s) connected to multiple PEs and load balancing of traffic to and from multi-homed CE(s).

[RFC7623] describes the use of Provider Backbone Bridging [802.1ah] with EVPN. PBB-EVPN maintains the C-MAC learning in data plane and

only advertises Provider Backbone MAC (B-MAC) addresses in control plane using BGP.

Procedures for simple and efficient mechanisms to detect data-plane failures using LSP Ping in MPLS network are well defined in [RFC8029][RFC6425]. This document defines procedures to detect data-plane failures using LSP Ping in MPLS networks deploying EVPN and PBB-EVPN. This draft defines 4 new Sub-TLVs for Target FEC Stack TLV with the purpose of identifying the FEC on the Peer PE.

2. Specification of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

AD: Auto Discovery

B-MAC: Backbone MAC Address

CE: Customer Edge Device

C-MAC: Customer MAC Address

DF: Designated Forwarder

ESI: Ethernet Segment Identifier

EVI: EVPN Instance Identifier that globally identifies the EVPN Instance

EVPN: Ethernet Virtual Private Network

MPLS-OAM: MPLS Operations, Administration, and Maintenance

P2MP: Point-to-Multipoint

PBB: Provider Backbone Bridge

PE: Provider Edge Device

4. Proposed Target FEC Stack Sub-TLVs

This document introduces four new Target FEC Stack sub-TLVs that are included in the LSP-Ping Echo Request packet sent for detecting

faults in data-plane connectivity in EVPN and PBB-EVPN networks. These Target FEC Stack sub-TLVs are described next.

4.1. EVPN MAC Sub-TLV

The EVPN MAC sub-TLV is used to identify the MAC for an EVI under test at a peer PE.

The EVPN MAC sub-TLV fields are derived from the MAC/IP advertisement route defined in [RFC7432] Section 7.2 and have the format as shown in Figure 1. This TLV is included in the Echo Request sent to the Peer PE by the PE that is the originator of the request.

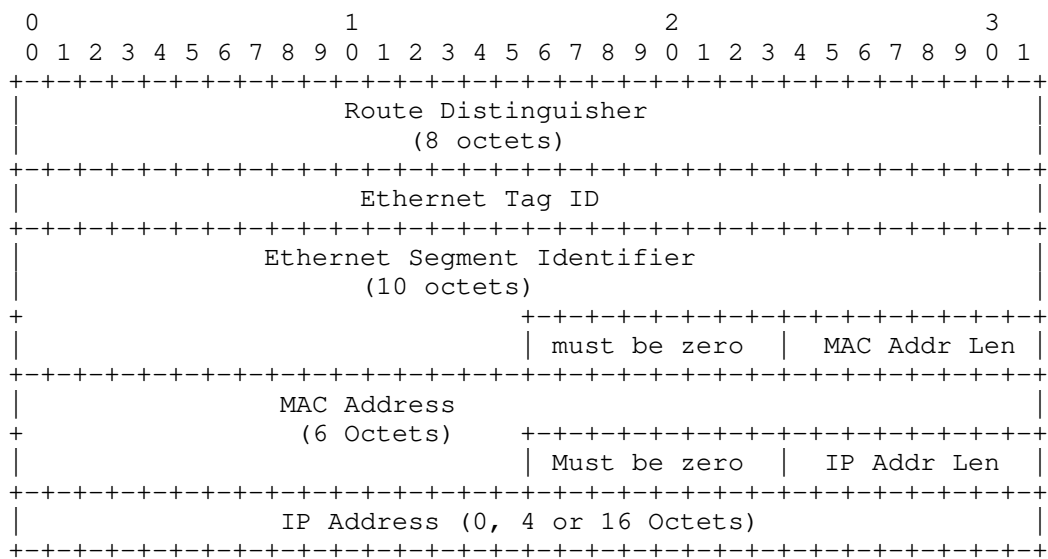


Figure 1: EVPN MAC sub-TLV format

The LSP Ping echo request is sent using the EVPN MPLS label(s) associated with the MAC route announced by a remote PE and the MPLS transport label(s) to reach the remote PE.

4.2. EVPN Inclusive Multicast Sub-TLV

The EVPN Inclusive Multicast sub-TLV fields are based on the EVPN Inclusive Multicast route defined in [RFC7432] Section 7.3.

The EVPN Inclusive Multicast sub-TLV has the format as shown in Figure 2. This TLV is included in the echo request sent to the EVPN

peer PE by the originator of request to verify the multicast connectivity state on the peer PE(s) in EVPN and PBB-EVPN.

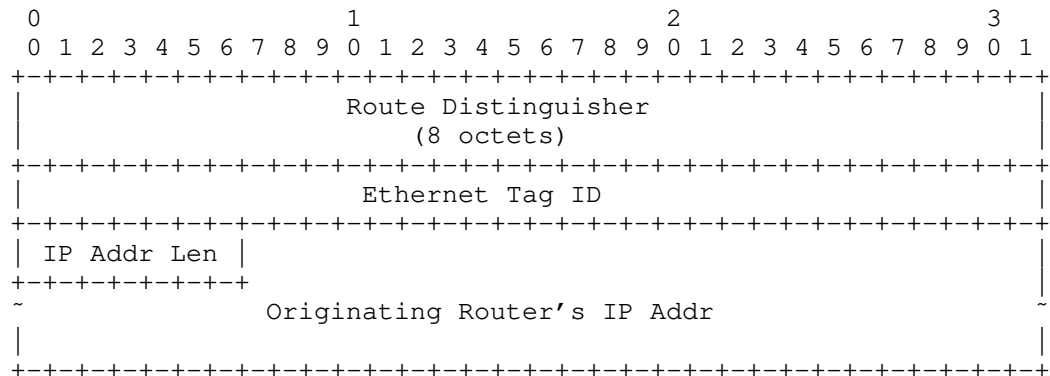


Figure 2: EVPN Inclusive Multicast sub-TLV format

Broadcast, multicast, and unknown unicast traffic can be sent using ingress replication or P2MP P-tree in EVPN and PBB-EVPN network. In case of ingress replication, the Echo Request is sent using a label stack of [Transport label, Inclusive Multicast label] to each remote PE participating in EVPN or PBB-EVPN. The inclusive multicast label is the downstream assigned label announced by the remote PE to which the Echo Request is being sent. The Inclusive Multicast label is the inner label in the MPLS label stack.

When using P2MP P-tree in EVPN or PBB-EVPN, the Echo Request is sent using P2MP P-tree transport label for inclusive P-tree arrangement or using a label stack of [P2MP P-tree transport label, upstream assigned EVPN Inclusive Multicast label] for the aggregate inclusive P2MP P-tree arrangement as described in Section 6.

In case of EVPN, an additional, EVPN Auto-Discovery sub-TLV and ESI MPLS label as the bottom label, may also be included in the Echo Request as is described in Section 6.

4.3. EVPN Auto-Discovery Sub-TLV

The EVPN Auto-Discovery (AD) sub-TLV fields are based on the Ethernet AD route advertisement defined in [RFC7432] Section 7.1. EVPN AD sub-TLV applies to only EVPN.

The EVPN AD sub-TLV has the format shown in Figure 3.

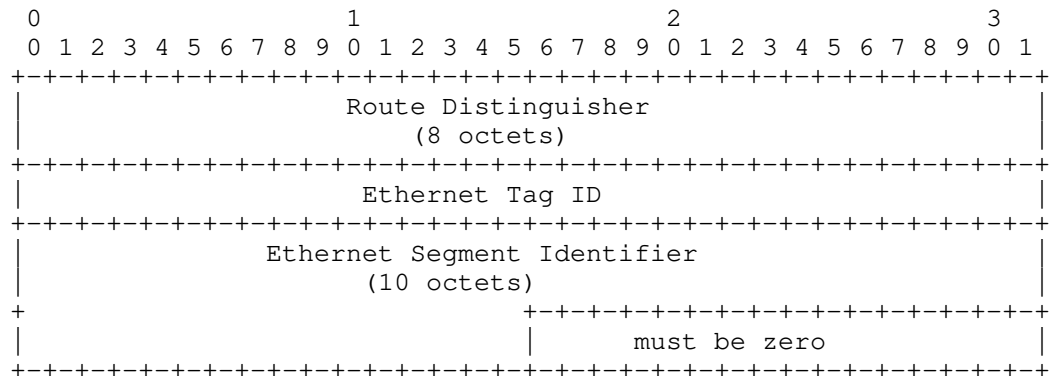


Figure 3: EVPN Auto-Discovery sub-TLV format

4.4. EVPN IP Prefix Sub-TLV

The EVPN IP Prefix sub-TLV is used to identify the IP Prefix for an EVI under test at a peer PE.

The EVPN IP Prefix sub-TLV fields are derived from the IP Prefix Route (RT-5) advertisement defined in [I-D.ietf-bess-evpn-prefix-advertisement] and has the format as shown in Figure 4. This TLV is included in the Echo Request sent to the Peer PE by the PE that is the originator of the request.

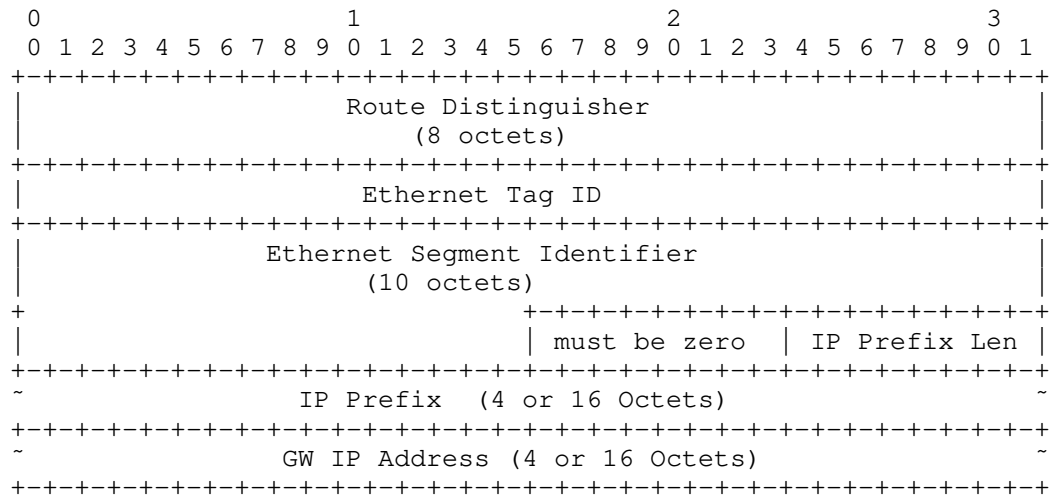


Figure 4: EVPN IP Prefix sub-TLV format

The LSP Ping echo request is sent using the EVPN MPLS label(s) associated with the IP Prefix route announced by a remote PE and the MPLS transport label(s) to reach the remote PE.

5. Encapsulation of OAM Ping Packets

The LSP Ping Echo request IPv4/UDP packets are encapsulated with the Transport and EVPN Label(s) followed by the Generic Associated Channel Label (GAL) [RFC6426] which is the bottom most label. The GAL label is followed by IPv4(0x0021) or IPv6(0x0057) Associated Channel Header (ACH) [RFC4385].

6. Operations

6.1. Unicast Data-plane connectivity checks

Figure 5 is an example of a PBB-EVPN network. CE1 is dual-homed to PE1 and PE2. Assume, PE1 announced a MAC route with RD 1.1.1.1:00 and B-MAC 00aa.00bb.00cc and with MPLS label 16001 for EVI 10. Similarly, PE2 announced a MAC route with RD 2.2.2.2:00 and B-MAC 00aa.00bb.00cc and with MPLS label 16002.

On PE3, when an operator performs a connectivity check for the B-MAC address 00aa.00bb.00cc on PE1, the operator initiates an LSP Ping request with the target FEC stack TLV containing EVPN MAC sub-TLV in the Echo Request packet. The Echo Request packet is sent with the {Transport Label(s) to reach PE1 + EVPN Label = 16001 + GAL} MPLS

label stack and IP ACH Channel header. Once the echo request packet reaches PE1, PE1 will use the GAL label and the IP ACH Channel header to determine that the packet is IPv4 OAM Packet. The PE1 will process the packet and perform checks for the EVPN MAC sub-TLV present in the Target FEC Stack TLV as described in Section 4.4 in [RFC8029] and respond according to [RFC8029] processing rules.

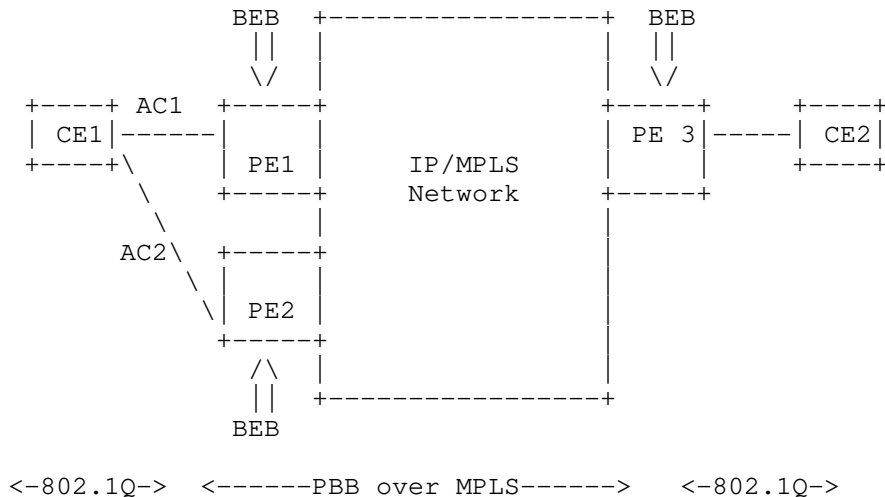


Figure 5: PBB EVPN network

Similarly, on PE3, when an operator performs a connectivity check for the B-MAC address 00aa.00bb.00cc on PE2, the operator initiates an LSP Ping request with the target FEC stack TLV containing EVPN MAC sub-TLV in the echo request packet. The echo request packet is sent with the {MPLS transport Label(s) to reach PE2 + EVPN Label = 16002 + GAL} MPLS label stack and IP ACH Channel header.

LSP Ping operation for unicast data-plane connectivity checks in E-VPN, are similar to those described above for PBB-EVPN except that the checks are for C-MAC addresses instead of B-MAC addresses.

6.2. Inclusive Multicast Data-plane Connectivity Checks

6.2.1. Ingress Replication

Assume PE1 announced an Inclusive Multicast route for EVI 10, with RD 1.1.1.1:00, Ethernet Tag (ISID 10), PMSI tunnel attribute Tunnel type set to ingress replication and downstream assigned inclusive multicast MPLS label 17001. Similarly, PE2 announced an Inclusive Multicast route for EVI 10, with RD 2.2.2.2:00, Ethernet Tag (ISID 10), PMSI tunnel attribute Tunnel type set to ingress replication and downstream assigned inclusive multicast MPLS label 17002.

Given CE1 is dual-homed to PE1 and PE2, assume that PE1 is the DF for ISID 10 for the port corresponding to the ESI 11aa.22bb.33cc.44dd.5500.

When an operator at PE3 initiates a connectivity check for the inclusive multicast on PE1, the operator initiates an LSP Ping request with the target FEC stack TLV containing EVPN Inclusive Multicast sub-TLV in the Echo Request packet. The Echo Request packet is sent with the {Transport Label(s) to reach PE1 + EVPN Incl. Multicast Label = 17001 + GAL} MPLS label stack and IP ACH Channel header. Once the echo request packet reaches PE1, PE1 will use the GAL label and the IP ACH Channel header to determine that the packet is IPv4 OAM Packet. The packet will have EVPN Inclusive multicast label. PE1 will process the packet and perform checks for the EVPN Inclusive Multicast sub-TLV present in the Target FEC Stack TLV as described in Section 4.4 in [RFC8029] and respond according to [RFC8029] processing rules.

An operator at PE3, may similarly also initiate an LSP Ping to PE2 with the target FEC stack TLV containing EVPN Inclusive Multicast sub-TLV in the echo request packet. The echo request packet is sent with the {transport Label(s) to reach PE2 + EVPN Incl. Multicast Label = 17002 + GAL} MPLS label stack and IP ACH Channel header. Once the echo request packet reaches PE2, PE2 will use the GAL label and the IP ACH Channel header to determine that the packet is IPv4 OAM Packet. Since PE2 is not the DF for ISID 10 for the port corresponding to the ESI value in the Inclusive Multicast sub-TLV in the Echo Request, PE2 will reply with the special code indicating that FEC exists on the router and the behavior is to drop the packet because of not DF as described in Section 8.

In case of EVPN, in the Echo Request packet, an Ethernet AD sub-TLV and the associated MPLS Split Horizon Label above the GAL label in the MPLS label stack, may be added to emulate traffic coming from a MH site, this label is used by leaf PE(s) attached to the same MH site not to forward packets back to the MH site. If the behavior on a leaf PE is to drop the packet because of Split Horizon filtering, the PE2 will reply with the special code indicating that FEC exists

on the router and the behavior is to drop the packet because of Split Horizon Filtering as described in Section 8.

6.2.2. Using P2MP P-tree

Both inclusive P-Tree and aggregate inclusive P-tree can be used in EVPN or PBB-EVPN networks.

When using an inclusive P-tree arrangement, p2mp p-tree transport label itself is used to identify the L2 service associated with the Inclusive Multicast Route, this L2 service could be a customer Bridge, or a Provider Backbone Bridge.

For an Inclusive P-tree arrangement, when an operator performs a connectivity check for the multicast L2 service, the operator initiates an LSP Ping request with the target FEC stack TLV containing EVPN Inclusive Multicast sub-TLV in the echo request packet. The echo request packet is sent over P2MP LSP with the {P2MP P-tree label, GAL} MPLS label stack and IP ACH Channel header.

When using Aggregate Inclusive P-tree, a PE announces an upstream assigned MPLS label along with the P-tree ID, in that case both the p2mp p-tree MPLS transport label and the upstream MPLS label can be used to identify the L2 service.

For an Aggregate Inclusive P-tree arrangement, when an operator performs a connectivity check for the multicast L2 service, the operator initiates an LSP Ping request with the target FEC stack TLV containing EVPN Inclusive Multicast sub-TLV in the echo request packet. The echo request packet is sent over P2MP LSP using the IP-ACH Control channel with the {P2MP P-tree label, EVPN Upstream assigned Multicast Label, GAL} MPLS label stack and IP ACH Channel header.

The Leaf PE(s) of the p2mp tree will process the packet and perform checks for the EVPN Inclusive Multicast sub-TLV present in the Target FEC Stack TLV as described in Section 4.4 in [RFC8029] and respond according to [RFC8029] processing rules. A PE that is not the DF for the EVI on the ESI in the Inclusive Multicast sub-TLV, will reply with a special code indicating that FEC exists on the router and the behavior is to drop the packet because of not DF as described in Section 8.

In case of EVPN, in the Echo Request packet, an Ethernet AD sub-TLV and the associated MPLS Split Horizon Label above the GAL Label in MPLS label stack, may be added to emulate traffic coming from a MH site, this label is used by leaf PE(s) attached to the same MH site not to forward packets back to the MH site. If the behavior on a

leaf PE is to drop the packet because of Split Horizon filtering, the PE2 will reply with special code indicating that FEC exists on the router and the behavior is to drop the packet because of Split Horizon Filtering as described in Section 8.

6.2.3. Controlling Echo Responses when using P2MP P-tree

The procedures described in [RFC6425] for preventing congestion of Echo Responses (Echo Jitter TLV) and limiting the echo reply to a single egress node (Node Address P2MP Responder Identifier TLV) can be applied to LSP Ping in PBB EVPN and EVPN when using P2MP P-trees for broadcast, multicast, and unknown unicast traffic.

6.3. EVPN Aliasing Data-plane connectivity check

Assume PE1 announced an Ethernet Auto discovery Route with the ESI set to CE1 system ID and MPLS label 19001, and PE2 an Ethernet Auto discovery Route with the ESI set to CE1 system ID and MPLS label 19002.

When an operator performs at PE3 a connectivity check for the aliasing aspect of the Ethernet AD route to PE1, the operator initiates an LSP Ping request with the target FEC stack TLV containing EVPN Ethernet AD sub-TLV in the echo request packet. The echo request packet is sent with the {Transport label(s) to reach PE1 + EVPN Ethernet AD Label 19001 + GAL} MPLS label stack and IP ACH Channel header.

When PE1 receives the packet it will process the packet and perform checks for the EVPN Ethernet AD sub-TLV present in the Target FEC Stack TLV as described in Section 4.4 in [RFC8029] and respond according to [RFC8029] processing rules.

6.4. EVPN IP Prefix (RT-5) Data-plane connectivity check

Assume PE1 in Figure 5, announced an IP Prefix Route (RT-5) with an IP prefix reachable behind CE1 and MPLS label 20001. When an operator on PE3 performs a connectivity check for the IP prefix on PE1, the operator initiates an LSP Ping request with the target FEC stack TLV containing EVPN IP Prefix sub-TLV in the echo request packet. The echo request packet is sent with the {Transport label(s) to reach PE1 + EVPN IP Prefix Label 20001 } MPLS label stack.

When PE1 receives the packet it will process the packet and perform checks for the EVPN IP Prefix sub-TLV present in the Target FEC Stack TLV as described in Section 4.4 in [RFC8029] and respond according to [RFC8029] processing rules.

7. Security Considerations

The proposal introduced in this document does not introduce any new security considerations beyond that already apply to [RFC7432], [RFC7623] and [RFC6425].

8. IANA Considerations

8.1. Sub-TLV Type

This document defines 4 new sub-TLV type to be included in Target FEC Stack TLV (TLV Type 1) [RFC8029] in LSP Ping.

