

INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros
VMware
Ali Sajassi
Cisco Systems
John Drake
Juniper Networks

Expires: December 21, 2017

June 19, 2017

EVPN control plane for Geneve
draft-boutros-bess-evpn-geneve-00.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 in NVO3 solutions. EVPN control plane can be used by a Network Virtualization Endpoints (NVEs) to express as well what Geneve tunnel option TLV(s) that they can transmit and/or receive.

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	3
1.1	Terminology	3
2.	BGP Extensions	4
2.1	Geneve Tunnel Option Types sub-TLV	4
3.	Operation	5
3.1	Negotiating TLV ordering, Size and total option length	6
4.	Security Considerations	6
5.	IANA Considerations	6
6.	Acknowledgements	6
7	References	6
7.1	Normative References	6
7.2	Informative References	7
	Authors' Addresses	7

1 Introduction

The Network Virtualization over Layer 3 (NVO3) develop solutions for network virtualization within a data center (DC) environment that assumes 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 signals Geneve encapsulation type in the BGP Tunnel Encapsulation Extended Community. The also 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].

The NVO3 encapsulation design team has made a recommendation in [DT-ENCAP] for a control plane to negotiate a subset of option TLVs and certain TLV ordering, as well can limit the total number of option TLVs present in the packet, for example, to allow hardware capable of processing fewer options.

This EVPN control plane extension will allow a Network Virtualization Endpoint (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", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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 Endpoint.

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. BGP Extensions

As per [ietf-evpn-overlay] the BGP Encapsulation extended community defined in [TUNNEL-ENCAP] and [RFC5512] 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.

2.1 Geneve Tunnel Option Types sub-TLV

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

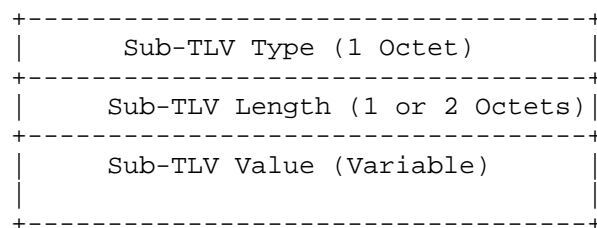


Figure 1: Geneve tunnel option types sub-TLV

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

Sub-TLV value will be the Geneve option TLV types, each type will be encoded as a 24 bit value.

3. Operation

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

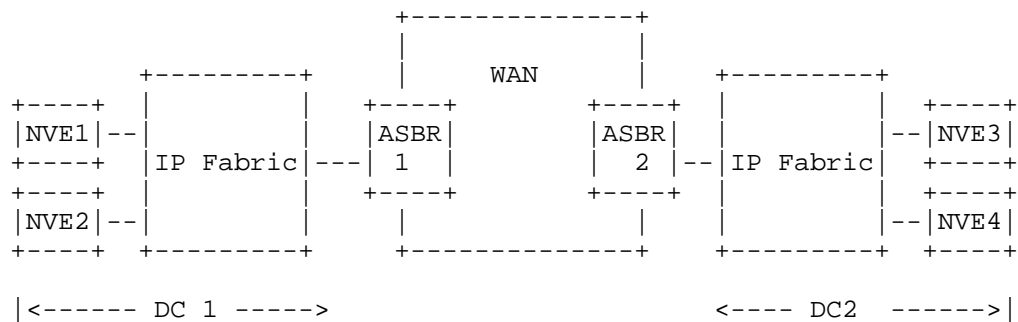


Figure 2: Data Center Interconnect with ASBR

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.

3.1 Negotiating TLV ordering, Size and total option length

TBD

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 IP-based tunnel technologies to support data plane transport. Security considerations described in [RFC7432] and in [ietf-evpn-overlay] are equally applicable.

5. 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.

6. Acknowledgements

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

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, <<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>>.

[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-04, work in progress, March, 2017.

[DT-ENCAP] Boutros, et al. "NVO3 Encapsulation Considerations", draft-ietf-nvo3-encap-00, work in progress, June, 2017.

7.2 Informative References

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

[TUNNEL-ENCAP] Rosen et al., "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-03, work in progress, May 31, 2016.

[ietf-evpn-overlay] Sajassi-Drake et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-07.txt, work in progress, December, 2016

Authors' Addresses

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

Ali Sajassi
Cisco
Email: sajassi@cisco.com

John Drake
Juniper Networks
Email: jdrake@juniper.net

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.

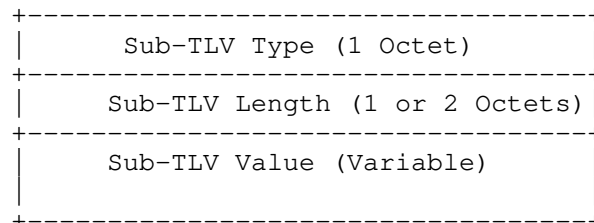


Figure 1: Geneve tunnel option types sub-TLV

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.

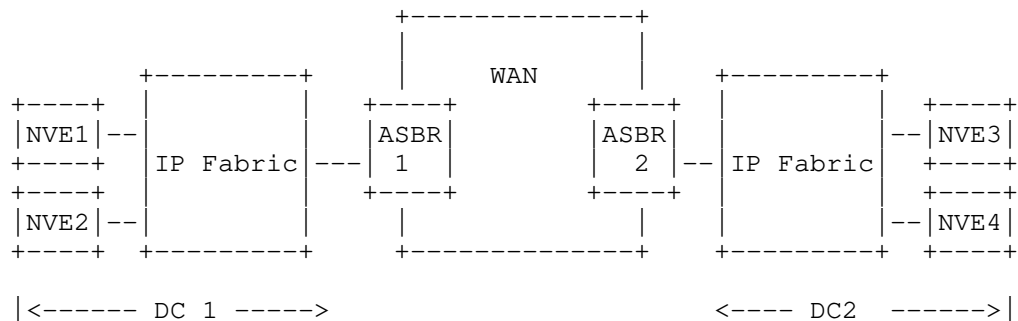


Figure 2: Data Center Interconnect with ASBR

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 2, 2018

J. Drake
A. Farrel
E. Rosen
Juniper Networks
K. Patel
Arrcus, Inc.
L. Jalil
Verizon
July 1, 2017

Gateway Auto-Discovery and Route Advertisement for Segment Routing
Enabled Domain Interconnection
draft-drake-bess-datacenter-gateway-04

Abstract

Data centers have become critical components of the infrastructure used by network operators to provide services to their customers. Data centers are attached to the Internet or a backbone network by gateway routers. One data center typically has more than one gateway for commercial, load balancing, and resiliency reasons.

Segment routing is a popular protocol mechanism for operating within a data center, but also for steering traffic that flows between two data center sites. In order that one data center site may load balance the traffic it sends to another data center site it needs to know the complete set of gateway routers at the remote data center, the points of connection from those gateways to the backbone network, and the connectivity across the backbone network.

Segment routing may also be operated in other domains, such as access networks. Those domains also need to be connected across backbone networks through gateways.

This document defines a mechanism using the BGP Tunnel Encapsulation attribute to allow each gateway router to advertise the routes to the prefixes in the segment routing domains to which it provides access, and also to advertise on behalf of each other gateway to the same segment routing domain.

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 [RFC2119].

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 <http://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 2, 2018.

Copyright 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	3
2. SR Domain Gateway Auto-Discovery	5
3. Relationship to BGP Link State and Egress Peer Engineering	6
4. Advertising an SR Domain Route Externally	6
5. Encapsulation	7
6. IANA Considerations	7
7. Security Considerations	7
8. Manageability Considerations	7
9. Acknowledgements	7
10. References	8
10.1. Normative References	8
10.2. Informative References	8
Authors' Addresses	9

1. Introduction

Data centers (DCs) have become critical components of the infrastructure used by network operators to provide services to their customers. DCs are attached to the Internet or a backbone network by gateway routers (GWs). One DC typically has more than one GW for various reasons including commercial preferences, load balancing, and resiliency against connection of device failure.

Segment routing (SR) [I-D.ietf-spring-segment-routing] is a popular protocol mechanism for operating within a DC, but also for steering traffic that flows between two DC sites. In order for an ingress DC that uses SR to load balance the flows it sends to an egress DC, it needs to know the complete set of entry nodes (i.e., GWs) for that egress DC from the backbone network connecting the two DCs. Note that it is assumed that the connected set of DCs and the backbone network connecting them are part of the same SR BGP Link State (LS) instance ([RFC7752] and [I-D.ietf-idr-bgpls-segment-routing-epe]) so that traffic engineering using SR may be used for these flows.

Segment routing may also be operated in other domains, such as access networks. Those domains also need to be connected across backbone networks through gateways.

Suppose that there are two gateways, GW1 and GW2 as shown in Figure 1, for a given egress segment routing domain and that they each advertise a route to prefix X which is located within the egress segment routing domain with each setting itself as next hop. One might think that the GWs for X could be inferred from the routes' next hop fields, but typically it is not the case that both routes get distributed across the backbone: rather only the best route, as selected by BGP, is distributed. This precludes load balancing flows across both GWs.

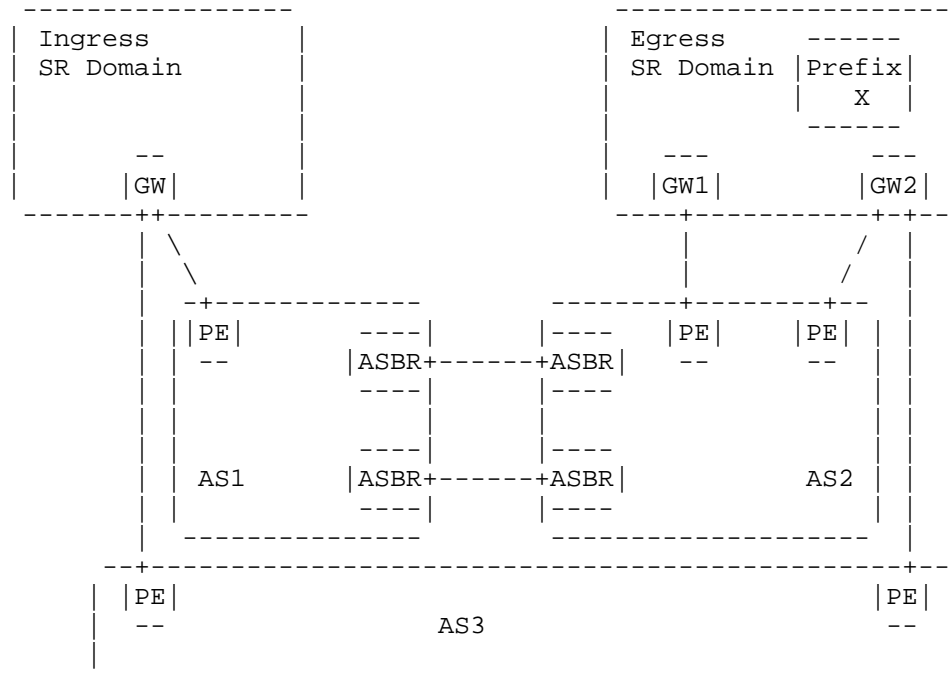


Figure 1: Example Segment Routing Domain Interconnection

The obvious solution to this problem is to use the BGP feature that allows the advertisement of multiple paths in BGP (known as Add-Paths) [RFC7911] to ensure that all routes to X get advertised by BGP. However, even if this is done, the identity of the GWs will be lost as soon as the routes get distributed through an Autonomous System Border Router (ASBR) that will set itself to be the next hop. And if there are multiple Autonomous Systems (ASes) in the backbone, not only will the next hop change several times, but the Add-Paths technique will experience scaling issues. This all means that this approach is limited to SR domains connected over a single AS.

This document defines a solution that overcomes this limitation and works equally well with a backbone constructed from one or more ASes. This solution uses the Tunnel Encapsulation attribute [I-D.ietf-idr-tunnel-encaps] as follows:

We define a new tunnel type, "SR tunnel". When the GWs to a given SR domain advertise a route to a prefix X within the SR domain, they will each include a Tunnel Encapsulation attribute with multiple tunnel instances each of type "SR tunnel", one for each

GW, and each containing a Remote Endpoint sub-TLV with that GW's address.

In other words, each route advertised by any GW identifies all of the GWs to the same SR domain (see Section 2 for a discussion of how GWs discover each other). Therefore, even if only one of the routes is distributed to other ASes, it will not matter how many times the next hop changes, as the Tunnel Encapsulation attribute (and its remote endpoint sub-TLVs) will remain unchanged.

To put this in the context of Figure 1, GW1 and GW2 discover each other as gateways for the egress SR domain. Both GW1 and GW2 advertise themselves as having routes to prefix X. Furthermore, GW1 includes a Tunnel Encapsulation attribute with a tunnel instance of type "SR tunnel" for itself and another for GW2. Similarly, GW2 includes a Tunnel Encapsulation for itself and another for GW1. The gateway in the ingress SR domain can now see all possible paths to the egress SR domain regardless of which route advertisement is propagated to it, and it can choose one or balance traffic flows as it sees fit.

2. SR Domain Gateway Auto-Discovery

To allow a given SR domain's GWs to auto-discover each other and to coordinate their operations, the following procedures are implemented:

- o Each GW is configured with an identifier for the SR domain that is common across all GWs to the domain (i.e., across all GWs to all SR domains that are interconnected) and unique across all SR domains that are connected.
- o A route target ([RFC4360]) is attached to each GW's auto-discovery route and has its value set to the SR domain identifier.
- o Each GW constructs an import filtering rule to import any route that carries a route target with the same SR domain identifier that the GW itself uses. This means that only these GWs will import those routes and that all GWs to the same SR domain will import each other's routes and will learn (auto-discover) the current set of active GWs for the SR domain.

The auto-discovery route each GW advertises consists of the following:

- o An IPv4 or IPv6 NLRI containing one of the GW's loopback addresses (that is, with AFI/SAFI that is one of 1/1, 2/1, 1/4, or 2/4).

- o A Tunnel Encapsulation attribute containing the GW's encapsulation information, which at a minimum consists of an SR tunnel TLV (type to be allocated by IANA) with a Remote Endpoint sub-TLV as specified in [I-D.ietf-idr-tunnel-encaps].

To avoid the side effect of applying the Tunnel Encapsulation attribute to any packet that is addressed to the GW itself, the GW SHOULD use a different loopback address for the two cases.

As described in Section 1, each GW will include a Tunnel Encapsulation attribute for each GW that is active for the SR domain (including itself), and will include these in every route advertised externally to the SR domain by each GW. As the current set of active GWs changes (due to the addition of a new GW or the failure/removal of an existing GW) each externally advertised route will be re-advertised with the set of SR tunnel instances reflecting the current set of active GWs.

If a gateway becomes disconnected from the backbone network, or if the SR domain operator decides to terminate the gateway's activity, it withdraws the advertisements described above. This means that remote gateways at other sites will stop seeing advertisements from this gateway. It also means that other local gateways at this site will "unlearn" the removed gateway and stop including a Tunnel Encapsulation attribute for the removed gateway in their advertisements.