IANA is requested to assign a sub-TLV type value to the following sub-TLV from the "Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) Parameters - TLVs" registry, "TLVs and sub- TLVs" sub-registry:

- o EVPN MAC route sub-TLV
- o EVPN Inclusive Multicast route sub-TLV
- o EVPN Auto-Discovery Route sub-TLV
- o EVPN IP Prefix Route sub-TLV

8.2. Proposed new Return Codes

[RFC8029] defines values for the Return Code field of Echo Reply. This document proposes two new Return Codes, which SHOULD be included in the Echo Reply message by a PE in response to LSP Ping Echo Request message:

1. The FEC exists on the PE and the behavior is to drop the packet because of not DF.
2. The FEC exists on the PE and the behavior is to drop the packet because of Split Horizon Filtering.

9. Acknowledgments

The authors would like to thank Patrice Brissette and Weiguo Hao for their comments.

10. References

10.1. Normative References

- [I-D.ietf-bess-evpn-prefix-advertisement]
Rabadan, J., Henderickx, W., Drake, J., Lin, W., and A. Sajassi, "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-11 (work in progress), May 2018.
- [RFC6425] Saxena, S., Ed., Swallow, G., Ali, Z., Farrel, A., Yasukawa, S., and T. Nadeau, "Detecting Data-Plane Failures in Point-to-Multipoint MPLS - Extensions to LSP Ping", RFC 6425, DOI 10.17487/RFC6425, November 2011, <<https://www.rfc-editor.org/info/rfc6425>>.
- [RFC6426] Gray, E., Bahadur, N., Boutros, S., and R. Aggarwal, "MPLS On-Demand Connectivity Verification and Route Tracing", RFC 6426, DOI 10.17487/RFC6426, November 2011, <<https://www.rfc-editor.org/info/rfc6426>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.
- [RFC7623] Sajassi, A., Ed., Salam, S., Bitar, N., Isaac, A., and W. Henderickx, "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, DOI 10.17487/RFC7623, September 2015, <<https://www.rfc-editor.org/info/rfc7623>>.
- [RFC8029] Kompella, K., Swallow, G., Pignataro, C., Ed., Kumar, N., Aldrin, S., and M. Chen, "Detecting Multiprotocol Label Switched (MPLS) Data-Plane Failures", RFC 8029, DOI 10.17487/RFC8029, March 2017, <<https://www.rfc-editor.org/info/rfc8029>>.

10.2. Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC5085] Nadeau, T., Ed. and C. Pignataro, Ed., "Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowires", RFC 5085, DOI 10.17487/RFC5085, December 2007, <<https://www.rfc-editor.org/info/rfc5085>>.
- [RFC6338] Giralt, V. and R. McDuff, "Definition of a Uniform Resource Name (URN) Namespace for the Schema for Academia (SCHAC)", RFC 6338, DOI 10.17487/RFC6338, August 2011, <<https://www.rfc-editor.org/info/rfc6338>>.

Authors' Addresses

Parag Jain (editor)
Cisco Systems, Inc.
2000 Innovation Drive
Kanata, ON K2K 3E8
Canada

Email: paragj@cisco.com

Samer Salam
Cisco Systems, Inc.
595 Burrard Street, Suite 2123
Vancouver, BC V7X 1J1
Canada

Email: ssalam@cisco.com

Ali Sajassi
Cisco Systems, Inc.
USA

Email: sajassi@cisco.com

Sami Boutros
VmWare, Inc.
USA

Email: sboutros@vmware.com

Greg Mirsky
ZTE Corporation.
USA

Email: gregmirsky@gmail.com>

BESS Working Group
Internet Draft
Category: Standard Track

A. Sajassi
K. Thiruvengatasamy
S. Thoria
Cisco
A. Gupta
Avi Networks
L. Jalil
Verizon

Expires: January 06, 2020

July 05, 2019

Seamless Multicast Interoperability between EVPN and MVPN PEs
draft-sajassi-bess-evpn-mvpn-seamless-interop-04

Abstract

Ethernet Virtual Private Network (EVPN) solution is becoming pervasive for Network Virtualization Overlay (NVO) services in data center (DC) networks and as the next generation VPN services in service provider (SP) networks.

As service providers transform their networks in their COs toward next generation data center with Software Defined Networking (SDN) based fabric and Network Function Virtualization (NFV), they want to be able to maintain their offered services including Multicast VPN (MVPN) service between their existing network and their new Service Provider Data Center (SPDC) network seamlessly without the use of gateway devices. They want to have such seamless interoperability between their new SPDCs and their existing networks for a) reducing cost, b) having optimum forwarding, and c) reducing provisioning. This document describes a unified solution based on RFCs 6513 & 6514 for seamless interoperability of Multicast VPN between EVPN and MVPN PEs. Furthermore, it describes how the proposed solution can be used as a routed multicast solution in data centers with only EVPN PEs.

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any

time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	4
2. Requirements Language	5
3. Terminology	5
4. Requirements	6
4.1. Optimum Forwarding	7
4.2. Optimum Replication	7
4.3. All-Active and Single-Active Multi-Homing	7
4.4. Inter-AS Tree Stitching	7
4.5. EVPN Service Interfaces	8
4.6. Distributed Anycast Gateway	8
4.7. Selective & Aggregate Selective Tunnels	8
4.8. Tenants' (S,G) or (*,G) states	8
4.9. Zero Disruption upon BD/Subnet Addition	8
4.10. No Changes to Existing EVPN Service Interface Models	8
4.11. External source and receivers	9
4.12. Tenant RP placement	9
5. IRB Unicast versus IRB Multicast	9
5.1. Emulated Virtual LAN Service	9
6. Solution Overview	10
6.1. Operational Model for EVPN IRB PEs	10

6.2.	Unicast Route Advertisements for IP multicast Source . . .	12
6.3.	Multi-homing of IP Multicast Source and Receivers . . .	13
6.3.1.	Single-Active Multi-Homing . . .	14
6.3.2.	All-Active Multi-Homing . . .	15
6.4.	Mobility for Tenant's Sources and Receivers . . .	17
6.5.	Intra-Subnet BUM Traffic Handling . . .	17
6.6	EVPN and MVPN interworking with gateway model . . .	17
7.	Control Plane Operation . . .	18
7.1.	Intra-ES/Intra-Subnet IP Multicast Tunnel . . .	18
7.2.	Intra-Subnet BUM Tunnel . . .	19
7.3.	Inter-Subnet IP Multicast Tunnel . . .	20
7.4.	IGMP Hosts as TSes . . .	20
7.5.	TS PIM Routers . . .	21
8	Data Plane Operation . . .	21
8.1	Intra-Subnet L2 Switching . . .	22
8.2	Inter-Subnet L3 Routing . . .	22
9.	DCs with only EVPN PEs . . .	23
9.1.	Setup of overlay multicast delivery . . .	23
9.2.	Handling of different encapsulations . . .	25
9.2.1.	MPLS Encapsulation . . .	25
9.2.2	VxLAN Encapsulation . . .	25
9.2.3.	Other Encapsulation . . .	26
10.	DCI with MPLS in WAN and VxLAN in DCs . . .	26
10.1.	Control plane inter-connect . . .	26
10.2.	Data plane inter-connect . . .	27
11.	Supporting application with TTL value 1 . . .	28
11.1.	Policy based model . . .	28
11.2.	Exercising BUM procedure for VLAN/BD . . .	28
11.3.	Intra-subnet bridging . . .	28
12.	Interop with L2 EVPN PEs . . .	30
13.	Connecting external Multicast networks or PIM routers. . .	30
14.	RP handling . . .	30
14.1.	Various RP deployment options . . .	30
14.1.1.	RP-less mode . . .	30
14.1.2.	Fabric anycast RP . . .	31
14.1.3.	Static RP . . .	31
14.1.4.	Co-existence of Fabric anycast RP and external RP . .	31
14.2.	RP configuration options . . .	31
15.	IANA Considerations . . .	32
16.	Security Considerations . . .	32
17.	Acknowledgements . . .	32
18.	References . . .	32
18.1.	Normative References . . .	32
18.2.	Informative References . . .	33
19.	Authors' Addresses . . .	34
Appendix A.	Use Cases . . .	34
A.1.	DCs with only IGMP/MLD hosts w/o tenant router . . .	34

1. Introduction

Ethernet Virtual Private Network (EVPN) solution is becoming pervasive for Network Virtualization Overlay (NVO) services in data center (DC) networks and as the next generation VPN services in service provider (SP) networks.

As service providers transform their networks in their COs toward next generation data center with Software Defined Networking (SDN) based fabric and Network Function Virtualization (NFV), they want to be able to maintain their offered services including Multicast VPN (MVPN) service between their existing network and their new SPDC network seamlessly without the use of gateway devices. There are several reasons for having such seamless interoperability between their new DCs and their existing networks:

- Lower Cost: gateway devices need to have very high scalability to handle VPN services for their DCs and as such need to handle large number of VPN instances (in tens or hundreds of thousands) and very large number of routes (e.g., in tens of millions). For the same speed and feed, these high scale gateway boxes are relatively much more expensive than the edge devices (e.g., PEs and TORs) that support much lower number of routes and VPN instances.
- Optimum Forwarding: in a given CO, both EVPN PEs and MVPN PEs can be connected to the same fabric/network (e.g., same IGP domain). In such scenarios, the service providers want to have optimum forwarding among these PE devices without the use of gateway devices. Because if gateway devices are used, then the IP multicast traffic between an EVPN and MVPN PEs can no longer be optimum and in some case, it may even get tromboned. Furthermore, when an SPDC network spans across multiple LATA (multiple geographic areas) and gateways are used between EVPN and MVPN PEs, then with respect to IP multicast traffic, only one GW can be designated forwarder (DF) between EVPN and MVPN PEs. Such scenarios not only results in non-optimum forwarding but also it can result in tromboing of IP multicast traffic between the two LATAs when both source and destination PEs are in the same LATA and the DF gateway is elected to be in a different LATA.
- Less Provisioning: If gateways are used, then the operator need to configure per-tenant info on the gateways. In other words, for each tenant that is configured, one (or maybe two) additional touch points are needed.

This document describes a unified solution based on [RFC6513] and [RFC6514] for seamless interoperability of multicast VPN between EVPN and MVPN PEs. Furthermore, it describes how the proposed solution can be used as a routed multicast solution in data centers with only EVPN

PEs (e.g., routed multicast VPN only among EVPN PEs).

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in [RFC2119] only when they appear in all upper case. They may also appear in lower or mixed case as English words, without any normative meaning.

3. Terminology

Most of the terminology used in this documents comes from [RFC8365]

Broadcast Domain (BD): In a bridged network, the broadcast domain corresponds to a Virtual LAN (VLAN), where a VLAN is typically represented by a single VLAN ID (VID) but can be represented by several VIDs where Shared VLAN Learning (SVL) is used per [802.1Q].

Bridge Table (BT): An instantiation of a broadcast domain on a MAC-VRF.

VXLAN: Virtual Extensible LAN

POD: Point of Delivery

NV: Network Virtualization

NVO: Network Virtualization Overlay

NVE: Network Virtualization Endpoint

VNI: Virtual Network Identifier (for VXLAN)

EVPN: Ethernet VPN

EVI: An EVPN instance spanning the Provider Edge (PE) devices participating in that EVPN

MAC-VRF: A Virtual Routing and Forwarding table for Media Access Control (MAC) addresses on a PE

IP-VRF: A Virtual Routing and Forwarding table for Internet Protocol (IP) addresses on a PE

Ethernet Segment (ES): When a customer site (device or network) is

connected to one or more PEs via a set of Ethernet links, then that set of links is referred to as an 'Ethernet segment'.

Ethernet Segment Identifier (ESI): A unique non-zero identifier that identifies an Ethernet segment is called an 'Ethernet Segment Identifier'.

Ethernet Tag: An Ethernet tag identifies a particular broadcast domain, e.g., a VLAN. An EVPN instance consists of one or more broadcast domains.

PE: Provider Edge device.

Single-Active Redundancy Mode: When only a single PE, among all the PEs attached to an Ethernet segment, is allowed to forward traffic to/from that Ethernet segment for a given VLAN, then the Ethernet segment is defined to be operating in Single-Active redundancy mode.

All-Active Redundancy Mode: When all PEs attached to an Ethernet segment are allowed to forward known unicast traffic to/from that Ethernet segment for a given VLAN, then the Ethernet segment is defined to be operating in All-Active redundancy mode.

PIM-SM: Protocol Independent Multicast - Sparse-Mode

PIM-SSM: Protocol Independent Multicast - Source Specific Multicast

Bidir PIM: Bidirectional PIM

FHR: First Hop Router

LHR: Last Hop Router

CO: Central Office of a service provider

SPDC: Service Provider Data Center

LATA: Local Access and Transport Area

Border Leafs: A set of EVPN-PE acting as exit point for EVPN fabric.

L3VNI: A VNI in the tenant VRF, which is associated with the core facing interface.

4. Requirements

This section describes the requirements specific in providing

seamless multicast VPN service between MVPN and EVPN capable networks.

4.1. Optimum Forwarding

The solution SHALL support optimum multicast forwarding between EVPN and MVPN PEs within a network. The network can be confined to a CO or it can span across multiple LATAs. The solution SHALL support optimum multicast forwarding with both ingress replication tunnels and P2MP tunnels.

4.2. Optimum Replication

For EVPN PEs with IRB capability, the solution SHALL use only a single multicast tunnel among EVPN and MVPN PEs for IP multicast traffic, when both PEs use the same tunnel type. Multicast tunnels can be either ingress replication tunnels or P2MP tunnels. The solution MUST support optimum replication for both Intra-subnet and Inter-subnet IP multicast traffic:

- Non-IP traffic SHALL be forwarded per EVPN baseline [RFC7432] or [RFC8365]
- If a Multicast VPN spans across both Intra and Inter subnets, then for Ingress replication regardless of whether the traffic is Intra or Inter subnet, only a single copy of IP multicast traffic SHALL be sent from the source PE to the destination PE.
- If a Multicast VPN spans across both Intra and Inter subnets, then for P2MP tunnels regardless of whether the traffic is Intra or Inter subnet, only a single copy of multicast data SHALL be transmitted by the source PE. Source PE can be either EVPN or MVPN PE and receiving PEs can be a mix of EVPN and MVPN PEs - i.e., a multicast VPN can be spread across both EVPN and MVPN PEs.

4.3. All-Active and Single-Active Multi-Homing

The solution MUST support multi-homing of source devices and receivers that are sitting in the same subnet (e.g., VLAN) and are multi-homed to EVPN PEs. The solution SHALL allow for both Single-Active and All-Active multi-homing. The solution MUST prevent loop during steady and transient states just like EVPN baseline solution [RFC7432] and [RFC8365] for all multi-homing types.

4.4. Inter-AS Tree Stitching

The solution SHALL support multicast tree stitching when the tree

spans across multiple Autonomous Systems.

4.5. EVPN Service Interfaces

The solution MUST support all EVPN service interfaces listed in section 6 of [RFC7432]:

- VLAN-based service interface
- VLAN-bundle service interface
- VLAN-aware bundle service interface

4.6. Distributed Anycast Gateway

The solution SHALL support distributed anycast gateways for tenant workloads on NVE devices operating in EVPN-IRB mode.

4.7. Selective & Aggregate Selective Tunnels

The solution SHALL support selective and aggregate selective P-tunnels as well as inclusive and aggregate inclusive P-tunnels. When selective tunnels are used, then multicast traffic SHOULD only be forwarded to the remote PE which have receivers - i.e., if there are no receivers at a remote PE, the multicast traffic SHOULD NOT be forwarded to that PE and if there are no receivers on any remote PEs, then the multicast traffic SHOULD NOT be forwarded to the core.

4.8. Tenants' (S,G) or (*,G) states

The solution SHOULD store (C-S,C-G) and (C-*,C-G) states only on PE devices that have interest in such states hence reducing memory and processing requirements - i.e., PE devices that have sources and/or receivers interested in such multicast groups.

4.9. Zero Disruption upon BD/Subnet Addition

In DC environments, various Bridge Domains are provisioned and removed on regular basis due to host mobility, policy and tenant changes. Such change in BD configuration should not affect existing flows within the same BD or any other BD in the network.

4.10. No Changes to Existing EVPN Service Interface Models

VLAN-aware bundle service as defined in [RFC7432] typically does not require any VLAN ID translation from one tenant site to another - i.e., the same set of VLAN IDs are configured consistently on all tenant segments. In such scenarios, EVPN-IRB multicast service MUST maintain the same mode of operation and SHALL NOT require any VLAN ID translation.

4.11. External source and receivers

The solution SHALL support sources and receivers external to the tenant domain. i.e., multicast source inside the tenant domain can have receiver outside the tenant domain and vice versa.

4.12. Tenant RP placement

The solution SHALL support a tenant to have RP anywhere in the network. RP can be placed inside the EVPN network or MVPN network or external domain.

5. IRB Unicast versus IRB Multicast

[EVPN-IRB] describes the operation for EVPN PEs in IRB mode for unicast traffic. The same IRB model used for unicast traffic in [EVPN-IRB], where an IP-VRF in an EVPN PE is attached to one or more bridge tables (BTs) via virtual IRB interfaces, is also applicable for multicast traffic. However, there are some noticeable differences between the IRB operation for unicast traffic described in [EVPN-IRB] versus for multicast traffic described in this document. For unicast traffic, the intra-subnet traffic, is bridged within the MAC-VRF associated with that subnet (i.e., a lookup based on MAC-DA is performed); whereas, the inter-subnet traffic is routed in the corresponding IP-VRF (ie, a lookup based on IP-DA is performed). A given tenant can have one or more IP-VRFs; however, without loss of generality, this document assumes one IP-VRF per tenant. In context of a given tenant's multicast traffic, the intra-subnet traffic is bridged for non-IP traffic and it is Layer-2 switched for IP traffic. Whereas, the tenants's inter-subnet multicast traffic is always routed in the corresponding IP-VRF. The difference between bridging and L2-switching for multicast traffic is that the former uses MAC-DA lookup for forwarding the multicast traffic; whereas, the latter uses IP-DA lookup for such forwarding where the forwarding states are built in the MAC-VRF using IGMP/MLD or PIM snooping.

5.1. Emulated Virtual LAN Service

EVPN does not provide a Virtual LAN (VLAN) service per [IEEE802.1Q] but rather an emulated VLAN service. This VLAN service emulation is not only done for unicast traffic but also is extended for intra-subnet multicast traffic described in [EVPN-IGMP-PROXY] and [EVPN-PIM-PROXY]. For intra-subnet multicast, an EVPN PE builds multicast forwarding states in its bridge table (BT) based on snooping of IGMP/MLD and/or PIM messages and the forwarding is performed based on destination IP multicast address of the Ethernet frame rather than destination MAC address as noted above. In order to enable seamless integration of EVPN and MVPN PEs, this document extends the concept

of an emulated VLAN service for multicast IRB applications such that the intra-subnet IP multicast traffic can get treated same as inter-subnet IP multicast traffic which means intra-subnet IP multicast traffic destined to remote PEs gets routed instead of being L2-switched - i.e., TTL value gets decremented and the Ethernet header of the L2 frame is de-capsulated and encapsulated at both ingress and egress PEs. It should be noted that the non-IP multicast or L2 broadcast traffic still gets bridged and frames get forwarded based on their destination MAC addresses.

6. Solution Overview

This section describes a multicast VPN solution based on [RFC6513] and [RFC6514] for EVPN PEs operating in IRB mode that want to perform seamless interoperability with their counterparts MVPN PEs.

6.1. Operational Model for EVPN IRB PEs

Without the loss of generality, this section assumes that all EVPN PEs have IRB capability and operating in IRB mode for both unicast and multicast traffic (e.g., all EVPN PEs are homogenous in terms of their capabilities and operational modes). As it will be seen later, an EVPN network can consist of a mix of PEs where some are capable of multicast IRB and some are not and the multicast operation of such heterogeneous EVPN network will be an extension of an EVPN homogenous network. Therefore, we start with the multicast IRB solution description for the EVPN homogenous network.

The EVPN PEs terminate IGMP/MLD messages from tenant host devices or PIM messages from tenant routers on their IRB interfaces, thus avoid sending these messages over MPLS/IP core. A tenant virtual/physical router (e.g., CE) attached to an EVPN PE becomes a multicast routing adjacency of that PE. Furthermore, the PE uses MVPN BGP protocol and procedures per [RFC6513] and [RFC6514]. With respect to multicast routing protocol between tenant's virtual/physical router and the PE that it is attached to, any of the following PIM protocols is supported per [RFC6513]: PIM-SM with Any Source Multicast (ASM) mode, PIM-SM with Source Specific Multicast (SSM) mode, and PIM Bidirectional (BIDIR) mode. Support of PIM-DM (Dense Mode) is excluded in this document per [RFC6513].

The EVPN PEs use MVPN BGP routes defined in [RFC6514] to convey tenant (S,G) or (*,G) states to other MVPN or EVPN PEs and to set up overlay trees (inclusive or selective) for a given MVPN instance. The root or a leaf of such an overlay tree is terminated on an EVPN or MVPN PE. Furthermore, this inclusive or selective overlay tree is terminated on a single IP-VRF of the EVPN or MVPN PE. In case of EVPN PE, these overlay trees never get terminated on MAC-VRFs of that PE.

Overlay trees are instantiated by underlay provider tunnels (P-tunnels) - e.g., P2MP, MP2MP, or unicast tunnels per [RFC 6513]. When there are several overlay trees mapped to a single underlay P-tunnel, the tunnel is referred to as an aggregate tunnel.

Figure-1 below depicts a scenario where a tenant's MVPN spans across both EVPN and MVPN PEs; where all EVPN PEs have multicast IRB capability. An EVPN PE (with multicast IRB capability) can be modeled as a MVPN PE where the virtual IRB interface of an EVPN PE (virtual interface between a BT and IP-VRF) can be considered a routed interface for the MVPN PE.