3. Relationship to BGP Link State and Egress Peer Engineering

When a remote GW receives a route to a prefix X it can use the SR tunnel instances within the contained Tunnel Encapsulation attribute to identify the GWs through which X can be reached. It uses this information to compute SR TE paths across the backbone network looking at the information advertised to it in SR BGP Link State (BGP-LS) [I-D.gredler-idr-bgp-ls-segment-routing-ext] and correlated using the SR domain identity. SR Egress Peer Engineering (EPE) [I-D.ietf-idr-bgpls-segment-routing-epe] can be used to supplement the information advertised in the BGP-LS.

4. Advertising an SR Domain Route Externally

When a packet destined for prefix X is sent on an SR TE path to a GW for the SR domain containing X, it needs to carry the receiving GW's label for X such that this label rises to the top of the stack before the GW completes its processing of the packet. To achieve this we place a prefix-SID sub-TLV for X in each SR tunnel instance in the Tunnel Encapsulation attribute in the externally advertised route for X.

Alternatively, if the GWs for a given SR domain are configured to allow remote GWs to perform SR TE through that SR domain for a prefix X, then each GW computes an SR TE path through that SR domain to X from each of the currently active GWs, and places each in an MPLS label stack sub-TLV [I-D.ietf-idr-tunnel-encaps] in the SR tunnel instance for that GW.

5. Encapsulation

If the GWs for a given SR domain are configured to allow remote GWs to send them a packet in that SR domain's native encapsulation, then each GW will also include multiple instances of a tunnel TLV for that native encapsulation in externally advertised routes: one for each GW and each containing a remote endpoint sub-TLV with that GW's address. A remote GW may then encapsulate a packet according to the rules defined via the sub-TLVs included in each of the tunnel TLV instances.

6. IANA Considerations

IANA maintains a registry called "BGP parameters" with a sub-registry called "BGP Tunnel Encapsulation Tunnel Types." The registration policy for this registry is First-Come First-Served.

IANA is requested to assign a codepoint from this sub-registry for "SR Tunnel". The next available value may be used and reference should be made to this document.

[[Note: This text is likely to be replaced with a specific code point value once FCFS allocation has been made.]]

7. Security Considerations

TBD

8. Manageability Considerations

TBD

9. Acknowledgements

Thanks to Bruno Rijsman for review comments, and to Robert Raszuk for useful discussions.

10. References

10.1. Normative References

- [I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., Filsfils, C., Patel, K., Ray, S., and J. Dong, "BGP-LS extensions for Segment Routing BGP Egress Peer Engineering", draft-ietf-idr-bgpls-segment-routing-epe-13 (work in progress), June 2017.
- [I-D.ietf-idr-tunnel-encaps]
Rosen, E., Patel, K., and G. Velde, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-06 (work in progress), June 2017.
- [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>>.
- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <<http://www.rfc-editor.org/info/rfc4360>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<http://www.rfc-editor.org/info/rfc7752>>.

10.2. Informative References

- [I-D.gredler-idr-bgp-ls-segment-routing-ext]
Previdi, S., Psenak, P., Filsfils, C., Gredler, H., Chen, M., and j. jeffrant@gmail.com, "BGP Link-State extensions for Segment Routing", draft-gredler-idr-bgp-ls-segment-routing-ext-04 (work in progress), October 2016.
- [I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-12 (work in progress), June 2017.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<http://www.rfc-editor.org/info/rfc7911>>.

Authors' Addresses

John Drake
Juniper Networks

Email: jdrake@juniper.net

Adrian Farrel
Juniper Networks

Email: afarrel@juniper.net

Eric Rosen
Juniper Networks

Email: erosen@juniper.net

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 23, 2018

J. Drake
A. Farrel
E. Rosen
Juniper Networks
K. Patel
Arrcus, Inc.
L. Jalil
Verizon
September 19, 2017

Gateway Auto-Discovery and Route Advertisement for Segment Routing
Enabled Domain Interconnection
draft-drake-bess-datacenter-gateway-05

Abstract

Data centers have become critical components of the infrastructure used by network operators to provide services to their customers. Data centers are attached to the Internet or a backbone network by gateway routers. One data center typically has more than one gateway for commercial, load balancing, and resiliency reasons.

Segment routing is a popular protocol mechanism for operating within a data center, but also for steering traffic that flows between two data center sites. In order that one data center site may load balance the traffic it sends to another data center site it needs to know the complete set of gateway routers at the remote data center, the points of connection from those gateways to the backbone network, and the connectivity across the backbone network.

Segment routing may also be operated in other domains, such as access networks. Those domains also need to be connected across backbone networks through gateways.

This document defines a mechanism using the BGP Tunnel Encapsulation attribute to allow each gateway router to advertise the routes to the prefixes in the segment routing domains to which it provides access, and also to advertise on behalf of each other gateway to the same segment routing domain.

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 [RFC2119].

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 March 23, 2018.

Copyright 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 (<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. SR Domain Gateway Auto-Discovery	5
3. Relationship to BGP Link State and Egress Peer Engineering .	6
4. Advertising an SR Domain Route Externally	7
5. Encapsulation	7
6. IANA Considerations	7
7. Security Considerations	7
8. Manageability Considerations	9
9. Acknowledgements	9
10. References	9
10.1. Normative References	9
10.2. Informative References	10
Authors' Addresses	11

1. Introduction

Data centers (DCs) have become critical components of the infrastructure used by network operators to provide services to their customers. DCs are attached to the Internet or a backbone network by gateway routers (GWs). One DC typically has more than one GW for various reasons including commercial preferences, load balancing, and resiliency against connection of device failure.

Segment routing (SR) [I-D.ietf-spring-segment-routing] is a popular protocol mechanism for operating within a DC, but also for steering traffic that flows between two DC sites. In order for an ingress DC that uses SR to load balance the flows it sends to an egress DC, it needs to know the complete set of entry nodes (i.e., GWs) for that egress DC from the backbone network connecting the two DCs. Note that it is assumed that the connected set of DCs and the backbone network connecting them are part of the same SR BGP Link State (LS) instance ([RFC7752] and [I-D.ietf-idr-bgpls-segment-routing-epe]) so that traffic engineering using SR may be used for these flows.

Segment routing may also be operated in other domains, such as access networks. Those domains also need to be connected across backbone networks through gateways.

Suppose that there are two gateways, GW1 and GW2 as shown in Figure 1, for a given egress segment routing domain and that they each advertise a route to prefix X which is located within the egress segment routing domain with each setting itself as next hop. One might think that the GWs for X could be inferred from the routes' next hop fields, but typically it is not the case that both routes get distributed across the backbone: rather only the best route, as selected by BGP, is distributed. This precludes load balancing flows across both GWs.

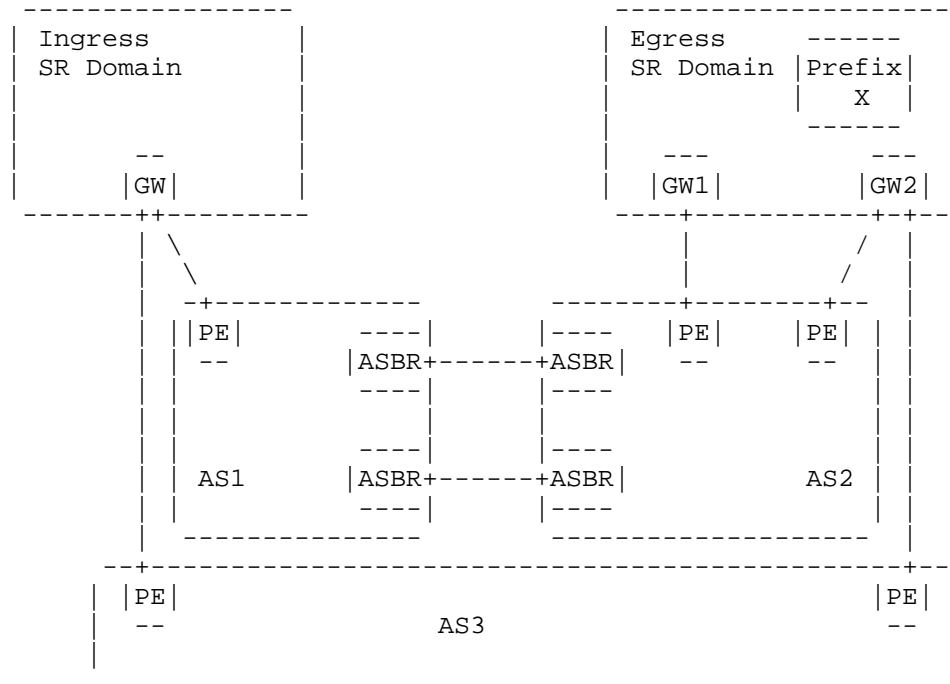


Figure 1: Example Segment Routing Domain Interconnection

The obvious solution to this problem is to use the BGP feature that allows the advertisement of multiple paths in BGP (known as Add-Paths) [RFC7911] to ensure that all routes to X get advertised by BGP. However, even if this is done, the identity of the GWs will be lost as soon as the routes get distributed through an Autonomous System Border Router (ASBR) that will set itself to be the next hop. And if there are multiple Autonomous Systems (ASes) in the backbone, not only will the next hop change several times, but the Add-Paths technique will experience scaling issues. This all means that this approach is limited to SR domains connected over a single AS.

This document defines a solution that overcomes this limitation and works equally well with a backbone constructed from one or more ASes. This solution uses the Tunnel Encapsulation attribute [I-D.ietf-idr-tunnel-encaps] as follows:

We define a new tunnel type, "SR tunnel". When the GWs to a given SR domain advertise a route to a prefix X within the SR domain, they will each include a Tunnel Encapsulation attribute with multiple tunnel instances each of type "SR tunnel", one for each

GW, and each containing a Remote Endpoint sub-TLV with that GW's address.

In other words, each route advertised by any GW identifies all of the GWs to the same SR domain (see Section 2 for a discussion of how GWs discover each other). Therefore, even if only one of the routes is distributed to other ASes, it will not matter how many times the next hop changes, as the Tunnel Encapsulation attribute (and its remote endpoint sub-TLVs) will remain unchanged.

To put this in the context of Figure 1, GW1 and GW2 discover each other as gateways for the egress SR domain. Both GW1 and GW2 advertise themselves as having routes to prefix X. Furthermore, GW1 includes a Tunnel Encapsulation attribute with a tunnel instance of type "SR tunnel" for itself and another for GW2. Similarly, GW2 includes a Tunnel Encapsulation for itself and another for GW1. The gateway in the ingress SR domain can now see all possible paths to the egress SR domain regardless of which route advertisement is propagated to it, and it can choose one or balance traffic flows as it sees fit.

The protocol extensions defined in this document are put into the broader context of SR domain interconnection by [I-D.farrel-spring-sr-domain-interconnect]. That document shows how other existing protocol elements may be combined with the extensions defined in this document to provide a full system.

2. SR Domain Gateway Auto-Discovery

To allow a given SR domain's GWs to auto-discover each other and to coordinate their operations, the following procedures are implemented:

- o Each GW is configured with an identifier for the SR domain that is common across all GWs to the domain (i.e., across all GWs to all SR domains that are interconnected) and unique across all SR domains that are connected.
- o A route target ([RFC4360]) is attached to each GW's auto-discovery route and has its value set to the SR domain identifier.
- o Each GW constructs an import filtering rule to import any route that carries a route target with the same SR domain identifier that the GW itself uses. This means that only these GWs will import those routes and that all GWs to the same SR domain will import each other's routes and will learn (auto-discover) the current set of active GWs for the SR domain.

The auto-discovery route each GW advertises consists of the following:

- o An IPv4 or IPv6 NLRI containing one of the GW's loopback addresses (that is, with AFI/SAFI that is one of 1/1, 2/1, 1/4, or 2/4).
- o A Tunnel Encapsulation attribute containing the GW's encapsulation information, which at a minimum consists of an SR tunnel TLV (type to be allocated by IANA) with a Remote Endpoint sub-TLV as specified in [I-D.ietf-idr-tunnel-encaps].

To avoid the side effect of applying the Tunnel Encapsulation attribute to any packet that is addressed to the GW itself, the GW SHOULD use a different loopback address for the two cases.

As described in Section 1, each GW will include a Tunnel Encapsulation attribute for each GW that is active for the SR domain (including itself), and will include these in every route advertised externally to the SR domain by each GW. As the current set of active GWs changes (due to the addition of a new GW or the failure/removal of an existing GW) each externally advertised route will be re-advertised with the set of SR tunnel instances reflecting the current set of active GWs.

If a gateway becomes disconnected from the backbone network, or if the SR domain operator decides to terminate the gateway's activity, it withdraws the advertisements described above. This means that remote gateways at other sites will stop seeing advertisements from this gateway. It also means that other local gateways at this site will "unlearn" the removed gateway and stop including a Tunnel Encapsulation attribute for the removed gateway in their advertisements.

3. Relationship to BGP Link State and Egress Peer Engineering

When a remote GW receives a route to a prefix X it can use the SR tunnel instances within the contained Tunnel Encapsulation attribute to identify the GWs through which X can be reached. It uses this information to compute SR TE paths across the backbone network looking at the information advertised to it in SR BGP Link State (BGP-LS) [I-D.gredler-idr-bgp-ls-segment-routing-ext] and correlated using the SR domain identity. SR Egress Peer Engineering (EPE) [I-D.ietf-idr-bgp-ls-segment-routing-epe] can be used to supplement the information advertised in the BGP-LS.

4. Advertising an SR Domain Route Externally

When a packet destined for prefix X is sent on an SR TE path to a GW for the SR domain containing X, it needs to carry the receiving GW's label for X such that this label rises to the top of the stack before the GW completes its processing of the packet. To achieve this we place a prefix-SID sub-TLV for X in each SR tunnel instance in the Tunnel Encapsulation attribute in the externally advertised route for X.

Alternatively, if the GWs for a given SR domain are configured to allow remote GWs to perform SR TE through that SR domain for a prefix X, then each GW computes an SR TE path through that SR domain to X from each of the currently active GWs, and places each in an MPLS label stack sub-TLV [I-D.ietf-idr-tunnel-encaps] in the SR tunnel instance for that GW.

5. Encapsulation

If the GWs for a given SR domain are configured to allow remote GWs to send them a packet in that SR domain's native encapsulation, then each GW will also include multiple instances of a tunnel TLV for that native encapsulation in externally advertised routes: one for each GW and each containing a remote endpoint sub-TLV with that GW's address. A remote GW may then encapsulate a packet according to the rules defined via the sub-TLVs included in each of the tunnel TLV instances.

6. IANA Considerations

IANA maintains a registry called "BGP parameters" with a sub-registry called "BGP Tunnel Encapsulation Tunnel Types." The registration policy for this registry is First-Come First-Served.

IANA is requested to assign a codepoint from this sub-registry for "SR Tunnel". The next available value may be used and reference should be made to this document.

[[Note: This text is likely to be replaced with a specific code point value once FCFS allocation has been made.]]

7. Security Considerations

From a protocol point of view, the mechanisms described in this document can leverage the security mechanisms already defined for BGP. Further discussion of security considerations for BGP may be found in the BGP specification itself [RFC4271] and in the security analysis for BGP [RFC4272]. The original discussion of the use of

the TCP MD5 signature option to protect BGP sessions is found in [RFC5925], while [RFC6952] includes an analysis of BGP keying and authentication issues.