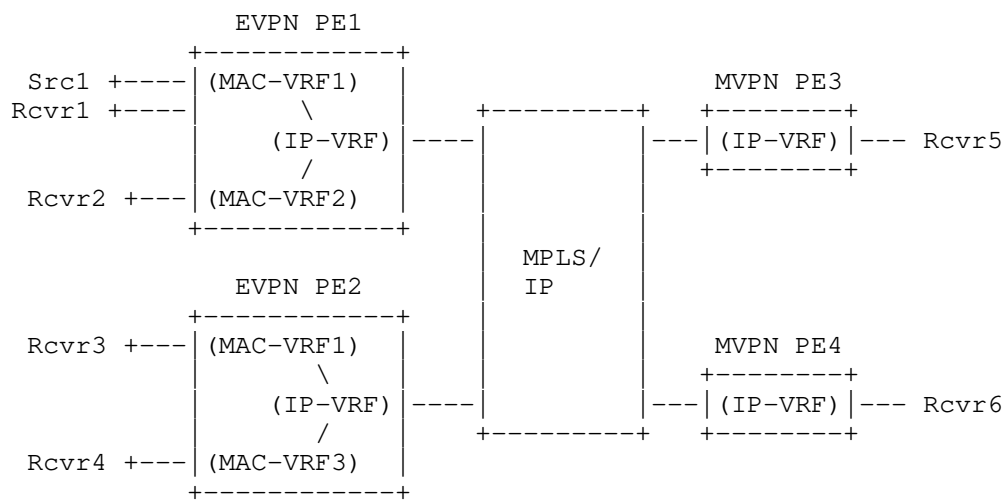


Figure-1: EVPN & MVPN PEs Seamless Interop

Figure 2 depicts the modeling of EVPN PEs based on MVPN PEs where an EVPN PE can be modeled as a PE that consists of a MVPN PE whose routed interfaces (e.g., attachment circuits) are replaced with IRB interfaces connecting each IP-VRF of the MVPN PE to a set of BTs. Similar to a MVPN PE where an attachment circuit serves as a routed multicast interface for an IP-VRF associated with a MVPN instance, an IRB interface serves as a routed multicast interface for the IP-VRF associated with the MVPN instance. Since EVPN PEs run MVPN protocols (e.g., [RFC6513] and [RFC6514]), for all practical purposes, they look just like MVPN PEs to other PE devices. Such modeling of EVPN PEs, transforms the multicast VPN operation of EVPN PEs to that of MVPN and thus simplifies the interoperability between EVPN and MVPN PEs to that of running a single unified solution based on MVPN.

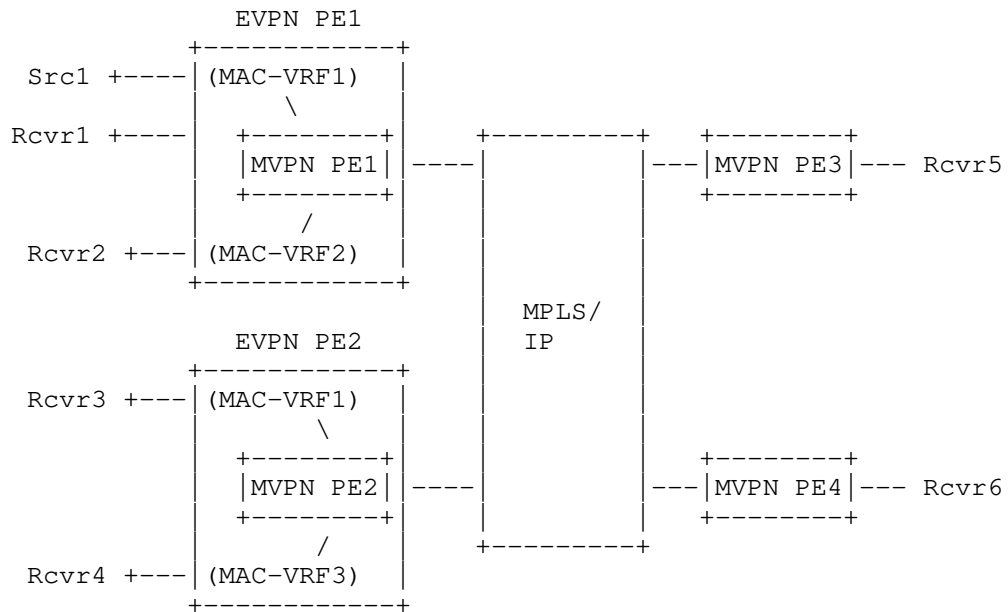


Figure-2: Modeling EVPN PEs as MVPN PEs

Although modeling an EVPN PE as a MVPN PE, conceptually simplifies the operation to that of a solution based on MVPN, the following operational aspects of EVPN need to be factored in when considering seamless integration between EVPN and MVPN PEs.

- 1) Unicast route advertisements for IP multicast source
- 2) Multi-homing of IP multicast sources and receivers
- 3) Mobility for Tenant's sources and receivers
- 4) non-IP multicast traffic handling

6.2. Unicast Route Advertisements for IP multicast Source

When an IP multicast source is attached to an EVPN PE, the unicast route for that IP multicast source needs to be advertised. When the source is attached to a Single-Active multi-homed ES, then the EVPN DF PE is the PE that advertises a unicast route corresponding to the source IP address with VRF Route Import extended community which in turn is used as the Route Target for Join (S,G) messages sent toward the source PE by the remote PEs. The EVPN PE advertises this unicast route using EVPN route type 2 and IPVPN unicast route along with VRF Route Import extended community. EVPN route type 2 is advertised with the Route Targets corresponding to both IP-VRF and MAC-VRF/BT; whereas, IPVPN unicast route is advertised with RT corresponding to the IP-VRF. When unicast routes are advertised by MVPN PEs, they are

advertised using IPVPN unicast route along with VRF Route Import extended community per [RFC6514].

When the source is attached to an All-Active multi-homed ES, then the PE that learns the source advertises the unicast route for that source using EVPN route type 2 and IPVPN unicast route along with VRF Route Import extended community. EVPN route type 2 is advertised with the Route Targets corresponding to both IP-VRF and MAC-VRF/BT; whereas, IPVPN unicast route is advertised with RT corresponding to the IP-VRF. When the other multi-homing EVPN PEs for that ES receive this unicast EVPN route, they import the route and check to see if they have learned the route locally for that ES, if they have, then they do nothing. But if they have not, then they add the IP and MAC addresses to their IP-VRF and MAC-VRF/BT tables respectively with the local interface corresponding to that ES as the corresponding route adjacency. Furthermore, these PEs advertise an IPVPN unicast route along with VRF Route Import extended community and Route Target corresponding to IP-VRF to other remote PEs for that MVPN. Therefore, the remote PEs learn the unicast route corresponding to the source from all multi-homing PEs associated with that All-Active Ethernet Segment even though one of the multi-homing PEs may only have directly learned the IP address of the source.

EVPN-PEs advertise unicast routes as host routes using EVPN route type 2 for sources that are directly attached to a tenant BD that has been extended in the EVPN fabric. EVPN-PE may summarize sources (IP networks) behind a router that are attached to EVPN-PE or sources that are connected to a BD, which is not extended across EVPN fabric and advertises those routes with EVPN route type 5. EVPN host-routes are advertised as IPVPN host-routes to MVPN-PEs only in case of seamless interop mode.

Section 6.6 discusses connecting EVPN and MVPN networks with gateway model. Section 9 extends seamless interop procedures to EVPN only fabrics as an IRB solution for multicast.

EVPN-PEs only need to advertise unicast routes using EVPN route-type 2 or route-type 5 and don't need to advertise IPVPN routes within EVPN only fabric. No L3VPN provisioning is needed between EVPN-PEs.

In gateway model, EVPN-PE advertises unicast routes as IPVPN routes along with VRI extended community for all multicast sources attached behind EVPN-PEs. All IPVPN routes SHOULD be summarized while advertising to MVPN-PEs.

6.3. Multi-homing of IP Multicast Source and Receivers

EVPN [RFC7432] has extensive multi-homing capabilities that allows

TSes to be multi-homed to two or more EVPN PEs in Single-Active or All-Active mode. In Single-Active mode, only one of the multi-homing EVPN PEs can receive/transmit traffic for a given subnet (a given BD) for that multi-homed Ethernet Segment (ES). In All-Active mode, any of the multi-homing EVPN PEs can receive/transmit unicast traffic but only one of them (the DF PE) can send BUM traffic to the multi-homed ES for a given subnet.

The multi-homing mode (Single-Active versus All-Active) of a TS source can impact the MVPN procedures as described below.

6.3.1. Single-Active Multi-Homing

When a TS source reside on an ES that is multi-homed to two or more EVPN PEs operating in Single-Active mode, only one of the EVPN PEs can be active for the source subnet on that ES. Therefore, only one of the multi-homing PE learns the unicast route of the TS source and advertises that using EVPN and IPVPN to other PEs as described previously.

A downstream PE that receives a Join/Prune message from a TS host/router, selects a Upstream Multicast Hop (UMH) which is the upstream PE that receives the IP multicast flow in case of Single-Active multi-homing. An IP multicast flow belongs to either a source-specific tree (S,G) or to a shared tree (*,G). We use the notation (X,G) to refer to either (S,G) or (*,G); where X refers to S in case of (S,G) and X refers to the Rendezvous Point (RP) for G in case of (*,G). Since the active PE (which is also the UMH PE) has advertised unicast route for X along with the VRF Route Import EC, the downstream PEs selects the UMH without any ambiguity based on MVPN procedures described in section 5.1 of [RFC6513]. Any of the three algorithms described in that section works fine.

The multi-homing PE that receives the IP multicast flow on its local AC, performs the following tasks:

- L2 switches the multicast traffic in its BT associated with the local AC over which it received the flow if there are any interested receivers for that subnet.
- L3 routes the multicast traffic to other BTs for other subnets if there are any interested receivers for those subnets.
- L3 routes the multicast traffic to other PEs per MVPN procedures.

The multicast traffic can be sent on Inclusive, Selective, or Aggregate-Selective tree. Regardless what type of tree is used, only a single copy of the multicast traffic is received by the downstream

PEs and the multicast traffic is forwarded optimally from the upstream PE to the downstream PEs.

6.3.2. All-Active Multi-Homing

When a TS source reside on an ES that is multi-homed to two or more EVPN PEs operating in All-Active mode, then any of the multi-homing PEs can learn the TS source's unicast route; however, that PE may not be the same PE that receives the IP multicast flow. Therefore, the procedures for Single-Active Multi-homing need to be augmented for All-Active scenario as below.

The multi-homing EVPN PE that receives the IP multicast flow on its local AC, needs to do the following task in additions to the ones listed in the previous section for Single-Active multi-homing: L2 switch the multicast traffic to other multi-homing EVPN PEs for that ES via a multicast tunnel which it is called intra-ES tunnel. There will be a dedicated tunnel for this purpose which is different from inter-subnet overlay tree/tunnel setup by MVPN procedures.

When the multi-homing EVPN PEs receive the IP multicast flow via this tunnel, they treat it as if they receive the flow via their local ACs and thus perform the tasks mentioned in the previous section for Single-Active multi-homing. The tunnel type for this intra-ES tunnel can be any of the supported tunnel types such as ingress-replication, P2MP tunnel, BIER, and Assisted Replication; however, given that vast majority of multi-homing ESes are just dual-homing, a simple ingress replication tunnel can serve well. For a given ES, since multicast traffic that is locally received by one multi-homing PE is sent to other multi-homing PEs via this intra-ES tunnel, there is no need for sending the multicast tunnel via MVPN tunnel to these multi-homing PEs - i.e., MVPN multicast tunnels are used only for remote EVPN and MVPN PEs. Multicast traffic sent over this intra-ES tunnel to other multi-homing PEs (only one other in case of dual-homing) for a given ES can be either fixed or on demand basis. If on-demand basis, then one of the other multi-homing PEs that is selected as a UMH upon receiving a join message from a downstream PE, sends a request to receive this multicast flow from the source multi-homing PE over the special intra-ES tunnel.

By feeding IP multicast flow received on one of the EVPN multi-homing PEs to the interested EVPN PEs in the same multi-homing group, we have essentially enabled all the EVPN PEs in the multi-homing group to serve as UMH for that IP multicast flow. Each of these UMH PEs advertises unicast route for X in (X,G) along with the VRF Route Import EC to all PEs for that MVPN instance. The downstream PEs build a candidate UMH set based on procedures described in section 5.1 of [RFC6513] and pick a UMH from the set. It should be noted that both

the default UMH selection procedure based on highest UMH PE IP address and the UMH selection algorithm based on hash function specified in section 5.1.3 of [RFC6513] (which is also a MUST implement algorithm) result in the same UMH PE be selected by all downstream PEs running the same algorithm. However, in order to allow a form of "equal cost load balancing", the hash algorithm is recommended to be used among all EVPN and MVPN PEs. This hash algorithm distributes UMH selection for different IP multicast flows among the multi-homing PEs for a given ES.

Since all downstream PEs (EVPN and MVPN) use the same hash-based algorithm for UMH determination, they all choose the same upstream PE as their UMH for a given (X,G) flow and thus they all send their (X,G) join message via BGP to the same upstream PE. This results in one of the multi-homing PEs to receive the join message and thus send the IP multicast flow for (X,G) over its associated overlay tree even though all of the multi-homing PEs in the All-Active redundancy group have received the IP multicast flow (one of them directly via its local AC and the rest indirectly via the associated intra-ES tunnel). Therefore, only a single copy of routed IP multicast flow is sent over the network regardless of overlay tree type supported by the PEs - i.e., the overlay tree can be of type selective or aggregate selective or inclusive tree. This gives the network operator the maximum flexibility for choosing any overlay tree type that is suitable for its network operation and still be able to deliver only a single copy of the IP multicast flows to the egress PEs. In other words, an egress PE only receives a single copy of the IP multicast flow over the network, because it either receives it via the EVPN intra-ES tunnel or MVPN inter-subnet tunnel. Furthermore, if it receives it via MVPN inter-subnet tunnel, then only one of the multi-homing PEs associated with the source ES, sends the IP multicast traffic.

Since the network of interest for seamless interoperability between EVPN and MVPN PEs is MPLS, the EVPN handling of BUM traffic for MPLS network needs to be considered. EVPN [RFC7432] uses ESI MPLS label for split-horizon filtering of Broadcast/Unknown unicast/multicast (BUM) traffic from an All-Active multi-homing Ethernet Segment to ensure that BUM traffic doesn't get loop back to the same Ethernet Segment that it came from. This split-horizon filtering mechanism applies as-is for multicast IRB scenario because of using the intra-ES tunnel among multi-homing PEs. Since the multicast traffic received from a TS source on an All-Active ES by a multi-homing PE is bridged to all other multi-homing PEs in that group, the standard EVPN split-horizon filtering described in [RFC7432] applies as-is. Split-horizon filtering for non-MPLS encapsulations such as VxLAN is described in section 9.2.2 that deals with a DC network that consists of only EVPN PEs.

6.4. Mobility for Tenant's Sources and Receivers

When a tenant system (TS), source or receiver, is multi-homed behind a group of multi-homing EVPN PEs, then TS mobility SHALL be supported among EVPN PEs. Furthermore, such TS mobility SHALL only cause an temporary disruption to the related multicast service among EVPN and MVPN PEs. If a source is moved from one EVPN PE to another one, then the EVPN mobility procedure SHALL discover this move and a new unicast route advertisement (using both EVPN and IP-VPN routes) is made by the EVPN PE where the source has moved to per section 6.3 above and unicast route withdraw (for both EVPN and IP-VPN routes) is performed by the EVPN PE where the source has moved from.

The move of a source results in disruption of the IP multicast flow for the corresponding (S,G) flow till the new unicast route associated with the source is advertised by the new PE along with the VRF Route Import EC, the join messages sent by the egress PEs are received by the new PE, the multicast state for that flow is installed in the new PE and a new overlay tree is built for that source from the new PE to the egress PEs that are interested in receiving that IP multicast flow.

The move of a receiver results in disruption of the IP multicast flow to that receiver only till the new PE for that receiver discovers the source and joins the overlay tree for that flow.

6.5. Intra-Subnet BUM Traffic Handling

Link local IP multicast traffic consists IPv4 traffic with a destination address prefix of 224/8 and IPv6 traffic with a destination address prefix of FF02/16. Such IP multicast traffic as well as non-IP multicast/broadcast traffic are sent per EVPN [RF7432] BUM procedures and does not get routed via IP-VRF for multicast addresses. So, such BUM traffic will be limited to a given EVI/VLAN (e.g., a give subnet); whereas, IP multicast traffic, will be locally L2 switched for local interfaces attached on the same subnet and will be routed for local interfaces attached on a different subnet or for forwarding traffic to other EVPN PEs (refer to section 8 for data plane operation).

6.6 EVPN and MVPN interworking with gateway model

The procedures specified in this document offers optimal multicast forwarding within a data center and also enables seamless interoperability of multicast traffic between EVPN and MVPN networks, when same tunnel types are used in the data plane.

There are few other use cases in connecting MVPN networks in the EVPN fabric other than seamless interop model, where gateway model is used to interconnect both networks.

Case1: All EVPN-PEs in the fabric can be made as MVPN exit points

Case2: MVPN network can be attached behind a EVPN PE or subset of EVPN-PEs

Case3: MVPN network (MVPN-PEs) which uses different tunnel model can be directly attached to EVPN fabric.

In gateway model, MVPN routes from one domain are terminated at the gateway PE and re-originated for another domain.

With use case 1 & 2, All PEs connected to an EVPN fabric can use one data plane to send & receive traffic within the fabric/data center. Also, IPVPN routes need not be advertised inside the fabric. Instead, PE where MVPN is terminated should advertise IPVPN as EVPN routes.

With use case 3, Fabric will get two copies per multicast flow, if receivers exist both MVPN and EVPN networks. (Two different data planes are used to send the traffic in the fabric; one for EVPN network and one for MVPN network).

7. Control Plane Operation

In seamless interop between EVPN and MVPN PEs, the control plane may need to setup the following three types of multicast tunnels. The first two are among EVPN PEs only but the third one is among EVPN and MVPN PEs.

- 1) Intra-ES IP multicast tunnel
- 2) Intra-subnet BUM tunnel
- 3) Inter-subnet IP multicast tunnel

7.1. Intra-ES/Intra-Subnet IP Multicast Tunnel

As described in section 6.3.2, when a multicast source is sitting behind an All-Active ES, then an intra-subnet multicast tunnel is needed among the multi-homing EVPN PEs for that ES to carry multicast flow received by one of the multi-homing PEs to the other PEs in that ES. We refer to this multicast tunnel as Intra-ES/Intra-Subnet tunnel. Vast majority of All-Active multi-homing for TOR devices in DC networks are just dual-homing which means the multicast flow received by one of the dual-homing PE only needs to be sent to the

other dual-homing PE. Therefore, a simple ingress replication tunnel is all that is needed. In case of multi-homing to three or more EVPN PEs, then other tunnel types such as P2MP, MP2MP, BIER, and Assisted Replication can be considered. It should be noted that this intra-ES tunnel is only needed for All-Active multi-homing and it is not required for Single-Active multi-homing.

The EVPN PEs belonging to a given All-Active ES discover each other using EVPN Ethernet Segment route per procedures described in [RFC7432]. These EVPN PEs perform DF election per [RFC7432], [EVPN-DF-Framework], or other DF election algorithms to decide who is a DF for a given BD. If the BD belongs to a tenant that has IRB IP multicast enabled for it, then for fixed-mode, each PE sets up an intra-ES tunnel to forward IP multicast traffic received locally on that BD to other multi-homing PE(s) for that ES. Therefore, IP multicast traffic received via a local attachment circuit is sent on this tunnel and on the associated IRB interface for that BT and other local attachment circuits if there are interested receivers for them. The other multi-homing EVPN PEs treat this intra-ES tunnel just like their local ACs - i.e., the multicast traffic received over this tunnel is treated as if it is received via its local AC. Thus, the multi-homing PEs cannot receive the same IP multicast flow from an MVPN tunnel (e.g., over an IRB interface for that BD) because between a source behind a local AC versus a source behind a remote PE, the PE always chooses its local AC.

When ingress replication is used for intra-ES tunnel, every PE in the All-Active multi-homing ES has all the information to setup these tunnels - i.e., a) each PE knows what are the other multi-homing PEs for that ES via EVPN Ethernet Segment route and can use this information to setup intra-ES/Intra-Subnet IP multicast tunnel among themselves.

7.2. Intra-Subnet BUM Tunnel

As the name implies, this tunnel is setup to carry BUM traffic for a given subnet/BD among EVNP PEs. In [RFC7432], this overlay tunnel is used for transmission of all BUM traffic including user IP multicast traffic. However, for multicast traffic handling in EVPN-IRB PEs, this tunnel is used for all broadcast, unknown-unicast, non-IP multicast traffic, and link-local IP multicast traffic - i.e., it is used for all BUM traffic except user IP multicast traffic. This tunnel is setup using IMET route for a given EVI/BD. The composition and advertisement of IMET routes are exactly per [RFC7432]. It should be noted that when an EVPN All-Active multi-homing PE uses both this tunnel as well as intra-ES tunnel, there SHALL be no duplication of multicast traffic over the network because they carry different types of multicast traffic - i.e., intra-ES tunnel among multi-homing PEs

carries only user IP multicast traffic; whereas, intra-subnet BUM tunnel carries link-local IP multicast traffic and BUM traffic (w/ non-IP multicast).

7.3. Inter-Subnet IP Multicast Tunnel

As its name implies, this tunnel is setup to carry IP-only multicast traffic for a given tenant across all its subnets (BDs) among EVPN and MVPN PEs.

The following NLRIs from [RFC6514] is used for setting up this inter-subnet tunnel in the network.

Intra-AS I-PMSI A-D route is used for the setup of default underlay tunnel (also called inclusive tunnel) for a tenant IP-VRF. The tunnel attributes are indicated using PMSI attribute with this route.

S-PMSI A-D route is used for the setup of Customer flow specific underlay tunnels. This enables selective delivery of data to PEs having active receivers and optimizes fabric bandwidth utilization. The tunnel attributes are indicated using PMSI attribute with this route.

Each EVPN PE supporting a specific MVPN instance discovers the set of other PEs in its AS that are attached to sites of that MVPN using Intra-AS I-PMSI A-D route (route type 1) per [RFC6514]. It can also discover the set of other ASes that have PEs attached to sites of that MVPN using Inter-AS I-PMSI A-D route (route type 2) per [RFC6514]. After the discovery of PEs that are attached to sites of the MVPN, an inclusive overlay tree (I-PMSI) can be setup for carrying tenant multicast flows for that MVPN; however, this is not a requirement per [RFC6514] and it is possible to adopt a policy in which all tenant flows are carried on S-PMSIs.

An EVPN-IRB PE sends a user IP multicast flow to other EVPN and MVPN PEs over this inter-subnet tunnel that is instantiated using MVPN I-PMSI or S-PMSI. This tunnel can be considered as being originated and terminated from/to among IP-VRFs of EVPN/MVPN PEs; whereas, intra-subnet tunnel is originated/terminated among MAC-VRFs of EVPN PEs.

7.4. IGMP Hosts as TSes

If a tenant system which is an IGMP host is multi-homed to two or more EVPN PEs using All-Active multi-homing, then IGMP join and leave messages are synchronized between these EVPN PEs using EVPN IGMP Join Synch route (route type 7) and EVPN IGMP Leave Synch route (route type 8) per [IGMP-PROXY]. IGMP states are built in the corresponding

BDs of the multi-homing EVPN PEs. In [IGMP-PROXY] the DF PE for that BD originates an EVPN Selective Multicast Tag route (SMET route) route to other EVPN PEs. However, in here there is no need to use SMET because the IGMP messages are terminated by the EVPN-IRB PE and tenant (*,G) or (S,G) join messages are sent via MVPN Shared Tree Join route (route type 6) or Source Tree Join route (route type 7) respectively of MCAST-VPN NLRI per [RFC6514]. In case of a network with only IGMP hosts, the preferred mode of operation is that of Shortest Path Tree (SPT) per section 14 of [RFC6514]. This mode is only supported for PIM-SM and avoids the RP configuration overhead. Such mode is chosen by provisioning/ configuration.

7.5. TS PIM Routers

Just like a MVPN PE, an EVPN PE runs a separate tenant multicast routing instance (VPN-specific) per MVPN instance and the following tenant multicast routing instances are supported:

- PIM Sparse Mode (PIM-SM) with the ASM service model
- PIM Sparse Mode with the SSM service model
- PIM Bidirectional Mode (BIDIR-PIM), which uses bidirectional tenant-trees to support the ASM service model

A given tenant's PIM join messages for (*,G) or (S, G) are processed by the corresponding tenant multicast routing protocol and they are advertised over MPLS/IP network using Shared Tree Join route (route type 6) and Source Tree Join route (route type 7) respectively of MCAST-VPN NLRI per [RFC6514].

8 Data Plane Operation

When an EVPN-IRB PE receives an IGMP/MLD join message over one of its Attachment Circuits (ACs), it adds that AC to its Layer-2 (L2) OIF list. This L2 OIF list is associated with the MAC-VRF/BT corresponding to the subnet of the tenant device that sent the IGMP/MLD join. Therefore, tenant (S,G) or (*,G) forwarding entries are created/updated for the corresponding MAC-VRF/BT based on these source and group IP addresses. Furthermore, the IGMP/MLD join message is propagated over the corresponding IRB interface and it is processed by the tenant multicast routing instance which creates the corresponding tenant (S,G) or (*,G) Layer-3 (L3) forwarding entries. It adds this IRB interface to the L3 OIF list. An IRB is removed as a L3 OIF when all L2 tenant (S,G) or (*,G) forwarding states is removed for the MAC-VRF/BT associated with that IRB. Furthermore, tenant (S,G) or (*,G) L3 forwarding state is removed when all of its L3 OIFs are removed - i.e., all the IRB and L3 interfaces associated with that tenant (S,G) or (*,G) are removed.

When an EVPN PE receives IP multicast traffic from one of its AC, if it has any attached receivers for that subnet, it performs L2 switching of the intra-subnet traffic within the BT attached to that AC. If the multicast flow is received over an AC that belongs to an All-Active ES, then the multicast flow is also sent over the intra-ES/Intra-Subnet tunnel among multi-homing PEs. The EVPN PE then sends the multicast traffic over the corresponding IRB interface. The multicast traffic then gets routed in the corresponding IP-VRF and it gets forwarded to interfaces in the L3 OIF list which can include other IRB interfaces, other L3 interfaces directly connected to TSeS, and the MVPN Inter-Subnet tunnel which is instantiated by an I-PMSI or S-PMSI tunnel. When the multicast packet is routed within the IP-VRF of the EVPN PE, its Ethernet header is stripped and its TTL gets decremented as the result of this IP routing. When the multicast traffic is received on an IRB interface by the BT corresponding to that interface, it gets L2 switched and sent over ACs that belong to the L2 OIF list.

8.1 Intra-Subnet L2 Switching

Rcvr1 in Figure 1 is connected to PE1 in MAC-VRF1 (same as Src1) and sends IGMP join for (C-S, C-G), IGMP snooping will record this state in local bridging entry. A routing entry will be formed as well which will point to MAC-VRF1 as RPF for Src1. We assume that Src1 is known via ARP or similar procedures. Rcvr1 will get a locally bridged copy of multicast traffic from Src1. Rcvr3 is also connected in MAC-VRF1 but to PE2 and hence would send IGMP join which will be recorded at PE2. PE2 will also form routing entry and RPF will be assumed as Tenant Tunnel "Tenant1" formed beforehand using MVPN procedures. Also this would cause multicast control plane to initiate a BGP MCAST-VPN type 7 route which would include VRI for PE1 and hence be accepted on PE1. PE1 will include Tenant1 tunnel as Outgoing Interface (OIF) in the routing entry. Now, since it has knowledge of remote receivers via MVPN control plane it will encapsulate original multicast traffic in Tenant1 tunnel towards core.

8.2 Inter-Subnet L3 Routing

Rcvr2 in Figure 1 is connected to PE1 in MAC-VRF2 and hence PE1 will record its membership in MAC-VRF2. Since MAC-VRF2 is enabled with IRB, it gets added as another OIF to routing entry formed for (C-S, C-G). Rcvr2 and Rcvr4 are also in different MAC-VRFs than multicast speaker Src1 and hence need Inter-subnet forwarding. PE2 will form local bridging entry in MAC-VRF2 due to IGMP joins received from Rcvr3 and Rcvr4 respectively. PE2 now adds another OIF 'MAC-VRF2' to its existing routing entry. But there is no change in control plane states since its already sent MVPN route and no further signaling is

required. Also since Src1 is not part of MAC-VRF2 subnet, it is treated as routing OIF and hence MAC header gets modified as per normal procedures for routing. PE3 forms routing entry very similar to PE2. It is to be noted that PE3 does not have MAC-VRF1 configured locally but still can receive the multicast data traffic over Tenant1 tunnel formed due to MVPN procedures

9. DCs with only EVPN PEs

As mentioned earlier, the proposed solution can be used as a routed multicast solution in data center networks with only EVPN PEs (e.g., routed multicast VPN only among EVPN PEs). It should be noted that the scope of intra-subnet forwarding for the solution described in this document, is limited to a single EVPN PE for Single-Active multi-homing and to multi-homing PEs for All-Active multi-homing. In other words, the IP multicast traffic that needs to be forwarded from the source PE to remote PEs is routed to remote PEs regardless of whether the traffic is intra-subnet or inter-subnet. As the result, the TTL value for intra-subnet traffic that spans across two or more PEs get decremented.

However, if there are applications that require intra-subnet multicast traffic to be L2 forwarded, Section 11 discusses some options to support applications having TTL value 1. The procedure discussed in Section 11 may be used to support applications that require intra-subnet multicast traffic to be L2 forwarded.

9.1. Setup of overlay multicast delivery

It must be emphasized that this solution poses no restriction on the setup of the tenant BDs and that neither the source PE, nor the receiver PEs do not need to know/learn about the BD configuration on other PEs in the MVPN. The Reverse Path Forwarder (RPF) is selected per the tenant multicast source and the IP-VRF in compliance with the procedures in [RFC6514], using the incoming EVPN route type 2 or 5 NLRI per [RFC7432].

The VRF Route Import (VRI) extended community that is carried with the IP-VPN routes in [RFC6514] MUST be carried with the EVPN unicast routes when these routes are used. The construction and processing of the VRI are consistent with [RFC6514]. The VRI MUST uniquely identify the PE which is advertising a multicast source and the IP-VRF it resides in.

VRI is constructed as following:

- The 4-octet Global Administrator field MUST be set to an IP

address of the PE. This address SHOULD be common for all the IP-VRFs on the PE (e.g., this address may be the PE's loopback address or VTEP address).

- The 2-octet Local Administrator field associated with a given IP-VRF contains a number that uniquely identifies that IP-VRF within the PE that contains the IP-VRF.

EVPN PE MUST have Route Target Extended Community to import/export MVPN routes. In data center environment, it is desirable to have this RT configured using auto-generated method than static configuration.

The following is one recommended model to auto-generate MVPN RT:

- The Global Administrator field of the MVPN RT MAY be set to BGP AS Number.
- The Local Administrator field of the MVPN RT MAY be set to the VNI associated with the tenant VRF.

Every PE which detects a local receiver via a local IGMP join or a local PIM join for a specific source (overlay SSM mode) MUST terminate the IGMP/PIM signaling at the IP-VRF and generate a (C-S,C-G) via the BGP MCAST-VPN route type 7 per [RFC6514] if and only if the RPF for the source points to the fabric. If the RPF points to a local multicast source on the same MAC-VRF or a different MAC-VRF on that PE, the MCAST-VPN MUST NOT be advertised and data traffic will be locally routed/bridged to the receiver as detailed in section 6.2.

The VRI received with EVPN route type 2 or 5 NLRI from source PE will be appended as an export route-target extended community. More details about handling of various types of local receivers are in section 10. The PE which has advertised the unicast route with VRI, will import the incoming MCAST-VPN NLRI in the IP-VRF with the same import route-target extended-community and other PEs SHOULD ignore it. Following such procedure the source PE learns about the existence of at least one remote receiver in the tenant overlay and programs data plane accordingly so that a single copy of multicast data is forwarded into the fabric using tenant VRF tunnel.

If the multicast source is unknown (overlay ASM mode), the MCAST-VPN route type 6 (C-*,C-G) join SHOULD be targeted towards the designated overlay Rendezvous Point (RP) by appending the received RP VRI as an export route-target extended community. Every PE which detects a local source, registers with its RP PE. That is how the RP learns about the tenant source(s) and group(s) within the MVPN. Once the overlay RP PE receives either the first remote (C-RP,C-G) join or a local IGMP/PIM join, it will trigger an MCAST-VPN route type 7 (C-

S,C-G) towards the actual source PE for which it has received PIM register message in full compliance with regular PIM procedures. This involves the source PE to advertise the MCAST-VPN Source Active A-D route (MCAST-VPN route-type 5) towards all PEs. The Source Active A-D route is used to inform all PEs in a given MVPN about the active multicast source for switching from RPT to SPT when MVPNs use tenant RP-shared trees (i.e., rooted at tenant's RP) per section 13 of [RFC6514]. This is done in order to choose a single forwarder PE and to suppress receiving duplicate traffic. In such scenarios, the active multicast source is used by the receiver PEs to join the SPT if they have not received tenant (S,G) joins and by the RPT PEs to prune off the tenant (S,G) state from the RPT. The Source Active A-D route is also used for MVPN scenarios without tenant RP-shared trees. In such scenarios, the receiver PEs with tenant (*,G) states use the Source Active A-D route to know which upstream PEs with sources behind them to join per section 14 of [RFC6514] - i.e., to suppress joining Overlay shared tree.

9.2. Handling of different encapsulations

Just as in [RFC6514] the MVPN I-PMSI and S-PMSI A-D routes are used to form the overlay multicast tunnels and signal the tunnel type using the P-Multicast Service Interface Tunnel (PMSI Tunnel) attribute.

9.2.1. MPLS Encapsulation

The [RFC6514] assumes MPLS/IP core and there is no modification to the signaling procedures and encoding for PMSI tunnel formation therein. Also, there is no need for a gateway to inter-operate with non-EVPN PEs supporting [RFC6514] based MVPN over IP/MPLS.

9.2.2 VxLAN Encapsulation

In order to signal VXLAN, the corresponding BGP encapsulation extended community [TUNNEL-ENCAP] SHOULD be appended to the MVPN I-PMSI and S-PMSI A-D routes. The MPLS label in the PMSI Tunnel Attribute MUST be the Virtual Network Identifier (VNI) associated with the customer MVPN. The supported PMSI tunnel types with VXLAN encapsulation are: PIM-SSM Tree, PIM-SM Tree, BIDIR-PIM Tree, Ingress Replication [RFC6514]. Further details are in [RFC8365].

In this case, a gateway is needed for inter-operation between the EVPN PEs and non-EVPN MVPN PEs. The gateway should re-originate the control plane signaling with the relevant tunnel encapsulation on either side. In the data plane, the gateway terminates the tunnels formed on either side and performs the relevant stitching/re-

encapsulation on data packets.

9.2.3. Other Encapsulation

In order to signal a different tunneling encapsulation such as NVGRE, GPE, or GENEVE the corresponding BGP encapsulation extended community [TUNNEL-ENCAP] SHOULD be appended to the MVPN I-PMSI and S-PMSI A-D routes. If the Tunnel Type field in the encapsulation extended-community is set to a type which requires Virtual Network Identifier (VNI), e.g., VXLAN-GPE or NVGRE [TUNNEL-ENCAP], then the MPLS label in the PMSI Tunnel Attribute MUST be the VNI associated with the customer MVPN. Same as in VXLAN case, a gateway is needed for inter-operation between the EVPN-IRB PEs and non-EVPN MVPN PEs.

10. DCI with MPLS in WAN and VxLAN in DCs

This section describes the inter-operation between MVPN PEs in WAN using MPLS encapsulation with EVPN PEs in a DC network using VxLAN encapsulation. Since the tunnel encapsulation between these networks are different, we must have at least one gateway in between. Usually, two or more are required for redundancy and load balancing purpose. In such scenarios, a DC network can be represented as a customer network that is multi-homed to two or more MVPN PEs via L3 interfaces and thus standard MVPN multi-homing procedures are applicable here. It should be noted that a MVPN overlay tunnel over the DC network is terminated on the IP-VRF of the gateway and not the MAC-VRF/BTs. Therefore, the considerations for loop prevention and split-horizon filtering described in [INTERCON-EVPN] are not applicable here. Some aspects of the multi-homing between VxLAN DC networks and MPLS WAN is in common with [INTERCON-EVPN].

10.1. Control plane inter-connect

The gateway(s) MUST be setup with the inclusive set of all the IP-VRFs that span across the two domains. On each gateway, there will be at least two BGP sessions: one towards the DC side and the other towards the WAN side. Usually for redundancy purpose, more sessions are setup on each side. The unicast route propagation follows the exact same procedures in [INTERCON-EVPN]. Hence, a multicast host located in either domain, is advertised with the gateway IP address as the next-hop to the other domain. As a result, PEs view the hosts in the other domain as directly attached to the gateway and all inter-domain multicast signaling is directed towards the gateway(s). Received MVPN routes type 1-7 from either side of the gateway(s), MUST NOT be reflected back to the same side but processed locally and re-advertised (if needed) to the other side:

- Intra-AS I-PMSI A-D Route: these are distributed within

each domain to form the overlay tunnels which terminate at gateway(s). They are not passed to the other side of the gateway(s).

- C-Multicast Route: joins are imported into the corresponding IP-VRF on each gateway and advertised as a new route to the other side with the following modifications (the rest of NLRI fields and path attributes remain on-touched):

- * Route-Distinguisher is set to that of the IP-VRF

- * Route-target is set to the exported route-target list on IP-VRF

- * The PMSI tunnel attribute and BGP Encapsulation extended community will be modified according to section 8

- * Next-hop will be set to the IP address which represents the gateway on either domain

- Source Active A-D Route: same as joins

- S-PMSI A-D Route: these are passed to the other side to form selective PMSI tunnels per every (C-S,C-G) from the gateway to the PEs in the other domain provided it contains receivers for the given (C-S, C-G). Similar modifications made to joins are made to the newly originated S-PMSI.

In addition, the Originating Router's IP address is set to GW's IP address. Multicast signaling from/to hosts on local ACs on the gateway(s) are generated and propagated in both domains (if needed) per the procedures in section 7 in this document and in [RFC6514] with no change. It must be noted that for a locally attached source, the gateway will program an OIF per every domain from which it receives a remote join in its forwarding plane and different encapsulation will be used on the data packets.

10.2. Data plane inter-connect

Traffic forwarding procedures on gateways are same as those described for PEs in section 5 and 6 except that, unlike a non-border leaf PE, the gateway will not only route the incoming traffic from one side to its local receivers, but will also send it to the remote receivers in the the other domain after de-capsulation and appending the right encapsulation. The OIF and IIF are programmed in FIB based on the received joins from either side and the RPF calculation to the source or RP. The de-capsulation and encapsulation actions are programmed based on the received I-PMSI or S-PMSI A-D routes from either sides. If there are more than one gateway between two domains, the multi-

homing procedures described in the following section must be considered so that incoming traffic from one side is not looped back to the other gateway.

The multicast traffic from local sources on each gateway flows to the other gateway with the preferred WAN encapsulation.

11. Supporting application with TTL value 1

It is possible that some deployments may have a host on the tenant domain that sends multicast traffic with TTL value 1. The interested receiver for that traffic flow may be attached to different PEs on the same subnet. The procedures specified in section 6 always routes the traffic between PEs for both intra and inter subnet traffic. Hence traffic with TTL value 1 is dropped due to the nature of routing.

This section discusses few possible ways to support traffic having TTL value 1. Implementation MAY support any of the following model.

11.1. Policy based model

Policies may be used to enforce EVPN BUM procedure for traffic flows with TTL value 1. Traffic flow that matches the policy is excluded from seamless interop procedure specified in this document, hence TTL decrement issue will not apply.

11.2. Exercising BUM procedure for VLAN/BD

Servers/hosts sending the traffic with TTL value 1 may be attached to a separate VLAN/BD, where multicast routing is disabled. When multicast routing is disabled, EVPN BUM procedure may be applied to all traffic ingressing on that VLAN/BD. On the Egress PE, the RPF for such traffic may be set to BD interface, where the source is attached.

11.3. Intra-subnet bridging

The procedure specified in the section enables a PE to detect an attached subnet source (i.e., source that is directly attached in the tenant BD/VLAN). By applying the following procedure for the attached source, Traffic flows having TTL value 1 can be supported.

- On the ingress PE, do the bridging on the interface towards the core interface
- On the egress side, make a decision whether to bridge or route at the outgoing interface (OIF) based on whether the source is

attached to the OIF's BD/VLAN or not.

Recent ASIC supports single lookup forwarding for brigading and routing (L2+L3). The procedure mentioned here leverages this ASIC capability.

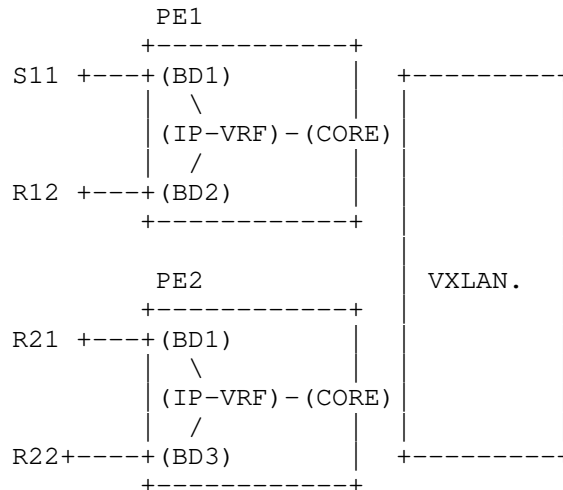


Figure 3 Intra-subnet bridging

Consider the above picture. In the picture

- PE1 and PE2 are seamless interop capable PEs
- S11 is a multicast host directly attached to PE1 in BD1
- Source S11 sends traffic to Group G11
- R21, R22 are IGMP receivers for group G11
- R21 and R22 are attached to BD1 and BD3 respectively at PE2.

When source S11 starts sending the traffic, PE1 learns the source and announces the source using MVPN procedures to the remote PEs.

At PE2, IGMP joins from R21, R22 result the creation of (*,G11) entry with outgoing OIF as IRB interface of BD1 and BD3. When PE2 learns the source information from PE1, it installs the route (S11, G11) at the tenant VRF with RPF as CORE interface.

PE2 inherits (*, G11) OIFs to (S11, G11) entry. While inheriting OIF, PE2 checks whether source is attached to OIF's subnet. OIF matching source subnet is added with flag indicating bridge only interface. In case of (S11, G11) entry, BD1 is added as the bridge only OIF, while BD3 is added as normal OIF(L3 OIF).

PEs (PE2) sends MVPN join (S11, G11) towards PE1, since it has local receivers.

At Ingress PE(PE1), CORE interface is added to (S11, G11) entry as an OIF (outgoing interface) with a flag indicating that bridge only interface. With this procedure, ingress PE(PE1) bridges the traffic on CORE interface. (PE1 retains the TTL and source-MAC). The traffic is encapsulated with VNI associated with CORE interface(L3VNI). PE1 also routes the traffic for R12 which is attached to BD2 on the same device.

PE2 decapsulates the traffic from PE1 and does inner lookup on the tenant VRF associated with incoming VNI. Traffic lookup on the tenant VRF yields (S11, G11) entry as the matching entry. Traffic gets bridged on BD1 (PE2 retains the TTL and source-MAC) since the OIF is marked as bridge only interface. Traffic gets routed on BD2.

12. Interop with L2 EVPN PEs

A gateway device is needed to do interop between EVPN PEs that support seamless interop procedure specified in this document and native EVPN-PEs(L2EVPN PE). The gateway device uses BUM tunnel when interworking with L2EVPN-PEs.

Interop procedure will be covered in the next version of the draft.