The mechanisms described in this document involve sharing routing or reachability information between domains: that may mean disclosing information that is normally contained within a domain. So it needs to be understood that normal security paradigms based on the boundaries of domains are weakened. Discussion of these issues with respect to VPNs can be found in [RFC4364] while [RFC7926] describes many of the issues associated with the exchange of topology or TE information between domains.

Particular exposures resulting from this work include:

- o Gateways to a domain will know about all other gateways to the same domain. This feature applies within a domain and so is not a substantial exposure, but it does mean that if the protocol BGP exchanges within a domain can be snooped or if a gateway can be subverted then an attacker may learn the full set of gateways to a domain. This facilitates more effective attacks on that domain.
- o The existence of multiple gateways to a domain becomes more visible across the backbone and even into remote domains. This means that an attacker is able to prepare a more comprehensive attack than exists when only the locally attached backbone network (e.g., the AS that hosts the domain) can see all of the gateways to a site.
- o A node in a domain that does not have external BGP peering (i.e., is not really a domain gateway and cannot speak BGP into the backbone network) may be able to get itself advertised as a gateway by letting other genuine gateways discover it (by speaking BGP to them within the domain) and so may get those genuine gateways to advertise it as a gateway into the backbone network.
- o If it is possible to modify a BGP message within the backbone, it may be possible to spoof the existence of a gateway. This could cause traffic to be attracted to a specific node and might result in blackholing of traffic.

All of the issues in the list above could cause disruption to domain interconnection, but are not new protocol vulnerabilities so much as new exposures of information that could be protected against using existing protocol mechanisms. Furthermore, it is a general observation that if these attacks are possible then it is highly likely that far more significant attacks can be made on the routing

system. It should be noted that BGP peerings are not discovered, but always arise from explicit configuration.

8. Manageability Considerations

TBD

9. Acknowledgements

Thanks to Bruno Rijsman for review comments, and to Robert Raszuk for useful discussions.

10. References

10.1. Normative References

- [I-D.ietf-idr-bgpls-segment-routing-epe]
Previdi, S., Filsfils, C., Patel, K., Ray, S., and J. Dong, "BGP-LS extensions for Segment Routing BGP Egress Peer Engineering", draft-ietf-idr-bgpls-segment-routing-epe-13 (work in progress), June 2017.
- [I-D.ietf-idr-tunnel-encaps]
Rosen, E., Patel, K., and G. Velde, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-07 (work in progress), July 2017.
- [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>>.
- [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>>.
- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <<https://www.rfc-editor.org/info/rfc4360>>.
- [RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <<https://www.rfc-editor.org/info/rfc5925>>.

- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.

10.2. Informative References

- [I-D.farrel-spring-sr-domain-interconnect]
Farrel, A. and J. Drake, "Interconnection of Segment Routing Domains - Problem Statement and Solution Landscape", draft-farrel-spring-sr-domain-interconnect-00 (work in progress), June 2017.
- [I-D.gredler-idr-bgp-ls-segment-routing-ext]
Previdi, S., Psenak, P., Filsfils, C., Gredler, H., Chen, M., and j. jefftant@gmail.com, "BGP Link-State extensions for Segment Routing", draft-gredler-idr-bgp-ls-segment-routing-ext-04 (work in progress), October 2016.
- [I-D.ietf-spring-segment-routing]
Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-12 (work in progress), June 2017.
- [RFC4272] Murphy, S., "BGP Security Vulnerabilities Analysis", RFC 4272, DOI 10.17487/RFC4272, January 2006, <<https://www.rfc-editor.org/info/rfc4272>>.
- [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>>.
- [RFC6952] Jethanandani, M., Patel, K., and L. Zheng, "Analysis of BGP, LDP, PCEP, and MSDP Issues According to the Keying and Authentication for Routing Protocols (KARP) Design Guide", RFC 6952, DOI 10.17487/RFC6952, May 2013, <<https://www.rfc-editor.org/info/rfc6952>>.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<https://www.rfc-editor.org/info/rfc7911>>.

[RFC7926] Farrel, A., Ed., Drake, J., Bitar, N., Swallow, G.,
Ceccarelli, D., and X. Zhang, "Problem Statement and
Architecture for Information Exchange between
Interconnected Traffic-Engineered Networks", BCP 206,
RFC 7926, DOI 10.17487/RFC7926, July 2016,
<<https://www.rfc-editor.org/info/rfc7926>>.

Authors' Addresses

John Drake
Juniper Networks

Email: jdrake@juniper.net

Adrian Farrel
Juniper Networks

Email: afarrel@juniper.net

Eric Rosen
Juniper Networks

Email: erosen@juniper.net

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.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
/nl:network-instances/nl:network-instance/nl:nl-type/l2vpn:l2vpn:
  +--rw evpn-instance?   evpn-instance-ref
augment
/nl:network-instances/nl:network-instance/nl:nl-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 unknoww 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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 11, 2019

Z. Zhang
Juniper Networks, Inc.
H. Tsunoda
Tohoku Institute of Technology
September 07, 2018

L2L3 VPN Multicast MIB
draft-ietf-bess-l2l3-vpn-mcast-mib-16

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes two MIB modules which will be used by other MIB modules for monitoring and/or configuring Layer 2 and Layer 3 Virtual Private Networks that support multicast.

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 <http://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 March 11, 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 (<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	2
1.1. Terminology	3
2. The Internet-Standard Management Framework	4
3. Summary of MIB Modules	4
4. Definitions	4
4.1. L2L3-VPN-MULTICAST-TC-MIB Object Definitions	4
4.2. L2L3-VPN-MULTICAST-MIB Object Definitions	9
5. Security Considerations	15
6. IANA Considerations	16
7. Acknowledgement	17
8. References	17
8.1. Normative References	17
8.2. Informative References	19
Authors' Addresses	20

1. Introduction

In BGP/MPLS Virtual Private Networks (VPNs), Border Gateway Protocol (BGP) is used for distributing routes and MultiProtocol Label Switching (MPLS) is used for forwarding packets across service provider networks.

The procedures for supporting multicast in BGP/MPLS Layer 3 (L3) VPN are specified in [RFC6513]. The procedures for supporting multicast in BGP/MPLS Layer 2 (L2) VPN are specified in [RFC7117]. Throughout this document, we will use the term "L2L3VpnMCast network" to mean BGP/MPLS L2 and L3 VPN that support multicast.

L2L3VpnMCast networks use various transport mechanisms for forwarding a packet to all or a subset of Provider Edge routers (PEs) across service provider networks. These transport mechanisms are abstracted as provider tunnels (P-tunnels). The type of a P-tunnel indicates the type of the tunneling technology used to establish the P-tunnel. The syntax and semantics of a Tunnel identifier is determined by the corresponding P-tunnel type [RFC6514]. P-tunnel type and P-tunnel identifier together identify a P-tunnel.

A BGP attribute that specifies information of a P-tunnel is called Provider Multicast Service Interface (PMSI) tunnel attribute. The PMSI tunnel attribute is advertised/received by PEs in BGP auto-discovery (A-D) routes. [RFC6514] defines the format of a PMSI tunnel attribute. P-tunnel type and the P-tunnel identifier are included in the corresponding PMSI tunnel attribute.

This document describes textual conventions (TCs) and common managed objects (MOs) which will be used by other Management Information Base (MIB) modules for monitoring and/or configuring L2L3VpnMCast networks.

This document defines two TCs to represent

- (a) the type of a P-tunnel and
- (b) the identifier of a P-tunnel

respectively.

The document also defines MOs that will provide the information contained in a PMSI tunnel attribute and corresponding P-tunnel information.

1.1. Terminology

This document adopts the definitions, acronyms and mechanisms described in [RFC6513] [RFC6514] [RFC7117] and other documents that they refer to. Familiarity with Multicast, MPLS, Layer 3 VPN, Multicast VPN concepts and/or mechanisms is assumed. Some terms specifically related to this document are explained below.

"Provider Multicast Service Interface (PMSI)" [RFC6513] is a conceptual interface instantiated by a P-tunnel, a transport mechanism used to deliver multicast traffic. A PE uses it to send customer multicast traffic to all or some PEs in the same VPN.

There are two kinds of PMSIs: "Inclusive PMSI (I-PMSI)" and "Selective PMSI (S-PMSI)" [RFC6513]. An I-PMSI is a PMSI that enables a PE attached to a particular Multicast VPN to transmit a message to all PEs in the same VPN. An S-PMSI is a PMSI that enables a PE attached to a particular Multicast VPN to transmit a message to some of the PEs in the same VPN.

Throughout this document, we will use the term "PMSI" to refer both "I-PMSI" and "S-PMSI."

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.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Summary of MIB Modules

This document defines two MIB modules: L2L3-VPN-MULTICAST-TC-MIB and L2L3-VPN-MULTICAST-MIB.

- o L2L3-VPN-MULTICAST-TC-MIB contains two Textual Conventions: L2L3VpnMcastProviderTunnelType and L2L3VpnMcastProviderTunnelId. L2L3VpnMcastProviderTunnelType provides an enumeration of the P-tunnel types. L2L3VpnMcastProviderTunnelId represents an identifier of a P-tunnel.
- o L2L3-VPN-MULTICAST-MIB defines a table l2l3VpnMcastPmsiTunnelAttributeTable. An entry in this table corresponds to the attribute information of a specific P-tunnel on a PE router. Entries in this table will be used by other MIB modules for monitoring and/or configuring L2L3VpnMcast network. The table index uniquely identifies a P-tunnel. It is composed of a type and identifier of a P-tunnel. The table may also be used in conjunction with other MIBs, such as MPLS Traffic Engineering MIB (MPLS-TE-STD-MIB) [RFC3812], to obtain further information about a P-tunnel. It may also be used in conjunction with the Interfaces Group MIB (IF-MIB) [RFC2863] to obtain further information about the interface corresponding to a P-tunnel.

4. Definitions

4.1. L2L3-VPN-MULTICAST-TC-MIB Object Definitions

```
L2L3-VPN-MULTICAST-TC-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, mib-2
        FROM SNMPv2-SMI
```

```
-- [RFC2578]
```


TEXTUAL-CONVENTION
FROM SNMPv2-TC;

-- [RFC2579]

12L3VpnMcastTCMIB MODULE-IDENTITY

LAST-UPDATED "201809071200Z" -- 7th September, 2018

ORGANIZATION "IETF BESS Working Group."

CONTACT-INFO

" Zhaohui Zhang
Juniper Networks, Inc.
10 Technology Park Drive
Westford, MA 01886
USA
Email: zzhang@juniper.net

Hiroshi Tsunoda
Tohoku Institute of Technology
35-1, Yagiyama Kasumi-cho
Taihaku-ku, Sendai, 982-8577
Japan
Email: tsuno@m.ieice.org

Comments and discussion to bess@ietf.org

"

DESCRIPTION

"This MIB module specifies textual conventions for
Border Gateway Protocol/MultiProtocol Label
Switching Layer 2 and Layer 3 Virtual Private Networks
that support multicast (L2L3VpnMCast networks).

Copyright (C) The Internet Society (2018).

"

-- Revision history.

REVISION "201809071200Z" -- 7th September, 2018

DESCRIPTION

"Initial version, published as RFC XXXX."

-- RFC Ed.: replace XXXX with actual RFC number and remove this note

::= { mib-2 AAAA }

-- IANA Reg.: Please assign a value for "AAAA" under the
-- 'mib-2' subtree and record the assignment in the SMI
-- Numbers registry.

-- RFC Ed.: When the above assignment has been made, please
-- remove the above note

```
-- replace "AAAA" here with the assigned value and
-- remove this note.
```

```
-- Textual convention
```

```
L2L3VpnMcastProviderTunnelType ::= TEXTUAL-CONVENTION
```

```
    STATUS          current
```

```
    DESCRIPTION
```

```
        "This textual convention enumerates values
        representing the type of a provider tunnel (P-tunnel)
        used for L2L3VpnMCast networks.
        These labeled numbers are aligned with the definition
        of Tunnel types in Section 5 of [RFC6514] and
        Section 14.1 of [RFC7524].
```

```
        The enumerated values and the corresponding P-tunnel types
        are as follows:
```

noTunnelInfo	(0) : no tunnel information present	[RFC6514]
rsvpP2mp	(1) : RSVP-TE P2MP LSP	[RFC4875]
ldpP2mp	(2) : mLDP P2MP LSP	[RFC6388]
pimSsm	(3) : PIM-SSM Tree	[RFC7761]
pimAsm	(4) : PIM-SM Tree	[RFC7761]
pimBidir	(5) : BIDIR-PIM Tree	[RFC5015]
ingressReplication	(6) : Ingress Replication	[RFC6513]
ldpMp2mp	(7) : mLDP MP2MP LSP	[RFC6388]
transportTunnel	(8) : Transport Tunnel	[RFC7524]

```
        These numbers are registered at IANA.
        A current list of assignments can be found at
        <https://www.iana.org/assignments/bgp-parameters/
        bgp-parameters.xhtml#pmsi-tunnel-types>.
```

```
"
```

```
REFERENCE
```

```
    "RFC4875
    RFC5015
    RFC6388
    RFC6513
    RFC6514, Section 5
    RFC7524, Section 14.1
    RFC7761
"
```

```

SYNTAX          INTEGER
{
    noTunnelInfo      (0),
    rsvpP2mp          (1),
    ldpP2mp           (2),
    pimSsm            (3),
    pimAsm            (4),
    pimBidir          (5),
    ingressReplication (6),
    ldpMp2mp          (7),
    transportTunnel   (8)
}

```

L2L3VpnMcastProviderTunnelId ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"This textual convention represents the tunnel identifier of a P-tunnel.

The size of the identifier depends on the address family (IPv4 or IPv6) and the value of the corresponding L2L3VpnMcastProviderTunnelType object.

The corresponding L2L3VpnMcastProviderTunnelType object represents the type of the tunneling technology used to establish the P-tunnel.

The size of the identifier for each tunneling technology is summarized below.

L2L3VpnMcastProviderTunnelType (tunneling technology)		Size (in octets)	
		IPv4	IPv6
noTunnelInfo	(No tunnel information)	0	0
rsvpP2mp	(RSVP-TE P2MP LSP)	12	24
ldpP2mp	(mLDP P2MP LSP)	17	29
pimSsm	(PIM-SSM Tree)	8	32
pimAsm	(PIM-SM Tree)	8	32
pimBidir	(BIDIR-PIM Tree)	8	32
ingressReplication	(Ingress Replication)	4	16
ldpMp2mp	(mLDP MP2MP LSP)	17	29
transportTunnel	(Transport Tunnel)	8	32

Tunnel type is set to 'No tunnel information present' when the PMSI Tunnel attribute carries no tunnel information (there is no Tunnel Identifier).

The value of the corresponding L2L3VpnMcastProviderTunnelId object will be a string of length zero.

For tunnel type `rsvpP2mp(1)`, the corresponding Tunnel Identifier is composed of Extended Tunnel ID (4 octets in IPv4, 16 octets in IPv6), Reserved (2 octets), Tunnel ID (2 octets), and P2MP ID (4 octets).
The size of the corresponding `L2L3VpnMcastProviderTunnelId` object will be 12 octets in IPv4 and 24 octets in IPv6.

For tunnel type `ldpP2mp(2)`, the corresponding Tunnel Identifier is the P2MP Forwarding Equivalence Class (FEC) Element [RFC6388]. The size of the corresponding `L2L3VpnMcastProviderTunnelId` object will be 17 octets in IPv4 and 29 octets in IPv6.

For tunnel type `pimSsm(3)`, `PimAsm(4)`, and `PimBidir(5)`, the corresponding Tunnel Identifier is composed of the source IP address and the group IP address.
The size of the corresponding `L2L3VpnMcastProviderTunnelId` object will be 8 octets in IPv4 and 32 octets in IPv6.

For tunnel type `ingressReplication(6)`, the Tunnel Identifier is the unicast tunnel endpoint IP address of the local PE.
The size of the corresponding `L2L3VpnMcastProviderTunnelId` object will be 4 octets in IPv4 and 16 octets in IPv6.

For tunnel type `ldpMp2mp(7)`, the Tunnel Identifier is MP2MP FEC Element [RFC6388].
The size of the corresponding `L2L3VpnMcastProviderTunnelId` object will be 17 octets in IPv4 and 29 octets in IPv6.

For tunnel type `transportTunnel(8)`, the Tunnel Identifier is a tuple of Source PE Address and Local Number, which is a number that is unique to the Source PE [RFC7524]. Both Source PE Address and Local Number are 4 octets in IPv4 and 16 octets in IPv6.
The size of the corresponding `L2L3VpnMcastProviderTunnelId` object will be 8 octets in IPv4 and 32 octets in IPv6.

"

REFERENCE

"RFC6514, Section 5
RFC4875, Section 19.1
RFC6388, Section 2.2 and 3.2
RFC7524, Section 14.1
"

SYNTAX OCTET STRING (SIZE (0|4|8|12|16|17|24|29|32))

END

4.2. L2L3-VPN-MULTICAST-MIB Object Definitions

```
L2L3-VPN-MULTICAST-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, mib-2, zeroDotZero
        FROM SNMPv2-SMI                                -- [RFC2578]
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF                                -- [RFC2580]
    RowPointer
        FROM SNMPv2-TC                                  -- [RFC2579]
    MplsLabel
        FROM MPLS-TC-STD-MIB                            -- [RFC3811]
    L2L3VpnMcastProviderTunnelType,
    L2L3VpnMcastProviderTunnelId
        FROM L2L3-VPN-MULTICAST-TC-MIB;                -- [RFCXXXX]

-- RFC Ed.: replace XXXX with actual RFC number and remove this note

12L3VpnMcastMIB MODULE-IDENTITY
    LAST-UPDATED "201809071200Z" -- 7th September, 2018
    ORGANIZATION "IETF BESS Working Group."
    CONTACT-INFO
        "
            Zhaohui Zhang
            Juniper Networks, Inc.
            10 Technology Park Drive
            Westford, MA 01886
            USA
            Email: zzhang@juniper.net

            Hiroshi Tsunoda
            Tohoku Institute of Technology
            35-1, Yagiyama Kasumi-cho
            Taihaku-ku, Sendai, 982-8577
            Japan
            Email: tsuno@m.ieice.org

            Comments and discussion to bess@ietf.org
        "
```

DESCRIPTION

"This MIB module defines a table representing the attribute information of the provider tunnels (P-tunnels) on a PE router. This MIB module will be used by other MIB modules designed for monitoring and/or configuring Border Gateway Protocol/MultiProtocol Label Switching Layer 2 and Layer 3 Virtual Private Network that support multicast (L2L3VpnMCast network). Copyright (C) The Internet Society (2018)."

-- Revision history.

REVISION "201809071200Z" -- 7th September, 2018

DESCRIPTION

"Initial version, published as RFC XXXX."

-- RFC Ed.: replace XXXX with actual RFC number and remove this note

::= { mib-2 BBBB }

-- IANA Reg.: Please assign a value for "BBBB" under the
-- 'mib-2' subtree and record the assignment in the SMI
-- Numbers registry.

-- RFC Ed.: When the above assignment has been made, please
-- remove the above note
-- replace "BBBB" here with the assigned value and
-- remove this note.

-- Top level components of this MIB.

12L3VpnMcastStates OBJECT IDENTIFIER
::= { 12L3VpnMcastMIB 1 }

12L3VpnMcastConformance OBJECT IDENTIFIER
::= { 12L3VpnMcastMIB 2 }

-- tables, scalars, conformance information
-- Table of PMSI Tunnel Attributes

12L3VpnMcastPmsiTunnelAttributeTable OBJECT-TYPE
SYNTAX SEQUENCE OF L2L3VpnMcastPmsiTunnelAttributeEntry
MAX-ACCESS not-accessible
STATUS current

DESCRIPTION

"An entry in this table corresponds to the attribute information of a specific P-tunnel on a PE router. A part of attributes correspond to fields in a Provider Multicast Service Interface (PMSI) Tunnel attribute advertised and received by a PE router. The entries will be referred to by other MIB modules for monitoring and/or configuring L2L3VpnMcast networks."

REFERENCE

"RFC6514, Section 5"

::= { l2L3VpnMcastStates 1 }

l2L3VpnMcastPmsiTunnelAttributeEntry OBJECT-TYPE

SYNTAX L2L3VpnMcastPmsiTunnelAttributeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual row corresponding to a specific P-tunnel on this router."

REFERENCE

"RFC6514, Section 5"

INDEX {
 l2L3VpnMcastPmsiTunnelAttributeType,
 l2L3VpnMcastPmsiTunnelAttributeId
}

::= { l2L3VpnMcastPmsiTunnelAttributeTable 1 }

l2L3VpnMcastPmsiTunnelAttributeEntry ::=

SEQUENCE {
 l2L3VpnMcastPmsiTunnelAttributeType
 L2L3VpnMcastProviderTunnelType,
 l2L3VpnMcastPmsiTunnelAttributeId
 L2L3VpnMcastProviderTunnelId,
 l2L3VpnMcastPmsiTunnelLeafInfoRequired
 INTEGER,
 l2L3VpnMcastPmsiTunnelAttributeMplsLabel
 MplsLabel,
 l2L3VpnMcastPmsiTunnelPointer
 RowPointer,
 l2L3VpnMcastPmsiTunnelIf
 RowPointer
}

l2L3VpnMcastPmsiTunnelAttributeType OBJECT-TYPE

SYNTAX L2L3VpnMcastProviderTunnelType

MAX-ACCESS not-accessible
STATUS current
DESCRIPTION

"This object indicates the type of the tunneling technology used to establish the P-tunnel corresponding to this entry.

When BGP-based PMSI signaling is used, the value of this object corresponds to the Tunnel Type field in the PMSI Tunnel attribute advertised/received in a PMSI auto-discovery (A-D) route.

"

REFERENCE

"RFC6514, Section 5"

::= { l2L3VpnMcastPmsiTunnelAttributeEntry 1 }

l2L3VpnMcastPmsiTunnelAttributeId OBJECT-TYPE

SYNTAX L2L3VpnMcastProviderTunnelId
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION

"This object represents the Tunnel Identifier field, which uniquely identifies a P-tunnel, in the PMSI Tunnel attribute of the P-tunnel corresponding to this entry.

The size of the identifier depends on the address family (IPv4 or IPv6) and the value of the corresponding l2L3VpnMcastPmsiTunnelAttributeType object i.e., the type of the tunneling technology used to establish the P-tunnel.

"

REFERENCE

"RFC6514, Section 5"

::= { l2L3VpnMcastPmsiTunnelAttributeEntry 2 }

l2L3VpnMcastPmsiTunnelLeafInfoRequired OBJECT-TYPE

SYNTAX INTEGER {
false (0),
true (1),
notAvailable (2)
}

MAX-ACCESS read-only
STATUS current
DESCRIPTION

"When the value of this object is set to 1 (true), it indicates that the PE which originated the PMSI Tunnel attribute of the P-tunnel corresponding to this entry requests receivers to originate a new Leaf A-D (Auto-Discovery) route.

A value of 0 (false) indicates that there is no such request.

When the P-tunnel does not have a corresponding PMSI tunnel attribute, the value of this object will be 2 (notAvailable).

In the case of Multicast in MPLS/BGP IP VPNs, this object represents the 'Leaf Information Required flag' [RFC6514] in the Flags field in the PMSI Tunnel attribute of the P-tunnel corresponding to this entry.

"

REFERENCE

"RFC6514, Section 5

"

::= { l2L3VpnMcastPmsiTunnelAttributeEntry 3 }

l2L3VpnMcastPmsiTunnelAttributeMplsLabel OBJECT-TYPE

SYNTAX MplsLabel

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This object represents the MPLS Label in the PMSI Tunnel attribute of the P-tunnel corresponding to this entry.

When BGP-based PMSI signaling is used, the PMSI Tunnel attribute of the P-tunnel will be advertised/received in a PMSI auto-discovery (A-D) route. The value of this object corresponds to the MPLS Label in the attribute.

When the P-tunnel does not have a PMSI tunnel attribute, the value of this object will be 0.

"

REFERENCE

"RFC6514, Section 5"

::= { l2L3VpnMcastPmsiTunnelAttributeEntry 4 }

l2L3VpnMcastPmsiTunnelPointer OBJECT-TYPE

SYNTAX RowPointer

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Details of a P-tunnel identified by l2L3VpnMcastPmsiTunnelAttributeId may be present in some other table, e.g., mplsTunnelTable [RFC3812]. This object specifies the pointer to the row that pertains to the entry in the table.

```
        If no such entry exists, the value of this object
        will be zeroDotZero.
    "
REFERENCE
    "RFC3812, Section 6.1 and Section 11"
DEFVAL      { zeroDotZero }

 ::= { l2L3VpnMcastPmsiTunnelAttributeEntry 5 }

l2L3VpnMcastPmsiTunnelIf OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "If the P-tunnel identified by
        l2L3VpnMcastPmsiTunnelAttributeId has a corresponding
        entry in ifXTable [RFC2863], this object will
        point to the row in ifXTable that pertains to the entry.
        Otherwise, the value of this object will be zeroDotZero.
        "
    REFERENCE
        "RFC2863, Section 6"
    DEFVAL      { zeroDotZero }
    ::= { l2L3VpnMcastPmsiTunnelAttributeEntry 6 }

-- Conformance Information

l2L3VpnMcastCompliances OBJECT IDENTIFIER
    ::= { l2L3VpnMcastConformance 1 }
l2L3VpnMcastGroups      OBJECT IDENTIFIER
    ::= { l2L3VpnMcastConformance 2 }

-- Compliance Statements

l2L3VpnMcastCoreCompliance MODULE-COMPLIANCE
    STATUS      current
    DESCRIPTION
        "The core compliance statement for SNMP entities
        which implement the L2L3-VPN-MULTICAST-MIB Module."
    MODULE      -- this module

    MANDATORY-GROUPS {
        l2L3VpnMcastCoreGroup
    }
    ::= { l2L3VpnMcastCompliances 1 }
```

```
12L3VpnMcastFullCompliance MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "The full compliance statement for SNMP entities
        which implement the L2L3-VPN-MULTICAST-MIB Module."
    MODULE -- this module

    MANDATORY-GROUPS {
        12L3VpnMcastCoreGroup,
        12L3VpnMcastOptionalGroup
    }
    ::= { 12L3VpnMcastCompliances 2 }

-- units of conformance

12L3VpnMcastCoreGroup OBJECT-GROUP
    OBJECTS {
        12L3VpnMcastPmsiTunnelLeafInfoRequired,
        12L3VpnMcastPmsiTunnelAttributeMplsLabel
    }
    STATUS current
    DESCRIPTION
        "Support of these objects is required."
    ::= { 12L3VpnMcastGroups 1 }

12L3VpnMcastOptionalGroup OBJECT-GROUP
    OBJECTS {
        12L3VpnMcastPmsiTunnelPointer,
        12L3VpnMcastPmsiTunnelIf
    }
    STATUS current
    DESCRIPTION
        "Support of these objects is optional."
    ::= { 12L3VpnMcastGroups 2 }

END
```

5. Security Considerations

There are no management objects defined in these MIB modules that have a MAX-ACCESS clause of read-write and/or read-create. So, if this MIB module is implemented correctly, then there is no risk that an intruder can alter or create any management objects of this MIB module via direct SNMP SET operations.

Some of the readable objects in these MIB modules (i.e., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to

control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- o the `l2L3VpnMcastPmsiTunnelAttributeTable` collectively shows the P-tunnel network topology and its performance characteristics. For instance, `l2L3VpnMcastPmsiTunnelAttributeId` in this table will contain the identifier that uniquely identifies a P-tunnel. This identifier may be composed of source and multicast group IP addresses. `l2L3VpnMcastPmsiTunnelPointer` and `l2L3VpnMcastPmsiTunnelIf` will point to the corresponding entries in other tables containing configuration and/or performance information of a P-tunnel and its interface. If an Administrator does not want to reveal this information, then these objects should be considered sensitive/vulnerable.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

6. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Name	Description	OBJECT IDENTIFIER value
-----	-----	-----
l2L3VpnMcastTCMIB	L2L3-VPN-MULTICAST-TC-MIB	{ mib-2 AAAA }
l2L3VpnMcastMIB	L2L3-VPN-MULTICAST-MIB	{ mib-2 BBBB }

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "AAAA" and "BBBB" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "AAAA" and "BBBB" (here and in the MIB module) with the assigned value and to remove this note.

7. Acknowledgement

Glenn Mansfield Keeni did the MIB Doctor review and provided valuable comments.

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>>.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <<https://www.rfc-editor.org/info/rfc2578>>.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <<https://www.rfc-editor.org/info/rfc2579>>.
- [RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <<https://www.rfc-editor.org/info/rfc2580>>.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<https://www.rfc-editor.org/info/rfc2863>>.

- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, DOI 10.17487/RFC3414, December 2002, <<https://www.rfc-editor.org/info/rfc3414>>.
- [RFC3811] Nadeau, T., Ed. and J. Cucchiara, Ed., "Definitions of Textual Conventions (TCs) for Multiprotocol Label Switching (MPLS) Management", RFC 3811, DOI 10.17487/RFC3811, June 2004, <<https://www.rfc-editor.org/info/rfc3811>>.
- [RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, DOI 10.17487/RFC3812, June 2004, <<https://www.rfc-editor.org/info/rfc3812>>.
- [RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, DOI 10.17487/RFC3826, June 2004, <<https://www.rfc-editor.org/info/rfc3826>>.
- [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>>.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <<https://www.rfc-editor.org/info/rfc5015>>.
- [RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 5591, DOI 10.17487/RFC5591, June 2009, <<https://www.rfc-editor.org/info/rfc5591>>.
- [RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, DOI 10.17487/RFC5592, June 2009, <<https://www.rfc-editor.org/info/rfc5592>>.