13. Connecting external Multicast networks or PIM routers.

External multicast networks or PIM routers can be attached to any seamless interop capable EVPN-PEs or set of EVPN-PEs. Multicast network or PIM router can also be attached to any IRB enabled BDI interface or L3 enabled interface or set of interfaces. The fabric can be used as a Transit network. All PIM signaling is terminated at EVPN-PEs.

No additional procedures are required while connecting external multicast networks.

14. RP handling

This section describes various RP models for a tenant VRF. The RP model SHOULD be consistent across all EVPN-PEs for given group/group range in the tenant VRF.

14.1. Various RP deployment options

14.1.1. RP-less mode

EVPN fabric without having any external multicast network/attached MVPN network, doesn't need RP configuration. A configuration option SHALL be provided to the end user to operate the fabric in RP less mode. When an EVPN-PE is operating in RP-less mode, EVPN-PE MUST advertise all attached sources to remote EVPN PEs using procedure specified in [RFC 6514].

In RP less mode, (C-*,C-G) RPF may be set to NULL or may be set to wild card interface(Any interface on the tenant VRF). In RP-less mode, traffic is always forwarded based on (C-S,C-G) state.

14.1.2. Fabric anycast RP

In this model, anycast GW IP address is configured as RP in all EVPN-PE. When an EVPN-PE is operating in Fabric anycast-RP mode, an EVPN-PE MUST advertise all sources behind that PE to other EVPN PEs using procedure specified in [RFC 6514]. In this model, Sources may be directly attached to tenant BDs or sources may be attached behind a PIM router (In that case EVPN-PE learns source information due to PIM register terminating at RP interface at the tenant VRF side)

In RP-less mode and Fabric anycast RP mode, EVPN-PE operates SPT-only mode as per section 14 of RFC 6514.

14.1.3. Static RP

The procedure specified in this document supports configuring EVPN fabric with static RP. RP can be configured in the EVPN-PE itself in the tenant VRF or in the external multicast networks connected behind an EVPN PE or in the MVPN network. When RPF is not local to EVPN-PE, EVPN-PE operates in rpt-spt mode as PER procedures specified in section 13 of RFC 6514.

14.1.4. Co-existence of Fabric anycast RP and external RP

External multicast network using its own RP may be connected to EVPN fabric operating with Fabric anycast RP mode. In this case, subset of EVPN-PEs may be designated as border leafs. Anycast RP may be configured between border leafs and external RP. Border leafs originates SA-AD routes for external sources towards fabric PEs. Border leaf acts as FHR for the sources inside the fabric. Configuration option may be provided to define the PE role as BL.

14.2. RP configuration options

PIM Bidir and PIM-SM ASM mode require Rendezvous point (RP) configuration, which acts as a shared root for a multicast shared tree. RP can be configured using static configuration or by using BSR

or Auto-RP procedures on the tenant VRF. This document only discusses static RP configuration. The use of BSR or Auto-RP procedure in the EVPN fabric is beyond the scope of this document.

15. IANA Considerations

IANA is requested to assign new flags in the "Multicast Flags Extended Community Flags" registry for the following.

- o Seamless interop capable PE

16. Security Considerations

All the security considerations in [RFC7432] apply directly to this document because this document leverages [RFC7432] control plane and their associated procedures.

17. Acknowledgements

The authors would like to thank Niloofar Fazlollahi, Aamod Vyavaharkar, Raunak Banthia, and Swadesh Agrawal for their discussions and contributions.

18. References

18.1. Normative References

- [RFC7432] A. Sajassi, et al., "BGP MPLS Based Ethernet VPN", RFC 7432, February 2015.
- [RFC8365] A. Sajassi, et al., "A Network Virtualization Overlay Solution using EVPN", RFC 8365, February 2018.
- [RFC6513] E. Rosen, et al., "Multicast in MPLS/BGP IP VPNs", RFC6513, February 2012.
- [RFC6514] R. Aggarwal, et al., "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC6514, February 2012.
- [EVPN-IRB] A. Sajassi, et al., "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-03, February 2017.
- [EVPN-IRB-MCAST] A. Rosen, et al., "EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding", draft-lin-bess-evpn-irb-

mcast-04, October 24, 2017.

18.2. Informative References

- [RFC7080] A. Sajassi, et al., "Virtual Private LAN Service (VPLS) Interoperability with Provider Backbone Bridges", RFC 7080, December 2013.
- [RFC7209] D. Thaler, et al., "Requirements for Ethernet VPN (EVPN)", RFC 7209, May 2014.
- [RFC4389] A. Sajassi, et al., "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, April 2006.
- [RFC4761] K. Kompella, et al., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.
- [INTERCON-EVPN] J. Rabadan, et al., "Interconnect Solution for EVPN Overlay networks", <https://tools.ietf.org/html/draft-ietf-bess-dci-evpn-overlay-04>, September 2016
- [TUNNEL-ENCAPS] E. Rosen, et al. "The BGP Tunnel Encapsulation Attribute", <https://tools.ietf.org/html/draft-ietf-idr-tunnel-encaps-06>, work in progress, June 2017.
- [EVPN-IGMP-PROXY] A. Sajassi, et. al., "IGMP and MLD Proxy for EVPN", draft-ietf-bess-evpn-igmp-mld-proxy-01, work in progress, March 2018.
- [EVPN-PIM-PROXY] J. Rabadan, et. al., "PIM Proxy in EVPN Networks", draft-skr-bess-evpn-pim-proxy-00, work in progress, July 3, 2017.

19. Authors' Addresses

Ali Sajassi
Cisco
170 West Tasman Drive
San Jose, CA 95134, US
Email: sajassi@cisco.com

Kesavan Thiruvengkatasamy
Cisco
170 West Tasman Drive
San Jose, CA 95134, US
Email: kethiruv@cisco.com

Samir Thoria
Cisco
170 West Tasman Drive
San Jose, CA 95134, US
Email: sthoria@cisco.com

Ashutosh Gupta
Avi Networks
Email: ashutosh@avinetworks.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com

Appendix A. Use Cases

A.1. DCs with only IGMP/MLD hosts w/o tenant router

In a EVPN network consisting of only IGMP/MLD hosts, PE's will receive IGMP (*, G) or (S, G) joins from their locally attached host and would originate MVPN C-Multicast Route Type 6 and 7 NLRI's respectively. As described in RFC 6514 these NLRI's are directed towards RP-PE for Type 6 or Source-PE for Type 7. In case of (*, G) join a Shared-Path Tree will be built in the core from RP-PE towards all Receiver-PE's. Once a Source starts to send Multicast data to specified multicast-group, the PE directly connected to Source will do PIM-registration with RP. Since there are existing receivers for the Group, RP will originate a PIM (S, G) join towards Source. This will

be converted to MVPN Type 7 NLRI by RP-PE. Please note that the router RP-PE would be the PE configured as RP (e.g., using static configuration or by using BSR or Auto-RP procedures). The detailed working of such protocols is beyond the scope of this document. Upon receiving Type 7 NLRI, Source-PE will include MVPN Tunnel in its Outgoing Interface List. Furthermore, Source-PE will follow the procedures in RFC-6514 to originate MVPN SA-AD route (RT 5) to avoid duplicate traffic and allow all Receiver-PE's to shift from Share-Tree to Shortest-Path-Tree rooted at Source-PE. Section 13 of [RFC6514] describes it.

However a network operator can chose to have only Shortest-Path-Tree built in MVPN core as described in section 14 of [RFC6514]. One way to achieve this, is for all PE's act as RP for its locally connected hosts and thus avoid sending any Shared-Tree Join (MVPN Type 6) into the core. In this scenario, there will be no PIM registration needed since all PE's are first-hop router as well as acting RP. Once a source starts to send multicast data, the PE directly connected to it originates Source-Active AD (RT 5) to all other PE's in network. Upon Receiving Source-Active AD route a PE must cache it in its local database and also look for any matching interest for (*, G) where G is the multicast group described in received Source-Active AD route. If it finds any such matching entry, it must originate a C-Multicast route (RT 7) in order to start receiving traffic from Source-PE. This procedure must be repeated on reception of any further Source-Active AD routes.

A.2. DCs with mixed of IGMP/MLD hosts & multicast routers running PIM-SSM

This scenario has multicast routers which can send PIM SSM (S, G) joins. Upon receiving these joins and if source described in join is learnt to be behind a MVPN peer PE, local PE will originate C-Multicast Join (RT 7) towards Source-PE. It is expected that PIM SSM group ranges are kept separate from ASM range for which IGMP hosts can send (*, G) joins. Hence both ASM and SSM groups shall operate without any overlap. There is no RP needed for SSM range groups and Shortest Path tree rooted at Source is built once a receiver interest is known.

A.3. DCs with mixed of IGMP/MLD hosts & multicast routers running PIM-ASM

This scenario includes reception of PIM (*, G) joins on PE's local AC. These joins are handled similar to IGMP (*, G) join as explained in sections above. Another interesting case can arise here is when one of the tenant routers can act as RP for some of the ASM Groups. In such scenario, a Upstream Multicast Hop (UMH) will be elected by other PE's in order to send C-Multicast Routes (RT 6). All procedures described in RFC 6513 with respect to UMH should be used to avoid traffic duplication due to incoherent selection of RP-PE by different Receiver-PE's.

A.4. DCs with mixed of IGMP/MLD hosts & multicast routers running PIM-Bidir

Creating Bidirectional (*, G) trees is useful when a customer wants least amount of control state in network. But on downside all receivers for a particular multicast group receive traffic from all sources sending to that group. However for the purpose of this document, all procedures as described in RFC 6513 and RFC 6514 apply when PIM-Bidir is used.

BESS Workgroup
Internet Draft

Intended status: Standards Track

J. Rabadan, Ed.
J. Kotalwar
S. Sathappan
Nokia

Z. Zhang
W. Lin
Juniper

E. Rosen
Individual

Expires: January 6, 2020

July 5, 2019

Multicast Source Redundancy in EVPN Networks
draft-skr-bess-evpn-redundant-mcast-source-01

Abstract

EVPN supports intra and inter-subnet IP multicast forwarding. However, EVPN (or conventional IP multicast techniques for that matter) do not have a solution for the case where: a) a given multicast group carries more than one flow (i.e., more than one source), and b) it is desired that each receiver gets only one of the several flows. Existing multicast techniques assume there are no redundant sources sending the same flow to the same IP multicast group, and, in case there were redundant sources, the receiver's application would deal with the received duplicated packets. This document extends the existing EVPN specifications and assumes that IP Multicast source redundancy may exist. It also assumes that, in case two or more sources send the same IP Multicast flows into the tenant domain, the EVPN PEs need to avoid that the receivers get packet duplication by following the described procedures.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering

Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>

This Internet-Draft will expire on January 6, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
1.1 Terminology	4
1.2 Background on IP Multicast Delivery in EVPN Networks	6
1.2.1 Intra-subnet IP Multicast Forwarding	6
1.2.2 Inter-subnet IP Multicast Forwarding	7
1.3 Multi-Homed IP Multicast Sources in EVPN	9
1.4 The Need for Redundant IP Multicast Sources in EVPN	11
2. Solution Overview	11
3. BGP EVPN Extensions	13
4. Warm Standby (WS) Solution for Redundant G-Sources	14
4.1 WS Example in an OISM Network	16
4.2 WS Example in a Single-BD Tenant Network	18

5. Hot Standby (HS) Solution for Redundant G-Sources	19
5.1 Use of BFD in the HS Solution	22
5.2 HS Example in an OISM Network	22
5.3 HS Example in a Single-BD Tenant Network	27
6. Security Considerations	27
7. IANA Considerations	27
8. References	27
8.1. Normative References	27
8.2. Informative References	28
9. Acknowledgments	28
10. Contributors	28
Authors' Addresses	29

1. Introduction

Intra and Inter-subnet IP Multicast forwarding are supported in EVPN networks. [IGMP-PROXY] describes the procedures required to optimize the delivery of IP Multicast flows when Sources and Receivers are connected to the same EVPN BD (Broadcast Domain), whereas [OISM] specifies the procedures to support Inter-subnet IP Multicast in a tenant network. Inter-subnet IP Multicast means that IP Multicast Source and Receivers of the same multicast flow are connected to different BDs of the same tenant.

[IGMP-PROXY], [OISM] or conventional IP multicast techniques do not have a solution for the case where a given multicast group carries more than one flow (i.e., more than one source) and it is desired that each receiver gets only one of the several flows. Multicast techniques assume there are no redundant sources sending the same flows to the same IP multicast group, and, in case there were redundant sources, the receiver's application would deal with the received duplicated packets.

As a workaround in conventional IP multicast (PIM or MVPN networks), if all the redundant sources are given the same IP address, each receiver will get only one flow. The reason is that, in conventional IP multicast, (S,G) state is always created by the RP, and sometimes by the Last Hop Router (LHR). The (S,G) state always binds the (S,G) flow to a source-specific tree, rooted at the source IP address. If multiple sources have the same IP address, one may end up with multiple (S,G) trees. However, the way the trees are constructed ensures that any given LHR or RP is on at most one of them. The use of an anycast address assigned to multiple sources may be useful for warm standby redundancy solutions. However, on one hand, it's not really helpful for hot standby redundancy solutions and on the other hand, configuring the same IP address (in particular IPv4 address) in

multiple sources may bring issues if the sources need to be reached by IP unicast traffic or if the sources are attached to the same Broadcast Domain.

In addition, in the scenario where several G-sources are attached via EVPN/OISM, there is not necessarily any (S,G) state created for the redundant sources. The LHRs may have only (*,G) state, and there may not be an RP (creating (S,G) state) either. Therefore, this document extends the above two specifications and assumes that IP Multicast source redundancy may exist. It also assumes that, in case two or more sources send the same IP Multicast flows into the tenant domain, the EVPN PEs need to avoid that the receivers get packet duplication.

The solution provides support for Warm Standby (WS) and Hot Standby (HS) redundancy. WS is defined as the redundancy scenario in which the upstream PEs attached to the redundant sources of the same tenant, make sure that only one source of the same flow can send multicast to the interested downstream PEs at the same time. In HS the upstream PEs forward the redundant multicast flows to the downstream PEs, and the downstream PEs make sure only one flow is forwarded to the interested attached receivers.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

- o OISM: Optimized Inter-Subnet Multicast, as in [OISM].
- o Broadcast Domain (BD): an emulated ethernet, such that two systems on the same BD will receive each other's link-local broadcasts. In this document, BD also refers to the instantiation of a Broadcast Domain on an EVPN PE. An EVPN PE can be attached to one or multiple BDs of the same tenant.
- o Designated Forwarder (DF): as defined in [RFC7432], an ethernet segment may be multi-homed (attached to more than one PE). An ethernet segment may also contain multiple BDs, of one or more EVIs. For each such EVI, one of the PEs attached to the segment becomes that EVI's DF for that segment. Since a BD may belong to only one EVI, we can speak unambiguously of the BD's DF for a given segment.
- o Upstream PE: in this document an Upstream PE is referred to as the

EVPN PE that is connected to the IP Multicast source or closest to it. It receives the IP Multicast flows on local ACs (Attachment Circuits).

- o Downstream PE: in this document a Downstream PE is referred to as the EVPN PE that is connected to the IP Multicast receivers and gets the IP Multicast flows from remote EVPN PEs.
- o G-traffic: any frame with an IP payload whose IP Destination Address (IP DA) is a multicast group G.
- o G-source: any system sourcing traffic to G.
- o SFG: Single Flow Group, i.e., a multicast group address G which represents traffic that contains only a single flow. However, multiple sources - with the same or different IP - may be transmitting an SFG.
- o Redundant G-source: a host or router that transmits an SFG in a tenant network where there are more hosts or routers transmitting the same SFG. Redundant G-sources for the same SFG SHOULD have different IP addresses, although they MAY have the same IP address when in different BDs of the same tenant network. Redundant G-sources are assumed NOT to be "bursty" in this document (typical example are Broadcast TV G-sources or similar).
- o P-tunnel: Provider tunnel refers to the type of tree a given upstream EVPN PE uses to forward multicast traffic to downstream PEs. Examples of P-tunnels supported in this document are Ingress Replication (IR), Assisted Replication (AR), BIER, mLDP or P2MP RSVP-TE.
- o Inclusive Multicast Tree or Inclusive Provider Multicast Service Interface (I-PMSI): defined in [RFC6513], in this document it is applicable only to EVPN and refers to the default multicast tree for a given BD. All the EVPN PEs that are attached to a specific BD belong to the I-PMSI for the BD. The I-PMSI trees are signaled by EVPN Inclusive Multicast Ethernet Tag (IMET) routes.
- o Selective Multicast Tree or Selective Provider Multicast Service Interface (S-PMSI): defined in [RFC6513], in this document it is applicable only to EVPN and refers to the multicast tree to which only the interested PEs of a given BD belong to. There are two types of EVPN S-PMSIs:
 - EVPN S-PMSIs that require the advertisement of S-PMSI AD routes from the upstream PE, as in [EVPN-BUM]. The interested downstream PEs join the S-PMSI tree as in [EVPN-BUM].

- EVPN S-PMSIs that don't require the advertisement of S-PMSI AD routes. They use the forwarding information of the IMET routes, but upstream PEs send IP Multicast flows only to downstream PEs issuing Selective Multicast Ethernet Tag (SMET) routes for the flow. These S-PMSIs are only supported with the following P-tunnels: Ingress Replication (IR), Assisted Replication (AR) and BIER.

This document also assumes familiarity with the terminology of [RFC7432], [RFC4364], [RFC6513], [RFC6514], [IGMP-PROXY], [OISM], [EVPN-RT5] and [EVPN-BUM].

1.2 Background on IP Multicast Delivery in EVPN Networks

IP Multicast is all about forwarding a single copy of a packet from a source S to a group of receivers G along a multicast tree. That multicast tree can be created in an EVPN tenant domain where S and the receivers for G are connected to the same BD or different BD. In the former case, we refer to Intra-subnet IP Multicast forwarding, whereas the latter case will be referred to as Inter-subnet IP Multicast forwarding.

1.2.1 Intra-subnet IP Multicast Forwarding

When the source S1 and receivers interested in G1 are attached to the same BD, the EVPN network can deliver the IP Multicast traffic to the receivers in two different ways (Figure 1):

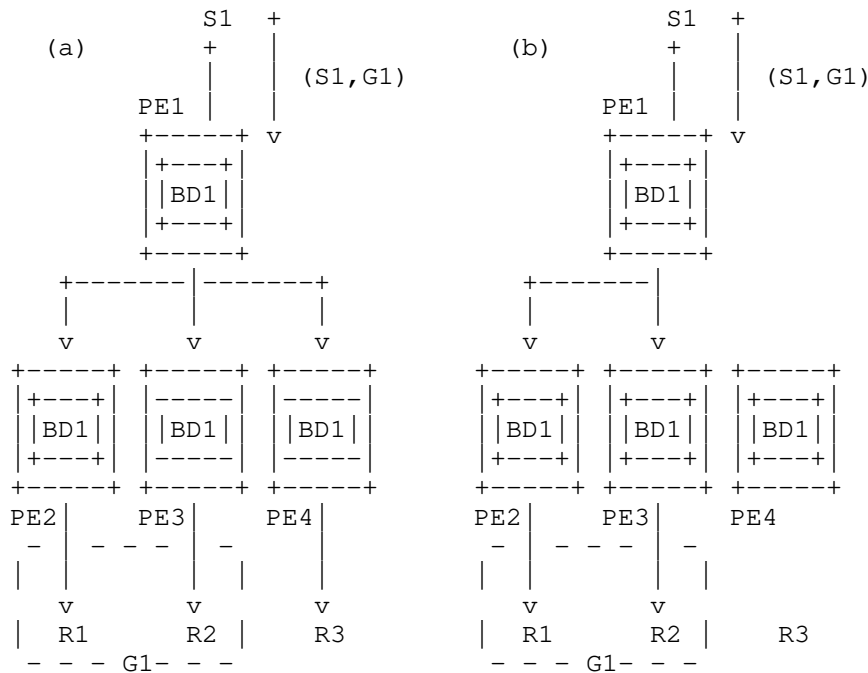


Figure 1 - Intra-subnet IP Multicast

Model (a) illustrated in Figure 1 is referred to as IP Multicast delivery as BUM traffic. This way of delivering IP Multicast traffic does not require any extensions to [RFC7432], however, it sends the IP Multicast flows to non-interested receivers, such as e.g., R3 in Figure 1. In this example, downstream PEs can snoop IGMP/MLD messages from the receivers so that layer-2 multicast state is created and, for instance, PE4 can avoid sending (S1,G1) to R3, since R3 is not interested in (S1,G1).

Model (b) in Figure 1 uses an S-PMSI to optimize the delivery of the (S1,G1) flow. For instance, assuming PE1 uses IR, PE1 sends (S1,G1) only to the downstream PEs that issued an SMET route for (S1,G1), that is, PE2 and PE3. In case PE1 uses any P-tunnel different than IR, AR or BIER, PE1 will advertise an S-PMSI A-D route for (S1,G1) and PE2/PE2 will join that tree.

Procedures for Model (b) are specified in [IGMP-PROXY].

1.2.2 Inter-subnet IP Multicast Forwarding

If the source and receivers are attached to different BDs of the same

tenant domain, the EVPN network can also use Inclusive or Selective Trees as depicted in Figure 2, models (a) and (b) respectively.