- [RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 6353, DOI 10.17487/RFC6353, July 2011, <<https://www.rfc-editor.org/info/rfc6353>>.
- [RFC6388] Wijnands, IJ., Ed., Minei, I., Ed., Kompella, K., and B. Thomas, "Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths", RFC 6388, DOI 10.17487/RFC6388, November 2011, <<https://www.rfc-editor.org/info/rfc6388>>.
- [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>>.
- [RFC7117] Aggarwal, R., Ed., Kamite, Y., Fang, L., Rekhter, Y., and C. Kodeboniya, "Multicast in Virtual Private LAN Service (VPLS)", RFC 7117, DOI 10.17487/RFC7117, February 2014, <<https://www.rfc-editor.org/info/rfc7117>>.
- [RFC7524] Rekhter, Y., Rosen, E., Aggarwal, R., Morin, T., Grosclaude, I., Leymann, N., and S. Saad, "Inter-Area Point-to-Multipoint (P2MP) Segmented Label Switched Paths (LSPs)", RFC 7524, DOI 10.17487/RFC7524, May 2015, <<https://www.rfc-editor.org/info/rfc7524>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [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>>.

8.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <<https://www.rfc-editor.org/info/rfc3410>>.

Authors' Addresses

Zhaohui (Jeffrey) Zhang
Juniper Networks, Inc.
10 Technology Park Drive
Westford, MA 01886
USA

Email: zzhang@juniper.net

Hiroshi Tsunoda
Tohoku Institute of Technology
35-1, Yagiyama Kasumi-cho
Taihaku-ku, Sendai 982-8577
Japan

Phone: +81-22-305-3411
Email: tsuno@m.ieice.org

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

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 15, 2021

D. Jain
Cisco
K. Patel
Arrcus, Inc
P. Brissette
Cisco
Z. Li
S. Zhuang
Huawei Technologies
X. Liu
Jabil
J. Haas
S. Esale
Juniper Networks
B. Wen
Comcast
April 13, 2021

Yang Data Model for BGP/MPLS L3 VPNs
draft-ietf-bess-l3vpn-yang-05

Abstract

This document defines a YANG data model that can be used to configure and manage BGP Layer 3 VPNs.

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].

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 October 15, 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. Introduction	2
2. Definitions and Acronyms	3
3. Design of BGP L3VPN Data Model	4
3.1. Overview	4
3.2. VRF Specific Configuration	4
3.2.1. VRF interface	4
3.2.2. Route distinguisher	4
3.2.3. Import and export route targets	4
3.2.4. Forwarding mode	5
3.2.5. Label security	5
3.2.6. Yang tree	5
3.3. BGP Specific Configuration	6
3.3.1. VPN peering	7
3.3.2. VPN prefix limits	7
3.3.3. Label Mode	7
3.3.4. ASBR options	7
3.3.5. Yang tree	7
4. BGP Yang Module	8
5. IANA Considerations	20
6. Security Considerations	20
7. Acknowledgements	20
8. References	20
Authors' Addresses	21

1. Introduction

YANG [RFC6020] 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 encodings other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

This document defines a YANG model that can be used to configure and manage BGP L3VPNs [RFC4364]. It contains VRF specific parameters as well as BGP specific parameters applicable for L3VPNs. The individual containers defined in this model contain control knobs for configuration for that purpose, as well as a few data nodes that can be used to monitor health and gather statistics.

2. Definitions and Acronyms

AF: Address Family

AS: Autonomous System

ASBR: Autonomous System Border Router

BGP: Border Gateway Protocol

CE: Customer Edge

PE: Provider Edge

L3VPN: Layer 3 VPN

NETCONF: Network Configuration Protocol

RD: Route Distinguisher

ReST: Representational State Transfer, a style of stateless interface and protocol that is generally carried over HTTP

RTFilter: Route Filter

VPN: Virtual Private Network

VRF: Virtual Routing and Forwarding

YANG: Data definition language for NETCONF

3. Design of BGP L3VPN Data Model

3.1. Overview

There are two parts of the BGP L3VPN yang data model. The first part of the model defines VRF specific parameters for L3VPN by augmenting the network-instance container defined in the network instance model [I-D.ietf-rtgwg-ni-model] and the second part of the model defines BGP specific parameters for the L3VPN by augmenting the base BGP data model defined in [I-D.ietf-idr-bgp-model] .

3.2. VRF Specific Configuration

IETF network instance model defines various ni-types, one of which is l3vpn. This provides an anchor point to add a new container l3vpn. Under this container per VPN parameters pertaining to L3VPN are added.

3.2.1. VRF interface

To associate a VRF instance with an interface, bind-network-instance config should be used. This is covered in the base network instance model [I-D.ietf-rtgwg-ni-model].

3.2.2. Route distinguisher

Route distinguisher (RD) is a unique identifier used in VPN routes to distinguish prefixes across different VPNs. RD is an 8 byte field as defined in the [RFC4364]. Where the first two bytes refer to type followed by 6 bytes of value. The format of the value is dependent on type. In the yang model, RD is defined under l3vpn container under a network-instance. Yang datatype for RD is imported from [RFC8294].

3.2.3. Import and export route targets

Route-target (RT) is an extended community used to specify the rules for importing and exporting the routes for each VRF as defined in [RFC4364]. This is applicable in the context of an address-family under the VRF. Under the l3vpn container, statements for import and export route-targets are added for ipv4 and ipv6 address family. Both import and export sets are modeled as a list of rout-targets, yang datatype for which is imported from [RFC8294]. An import rule is modeled as list of RTs or a leafref to the route policy [I-D.ietf-rtgwg-policy-model] specifying the list of RTs to be matched for importing the routes into the VRF. Similarly, an export rule is modeled as a list of RTs or a leafref the route policy [I-D.ietf-rtgwg-policy-model] specifying the list of RTs which should be

attached to routes exported from the VRF. In the case where policy is used to specify the RTs, a reference to the policy via leafref is used in this model, but actual definition of policy is outside the scope of this document. In addition, this section also defines parameters for the import from global routing table and export to global routing table, as well as route limit per VPN instance for ipv4 and ipv6 address family.

3.2.4. Forwarding mode

This configuration augments interface list under interface container under a network instance as defined in IETF network instance model [I-D.ietf-rtgwg-ni-model]. Forwarding mode configuration is required under the ASBR facing interface to enable mpls forwarding for directly connected BGP peers for inter-as option B peering.

3.2.5. Label security

For inter-as option-B peering across ASs, under the ASBR facing interface, mpls label security enables the checks for RPF label on incoming packets. Ietf-interface container is augmented to add this config.

3.2.6. Yang tree

```

module: ietf-bgp-l3vpn
module: ietf-bgp-l3vpn
augment /ni:network-instances/ni:network-instance/ni:ni-type:
  +--:(l3vpn)
    +--rw l3vpn
      +--rw rd?          bgp-rd-type
      +--ro auto-rd?     rt-types:route-distinguisher
      +--rw ipv4
        +--rw unicast
          +--rw vpn-targets
            +--rw vpn-target* [route-target]
              +--rw route-target          rt-types:route-target
              +--rw route-target-type      rt-types:route-target-type
            +--rw route-policy? -> /rt-pol:routing-policy/policy-definition/policy-definition/name
          +--rw import-from-global
            +--rw enable?                  boolean
            +--rw advertise-as-vpn?        boolean
            +--rw route-policy?            -> /rt-pol:routing-policy/policy-definition/policy-definition/name
          +--rw bgp-valid-route?           boolean
          +--rw protocol?                   enumeration
          +--rw instance?                   string
        +--rw export-to-global
          +--rw enable?                     boolean
        +--rw routing-table-limit
          +--rw routing-table-limit-number? uint32
          +--rw (routing-table-limit-action)?
            +--:(enable-alert-percent)
              +--rw alert-percent-value?    rt-types:percentage
            +--:(enable-simple-alert)
              +--rw simple-alert?           boolean
        +--rw tunnel-params
          +--rw tunnel-policy?             string
      +--rw ipv6
      ...

augment /if:interfaces/if:interface:
  +--rw forwarding-mode?      enumeration
  +--rw mpls-label-security
  +--rw rpf?                  boolean

```

3.3. BGP Specific Configuration

The BGP specific configuration for L3VPNs is defined by augmenting base BGP model [I-D.ietf-idr-bgp-model]. In particular, specific knobs are added under neighbor and address family containers to handle VPN routes and ASBR peering.

3.3.1. VPN peering

For peering between PE routers, specific VPN address family needs to be enabled under BGP container in the context of core instance. Base BGP draft [I-D.ietf-idr-bgp-model] has l3vpn address family in the list of identity refs for AFs under global and neighbor modes. The same is augmented here for additional knobs. For peering with CE routers the VRF specific BGP configurations such as neighbors and address-family are covered in base BGP config, except that such configuration will be in the context of a VRF. The instance of BGP in this case would be a separate instance in the context of vrf-root as defined in [I-D.ietf-rtgwg-ni-model].

3.3.2. VPN prefix limits

Limits for max number of VPN prefixes for a PE router is defined in the context of VPN address family under BGP. This would be the total number of prefixes in VPN table per AF in the context of BGP protocol. Route table limit for ipv4 and ipv6 address family for each VPN instance is also defined under BGP. The total prefix limit per VPN, including all the protocols is defined in the context of VRF address family under routing instance.

3.3.3. Label Mode

Label mode knobs control the label allocation behavior for VRF routes. Such as to specify Per-site, Per-vpn and Per-route label allocation. These knobs augment BGP global AF containers in the context of default routing instance.

3.3.4. ASBR options

This includes few specific knobs for ASBR peering methods illustrated in [RFC4364]. Such as route target retention on ASBRs for inter-as VPN peering across ASBRs with option-B method. Appropriate containers under BGP AF are augmented.

3.3.5. Yang tree

```
module: ietf-bgp-l3vpn
```

```

augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast:
  +--rw retain-route-targets
  |   +--rw all?          empty
  |   +--rw route-policy? -> /rt-pol:routing-policy/policy-definitions/policy-
definition/name
  +--rw vpn-prefix-limit
  +--rw prefix-limit-number? uint32
  +--rw (prefix-limit-action)?
  |   +--:(enable-alert-percent)
  |   |   +--rw alert-percent-value? rt-types:percentage
  |   |   +--rw route-unchanged?    boolean
  |   +--:(enable-simple-alert)
  |   |   +--rw simple-alert?        boolean
  ...
augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
  +--rw label-mode?          bgp-label-mode
  +--rw routing-table-limit
  +--rw routing-table-limit-number? uint32
  +--rw (routing-table-limit-action)?
  |   +--:(enable-alert-percent)
  |   |   +--rw alert-percent-value?          rt-types:percentage
  |   +--:(enable-simple-alert)
  |   |   +--rw simple-alert?                  boolean
  ...

```

4. BGP Yang Module

<CODE BEGINS> file "ietf-bgp-l3vpn@2018-04-17.yang"

```

module ietf-bgp-l3vpn {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-bgp-l3vpn";
  // replace with IANA namespace when assigned
  prefix l3vpn ;

  import ietf-network-instance {
    prefix ni;
  }

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

  import ietf-interfaces {
    prefix if;
  }

  import ietf-bgp {
    prefix bgp;
  }

```

```
}  
  
import ietf-routing-policy {  
    prefix rt-pol;  
}  
  
organization  
    "IETF BGP Enabled Services WG";  
  
contact  
    "BESS working group - bess@ietf.org";  
  
description  
    "This YANG module defines a YANG data model to configure and  
    manage BGP Layer3 VPNs. It augments the IETF bgp yang model  
    and IETF network instance model to add L3VPN specific  
    configuration and operational knobs."
```

Terms and Acronyms

AF : Address Family

AS : Autonomous System

ASBR : Autonomous Systems Border Router

BGP (bgp) : Border Gateway Protocol

CE : Customer Edge

IP (ip) : Internet Protocol

IPv4 (ipv4):Internet Protocol Version 4

IPv6 (ipv6): Internet Protocol Version 6

L3VPN: Layer 3 VPN

PE : Provider Edge

RT : Route Target

RD : Route Distinguisher

VPN : Virtual Private Network

VRF : Virtual Routing and Forwarding

```
    ";

    revision 2018-04-17 {
        description
            "Import latest revisions of ietf-network-instance" +
            "Added leafrefs to named policy defs from routing-policy model" +
            "Minor other text corrections";
        reference "";
    }

    revision 2017-10-15 {
        description
            "Removed state containers per NMDA alignment" +
            "Changes for network instance ni-type alignment" +
            "Other cleanups";
        reference "";
    }
    revision 2017-04-25 {
        description
            "Reused ietf-rotng-types.yang for vpn route-targets" +
            " and route distinguisher types";
        reference "";
    }

    revision 2016-09-09 {
        description
            "Initial revision.";
        reference
            "RFC XXXX: A YANG Data Model for BGP L3VPN config management";
    }

    // Local typedef for RD
    typedef bgp-rd-type {
        type union {
            // Either RD value as per IETF routing types or AUTO assigned value
            type rt-types:route-distinguisher;
            type enumeration {
                enum auto-assigned {
                    description "Assigned by system";
                }
            }
        }
        description "BGP RD type augmentation for configured and Auto RD value";
    }

    //Label mode

    typedef bgp-label-mode {
```

```
type enumeration {
  enum per-ce {
    description "Allocate labels per CE";
  }
  enum per-route {
    description "Allocate labels per prefix";
  }
  enum per-vpn {
    description "Allocate labels per VRF";
  }
}
description "BGP label allocation mode";
}

//RD
grouping route-distinguisher-params {
  description "Route distinguisher value as per RFC4364";
  leaf rd {
    type bgp-rd-type;
    description "Route distinguisher value as per RFC4364";
  }
  leaf auto-rd {
    type rt-types:route-distinguisher;
    config false;
    description
      "Automatically assigned RD value when rd AUTO is configured";
  }
}

//Fwding mode
grouping forwarding-mode {
  description "Forwarding mode of interface for ASBR scenario";
  leaf forwarding-mode {
    type enumeration {
      enum mpls {
        description "Forwarding mode mpls";
      }
    }
  }
  description "Forwarding mode of interface for ASBR scenario";
}

grouping label-security {
  description "Mpls label security for ASBR option B scenario";
  container mpls-label-security {
    description "MPLS label security";
    leaf rpf {
      type boolean;
    }
  }
}
```

```
        description "Enable MPLS label security rpf on interface";
    }
}

//per VPN instance table limit under BGP
grouping vpn-pfx-limit {
    description "Per VPN instance table limit under BGP";
    container vpn-prefix-limit {
        description
            "The prefix limit config sets a limit on the maximum
            number of prefixes supported in the existing VPN
            instance, preventing the PE from importing excessive
            VPN route prefixes.
            ";
        leaf prefix-limit-number {
            type uint32 {
                range "1..4294967295";
            }
            description
                "Specifies the maximum number of prefixes supported in the
                VPN instance IPv4 or IPv6 address family.";
        }

        choice prefix-limit-action {
            description ".";
            case enable-alert-percent {
                leaf alert-percent-value {
                    type rt-types:percentage;
                    description
                        "Specifies the proportion of the alarm threshold to the
                        maximum number of prefixes.";
                }
            }
            leaf route-unchanged {
                type boolean;
                default "false";
                description
                    "Indicates that the routing table remains unchanged.
                    By default, route-unchanged is not configured. When
                    the number of prefixes in the routing table is
                    greater than the value of the parameter number,
                    routes are processed as follows:
                    (1)If route-unchanged is configured, routes in the
                    routing table remain unchanged.
                    (2)If route-unchanged is not configured, all routes
                    in the routing table are deleted and then
                    re-added.";
```

```
    }
  }
  case enable-simple-alert {
    leaf simple-alert {
      type boolean;
      default "false";
      description
        "Indicates that when the number of VPN route prefixes
        exceeds number, prefixes can still join the VPN
        routing table and alarms are displayed.";
    }
  }
}

grouping global-imports {
  description "Grouping for imports from global routing table";
  container import-from-global {
    description "Import from global routing table";
    leaf enable {
      type boolean;
      description "Enable";
    }
    leaf advertise-as-vpn {
      type boolean;
      description
        "Advertise routes imported from global table as VPN routes";
    }
    leaf route-policy {
      type leafref {
        path "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
          "rt-pol:policy-definition/rt-pol:name";
        require-instance true;
      }
      description "Route policy as a filter for importing routes.";
    }
  }

  leaf bgp-valid-route {
    type boolean;
    description
      "Enable all valid routes (including non-best paths) to be
      candidate for import";
  }

  leaf protocol {
    type enumeration {
      enum ALL {
```

```
        value "0";
        description "ALL:";
    }
    enum Direct {
        value "1";
        description "Direct:";
    }
    enum OSPF {
        value "2";
        description "OSPF:";
    }
    enum ISIS {
        value "3";
        description "ISIS:";
    }
    enum Static {
        value "4";
        description "Static:";
    }
    enum RIP {
        value "5";
        description "RIP:";
    }
    enum BGP {
        value "6";
        description "BGP:";
    }
    enum OSPFV3 {
        value "7";
        description "OSPFV3:";
    }
    enum RIPNG {
        value "8";
        description "RIPNG:";
    }
}
description
    "Specifies the protocol from which routes are imported.
    At present, In the IPv4 unicast address family view,
    the protocol can be IS-IS, static, direct and BGP.";
}

leaf instance {
    type string;
    description
        "Specifies the instance id of the protocol";
}
}
```



```
}

grouping global-exports {
  description "Grouping for exports routes to global table";
  container export-to-global {
    description "Export to global routing table";
    leaf enable {
      type boolean;
      description "Enable";
    }
  }
}

grouping route-target-params {
  description "Grouping to specify rules for route import and export";
  container vpn-targets {
    description
      "Set of route-targets to match for import and export routes
      to/from VRF";
    uses rt-types:vpn-route-targets;
    leaf route-policy {
      type leafref {
        path "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
          "rt-pol:policy-definition/rt-pol:name";
        require-instance true;
      }
      description
        "Reference to the route policy containing set of route-targets.";
    }
  }
}

grouping route-tbl-limit-params {
  description "Grouping for VPN table prefix limit config";
  leaf routing-table-limit-number {
    type uint32 {
      range "1..4294967295";
    }
    description
      "Specifies the maximum number of routes supported by a
      VPN instance. ";
  }

  choice routing-table-limit-action {
    description ".";
    case enable-alert-percent {
      leaf alert-percent-value {
        type rt-types:percentage;
      }
    }
  }
}
```

```
        description
        "Specifies the percentage of the maximum number of
        routes. When the maximum number of routes that join
        the VPN instance is up to the value
        (number*alert-percent)/100, the system prompts
        alarms. The VPN routes can be still added to the
        routing table, but after the number of routes
        reaches number, the subsequent routes are
        dropped.";
    }
}
case enable-simple-alert {
    leaf simple-alert {
        type boolean;
        description
        "Indicates that when VPN routes exceed number, routes
        can still be added into the routing table, but the
        system prompts alarms.
        However, after the total number of VPN routes and
        network public routes reaches the unicast route limit
        specified in the License, the subsequent VPN routes
        are dropped.";
    }
}
}

grouping routing-tbl-limit {
    description ".";
    container routing-table-limit {
        description
        "The routing-table limit command sets a limit on the maximum
        number of routes that the IPv4 or IPv6 address family of a
        VPN instance can support.
        By default, there is no limit on the maximum number of
        routes that the IPv4 or IPv6 address family of a VPN
        instance can support, but the total number of private
        network and public network routes on a device cannot
        exceed the allowed maximum number of unicast routes.";

        uses route-tbl-limit-params;
    }
}

// Tunnel policy parameters
grouping tunnel-params {
    description "Tunnel parameters";
    container tunnel-params {
```

```
        description "Tunnel config parameters";
        leaf tunnel-policy {
            type string;
            description
                "Tunnel policy to steer the VPN traffic into specific tunnel";
        }
    }
}

// Grouping for the L3vpn specific parameters under VRF
// (network-instance)
grouping l3vpn-vrf-params {
    description "Specify route filtering rules for import/export";
    container ipv4 {
        description
            "Specify route filtering rules for import/export";
        container unicast {
            description
                "Specify route filtering rules for import/export";
            uses route-target-params;
            uses global-imports;
            uses global-exports;
            uses routing-tbl-limit;
            uses tunnel-params;
        }
    }
    container ipv6 {
        description
            "Ipv6 address family specific rules for import/export";
        container unicast {
            description "Ipv6 unicast address family";
            uses route-target-params;
            uses global-imports;
            uses global-exports;
            uses routing-tbl-limit;
            uses tunnel-params;
        }
    }
}

grouping bgp-label-mode {
    description "MPLS/VPN label allocation mode";
    leaf label-mode {
        type bgp-label-mode;
        description "Label allocation mode";
    }
}
```

```

grouping retain-route-targets {
    description "Grouping for route target accept";
    container retain-route-targets {
        description "Control route target acceptance behavior for ASBRs";
        leaf all {
            type empty;
            description "Accept all route targets.";
        }
        leaf route-policy {
            type leafref {
                path "/rt-pol:routing-policy/rt-pol:policy-definitions/" +
                    "rt-pol:policy-definition/rt-pol:name";
                require-instance true;
            }
            description "Reference to route policy containing set of route-targets to
accept.";
        }
    }
}

//
// VRF specific parameters.
// RD and RTs and route import-export rules are added under
// network instance container in network instance model, hence
// per VRF scoped
augment "/ni:network-instances/ni:network-instance/ni:ni-type" {
    description
        "Augment network instance for per VRF L3vpn parameters";
    case l3vpn {
        container l3vpn {
            description "Configuration of L3VPN specific parameters";

            uses route-distinguisher-params;
            uses l3vpn-vrf-params ;
        }
    }
}

// bgp mpls forwarding enable required for inter-as option AB.
augment "/if:interfaces/if:interface" {
    description
        "BGP mpls forwarding mode configuration on interface for
        ASBR scenario";
    uses forwarding-mode ;
    uses label-security;
}

//
// BGP Specific Paramters

```

```
//  
  
//  
// Retain route-target for inter-as option ASBR knob.  
// vpn prefix limits  
// vpnv4/vpnv6 address-family only.  
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +  
    "bgp:afi-safi/bgp:l3vpn-ipv4-unicast" {  
    description "Retain route targets for ASBR scenario";  
    uses retain-route-targets;  
    uses vpn-pfx-limit;  
}  
  
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +  
    "bgp:afi-safi/bgp:l3vpn-ipv6-unicast" {  
    description "Retain route targets for ASBR scenario";  
    uses retain-route-targets;  
    uses vpn-pfx-limit;  
}  
  
// Label allocation mode configuration. Certain AFs only.  
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +  
    "bgp:afi-safi/bgp:ipv4-unicast" {  
    description  
        "Augment BGP global AF mode for label allocation mode  
        configuration";  
    uses bgp-label-mode ;  
    uses routing-tbl-limit;  
}  
  
augment "/bgp:bgp/bgp:global/bgp:afi-safis/" +  
    "bgp:afi-safi/bgp:ipv6-unicast" {  
    description  
        "Augment BGP global AF mode for label allocation mode  
        configuration";  
    uses bgp-label-mode ;  
    uses routing-tbl-limit;  
}  
  
// TBD Additional oper state leafs  
  
// TBD RPCs  
  
}  
  
<CODE ENDS>
```

5. IANA Considerations

6. Security Considerations

The transport protocol used for sending the BGP L3VPN data MUST support authentication and SHOULD support encryption. The data-model by itself does not create any security implications. This draft does not change any underlying security issues inherent in [I-D.ietf-rtgwg-ni-model] and [I-D.ietf-idr-bgp-model].

7. Acknowledgements

The authors would like to thank TBD for their detail reviews and comments.

8. References

- [I-D.ietf-idr-bgp-model]
Jethanandani, M., Patel, K., Hares, S., and J. Haas, "BGP YANG Model for Service Provider Networks", draft-ietf-idr-bgp-model-10 (work in progress), November 2020.
- [I-D.ietf-rtgwg-ni-model]
Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Network Instances", draft-ietf-rtgwg-ni-model-12 (work in progress), March 2018.
- [I-D.ietf-rtgwg-policy-model]
Qu, Y., Tantsura, J., Lindem, A., and X. Liu, "A YANG Data Model for Routing Policy", draft-ietf-rtgwg-policy-model-27 (work in progress), January 2021.
- [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>>.
- [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>>.
- [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>>.

- [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>>.
- [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>>.

Authors' Addresses

Dhanendra Jain
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dhanendra.ietf@gmail.com

Keyur Patel
Arrcus, Inc

Email: keyur@arrcus.com

Patrice Brissette
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: pbrisset@cisco.com

Zhenbin Li
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing, 100095
China

Email: lizhenbin@huawei.com

Shunwan Zhuang
Huawei Technologies
156 Beiqing Road
Beijing, 100095
China

Email: zhuangshunwan@huawei.com

Xufeng Liu
Jabil
8281 Greensboro Drive, Suite 200
McLean, VA 22102
USA

Email: Xufeng_liu@jabil.com

Jeffrey Haas
Juniper Networks

Email: jhaas@juniper.net

Santosh Esale
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: sesale@juniper.net

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 21, 2019

H. Tsunoda
Tohoku Institute of Technology
September 17, 2018

BGP/MPLS Layer 3 VPN Multicast Management Information Base
draft-ietf-bess-mvpn-mib-12

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes managed objects to configure and/or monitor Multicast communication over IP Virtual Private Networks (VPNs) supported by MultiProtocol Label Switching/Border Gateway Protocol (MPLS/BGP) on a Provider Edge router.

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 <http://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 March 21, 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 (<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	2
1.1. Terminology	2
2. The Internet-Standard Management Framework	3
3. BGP-MPLS-LAYER3-VPN-MULTICAST-MIB	4
3.1. Summary of MIB Module	4
3.2. MIB Module Definitions	5
4. Security Considerations	50
5. IANA Considerations	53
6. Acknowledgement	53
7. References	54
7.1. Normative References	54
7.2. Informative References	56
Author's Address	56

1. Introduction

[RFC6513], [RFC6514], and [RFC6625] specify procedures for supporting multicast in Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Layer 3 (IP) Virtual Private Networks (VPNs). Throughout this document, we will use the term "Multicast VPN" (MVPN) [RFC6513] to refer to a BGP/MPLS IP VPN that supports multicast.

Provider Edge routers (PEs) attaching to a particular MVPN exchange customer multicast (C-multicast) routing information with neighboring PEs. In [RFC6513], two basic methods for exchanging C-multicast routing information are defined (1) Protocol Independent Multicast (PIM) [RFC7761] and (2) BGP.

In the rest of this document we will use the term "PIM-MVPN" to refer to the case where PIM is used for exchanging C-multicast routing information, and "BGP-MVPN" to refer to the case where BGP is used for exchanging C-multicast routing information.

This document describes managed objects to configure and/or monitor MVPNs. Most of the managed objects are common to both PIM-MVPN and BGP-MVPN, and some managed objects are BGP-MVPN specific.

1.1. Terminology

This document adopts the definitions, acronyms and mechanisms described in [RFC4364], [RFC6513], and [RFC6514]. Familiarity with Multicast, MPLS, Layer 3 (L3) VPN, MVPN concepts and/or mechanisms is

assumed. Some terms specifically related to this document are explained below.

An MVPN can be realized by using various kinds of transport mechanisms for forwarding a packet to all or a subset of PEs across service provider networks. Such transport mechanisms are referred to as provider tunnels (P-tunnels).

A "Provider Multicast Service Interface" (PMSI) [RFC6513] is a conceptual interface instantiated by a P-tunnel. A PE uses a PMSI to send customer multicast traffic to all or some PEs in the same VPN.

There are two kinds of PMSI: "Inclusive PMSI" (I-PMSI) and "Selective PMSI" (S-PMSI) [RFC6513]. An I-PMSI enables a PE attached to a particular MVPN to transmit a message to all PEs in the same MVPN. An S-PMSI enables a PE to transmit a message to a selected set of PEs in the same MVPN.

As described in [RFC4382], each PE maintains one default forwarding table and zero or more "Virtual Routing and Forwarding tables" (VRFs). Throughout this document, we will use the term "multicast VRF" (MVRF) to refer to a VRF that contains multicast routing information.

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.

2. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. BGP-MPLS-LAYER3-VPN-MULTICAST-MIB

This document defines BGP-MPLS-LAYER3-VPN-MULTICAST-MIB, a MIB module for monitoring and/or configuring MVPNs on PEs. This MIB module will be used in conjunction with MPLS-L3VPN-STD-MIB [RFC4382] and IPMCAST-MIB [RFC5132].

3.1. Summary of MIB Module

BGP-MPLS-LAYER3-VPN-MULTICAST-MIB provides the following functionalities.

- o Monitoring attributes of MVPNs on a PE
- o Configuring timers and thresholds related to an MVPN on a PE
- o Notifying creation, deletion, and modification of MVRFs on a PE
- o Monitoring PMSI attributes
- o Monitoring statistics of advertisements exchanged by a PE
- o Monitoring routing information for multicast destinations
- o Monitoring next-hops for each multicast destination

To provide these functionalities, BGP-MPLS-LAYER3-VPN-MULTICAST-MIB defines following tables.

- o mvpnGenericTable

This table contains generic information about MVPNs on a PE. Each entry in this table represents an instance of an MVPN on a PE and contains generic information related to the MVPN. For each entry in this table there MUST be a corresponding VRF in MPLS-L3VPN-STD-MIB [RFC4382].

- o mvpnBgpTable

This table contains information specific to BGP-MVPNs. Each BGP-MVPN on a PE will have an entry in this table.

- o mvpnPmsiTable

This table contains managed objects representing attribute information that is common to I-PMSIs and S-PMSIs on a PE.

- o mvpnSpmsiTable

This table contains managed objects representing attribute information specific to S-PMSIs. An S-PMSI represented in this table will have a corresponding entry in mvpnPmsiTable.

- o mvpnAdvtStatsTable

This table contains statistics pertaining to I-PMSI and S-PMSI advertisements sent/received.

- o mvpnMrouteTable

This table contains multicast routing information in MVRFs on a PE.

- o mvpnMrouteNextHopTable

This table contains information on the next-hops for routing IP multicast datagrams in MVPNs on a PE.