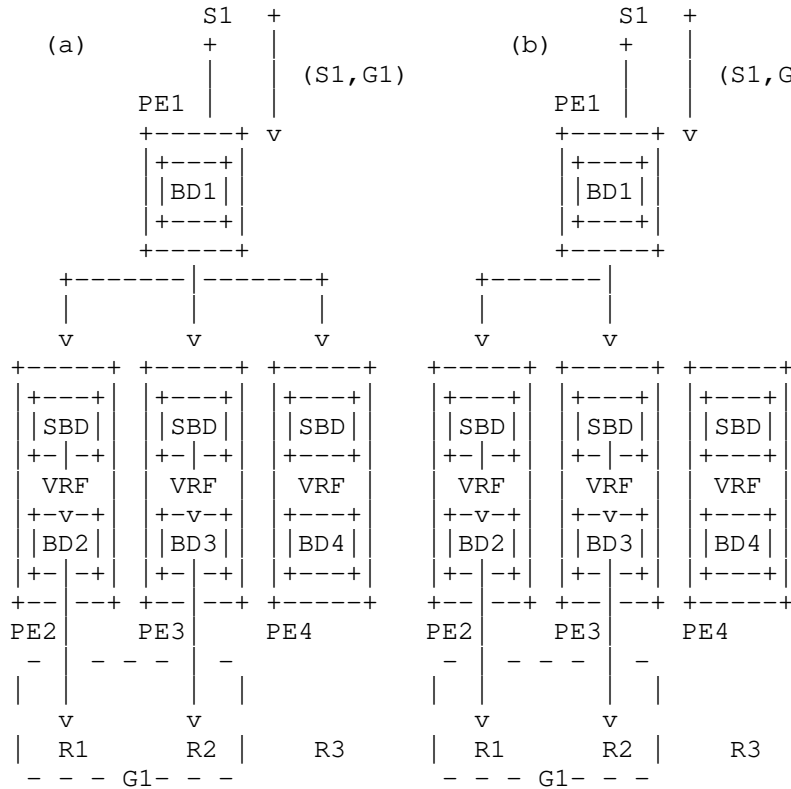


Figure 2 - Inter-subnet IP Multicast

[OISM] specifies the procedures to optimize the Inter-subnet Multicast forwarding in an EVPN network. The IP Multicast flows are always sent in the context of the source BD. As described in [OISM], if the downstream PE is not attached to the source BD, the IP Multicast flow is received on the SBD (Supplementary Broadcast Domain), as in the example in Figure 2.

[OISM] supports Inclusive or Selective Multicast Trees, and as explained in section 1.3.1 "Intra-subnet IP Multicast Forwarding", the Selective Multicast Trees are setup in a different way, depending on the P-tunnel being used by the source BD. As an example, model (a) in Figure 2 illustrates the use of an Inclusive Multicast Tree for

BD1 on PE1. Since the downstream PEs are not attached to BD1, they will all receive (S1,G1) in the context of the SBD and will locally route the flow to the local ACs. Model (b) uses a similar forwarding model, however PE1 sends the (S1,G1) flow in a Selective Multicast Tree. If the P-tunnel is IR, AR or BIER, PE1 does not need to advertise an S-PMSI A-D route.

[OISM] is a superset of the procedures in [IGMP-PROXY], in which sources and receivers can be in the same or different BD of the same tenant. [OISM] ensures every upstream PE attached to a source will learn of all other PEs (attached to the same Tenant Domain) that have interest in a particular set of flows. This is because the downstream PEs advertise SMET routes for a set of flows with the SBD's Route Target and they are imported by all the Upstream PEs of the tenant. As a result of that, inter-subnet multicasting can be done within the Tenant Domain, without requiring any Rendezvous Points (RP), shared trees, UMH selection or any other complex aspects of conventional multicast routing techniques.

1.3 Multi-Homed IP Multicast Sources in EVPN

Contrary to conventional multicast routing technologies, multi-homing PEs attached to the same source can never create IP Multicast packet duplication if the PEs use a multi-homed Ethernet Segment (ES). Figure 3 illustrates this by showing two multi-homing PEs (PE1 and PE2) that are attached to the same source (S1). We assume that S1 is connected to an all-active ES by a layer-2 switch (SW1) with a LAG to PE1 and PE2.

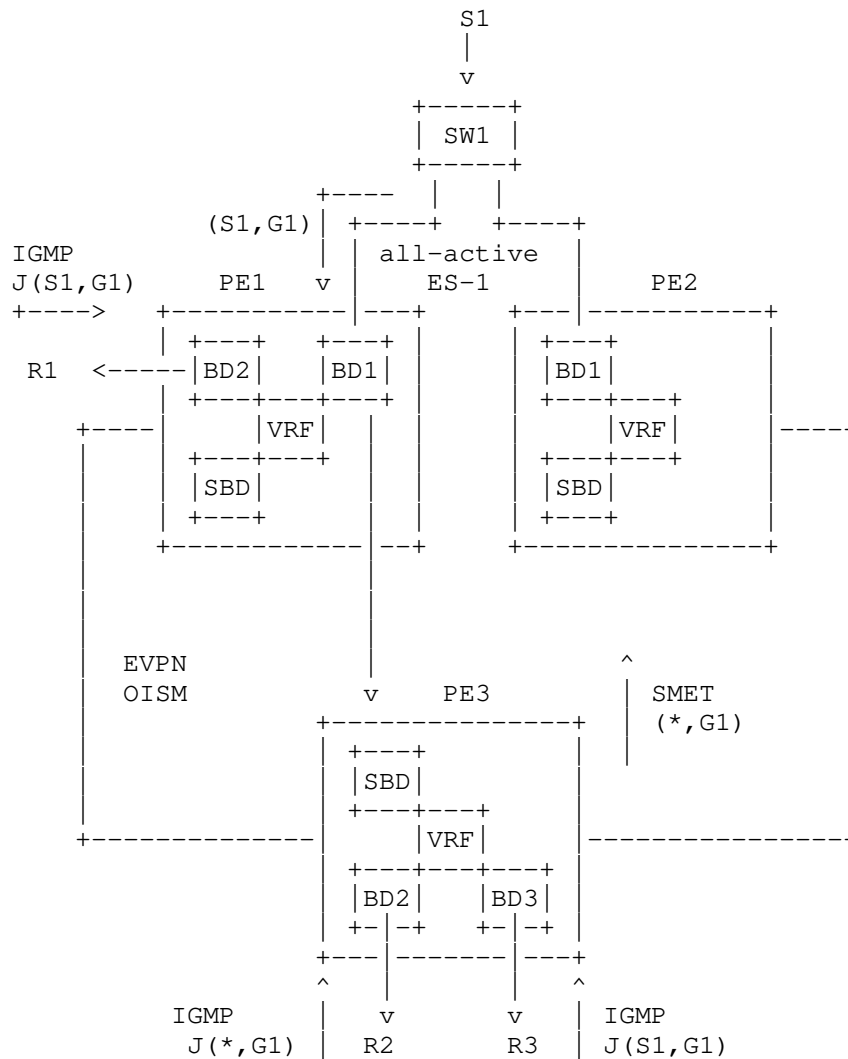


Figure 3 - All-active Multi-homing and OISM

When receiving the (S1,G1) flow from S1, SW1 will choose only one link to send the flow, as per [RFC7432]. Assuming PE1 is the receiving PE on BD1, the IP Multicast flow will be forwarded as soon as BD1 creates multicast state for (S1,G1) or (*,G1). In the example of Figure 3, receivers R1, R2 and R3 are interested in the multicast flow to G1. R1 will receive (S1,G1) directly via the IRB interface as per [OISM]. Upon receiving IGMP reports from R2 and R3, PE3 will issue an SMET (*,G1) route that will create state in PE1's BD1. PE1

will therefore forward the IP Multicast flow to PE3's SBD and PE3 will forward to R2 and R3, as per [OISM] procedures.

When IP Multicast source multi-homing is required, EVPN multi-homed Ethernet Segments MUST be used. EVPN multi-homing guarantees that only one Upstream PE will forward a given multicast flow at the time, avoiding packet duplication at the Downstream PEs. In addition, the SMET route for a given flow creates state in all the multi-homing Upstream PEs. Therefore, in case of failure on the Upstream PE forwarding the flow, the backup Upstream PE can forward the flow immediately.

This document assumes that multi-homing PEs attached to the same source always use multi-homed Ethernet Segments.

1.4 The Need for Redundant IP Multicast Sources in EVPN

While multi-homing PEs to the same IP Multicast G-source provides certain level of resiliency, multicast applications are often critical in the Operator's network and greater level of redundancy is required. This document assumes that:

- a) Redundant G-sources for an SFG may exist in the EVPN tenant network. A Redundant G-source is a host or a router that sends an SFG in a tenant network where there is another host or router sending traffic to the same SFG.
- b) Those redundant G-sources may be in the same BD or different BDs of the tenant. There must not be restrictions imposed on the location of the receiver systems either.
- c) The redundant G-sources can be single-homed to only one EVPN PE or multi-homed to multiple EVPN PEs.
- d) The EVPN PEs must avoid duplication of the same SFG on the receiver systems.

2. Solution Overview

An SFG is represented as (*,G) if any source that issues multicast traffic to G is a redundant G-source. Alternatively, this document allows an SFG to be represented as (S,G), where S is a prefix of any length. In this case, a source is considered a redundant G-source for the SFG if it is contained in the prefix. This document allows variable length prefixes in the Sources advertised in S-PMSI A-D

routes only for the particular application of redundant G-sources.

There are two redundant G-source solutions described in this document:

- o Warm Standby (WS) Solution
- o Hot Standby (HS) Solution

The WS solution is an upstream PE based solution (downstream PEs do not participate in the procedures), in which all the upstream PEs attached to redundant G-sources for an SFG represented by (*,G) or (S,G) will elect a "Single Forwarder" (SF) among themselves. Once a SF is elected, the upstream PEs add an RPF check to the (*,G) or (S,G) state for the SFG:

- A non-SF upstream PE discards any (*,G)/(S,G) packets received over a local AC.
- The SF accepts and forwards any (*,G)/(S,G) packets it receives over a single local AC (for the SFG). In case (*,G)/(S,G) packets for the SFG are received over multiple local ACs, they will be discarded in all the local ACs but one. The procedure to choose the local AC that accepts packets is a local implementation matter.

A failure on the SF will result in the election of a new SF. The Election requires BGP extensions on the existing EVPN routes. These extensions and associated procedures are described in Sections 3 and 4 respectively.

In the HS solution the downstream PEs are the ones avoiding the SFG duplication. The upstream PEs are aware of the locally attached G-sources and add a unique ESI-label per SFG to the SFG packets forwarded to downstream PEs. The downstream PEs pull the SFG from all the upstream PEs attached to the redundant G-sources and avoid duplication on the receiver systems by adding an RPF check to the (*,G) state for the SFG:

- A downstream PE discards any (*,G) packets it receives from the "wrong G-source".
- The wrong G-source is identified in the data path by an ESI-label that is different than the ESI-label used for the selected G-source.
- Note that the ESI-label is used here for "ingress filtering" (at the egress/downstream PE) as opposed to the [RFC7432] "egress filtering" (at the egress/downstream PE) used in the split-horizon procedures. In [RFC7432] the ESI-label indicates what egress ACs

must be skipped when forwarding BUM traffic to the egress. In this document, the ESI-label indicates what ingress traffic must be discarded at the downstream PE.

The use of ESI-labels for SFGs forwarded by upstream PEs require some control plane and data plane extensions in the procedures used by [RFC7432] for multi-homing. Upon failure of the selected G-source, the downstream PE will switch over to a different selected G-source, and will therefore change the RPF check for the (*,G) state. The extensions and associated procedures are described in Sections 3 and 5 respectively.

An operator should use the HS solution if they require a fast fail-over time and the additional bandwidth consumption is acceptable (SFG packets are received multiple times on the downstream PEs). Otherwise the operator should use the WS solution, at the expense of a slower fail-over time in case of a G-source or upstream PE failure. Besides bandwidth efficiency, another advantage of the WS solution is that only the upstream PEs attached to the redundant G-sources for the same SFG need to be upgraded to support the new procedures.

This document does not impose the support of both solutions on a system. If one solution is supported, the support of the other solution is OPTIONAL.

3. BGP EVPN Extensions

This document makes use of the following BGP EVPN extensions:

1. SFG flag in the Multicast Flags Extended Community

The Single Flow Group (SFG) flag is a new bit requested to IANA out of the registry Multicast Flags Extended Community Flag Values. This new flag is set for S-PMSI A-D routes that carry a (*,G)/(S,G) SFG in the NLRI.

2. ESI Label Extended Community is used in S-PMSI A-D routes

The HS solution requires the advertisement of one or more ESI Label Extended Communities [RFC7432] that encode the Ethernet Segment Identifier(s) associated to an S-PMSI A-D (*,G)/(S,G) route that advertises the presence of an SFG. Only the ESI Label value in the extended community is relevant to the procedures in this document. The Flags field in the extended community will be advertised as 0x00 and ignored on reception. [RFC7432] specifies that the ESI Label Extended Community is advertised along with the A-D per ES route. This documents extends the use of this extended

community so that it can be advertised multiple times (with different ESI values) along with the S-PMSI A-D route.

4. Warm Standby (WS) Solution for Redundant G-Sources

The general procedure is described as follows:

1. Configuration of the upstream PEs

Upstream PEs (possibly attached to redundant G-sources) need to be configured to know which groups are carrying only flows from redundant G-sources, that is, the SFGs in the tenant domain. They will also be configured to know which local BDs may be attached to a redundant G-source. The SFGs can be configured for any source, E.g., SFG for "*", or for a prefix that contains multiple sources that will issue the same SFG, i.e., "10.0.0.0/30". In the latter case sources 10.0.0.1 and 10.0.0.2 are considered as Redundant G-sources, whereas 10.0.0.10 is not considered a redundant G-source for the same SFG.

As an example:

- PE1 is configured to know that G1 is an SFG for any source and redundant G-sources for G1 may be attached to BD1 or BD2.
- Or PE1 can also be configured to know that G1 is an SFG for the sources contained in 10.0.0.0/30, and those redundant G-sources may be attached to BD1 or BD2.

2. Signaling the location of a G-source for a given SFG

Upon receiving G-traffic for a configured SFG on a BD, an upstream PE configured to follow this procedure, e.g., PE1:

- a. Originates an S-PMSI A-D (*,G)/(S,G) route for the SFG. An (*,G) route is advertised if the SFG is configured for any source, and an (S,G) route is advertised (where the Source can have any length) if the SFG is configured for a prefix.
- b. The S-PMSI A-D route is imported by all the PEs attached to the tenant domain. In order to do that, the route will use the SBD-RT (Supplementary Broadcast Domain Route-Target) in addition to the BD-RT of the BD over which the G-traffic is received. The route SHOULD also carry a DF Election Extended Community (EC) and a flag indicating that it conveys an SFG. The DF Election EC and its use is specified in [RFC8584].
- c. The above S-PMSI A-D route MAY be advertised with or without PMSI Tunnel Attribute (PTA):

- With no PTA if an I-PMSI or S-PMSI A-D with IR/AR/BIER are to be used.
 - With PTA in any other case.
- d. The S-PMSI A-D route is triggered by the first packet of the SFG and withdrawn when the flow is not received anymore. Detecting when the G-source is no longer active is a local implementation matter. The use of a timer is RECOMMENDED. The timer is started when the traffic to G1 is not received. Upon expiration of the timer, the PE will withdraw the route.

3. Single Forwarder (SF) Election

If the PE with a local G-source receives one or more S-PMSI A-D routes for the same SFG from a remote PE, it will run a Single Forwarder (SF) Election based on the information encoded in the DF Election EC. Two S-PMSI A-D routes are considered for the same SFG if they are advertised for the same tenant, and their Multicast Source Length, Multicast Source, Multicast Group Length and Multicast Group fields match.

- a. A given DF Alg can only be used if all the PEs running the DF Alg have consistent input. For example, in an OISM network, if the redundant G-sources for an SFG are attached to BDs with different Ethernet Tags, the Default DF Election Alg MUST NOT be used.
- b. In case there is a mismatch in the DF Election Alg or capabilities advertised by two PEs competing for the SF, the lowest PE IP address (given by the Originator Address in the S-PMSI A-D route) will be used as a tie-breaker.

4. RPF check on the PEs attached to a redundant G-source

All the PEs with a local G-source for the SFG will add an RPF check to the (*,G)/(S,G) state for the SFG. That RPF check depends on the SF Election result:

- a. The non-SF PEs discard any (*,G)/(S,G) packets for the SFG received over a local AC.
- b. The SF accepts any (*,G)/(S,G) packets for the SFG it receives over one (and only one) local AC.

The solution above provides redundancy for SFGs and it does not require an upgrade of the downstream PEs (PEs where there is

certainty that no redundant G-sources are connected). Other G-sources for non-SFGs may exist in the same tenant domain. This document does not change the existing procedures for non-SFG G-sources.

The redundant G-sources can be single-homed or multi-homed to a BD in the tenant domain. Multi-homing does not change the above procedures.

Sections 4.1 and 4.2 show two examples of the WS solution.

4.1 WS Example in an OISM Network

Figure 4 illustrates an example in which S1 and S2 are redundant G-sources for the SFG (*,G1).

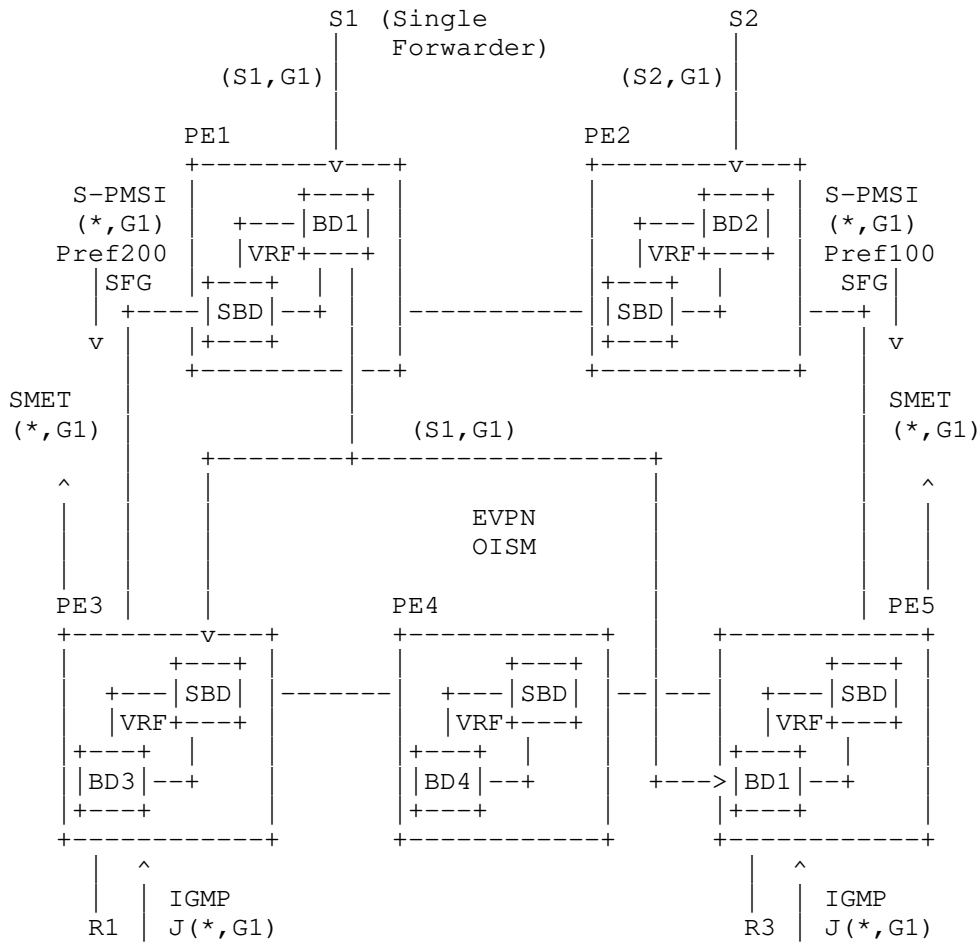


Figure 4 - WS Solution for Redundant G-Sources

The WS solution works as follows:

1. Configuration of the upstream PEs, PE1 and PE2

PE1 and PE2 are configured to know that G1 is an SFG for any source and redundant G-sources for G1 may be attached to BD1 or BD2, respectively.

2. Signaling the location of S1 and S2 for (*, G1)

Upon receiving (S1, G1) traffic on a local AC, PE1 and PE2 originate S-PMSI A-D (*, G1) routes with the SBD-RT, DF Election

Extended Community (EC) and a flag indicating that it conveys an SFG.

3. Single Forwarder (SF) Election

Based on the DF Election EC content, PE1 and PE2 elect an SF for (*,G1). Assuming both PEs agree on e.g., Preference based Election as the algorithm to use [DF-PREF], and PE1 has a higher preference, PE1 becomes the SF for (*,G1).

4. RPF check on the PEs attached to a redundant G-source

- a. The non-SF, PE2, discards any (*,G1) packets received over a local AC.
- b. The SF, PE1 accepts (*,G1) packets it receives over a one (and only one) local AC.

The end result is that, upon receiving reports for (*,G1) or (S,G1), the downstream PEs (PE3 and PE5) will issue SMET routes and will pull the multicast SFG from PE1, and PE1 only. A failure on S1, the AC connected to S1 or PE1 itself will trigger the S-PMSI A-D (*,G1) withdrawal from PE1 and PE2 will be promoted to SF.

4.2 WS Example in a Single-BD Tenant Network

Figure 5 illustrates an example in which S1 and S2 are redundant G-sources for the SFG (*,G1), however, now all the G-sources and receivers are connected to the same BD1 and there is no SBD.