3.2. MIB Module Definitions

```
BGP-MPLS-LAYER3-VPN-MULTICAST-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```

MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE,
Counter32, Counter64, Gauge32, Unsigned32, TimeTicks,
mib-2
    FROM SNMPv2-SMI                                -- [RFC2578]

MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
    FROM SNMPv2-CONF                                -- [RFC2580]

RowPointer, TimeStamp, DateAndTime
    FROM SNMPv2-TC                                    -- [RFC2579]

InterfaceIndex, InterfaceIndexOrZero
    FROM IF-MIB                                       -- [RFC2863]

InetAddress, InetAddressType, InetAddressPrefixLength
    FROM INET-ADDRESS-MIB                            -- [RFC4001]

mplsL3VpnVrfName, MplsL3VpnRouteDistinguisher
    FROM MPLS-L3VPN-STD-MIB                          -- [RFC4382]

IANAipRouteProtocol, IANAipMRouteProtocol
    FROM IANA-RTPROTO-MIB                            -- [RTPROTO]

L2L3VpnMcastProviderTunnelType
```

```
FROM L2L3-VPN-MULTICAST-TC-MIB;          -- [RFCXXXX]

-- RFC Ed.: replace XXXX here and in the References Section
-- with the actual RFC number assigned to
-- I-D ietf-bess-l2l3-vpn-mcast-mib and remove this note.

mvpnMIB MODULE-IDENTITY
  LAST-UPDATED "201809071200Z" -- 7th September 2018 12:00:00 GMT
  ORGANIZATION "IETF BESS Working Group."
  CONTACT-INFO
    "
      Hiroshi Tsunoda
      Tohoku Institute of Technology
      35-1, Yagiyama Kasumi-cho
      Taihaku-ku, Sendai, 982-8577
      Japan
      Email: tsuno@m.ieice.org

      Comments and discussion to bess@ietf.org"

  DESCRIPTION
    "This MIB module contains managed object definitions to
    configure and/or monitor Multicast communication over IP
    Virtual Private Networks (VPNs) supported by MultiProtocol
    Label Switching/Border Gateway Protocol (MPLS/BGP) on a
    Provider Edge router (PE).
    Copyright (C) The Internet Society (2018).
    "

-- Revision history.

REVISION "201809071200Z" -- 7th September, 2018
DESCRIPTION
  "Initial version, published as RFC YYYY."

-- RFC Ed.: replace YYYY with the actual RFC number and
-- remove this note

::= { mib-2 AAAA }

-- IANA Reg.: Please assign a value for "AAAA" under the
-- 'mib-2' subtree and record the assignment in the SMI
-- Numbers registry.

-- RFC Ed.: When the above assignment has been made, please
-- remove the above note
-- replace "AAAA" here with the assigned value and
-- remove this note.
```

```
-- Top level components of this MIB module.
mvpnNotifications OBJECT IDENTIFIER ::= { mvpnMIB 0 }

-- scalars, tables
mvpnObjects          OBJECT IDENTIFIER ::= { mvpnMIB 1 }

-- conformance information
mvpnConformance     OBJECT IDENTIFIER ::= { mvpnMIB 2 }

-- mvpn Objects
mvpnScalars          OBJECT IDENTIFIER ::= { mvpnObjects 1 }

-- Scalar Objects

mvpnMvrfs OBJECT-TYPE
    SYNTAX          Gauge32
    MAX-ACCESS       read-only
    STATUS           current
    DESCRIPTION
        "The total number of Multicast Virtual Routing and
        Forwarding tables (MVRFs) that are present on
        this Provider Edge router (PE). This includes MVRFs
        for IPv4, IPv6, and mLDP C-Multicast.
        "
    ::= { mvpnScalars 1 }

mvpnV4Mvrfs OBJECT-TYPE
    SYNTAX          Gauge32
    MAX-ACCESS       read-only
    STATUS           current
    DESCRIPTION
        "The number of MVRFs for IPv4 C-Multicast on this PE.
        "
    ::= { mvpnScalars 2 }

mvpnV6Mvrfs OBJECT-TYPE
    SYNTAX          Gauge32
    MAX-ACCESS       read-only
    STATUS           current
    DESCRIPTION
        "The number of MVRFs for IPv6 C-Multicast on this PE.
        "
    ::= { mvpnScalars 3 }

mvpnMldpMvrfs OBJECT-TYPE
    SYNTAX          Gauge32
    MAX-ACCESS       read-only
    STATUS           current
```

```
DESCRIPTION
    "The number of MVRFs on this PE that use BGP for
    exchanging Multipoint Label Distribution Protocol (mLDP)
    C-Multicast routing information.
    "
 ::= { mvpnScalars 4 }

mvpnPimV4Mvrf OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of MVRFs on this PE that use Provider
        Independent Multicast (PIM) for exchanging IPv4
        C-Multicast routing information.
        "
    ::= { mvpnScalars 5 }

mvpnPimV6Mvrf OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of MVRFs on this PE that use PIM for
        exchanging IPv6 C-Multicast routing information.
        "
    ::= { mvpnScalars 6 }

mvpnBgpV4Mvrf OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of MVRFs on this PE that use BGP for
        exchanging IPv4 C-Multicast routing information.
        "
    ::= { mvpnScalars 7 }

mvpnBgpV6Mvrf OBJECT-TYPE
    SYNTAX      Gauge32
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The number of MVRFs on this PE that use BGP for
        exchanging IPv6 C-Multicast routing information.
        "
    ::= { mvpnScalars 8 }
```


mvpnSPTunnelLimit OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"The maximum number of selective provider tunnels that
this PE allows for a particular MVPN on this PE.
"

REFERENCE

"RFC6513, Section 13"

::= { mvpnScalars 9 }

mvpnBgpCmcastRouteWithdrawalTimer OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milliseconds"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"A configurable timer to control the delay
of C-multicast route withdrawal advertisements.
"

REFERENCE

"RFC6514, Section 16.1.1"

::= { mvpnScalars 10 }

mvpnBgpSrcSharedTreeJoinTimer OBJECT-TYPE

SYNTAX Unsigned32

UNITS "milliseconds"

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"A configurable timer to control the delay
of Source/Shared Tree Join C-multicast route
advertisements.
"

REFERENCE

"RFC6514, Section 16.1.2"

::= { mvpnScalars 11 }

-- Generic MVRP Information Table

mvpnGenericTable OBJECT-TYPE

SYNTAX SEQUENCE OF MvpnGenericEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual table containing generic information about MVPNs
on this PE."

```

"
 ::= { mvpnObjects 2 }

mvpnGenericEntry OBJECT-TYPE
    SYNTAX      MvpnGenericEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "A conceptual row that represents an MVPN on this PE.
        The MVPN represented by this entry will have one or more
        corresponding P-Multicast Service Interfaces (PMSIs)
        and a corresponding VRF in MPLS-L3VPN-STD-MIB [RFC4382].
        "
    INDEX {
        mplsL3VpnVrfName
    }
 ::= { mvpnGenericTable 1 }

MvpnGenericEntry ::= SEQUENCE {
    mvpnGenMvrfLastAction      INTEGER,
    mvpnGenMvrfLastActionTime  DateAndTime,
    mvpnGenMvrfCreationTime    DateAndTime,
    mvpnGenCmcastRouteProtocol INTEGER,
    mvpnGenIpmsiInfo           RowPointer,
    mvpnGenInterAsPmsiInfo     RowPointer,
    mvpnGenUmhSelection         INTEGER,
    mvpnGenCustomerSiteType    INTEGER
}

mvpnGenMvrfLastAction OBJECT-TYPE
    SYNTAX      INTEGER {
                                createdMvrf      (1),
                                deletedMvrf      (2),
                                modifiedMvrfIpmsiConfig (3),
                                modifiedMvrfSpmsiConfig (4)
                            }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "This object describes the last action pertaining
        to the MVPN represented by this entry.

        The enumerated action types and the corresponding
        descriptions are as follows:

        createdMvrf:
            MVRP was created for this MVPN on the PE.

```

```

    deletedMvrf:
        MVRF for this MVPN was deleted from the PE.
        A conceptual row in this table will never have
        mvpnGenMvrfLastAction equal to deletedMvrf,
        because in that case the row itself will not exist
        in the table.
        This value for mvpnGenMvrfLastAction is defined
        solely for use in mvpnMvrfActionChange notification.

    modifiedMvrfIpmsiConfig:
        an I-PMSI for this MVPN was configured, deleted or
        changed.

    modifiedMvrfSpmsiConfig:
        an S-PMSI for this MVPN was configured, deleted or
        changed.
    "
 ::= { mvpnGenericEntry 2 }

mvpnGenMvrfLastActionTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The timestamp when the last action, given in
        the corresponding mvpnGenMvrfLastAction object,
        was carried out.
        "
 ::= { mvpnGenericEntry 3 }

mvpnGenMvrfCreationTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The timestamp when the MVRF was created for
        the MVPN represented by this entry.
        "
 ::= { mvpnGenericEntry 4 }

mvpnGenCmcastRouteProtocol OBJECT-TYPE
    SYNTAX      INTEGER {
                                pim (1),
                                bgp (2)
                            }
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION

```

"The protocol used to signal C-multicast routing information across the provider core for the MVPN represented by this entry.

The enumerated protocols and the corresponding descriptions are as follows:

```
pim : PIM (PIM-MVPN)
bgp : BGP (BGP-MVPN)
```

"

REFERENCE

"RFC6513, Section 5"

```
::= { mvpnGenericEntry 5 }
```

mvpnGenIpmsiInfo OBJECT-TYPE

```
SYNTAX      RowPointer
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"A pointer to a conceptual row representing the corresponding I-PMSI in mvpnPmsiTable. If there is no I-PMSI for the MVPN represented by this entry, the value of this object will be zeroDotZero.

"

```
::= { mvpnGenericEntry 6 }
```

mvpnGenInterAsPmsiInfo OBJECT-TYPE

```
SYNTAX      RowPointer
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"A pointer to a conceptual row representing the corresponding segmented Inter-AS I-PMSI in mvpnPmsiTable. If there is no segmented Inter-AS I-PMSI for the MVPN, the value of this object will be zeroDotZero.

"

```
::= { mvpnGenericEntry 7 }
```

mvpnGenUmhSelection OBJECT-TYPE

```
SYNTAX      INTEGER {
                                highestPeAddress  (1),
                                cRootGroupHashing (2),
                                ucastUmhRoute     (3)
                                }
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"The Upstream Multicast Hop (UMH) selection method for the MVPN represented by this entry.

The enumerated methods and the corresponding descriptions are as follows:

```

    highestPeAddress  : PE with the highest address
                        (see RFC6513, Section 5.1.3)
    cRootGroupHashing : hashing based on (c-root, c-group)
    ucastUmhRoute     : per unicast route towards c-root

```

"

REFERENCE

"RFC6513, Section 5.1"

::= { mvpnGenericEntry 8 }

mvpnGenCustomerSiteType OBJECT-TYPE

```

SYNTAX          INTEGER {
                                senderReceiver (1),
                                receiverOnly   (2),
                                senderOnly     (3)
                            }

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The type of the customer site, connected to the MVPN represented by this entry.

The enumerated types and the corresponding descriptions are as follows:

```

    senderReceiver : Site is both sender and receiver
    receiverOnly   : Site is receiver-only
    senderOnly     : Site is sender-only

```

"

REFERENCE

"RFC6513, Section 2.3"

::= { mvpnGenericEntry 9 }

-- Generic BGP-MVPN table

mvpnBgpTable OBJECT-TYPE

```

SYNTAX          SEQUENCE OF MvpnBgpEntry

```

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual table that supplements mvpnGenericTable with BGP-MVPN specific information for BGP-MVPNs on this PE.

"

```

 ::= { mvpnObjects 3 }

mvpnBgpEntry OBJECT-TYPE
    SYNTAX          MvpnBgpEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A conceptual row corresponding to a BGP-MVPN on this PE."
    INDEX {
        mplsL3VpnVrfName
    }
 ::= { mvpnBgpTable 1 }

MvpnBgpEntry ::= SEQUENCE {
    mvpnBgpMode                INTEGER,
    mvpnBgpVrfRouteImportExtendedCommunity MplsL3VpnRouteDistinguisher,
    mvpnBgpSrcASEExtendedCommunity Unsigned32,
    mvpnBgpMsgRateLimit        Unsigned32,
    mvpnBgpMaxSpmsiAdRoutes    Unsigned32,
    mvpnBgpMaxSpmsiAdRouteFreq Unsigned32,
    mvpnBgpMaxSrcActiveAdRoutes Unsigned32,
    mvpnBgpMaxSrcActiveAdRouteFreq Unsigned32
}

mvpnBgpMode OBJECT-TYPE
    SYNTAX          INTEGER {
                        other      (0),
                        rptSpt     (1),
                        sptOnly    (2)
                    }
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The inter-site C-tree mode used by the BGP-MVPN
        represented by this entry.

        other      : none of the following
        rptSpt     : inter-site shared tree mode
                     (Rendezvous Point Tree (RPT) and
                     source-specific shortest-path tree (SPT))
        sptOnly    : inter-site source-only tree mode
        "
    REFERENCE
        "RFC6513, Section 9.3.1"
 ::= { mvpnBgpEntry 1 }

mvpnBgpVrfRouteImportExtendedCommunity OBJECT-TYPE

```

SYNTAX MplsL3VpnRouteDistinguisher
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The VRF Route Import Extended Community added by this PE
to unicast VPN routes that it advertises for the BGP-MVPN
corresponding to this entry.
"

REFERENCE
"RFC6514, Section 7
"

::= { mvpnBgpEntry 2 }

mvpnBgpSrcASExtendedCommunity OBJECT-TYPE

SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The Source AS Extended Community added by this PE
to the unicast VPN routes that it advertises for
the BGP-MVPN represented by this entry.
"

REFERENCE
"RFC6514, Section 6
"

::= { mvpnBgpEntry 3 }

mvpnBgpMsgRateLimit OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)
UNITS "messages per second"
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"The configurable upper bound for the rate of BGP C-multicast
routing information message exchange between this PE and other
PEs in the BGP-MVPN corresponding to this entry.
"

REFERENCE
"RFC6514, Section 17"
::= { mvpnBgpEntry 4 }

mvpnBgpMaxSpmsiAdRoutes OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)
MAX-ACCESS read-write
STATUS current
DESCRIPTION
"The configurable upper bound for the number of
S-PMSI A-D routes for the BGP-MVPN corresponding to
"

```
        this entry.
    "
REFERENCE
    "RFC6514, Section 17"
 ::= { mvpnBgpEntry 5 }

mvpnBgpMaxSpmsiAdRouteFreq OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    UNITS       "routes per second"
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The configurable upper bound for the frequency of
         S-PMSI A-D route generation for the BGP-MVPN corresponding
         to this entry.
        "
REFERENCE
    "RFC6514, Section 17"
 ::= { mvpnBgpEntry 6 }

mvpnBgpMaxSrcActiveAdRoutes OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The configurable upper bound for the number of
         Source Active A-D routes for the BGP-MVPN corresponding
         to this entry.
        "
REFERENCE
    "RFC6514, Section 17"
 ::= { mvpnBgpEntry 7 }

mvpnBgpMaxSrcActiveAdRouteFreq OBJECT-TYPE
    SYNTAX      Unsigned32 (0..4294967295)
    UNITS       "routes per second"
    MAX-ACCESS   read-write
    STATUS      current
    DESCRIPTION
        "The configurable upper bound for the frequency of Source
         Active A-D route generation for the BGP-MVPN corresponding
         to this entry.
        "
REFERENCE
    "RFC6514, Section 17"
 ::= { mvpnBgpEntry 8 }

-- Table of PMSI information
```



```

mvpnPmsiTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MvpnPmsiEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual table containing information related
        to PMSIs on this PE.
        "
    ::= { mvpnObjects 4 }

mvpnPmsiEntry OBJECT-TYPE
    SYNTAX      MvpnPmsiEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row corresponding to a
        PMSI on this PE.
        "
    INDEX        {
                    mvpnPmsiTunnelIfIndex
                }
    ::= { mvpnPmsiTable 1 }

MvpnPmsiEntry ::= SEQUENCE {
    mvpnPmsiTunnelIfIndex      InterfaceIndex,
    mvpnPmsiRD                 MplsL3VpnRouteDistinguisher,
    mvpnPmsiTunnelType         L2L3VpnMcastProviderTunnelType,
    mvpnPmsiTunnelAttribute    RowPointer,
    mvpnPmsiTunnelPimGroupAddrType InetAddressType,
    mvpnPmsiTunnelPimGroupAddr InetAddress,
    mvpnPmsiEncapsulationType  INTEGER
}

mvpnPmsiTunnelIfIndex OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A unique value for this conceptual row. Its value
        will be the same as that of the ifIndex object instance
        for the corresponding PMSI in ifTable.
        "
    REFERENCE
        "RFC2863 Sec. 3.1.5
        "
    ::= { mvpnPmsiEntry 1 }

mvpnPmsiRD OBJECT-TYPE