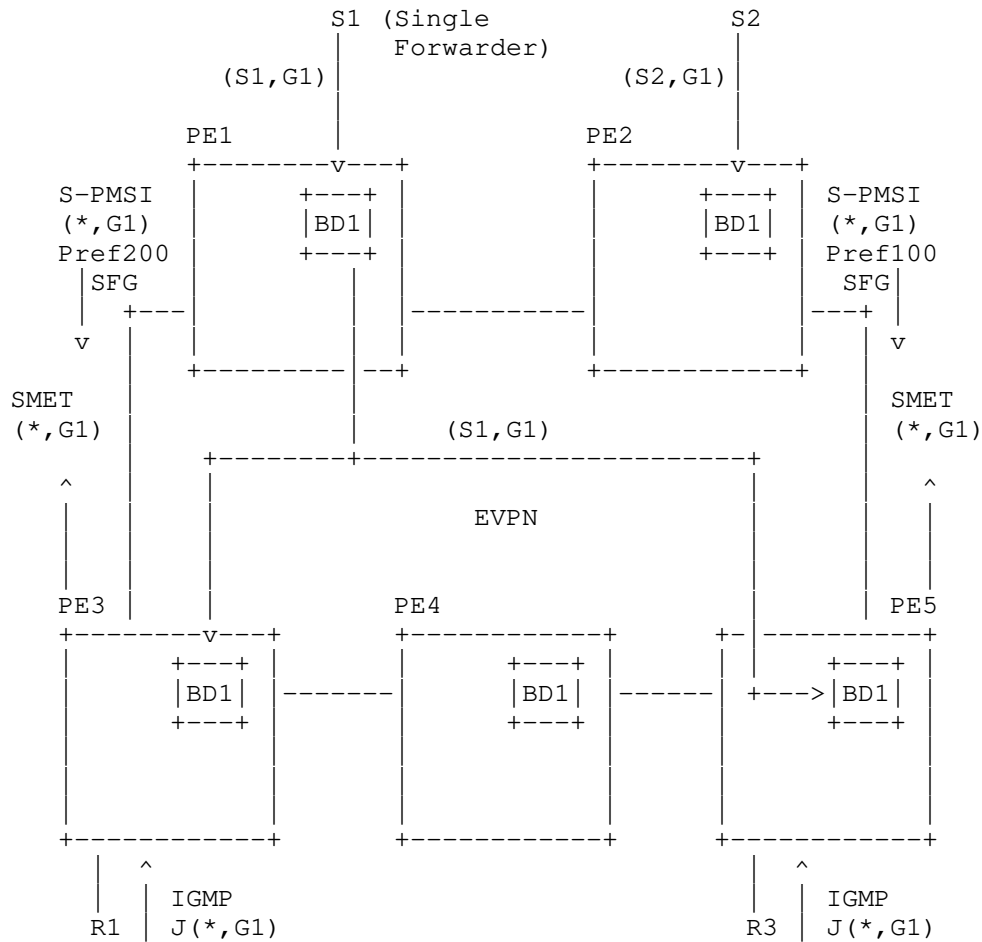


Figure 5 - WS Solution for Redundant G-Sources in the same BD

The same procedure as in Section 4.1 is valid here, being this a sub-case of the one in Section 4.1. Upon receiving traffic for the SFG G1, PE1 and PE2 advertise the S-PMSI A-D routes with BD1-RT only, since there is no SBD.

5. Hot Standby (HS) Solution for Redundant G-Sources

If fast-failover time is desired upon the failure of a G-source or PE attached to the G-source, and in spite of the extra bandwidth consumption in the tenant network, the HS solution should be used. The procedure is as follows:

1. Configuration of the PEs

As in the WS case, the upstream PEs where redundant G-sources may exist need to be configured to know which groups (for any source or a prefix containing the intended sources) are carrying only flows from redundant G-sources, that is, the SFGs in the tenant domain.

In addition (and this is not done in WS mode), the individual redundant G-sources for an SFG need to be associated with an Ethernet Segment (ES) on the upstream PEs. This is irrespective of the redundant G-source being multi-homed or single-homed. Even for single-homed redundant G-sources the HS procedure relies on the ESI labels for the RPF check on downstream PEs. The term "S-ESI" is used in this document to refer to an ESI associated to a redundant G-source.

Contrary to the WS method (that is transparent to the downstream PEs), the support for the HS procedure in all downstream PEs connected to the receivers in the tenant network is REQUIRED. The downstream PEs do not need to be configured to know the connected SFGs or their ESIs, since they get that information from the upstream PEs. The downstream PEs will locally select an ESI for a given SFG, and will program an RPF check to the (*,G)/(S,G) state for the SFG that will discard (*,G)/(S,G) packets from the rest of the ESIs. The selection of the ESI for the SFG is based on local policy.

2. Signaling the location of a G-source for a given SFG and its association to the local ESIs

Based on the configuration in step 1, an upstream PE configured to follow the HS procedures:

- a. Advertises an S-PMSI A-D (*,G)/(S,G) route per each configured SFG. These routes need to be imported by all the PEs of the tenant domain, therefore they will carry the BD-RT and SBD-RT (if the SBD exists). The route also carries the ESI Label Extended Communities needed to convey all the S-ESIs associated to the SFG in the PE.
- b. The S-PMSI A-D route will convey a PTA in the same cases as in the WS procedure.
- c. The S-PMSI A-D (*,G)/(S,G) route is triggered by the configuration of the SFG and not by the reception of G-traffic.

3. Distribution of DCB (Domain-wide Common Block) ESI-labels and G-

source ES routes

An upstream PE advertises the corresponding ES, A-D per EVI and A-D per ES routes for the local S-ESIs.

- a. ES routes are used for regular DF Election for the S-ES. This document does not introduce any change in the procedures related to the ES routes.
 - b. The A-D per EVI and A-D per ES routes MUST include the SBD-RT since they have to be imported by all the PEs in the tenant domain.
 - c. The A-D per ES routes convey the S-ESI labels that the downstream PEs use to add the RPF check for the (*,G)/(S,G) associated to the SFGs. This RPF check requires that all the packets for a given G-source are received with the same S-ESI label value on the downstream PEs. For example, if two redundant G-sources are multi-homed to PE1 and PE2 via S-ES-1 and S-ES-2, PE1 and PE2 MUST allocate the same ESI label "Lx" for S-ES-1 and they MUST allocate the same ESI label "Ly" for S-ES-2. In addition, Lx and Ly MUST be different. These ESI labels are Domain-wide Common Block (DCB) labels and follow the allocation procedures in [DCB].
4. Processing of A-D per ES/EVI routes and RPF check on the downstream PEs

The A-D per ES/EVI routes are received and imported in all the PEs in the tenant domain. The processing of the A-D per ES/EVI routes on a given PE depends on its configuration:

- a. The PEs attached to the same BD of the BD-RT that is included in the A-D per ES/EVI routes will process the routes as in [RFC7432] and [RFC8584]. If the receiving PE is attached to the same ES as indicated in the route, [RFC7432] split-horizon procedures will be followed and the DF Election candidate list may be modified as in [RFC8584] if the ES supports the AC-DF capability.
- b. The PEs that are not attached to the BD-RT but are attached to the SBD of the received SBD-RT, will import the A-D per ES/EVI routes and use them for redundant G-source mass withdrawal, as explained later.
- c. Upon importing A-D per ES routes corresponding to different S-ESes, a PE MUST select a primary S-ES and add an RPF check to the (*,G)/(S,G) state in the BD or SBD. This RPF check will

discard all ingress packets to $(*,G)/(S,G)$ that are not received with the ESI-label of the primary S-ES. The selection of the primary S-ES is a matter of local policy.

5. G-traffic forwarding for redundant G-sources and fault detection

Assuming there is $(*,G)$ or (S,G) state for the SFG with OIF list entries associated to remote EVPN PEs, upon receiving G-traffic on a S-ES, the upstream PE will add a S-ESI label at the bottom of the stack before forwarding the traffic to the remote EVPN PEs. This label is allocated from a DCB as described in step 3. If P2MP or BIER PMSIs are used, this is not adding any new data path procedures on the upstream PEs (except that the ESI-label is allocated from a DCB). However, if IR/AR are used, this document extends the [RFC7432] procedures by pushing the S-ESI labels not only on packets sent to the PEs that shared the ES but also to the rest of the PEs in the tenant domain. This allows the downstream PEs to receive all the multicast packets from the redundant G-sources with a S-ESI label (irrespective of the PMSI type and the local ESes), and discard any packet that conveys a S-ESI label different from the primary S-ESI label (that is, the label associated to the selected primary S-ES), as discussed in step 4.

If the last A-D per EVI or the last A-D per ES route for the primary S-ES is withdrawn, the downstream PE will immediately select a new primary S-ES and will change the RPF check. Note that if the S-ES is re-used for multiple tenant domains by the upstream PEs, the withdrawal of all the A-D per-ES routes for a S-ES provides a mass withdrawal capability that makes a downstream PE to change the RPF check in all the tenant domains using the same S-ES.

The withdrawal of the last S-PMSI A-D route for a given $(*,G)/(S,G)$ that represents a SFG SHOULD make the downstream PE remove the S-ESI label based RPF check on $(*,G)/(S,G)$.

5.1 Use of BFD in the HS Solution

The BGP-BFD Attribute (advertised along with the S-PMSI A-D routes) and similar procedures as the ones described in [MVPN-FAST-FAILOVER] MAY be used to bootstrap multipoint BFD sessions on the downstream PEs.

5.2 HS Example in an OISM Network

Figure 6 illustrates the HS model in an OISM network. Consider S1 and S2 are redundant G-sources for the SFG $(*,G1)$ in BD1 (any source

using G1 is assumed to transmit an SFG). S1 and S2 are (all-active) multi-homed to upstream PEs, PE1 and PE2. The receivers are attached to downstream PEs, PE3 and PE5, in BD3 and BD1, respectively. S1 and S2 are assumed to be connected by a LAG to the multi-homing PEs, and the multicast traffic can use the link to either upstream PE. The diagram illustrates how S1 sends the G-traffic to PE1 and PE1 forwards to the remote interested downstream PEs, whereas S2 sends to PE2 and PE2 forwards further. In this HS model, the interested downstream PEs will get duplicate G-traffic from the two G-sources for the same SFG. While the diagram shows that the two flows are forwarded by different upstream PEs, the all-active multi-homing procedures may cause that the two flows come from the same upstream PE. Therefore, finding out the upstream PE for the flow is not enough for the downstream PEs to program the required RPF check to avoid duplicate packets on the receiver.

ESI-label-1 and ESI-2 to use ESI-label-2.

The downstream PEs, PE3, PE4 and PE5 are configured to support HS mode and select the G-source with e.g., lowest ESI value.

2. PE1 and PE2 advertise S-PMSI A-D (*,G1) and ES/A-D per ES/EVI routes

Based on the configuration of step 1, PE1 and PE2 advertise an S-PMSI A-D (*,G1) route each. The route from each of the two PEs will include TWO ESI Label Extended Communities with ESI-1 and ESI-2 respectively, as well as BD1-RT plus SBD-RT and a flag that indicates that (*,G1) is an SFG.

In addition, PE1 and PE2 advertise ES and A-D per ES/EVI routes for ESI-1 and ESI-2. The A-D per ES and per EVI routes will include the SBD-RT so that they can be imported by the downstream PEs that are not attached to BD1, e.g., PE3 and PE4. The A-D per ES routes will convey ESI-label-1 for ESI-1 (on both PEs) and ESI-label-2 for ESI-2 (also on both PEs).

3. Processing of A-D per ES/EVI routes and RPF check

PE1 and PE2 received each other's ES and A-D per ES/EVI routes. Regular [RFC7432] [RFC8584] procedures will be followed for DF Election and programming of the ESI-labels for egress split-horizon filtering. PE3/PE4 import the A-D per ES/EVI routes in the SBD. Since PE3 has created a (*,G1) state based on local interest, PE3 will add an RPF check to (*,G1) so that packets coming with ESI-label-2 are discarded (lowest ESI value is assumed to give the primary S-ES).

4. G-traffic forwarding and fault detection

PE1 receives G-traffic (S1,G1) on ES-1 that is forwarded within the context of BD1. Irrespective of the tunnel type, PE1 pushes ESI-label-1 at the bottom of the stack and the traffic gets to PE3 and PE5 with the mentioned ESI-label (PE4 has no local interested receivers). The G-traffic with ESI-label-1 passes the RPF check and it is forwarded to R1. In the same way, PE2 sends (S2,G1) with ESI-label-2, but this G-traffic does not pass the RPF check and gets discarded at PE3/PE5.

If the link from S1 to PE1 fails, S1 will forward the (S1,G1) traffic to PE2 instead. PE1 withdraws the ES and A-D routes for ESI-1. Now both flows will be originated by PE2, however the RPF checks don't change in PE3/PE5.

If subsequently, the link from S1 to PE2 fails, PE2 also withdraws the ES and A-D routes for ESI-1. Since PE3 and PE5 have no longer A-D per ES/EVI routes for ESI-1, they immediately change the RPF check so that packets with ESI-label-2 are now accepted.

Figure 7 illustrates a scenario where S1 and S2 are single-homed to PE1 and PE2 respectively. This scenario is a sub-case of the one in Figure 6. Now ES-1 only exists in PE1, hence only PE1 advertises the A-D per ES/EVI routes for ESI-1. Similarly, ES-2 only exists in PE2 and PE2 is the only PE advertising A-D routes for ESI-2. The same procedures as in Figure 6 applies to this use-case.

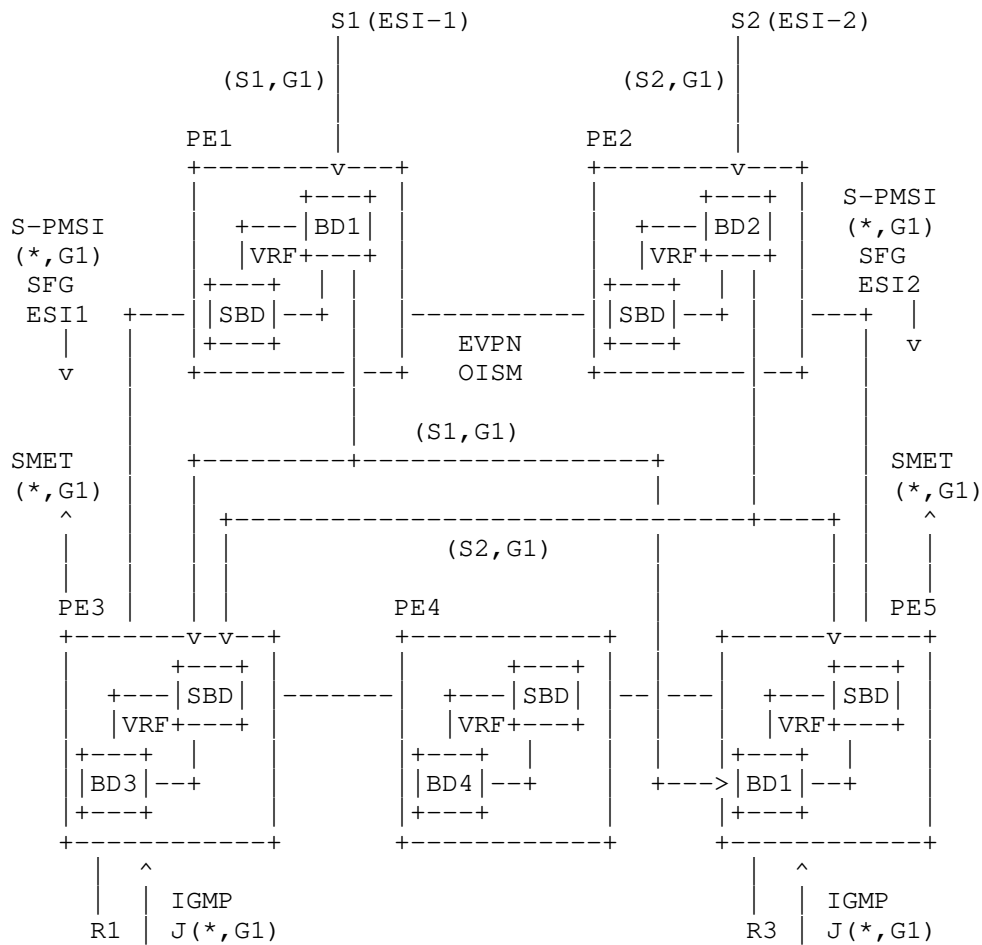


Figure 7 - HS Solution for single-homed Redundant G-Sources in OISM

5.3 HS Example in a Single-BD Tenant Network

Irrespective of the redundant G-sources being multi-homed or single-homed, if the tenant network has only one BD, e.g., BD1, the procedures of Section 5.2 still apply, only that routes do not include any SBD-RT and all the procedures apply to BD1 only.

6. Security Considerations

The same Security Considerations described in [OISM] are valid for this document.

7. IANA Considerations

IANA is requested to allocate a Bit in the Multicast Flags Extended Community to indicate that a given (*,G) or (S,G) in an S-PMSI A-D route is associated with an SFG.

8. References

8.1. Normative References

[RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.

[RFC6513] Rosen, E., Ed., and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <<https://www.rfc-editor.org/info/rfc6513>>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <<https://www.rfc-editor.org/info/rfc6514>>.

[IGMP-PROXY] Sajassi, A. et al, "IGMP and MLD Proxy for EVPN", June 2019, work-in-progress, draft-ietf-bess-evpn-igmp-ml-d-proxy-03.

[OISM] Rosen, E. et al, "EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding", January 2019, work-in-progress, draft-ietf-bess-evpn-irb-mcast-02.

[RFC8584] Rabadan, J., Mohanty, S., Sajassi, A., Drake, J., Nagaraj,

K., and S. Sathappan, "Framework for Ethernet VPN Designated Forwarder Election Extensibility", RFC 8584, DOI 10.17487/RFC8584, April 2019, <<https://rfc-editor.org/rfc/rfc8584.txt>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

[DCB] Zhang, Z. et al, "MVPN/EVPN Tunnel Aggregation with Common Labels", April 2018, work-in-progress, draft-zzhang-bess-mvpn-evpn-aggregation-label-01.

[MVPN-FAST-FAILOVER] Morin, T., Kebler, R., and G. Mirsky, "Multicast VPN fast upstream failover", draft-ietf-bess-mvpn-fast-failover-06 (work in progress), July 2019.

8.2. Informative References

[EVPN-RT5] Rabadan, J., Henderickx, W., Drake, J., Lin, W., and A. Sajassi, "IP Prefix Advertisement in EVPN", internet-draft ietf-bess-evpn-prefix-advertisement-11.txt, May 2018.

[EVPN-BUM] Zhang, Z., Lin, W., Rabadan, J., and K. Patel, "Updates on EVPN BUM Procedures", internet-draft ietf-bess-evpn-bum-procedure-updates-06, June 2019.

[DF-PREF] Rabadan, J., Sathappan, S., Przygienda, T., Lin, W., Drake, J., Sajassi, A., and S. Mohanty, "Preference-based EVPN DF Election", internet-draft ietf-bess-evpn-pref-df-04.txt, June 2019.

[RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.

9. Acknowledgments

The authors would like to thank Mankamana Mishra and Ali Sajassi for their review and valuable comments.

10. Contributors

Authors' Addresses

Jorge Rabadan
Nokia
777 E. Middlefield Road
Mountain View, CA 94043 USA
Email: jorge.rabadan@nokia.com

Senthil Sathappan
Nokia
701 E. Middlefield Road
Mountain View, CA 94043 USA
Email: senthil.sathappan@nokia.com

Jayant Kotalwar
Nokia
701 E. Middlefield Road
Mountain View, CA 94043 USA
Email: jayant.kotalwar@nokia.com

Eric C. Rosen
EMail: erosen52@gmail.com

Zhaohui Zhang
Juniper Networks
EMail: zzhang@juniper.net

Wen Lin
Juniper Networks, Inc.
EMail: wlin@juniper.net

IDR
Internet-Draft
Intended status: Standards Track
Expires: January 6, 2020

S. Sangli
R. Bonica
Juniper Networks Inc.
July 5, 2019

BGP based Virtual Private Network (VPN) Services over SRv6+ enabled IPv6
networks
draft-ssangli-idr-bgp-vpn-srv6-plus-01

Abstract

This document defines BGP protocol extensions for encoding and carrying SRv6+ Per-Path Service Instruction information to support Virtual Private Network services. This is applicable when the VPN services are offered in a SRv6+ enabled IPv6 network such that the VPN payload is transported over IPv6. The Per-Path Service Instruction information is encoded in the IPv6 Destination Option Header in the IPv6 data packets.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 6, 2020.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Requirements Language	3
3. Per-Path Service Instruction Information	3
4. Usage of Tunnel Encapsulation Attribute	4
5. Procedures for Egress BGP Speaker	6
6. Procedures for Ingress BGP Speaker	6
7. BGP based L3 VPN services over IPv6	7
7.1. IPv4 VPN on SRv6+ enabled IPv6 Core	7
7.2. IPv6 VPN on SRv6+ enabled IPv6 Core	8
7.3. IPv4 Global Routes on SRv6+ enabled IPv6 Core	8
8. BGP based Ethernet VPN services over IPv6	9
8.1. Ethernet Per ES Auto-Discovery (A-D) route	9
8.2. Ethernet per EVI Auto-Discovery (A-D) route	10
8.3. MAC/IP Advertisement route	10
8.4. Inclusive Multicast Ethernet Route	11
8.5. IP Prefix Route	11
9. Deployment Considerations	11
10. Backward Compatibility	13
11. Security Considerations	13
12. IANA Considerations	13
13. Acknowledgements	13
14. References	13
14.1. Normative References	13
14.2. Informative References	14
Authors' Addresses	16

1. Introduction

Virtual Private Network (VPN) technologies allow network providers to emulate private networks with shared infrastructure. For example, assume that a set of red sites, set of blue sites and a set of green sites connect to a provider network. Furthermore, assume that red sites and blue sites wish to interconnect, exchange packets. However, the green sites wish to communicate with green sites only. The provider should allow its infrastructure network to scale to both the requirements without having to create multiple parallel network infrastructures. The IETF has standardized many VPN technologies viz. Layer 3 VPN (L3VPN) [RFC4364], Layer 2 VPN (L2VPN) [RFC6624], Virtual Private LAN Service (VPLS) [RFC4761], [RFC4762], Ethernet VPN (EVPN) [RFC7432], Pseudowires [RFC8077] to enable Layer 3 and Layer 2 VPN services.

The aforementioned technologies leverage MPLS network architecture :

- o to establish a MPLS tunnel from ingress PE to egress PE, thus making all P routers agnostic of VPN state.
- o to provide demultiplexing abstraction in the tunnelled packet so the payload packet can be forwarded at the egress router based on Routing table and/or interface.

In pure IPv6 deployments where there may be non-MPLS capable routers, it would be desirable to have alternate mechanism to provide VPN connectivity. This document describes BGP extensions and procedures applicable for SRv6+ enabled IPv6 networks, to provide VPN services over BGP.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Per-Path Service Instruction Information

A SRv6+ [I-D.bonica-spring-srv6-plus] segment provides unidirectional connectivity from an ingress node to an egress node. A SRv6+ path contains one or more such segments. SRv6+ introduces the concept of Per-Segment Service Instruction and Per-Path Service Instruction. These instructions describe the additional packet processing performed on a node. The Per-Segment Service Instruction is executed on the segment egress node while the Per-Path Service Instruction is executed on the path egress node. The SR Path egress node advertises the reachability information to SR Path ingress node via Multi Protocol extensions in BGP [RFC4760].

For providing VPN services, aforementioned BGP extensions rely on MPLS architecture [RFC3031]. The BGP extensions specify the new encoding for Network Layer Reachability Information (NLRI) to include the MPLS VPN labels [RFC8277]. Such a MPLS VPN label is associated with a forwarding decision in the VPN Routing Instance on the egress BGP Router. The ingress BGP router will push the VPN label on the data packet destined to the egress BGP router. The transport tunnel from ingress router to egress router can be MPLS or GRE or L2TPv3, but inner payload is a MPLS packet as described in [RFC4023], [RFC4817], [RFC7510]. The intermediate routers do not process the VPN label [a.k.a.] embedded label as described in [I-D.ietf-idr-tunnel-encaps].

To provide BGP based VPN services on a non-MPLS IPv6 networks, it would be beneficial to retain the benefits of BGP protocol extensions while leveraging the benefits of IPv6 [RFC8200].

[I-D.bonica-6man-vpn-dest-opt] describes SRv6+ paths as programmable with Per-Path Service Instructions (PPSI) that determine how egress nodes process SRv6+ payloads. The PPSIs are carried in the PPSI Option encoded in the IPv6 Destination Option Header [RFC8200].

The Per-Path Service Instruction (PPSI) Identifier is defined as follows:

- o 32 bit quantity.

The PPSI Identifier have node-local significance and is assigned by the egress BGP router. The value of zero is reserved. The PPSI Identifier will serve 2 purposes.

- o It MUST uniquely identify the VPN Routing Instance for L3VPN or identify an Ethernet Segment for EVPN or identify a leaf property for EVPN TREE upon which forwarding decision can be taken.
- o It MAY provide information for special processing before the packet is forwarded.

The structure of 3 octet PPSI Identifier will be updated in the next version of this document.

The encoding of the Per-Path Service Instruction Identifier for VPNs is described in Section 7 and Section 8.

4. Usage of Tunnel Encapsulation Attribute

This document defines a new Tunnel type : SRv6+. The format is as per below.

- o Tunnel Type (2 Octets) : To be assigned
- o Tunnel Length (2 Octets) : 1
- o Value : List of Sub-TLVs

[I-D.ietf-idr-tunnel-encaps] defines many sub-TLVs for the tunnels. The encoding for them are as follows:

- o Remote Endpoint sub-TLV : As per [I-D.ietf-idr-tunnel-encaps]
- o Encapsulation sub-TLV : Not needed.

- o IPv4 DS Field sub-TLV : Not needed.
- o UDP Destination Port sub-TLV : Not needed.
- o Protocol type sub-TLV : As per [I-D.ietf-idr-tunnel-encaps].
- o Color Sub-TLV : As per [I-D.ietf-idr-tunnel-encaps].
- o Embedded Label Handling sub-TLV : 2.
- o MPLS Label Stack Sub-TLV : Not needed.
- o Prefix SID Sub-TLV : Not Needed.

The Tunnel Encapsulation Attribute is an Optional Transitive attribute as described in [I-D.ietf-idr-tunnel-encaps]. This attribute with SRv6+ tunnel type MUST be present in the BGP update carrying the Network Layer Reachability Information encoded with the PPSI Information. This document refers to the NLRI that is associated with SRv6+ Tunnel Encapsulation attribute as SRv6+_NLRI. The document [I-D.ietf-idr-tunnel-encaps-12] defines the encoding for sub-TLV as follows.

- o Sub-TLV Type : 1 octet
- o Sub-TLV Length : 1 or 2 octets
- o Sub-TLV Value : defined per Sub-TLV as per below.

The Remote Endpoint sub-TLV can specify the IPv6 address of the egress router as the final destination address of SRv6+ packet which is also referred to as SR Path destination address. The sub-fields on this sub-TLV is encoded as below.

- o Autonomous System Number : AS number of the IPv6 SR domain.
- o Address Family : 2 (refers to IPv6).
- o Address : IPv6 address of the egress interface present in SRv6+ domain.

The Value field may be set to 0 which indicates that next hop value in the NLRI should be chosen for the SRv6+ Path destination address.

The Embedded Label Handling sub-TLV describes how the label field in the NLRI should be interpreted.

- o Value : MUST be set to 2.

The value of 2 indicates that the label field in the NLRI MUST be ignored at the ingress router.

5. Procedures for Egress BGP Speaker

The PPSI Information instructs the egress router to de-encapsulate the packet and forward the newly exposed payload inner packet through the specified interface or forward using the specified Routing Instance. The PPSI Identifier described in Section 3 will be assigned by the egress BGP Router except in the case of EVPN per ES AD route when P2MP tunnel is used for delivering BUM traffic in EVPN. If P2MP tunnel is used to deliver BUM traffic for EVPN, the PPSI Identifier used to identify an Ethernet Segment is assigned by the upstream ingress BGP Router. Otherwise, it is downstream assigned by the egress BGP router.

When the egress BGP Speaker advertises the NLRI, it will include the PPSI Information in the encoding described in Section 7 and Section 8. The egress BGP Speaker MUST include the Tunnel Encapsulation Attribute with Route type SRv6+ as described in Section 4 in such BGP updates.

By tagging the BGP update with Tunnel Encapsulation attribute of SRv6+ type, the BGP Speaker informs how the SRv6+_NLRI should be decoded and processed by the receiving BGP Speaker.

Via the Remote Tunnel Endpoint Sub-TLV encoding, the egress BGP router may specify the SRv6+ Path Destination Address. The Protocol type Sub-TLV and the Color Sub-TLV may be used by the egress BGP router to influence the payload packets to be put on SRv6+ path. The Embedded Label Handling Sub-TLV MUST be set to 2 to inform that the MPLS label field should be ignored.

A single PPSI Identifier may be associated with all the prefixes in a Routing Instance or a unique PPSI Identifier may be associated for each prefix in the Routing Instance. Similarly, a PPSI Identifier may be assigned to identify an Ethernet segment or leaf AC property by EVPN. The choice is left to the Network Operator and is outside the scope of this document.

6. Procedures for Ingress BGP Speaker

Upon receiving a BGP update, the receiving BGP Speaker will look for Tunnel Encapsulation attribute. If the tunnel type carried in the Tunnel Encapsulation attribute is SRv6+, the BGP updates is said to be carrying the SRv6+_NLRI and the Label field in the Network Layer Reachability Information is treated as Per-Path Service Instruction (PPSI) Identifier.

The tuple (PPSI Identifier, Prefix) is programmed in the forwarding infrastructure of the router. The manner in which this tuple is stored in the router is outside the scope of this document. If SRv6+ has been enabled on the router, such a tuple SHOULD be used for encoding the Destination Options Header as described in [I-D.bonica-6man-vpn-dest-opt].

[I.D.ietf-idr-tunnel-encaps-12] describes how Remote Tunnel Endpoint Sub-TLV has to be processed. It also describes the usage of the Protocol type Sub-TLV and the Color Sub-TLV. This may be used by the ingress BGP router to select the payload packets that should be put on SRv6+ path.

The Embedded Label Handling Sub-TLV value that is set to 2 indicates that ingress BGP router to ignore the MPLS label field.

7. BGP based L3 VPN services over IPv6

The Egress and Ingress BGP speakers form a BGP peering session to exchange a set of prefixes described in [RFC4271] and Multi protocol extensions [RFC4760]. The BGP Router capable of SRv6+ that is enabled to carry L3 VPN services over IPv6 networks should follow the procedures mentioned in Section 5 and Section 6. The manner in which a BGP Router is configured for SRv6+ underlay and L3 VPN overlay is outside the scope of this document.

7.1. IPv4 VPN on SRv6+ enabled IPv6 Core

The IPv4 L3 VPN over IPv6 is defined in [RFC5549]. The MP_REACH NLRI and Tunnel Encapsulation attribute encoding is as per below:

- o AFI : 1; SAFI : 128
- o Length of the Next Hop : 16 (or 32 if Link Local)
- o Network address of the Next Hop : IPv6 address of the egress BGP Router
- o NLRI : IPv4-VPN routes
- o Label : Per-Path Service Instruction Identifier
- o Tunnel Encapsulation Path Attribute : SRv6+ Type as described in Section 4

The PPSI Identifier is associated with VPN Routing Instance on the Egress PE. The Tunnel Encapsulation attribute with SRv6+ type MUST be appended to the Path attributes associated with the NLRI.

7.2. IPv6 VPN on SRv6+ enabled IPv6 Core

The IPv6 L3 VPN over IPv6 is defined in [RFC4659]. The MP_REACH NLRI and Tunnel Encapsulation attribute encoding is as per below:

- o AFI : 2; SAFI : 128
- o Length of the Next Hop : 16 (or 32 if Link Local)
- o Network address of the Next Hop : IPv6 address of the egress BGP Router
- o NLRI : IPv6-VPN routes
- o Label : Per-Path Service Instruction Identifier
- o Tunnel Encapsulation Path Attribute : SRv6+ Type as described in Section 4

The PPSI Identifier is associated with VPN Routing Instance on the Egress PE. The Tunnel Encapsulation attribute with SRv6+ type MUST be appended to the Path attribute associated with the NLRI.

7.3. IPv4 Global Routes on SRv6+ enabled IPv6 Core

The IPv4 L3 VPN over IPv6 is defined in [RFC5549]. The MP_REACH NLRI and Tunnel Encapsulation attribute encoding is per below:

- o AFI : 1; SAFI : 1
- o Length of the Next Hop : 16 (or 32 if Link Local)
- o Network address of the Next Hop : IPv6 address of the egress BGP Router
- o NLRI : IPv4 routes
- o Label : Per-Path Service Instruction Identifier
- o Tunnel Encapsulation Path Attribute : SRv6+ Type as described in Section 4

The PPSI Identifier is associated with VPN Routing Instance on the Egress PE. The Tunnel Encapsulation attribute with SRv6+ type MUST be appended to the Path attribute associated with the NLRI.

8. BGP based Ethernet VPN services over IPv6

The [RFC7432] describes the BGP extensions for carrying the Ethernet Virtual Private Network Overlay on MPLS network. It defines 4 types of EVPN NLRI. This document specifies changes to certain fields for those NLRIs.

- o Ethernet Auto-Discovery (A-D) route
- o MAC/IP Advertisement route
- o Inclusive Multicast Ethernet Tag route
- o IP Prefix route

8.1. Ethernet Per ES Auto-Discovery (A-D) route

The MP_REACH and MP_UNREACH attributes will carry this route in the NLRI encoding described in [RFC7432]. In addition to Tunnel Encapsulation attribute encoding, this document recommends to follow the [RFC7432] encoding except the following. For MPLS label carried in the Ethernet A-D per ESI route:

- o MPLS label : Per [RFC7432], it is set to zero.
- o Tunnel Encapsulation Path Attribute : SRv6+ Type as described in Section 4

The MPLS label field is not part of the route but treated as route attribute. For procedures and usage of this route, refer to [RFC7432]. The Tunnel Encapsulation attribute with SRv6+ type MUST be appended to the Path attribute associated with the NLRI.

An EVPN Ethernet per ES A-D route is usually signaled together with an ESI label extended community. For ESI Label carried in the ESI label extended community:

- o ESI Label: Per-Path Service Instruction Identifier

The Per-Path Service Instruction Identifier is used to identify an Ethernet segment attached to the BGP PE for EVPN.

If P2MP tunnel is used to deliver BUM traffic, then this PPSI Identifier is upstream assigned by the ingress router, otherwise it is downstream assigned by the egress router.

8.2. Ethernet per EVI Auto-Discovery (A-D) route

The MP_REACH and MP_UNREACH attributes will carry this route in the NLRI encoding described in [RFC7432]. In addition to Tunnel Encapsulation attribute encoding, this document recommends to follow the [RFC4732] encoding except the following:

- o MPLS label : Per-Path Service Instruction Identifier
- o Tunnel Encapsulation Path Attribute : SRv6+ Type as described in Section 4

The MPLS label field is not part of the route but treated as route attribute. For procedures and usage of this route, refer to [RFC7432]. The Tunnel Encapsulation attribute with SRv6+ type MUST be appended to the Path attribute associated with the NLRI.

In addition, for EVPN E-tree service, this route may be signaled together with an E-Tree Extended Community as it is specified in [RFC8317]. For the leaf label carried in the E-Tree Extended Community:

- o Leaf Label: Per-Path Service Instruction Identifier

In case of EVPN E-tree service, the per-path service identifier carried in the E-Tree extended community is used to signal a leaf AC property.

In the data plane, this PPSI identifier specified in the Destination Option header is used by an egress router to identify that a data packet is ingressed from a leaf AC such that appropriate forwarding decision can be made.

If P2MP tunnel is used to deliver BUM traffic, then this PPSI Identifier is upstream assigned by the ingress router. Otherwise it is downstream assigned by the egress router.

8.3. MAC/IP Advertisement route

The MP_REACH and MP_UNREACH attributes will carry this route in the NLRI encoding described in [RFC7432]. In addition to Tunnel Encapsulation attribute encoding, this document recommends to follow the [RFC4732] encoding except the following.

- o MPLS label1 : Per-Path Service Instruction Identifier1
- o MPLS label2 : Per-Path Service Instruction Identifier2

- o Tunnel Encapsulation Path Attribute : SRv6+ Type as described in Section 4

The MPLS label field is not part of the route but treated as route attribute. For procedures and usage of this route, refer to [RFC7432]. The Tunnel Encapsulation attribute with SRv6+ type MUST be appended to the Path attribute associated with the NLRI.

8.4. Inclusive Multicast Ethernet Route

The MP_REACH and MP_UNREACH attributes will carry this route in the NLRI encoding described in [RFC7432]. In addition to Tunnel Encapsulation attribute encoding, this document recommends to follow the [RFC4732] encoding except the following.

- o If MPLS label field in the PMSI Tunnel Attribute is non-zero, it is set to Per-Path Service Instruction Identifier.
- o Tunnel Encapsulation Path Attribute : SRv6+ Type as described in Section 4

The Tunnel Encapsulation attribute with SRv6+ type MUST be appended to the Path attribute associated with the NLRI.

8.5. IP Prefix Route

The MP_REACH and MP_UNREACH attributes will carry this route in the NLRI encoding described in [I-D draft-ietf-bess-evpn-prefix-advertisement]. In addition to Tunnel Encapsulation attribute encoding, this document recommends the following change:

- o MPLS label: if it is non-zero, it is set to Per-Path Service Instruction Identifier.
- o Tunnel Encapsulation Path Attribute : SRv6+ Type as described in Section 4

The MPLS label field is not part of the route but treated as route attribute. For procedures and usage of this route, refer to [I-D draft-ietf-bess-evpn-prefix-advertisement]. The Tunnel Encapsulation attribute with SRv6+ type MUST be appended to the Path attribute associated with the NLRI.

9. Deployment Considerations

This document proposes to reuse the NLRI encoding for BGP L3VPN and EVPN Network Layer Routing Information. However, care should be taken when BGP VPN overlay services are enabled on SRv6+ underlay

such that Tunnel Encapsulation Path attribute with SRv6+ type MUST be appended. When a BGP router advertises SRv6+_NLRI, it MUST NOT remove the Tunnel Encapsulation Path attribute.

The SRv6+ underlay is similar to other "tunnel" technologies viz MPLS, GRE, IP-in-IP, L2TPv3. The egress and ingress BGP routers can be connected via one or more such underlay technologies. A BGP speaker can advertise the VPN NLRI with the nexthop reachable via one or more such underlay paths. Each such mechanism can co-exist together as ships-in-night. However, when SRv6+_NLRI is advertised by a egress BGP speaker and received by an ingress BGP speaker, they MUST follow the procedures mentioned in this document.

For migrating a BGP router to SRv6+ the following procedures can be followed.

- o Operator will enable SRv6+ underlay on the ingress and egress routers identifying the SRv6+ path from ingress router's interface to egress router's interface. The way to configure the ingress and egress routers are outside the scope of this document.
- o SRv6+ enabled ingress BGP router will setup the additional information in the forwarding table such that it can append an IPv6 tunnel header and encode the PPSI Option in the Destination Options Header.
- o SRv6+ enabled egress BGP router will setup the additional information in the forwarding table such that PPSI Identifier can be used to lookup to find the Routing Instance and make the forwarding decision.
- o Operator will enable BGP VPN overlay over SRv6+ underlay on ingress router. This means that ingress router will start looking for SRv6+_NLRI in the BGP updates. The way to enable the BGP VPN overlay over SRv6+ underlay is outside the scope of this document.
- o The operator will enable BGP VPN overlay over SRv6+ underlay on egress router. With this, the egress router will create PPSI Identifier and associate it with Routing Instances. It then advertises the SRv6+_NLRI to the ingress BGP router.
- o The ingress router will interpret the SRv6+_NLRI and use PPSI identifier and follow the procedures in [I.D. bonica-spring-srv6-plus-00.txt] to encode the Destination Options Header to forward the data packet.
- o Now that SRv6+ path is setup between ingress and egress BGP routers, on the egress BGP router the Operator can migrate the

Routing Instances from MPLS VPN set of Instances to SRv6+ enabled set of Instances. The way to configure Routing Instances to achieve the above is outside the scope of this document.

10. Backward Compatibility

The extension proposed in this document is backward compatible with procedures described for BGP enabled services.

11. Security Considerations

This document does not introduce any new security considerations beyond those already specified in [RFC4271], [RFC8277] and [I.D.ietf-idr-tunnel-encaps-12].

12. IANA Considerations

IANA is requested to assign a code point for SRv6+ Route Type for BGP Tunnel Encapsulation Path Attribute from BGP Tunnel Encapsulation Attribute Tunnel Types Registry.

13. Acknowledgements

The authors would like to thank Jeff Haas and Wen Lin for careful review and suggestions.

14. References

14.1. Normative References

[I-D.bonica-6man-vpn-dest-opt]

Bonica, R., Lenart, C., So, N., Xu, F., Presbury, G., Chen, G., Zhu, Y., Yang, G., and Y. Zhou, "The IPv6 Virtual Private Network (VPN) Context Information Option", draft-bonica-6man-vpn-dest-opt-05 (work in progress), March 2019.

[I-D.bonica-spring-srv6-plus]

Bonica, R., Hegde, S., Kamite, Y., Alston, A., Henriques, D., Halpern, J., and J. Linkova, "IPv6 Support for Segment Routing: SRv6+", draft-bonica-spring-srv6-plus-01 (work in progress), July 2019.

[I-D.ietf-idr-tunnel-encaps]

Patel, K., Velde, G., Ramachandra, S., and E. Rosen, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-12 (work in progress), May 2019.

- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.

14.2. Informative References

- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <<https://www.rfc-editor.org/info/rfc4023>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.

- [RFC4659] De Clercq, J., Ooms, D., Carugi, M., and F. Le Faucheur, "BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN", RFC 4659, DOI 10.17487/RFC4659, September 2006, <<https://www.rfc-editor.org/info/rfc4659>>.
- [RFC4732] Handley, M., Ed., Rescorla, E., Ed., and IAB, "Internet Denial-of-Service Considerations", RFC 4732, DOI 10.17487/RFC4732, December 2006, <<https://www.rfc-editor.org/info/rfc4732>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<https://www.rfc-editor.org/info/rfc4760>>.
- [RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <<https://www.rfc-editor.org/info/rfc4761>>.
- [RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <<https://www.rfc-editor.org/info/rfc4762>>.
- [RFC4817] Townsley, M., Pignataro, C., Wainner, S., Seely, T., and J. Young, "Encapsulation of MPLS over Layer 2 Tunneling Protocol Version 3", RFC 4817, DOI 10.17487/RFC4817, March 2007, <<https://www.rfc-editor.org/info/rfc4817>>.
- [RFC5549] Le Faucheur, F. and E. Rosen, "Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop", RFC 5549, DOI 10.17487/RFC5549, May 2009, <<https://www.rfc-editor.org/info/rfc5549>>.
- [RFC6624] Kompella, K., Kothari, B., and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, DOI 10.17487/RFC6624, May 2012, <<https://www.rfc-editor.org/info/rfc6624>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.

- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black,
"Encapsulating MPLS in UDP", RFC 7510,
DOI 10.17487/RFC7510, April 2015,
<<https://www.rfc-editor.org/info/rfc7510>>.
- [RFC8077] Martini, L., Ed. and G. Heron, Ed., "Pseudowire Setup and
Maintenance Using the Label Distribution Protocol (LDP)",
STD 84, RFC 8077, DOI 10.17487/RFC8077, February 2017,
<<https://www.rfc-editor.org/info/rfc8077>>.
- [RFC8277] Rosen, E., "Using BGP to Bind MPLS Labels to Address
Prefixes", RFC 8277, DOI 10.17487/RFC8277, October 2017,
<<https://www.rfc-editor.org/info/rfc8277>>.
- [RFC8317] Sajassi, A., Ed., Salam, S., Drake, J., Uttaro, J.,
Boutros, S., and J. Rabadan, "Ethernet-Tree (E-Tree)
Support in Ethernet VPN (EVPN) and Provider Backbone
Bridging EVPN (PBB-EVPN)", RFC 8317, DOI 10.17487/RFC8317,
January 2018, <<https://www.rfc-editor.org/info/rfc8317>>.

Authors' Addresses

Srihari Sangli
Juniper Networks Inc.
Exora Business Park
Bangalore, KA 560103
India

Email: ssangli@juniper.net

Ron Bonica
Juniper Networks Inc.
2251 Corporate Park Drive
Herndon, Virginia 20171
USA

Email: rbonica@juniper.net