```

SYNTAX MplsL3VpnRouteDistinguisher
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The Route Distinguisher for this I-PMSI."
 ::= { mvpnPmsiEntry 3 }

mvpnPmsiTunnelType OBJECT-TYPE

SYNTAX L2L3VpnMcastProviderTunnelType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The type of tunnel used to
 instantiate the PMSI corresponding to this entry."
"
REFERENCE
"RFC6513, Sec. 2.6
"
 ::= { mvpnPmsiEntry 4 }

mvpnPmsiTunnelAttribute OBJECT-TYPE

SYNTAX RowPointer
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"A pointer to a conceptual row representing
 the P-tunnel used by the PMSI in
 l2L3VpnMcastPmsiTunnelAttributeTable."
"
 ::= { mvpnPmsiEntry 5 }

mvpnPmsiTunnelPimGroupAddrType OBJECT-TYPE

SYNTAX InetAddressType
MAX-ACCESS read-only
STATUS current
DESCRIPTION
"The InetAddressType of the mvpnPmsiTunnelPimGroupAddr object
 that follows.
 When the PMSI corresponding to this entry does not use
 the PIM provider tunnel, i.e.,
 the value of mvpnPmsiTunnelType is not one of
 pimSsm(3), pimAsm(4), or pimBidir(5),
 this object should be unknown(0)."
"
 ::= { mvpnPmsiEntry 6 }

mvpnPmsiTunnelPimGroupAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The tunnel address which is used by the PMSI corresponding to this entry. When the PMSI corresponding to this entry does not use PIM provider tunnel, i.e., the value of mvpnPmsiTunnelType is not one of pimSsm(3), pimAsm(4), or pimBidir(5), this object should be a zero-length octet string."

::= { mvpnPmsiEntry 7 }

mvpnPmsiEncapsulationType OBJECT-TYPE

SYNTAX INTEGER {
 greIp (1),
 ipIp (2),
 mpls (3)
 }

MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"The encapsulation type used for sending packets through the PMSI corresponding to this entry.

The enumerated encapsulation types and the corresponding descriptions are as follows:

greIp : GRE (Generic Routing Encapsulation)
 encapsulation [RFC2784]
 ipIp : IP-in-IP encapsulation [RFC2003]
 mpls : MPLS encapsulation [RFC3032]

"

REFERENCE

"RFC2003
 RFC2784
 RFC3032
 RFC6513, Sec. 12.1"

::= { mvpnPmsiEntry 8 }

-- Table of S-PMSI specific information

mvpnSpmsiTable OBJECT-TYPE

SYNTAX SEQUENCE OF MvpnSpmsiEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

```

    "A conceptual table containing information related
    to S-PMSIs on this PE.
    This table stores only S-PMSI specific attribute
    information. Generic PMSI attribute information of
    S-PMSIs is stored in mvpnPmsiTable.
    "
 ::= { mvpnObjects 5 }

mvpnSpmsiEntry OBJECT-TYPE
    SYNTAX      MvpnSpmsiEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row corresponding to an S-PMSI on this PE.
        Implementers need to be aware that if the total number of
        octets in mplsL3VpnVrfName, mvpnSpmsiCmcastGroupAddr and
        mvpnSpmsiCmcastSourceAddr exceeds 113, the OIDs of column
        instances in this row will have more than 128 sub-identifiers
        and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3.
        "
    INDEX
        {
            mplsL3VpnVrfName,
            mvpnSpmsiCmcastGroupAddrType,
            mvpnSpmsiCmcastGroupAddr,
            mvpnSpmsiCmcastGroupPrefixLen,
            mvpnSpmsiCmcastSourceAddrType,
            mvpnSpmsiCmcastSourceAddr,
            mvpnSpmsiCmcastSourcePrefixLen
        }
 ::= { mvpnSpmsiTable 1 }

MvpnSpmsiEntry ::= SEQUENCE {
    mvpnSpmsiCmcastGroupAddrType  InetAddressType,
    mvpnSpmsiCmcastGroupAddr      InetAddress,
    mvpnSpmsiCmcastGroupPrefixLen  InetAddressPrefixLength,
    mvpnSpmsiCmcastSourceAddrType  InetAddressType,
    mvpnSpmsiCmcastSourceAddr      InetAddress,
    mvpnSpmsiCmcastSourcePrefixLen  InetAddressPrefixLength,
    mvpnSpmsiPmsiPointer           RowPointer
}

mvpnSpmsiCmcastGroupAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "The InetAddressType of the mvpnSpmsiCmcastGroupAddr object
        that follows.

```

```
"
 ::= { mvpnSpmsiEntry 1 }

mvpnSpmsiCmcastGroupAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The group address of the C-flow assigned to the
        S-PMSI corresponding to this entry."
    REFERENCE
        "RFC6513, Sec. 3.1"
    ::= { mvpnSpmsiEntry 2 }

mvpnSpmsiCmcastGroupPrefixLen OBJECT-TYPE
    SYNTAX      InetAddressPrefixLength
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The prefix length of the corresponding
        mvpnSpmsiCmcastGroupAddr object."
    ::= { mvpnSpmsiEntry 3 }

mvpnSpmsiCmcastSourceAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The InetAddressType of the mvpnSpmsiCmcastSourceAddr object
        that follows."
    ::= { mvpnSpmsiEntry 4 }

mvpnSpmsiCmcastSourceAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The source address of the C-flow assigned to the
        S-PMSI corresponding to this entry."
    ::= { mvpnSpmsiEntry 5 }

mvpnSpmsiCmcastSourcePrefixLen OBJECT-TYPE
    SYNTAX      InetAddressPrefixLength
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
```

```

        "The prefix length of the corresponding
        mvpnSpmsiCmcastSourceAddr object.
        "
 ::= { mvpnSpmsiEntry 6 }

mvpnSpmsiPmsiPointer OBJECT-TYPE
    SYNTAX      RowPointer
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "A pointer to a conceptual row representing
        generic information of this S-PMSI in mvpnPmsiTable.
        "
 ::= { mvpnSpmsiEntry 7 }

-- Table of statistics pertaining to
-- advertisements sent/received

mvpnAdvtStatsTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF MvpnAdvtStatsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual table containing statistics pertaining to
        I-PMSI and S-PMSI advertisements sent/received by this PE.
        "
 ::= { mvpnObjects 6 }

mvpnAdvtStatsEntry OBJECT-TYPE
    SYNTAX      MvpnAdvtStatsEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A conceptual row corresponding to statistics
        pertaining to advertisements sent/received
        for a particular MVPN on this PE.

        Implementers need to be aware that if the total number of
        octets in mplsL3VpnVrfName and mvpnAdvtPeerAddr exceeds 115,
        then OIDs of column instances in this row will have more than
        128 sub-identifiers and cannot be accessed using SNMPv1,
        SNMPv2c, or SNMPv3.
        "
    INDEX      {
        mplsL3VpnVrfName,
        mvpnAdvtType,
        mvpnAdvtPeerAddrType,
        mvpnAdvtPeerAddr
    }

```

```

    }
    ::= { mvpnAdvtStatsTable 1 }

MvpnAdvtStatsEntry ::= SEQUENCE {
    mvpnAdvtType                INTEGER,
    mvpnAdvtPeerAddrType        InetAddressType,
    mvpnAdvtPeerAddr            InetAddress,
    mvpnAdvtSent                 Counter32,
    mvpnAdvtReceived            Counter32,
    mvpnAdvtReceivedError       Counter32,
    mvpnAdvtReceivedMalformedTunnelType Counter32,
    mvpnAdvtReceivedMalformedTunnelId Counter32,
    mvpnAdvtLastSentTime        DateAndTime,
    mvpnAdvtLastReceivedTime    DateAndTime,
    mvpnAdvtCounterDiscontinuityTime TimeStamp
}

mvpnAdvtType OBJECT-TYPE
    SYNTAX          INTEGER {
                                intraAsIpmsi (0),
                                interAsIpmsi (1),
                                sPmsi        (2)
                            }
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The PMSI type.

        The enumerated PMSI types and corresponding
        descriptions are as follows:

            intraAsIpmsi : Intra-AS Inclusive PMSI
            interAsIpmsi : Inter-AS Inclusive PMSI
            sPmsi        : Selective PMSI
        "
    REFERENCE
        "RFC6513, Sec. 3.2.1"
    ::= { mvpnAdvtStatsEntry 1 }

mvpnAdvtPeerAddrType OBJECT-TYPE
    SYNTAX          InetAddressType
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The InetAddressType of the mvpnAdvtPeerAddr object
        that follows.
        "

```

```
 ::= { mvpnAdvtStatsEntry 2 }

mvpnAdvtPeerAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The address of a peer PE that exchanges advertisement with
        this PE.
        "
    ::= { mvpnAdvtStatsEntry 3 }

mvpnAdvtSent OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of advertisements successfully
        sent to the peer PE specified by the corresponding
        mvpnAdvtPeerAddr.

        Discontinuities in the value of this counter can
        occur at re-initialization of the management system,
        and at other times as indicated by the corresponding
        mvpnAdvtCounterDiscontinuityTime object.
        "
    ::= { mvpnAdvtStatsEntry 4 }

mvpnAdvtReceived OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The number of advertisements received from the peer PE
        specified by the corresponding mvpnAdvtPeerAddr object.
        This includes advertisements that were discarded.

        Discontinuities in the value of this counter can
        occur at re-initialization of the management system,
        and at other times as indicated by the corresponding
        mvpnAdvtCounterDiscontinuityTime object.
        "
    ::= { mvpnAdvtStatsEntry 5 }

mvpnAdvtReceivedError OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
```


DESCRIPTION

"The total number of advertisements received from a peer PE, specified by the corresponding mvpnAdvtPeerAddr object, that were rejected due to error(s) in the advertisement. The value of this object includes the error cases counted in the corresponding mvpnAdvtReceivedMalformedTunnelType and mvpnAdvtReceivedMalformedTunnelId objects.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnAdvtCounterDiscontinuityTime object.

"

::= { mvpnAdvtStatsEntry 6 }

mvpnAdvtReceivedMalformedTunnelType OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of advertisements received from the peer PE specified by the corresponding mvpnAdvtPeerAddr object, that were rejected due to malformed Tunnel Type in the PMSI Tunnel attribute.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnAdvtCounterDiscontinuityTime object.

"

REFERENCE

"RFC6514 Sec.5"

::= { mvpnAdvtStatsEntry 7 }

mvpnAdvtReceivedMalformedTunnelId OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The total number of advertisements received from the peer PE specified by the corresponding mvpnAdvtPeerAddr object, that were rejected due to malformed Tunnel Identifier in the PMSI Tunnel attribute.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnAdvtCounterDiscontinuityTime object.

```
"
REFERENCE
  "RFC6514 Sec.5"
 ::= { mvpnAdvtStatsEntry 8 }

mvpnAdvtLastSentTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The timestamp when the last advertisement
         was successfully sent by this PE.
         If no advertisement has been sent since the
         last re-initialization of this PE, then this
         object will have a zero-length string."
    ::= { mvpnAdvtStatsEntry 9 }

mvpnAdvtLastReceivedTime OBJECT-TYPE
    SYNTAX      DateAndTime
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The timestamp when the last advertisement
         was successfully received from the peer PE specified
         by the corresponding mvpnAdvtPeerAddr object and
         processed by this PE.
         If no advertisement has been received since the
         last re-initialization of this PE, then this
         object will have a zero-length string."
    ::= { mvpnAdvtStatsEntry 10 }

mvpnAdvtCounterDiscontinuityTime OBJECT-TYPE
    SYNTAX      TimeStamp
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "The value of sysUpTime on the most recent occasion
         at which any one or more of this application's
         counters, viz., counters with OID prefix
         'mvpnAdvtSent' or
         'mvpnAdvtReceived' or
         'mvpnAdvtReceivedError' or
         'mvpnAdvtReceivedMalformedTunnelType' or
         'mvpnAdvtReceivedMalformedTunnelId' suffered a
         discontinuity.
         If no such discontinuities have occurred since the
```

```

        last re-initialization of the local management
        subsystem, then this object will have a zero value.
    "
 ::= { mvpnAdvtStatsEntry 11 }

-- Table of multicast routes in an MVPN

mvpnMrouteTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF MvpnMrouteEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A conceptual table containing multicast routing information
        corresponding to the MVRFs present on the PE.
    "
 ::= { mvpnObjects 7 }

mvpnMrouteEntry OBJECT-TYPE
    SYNTAX          MvpnMrouteEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "A conceptual row corresponding to a route for IP datagrams
        from a particular source and addressed to a particular IP
        multicast group address.

        Implementers need to be aware that if the total number of
        octets in mplsL3VpnVrfName, mvpnMrouteCmcastGroupAddr and
        mvpnMrouteCmcastSourceAddrs exceeds 113, the OIDs of column
        instances in this row will have more than 128 sub-identifiers
        and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3.
    "
    INDEX {
        mplsL3VpnVrfName,
        mvpnMrouteCmcastGroupAddrType,
        mvpnMrouteCmcastGroupAddr,
        mvpnMrouteCmcastGroupPrefixLength,
        mvpnMrouteCmcastSourceAddrType,
        mvpnMrouteCmcastSourceAddrs,
        mvpnMrouteCmcastSourcePrefixLength
    }
 ::= { mvpnMrouteTable 1 }

MvpnMrouteEntry ::= SEQUENCE {
    mvpnMrouteCmcastGroupAddrType      InetAddressType,
    mvpnMrouteCmcastGroupAddr          InetAddress,
    mvpnMrouteCmcastGroupPrefixLength  InetAddressPrefixLength,
    mvpnMrouteCmcastSourceAddrType     InetAddressType,

```

```

    mvpnMrouteCmcastSourceAddr      InetAddress,
    mvpnMrouteCmcastSourcePrefixLength InetAddressPrefixLength,
    mvpnMrouteUpstreamNeighborAddrType InetAddressType,
    mvpnMrouteUpstreamNeighborAddr   InetAddress,
    mvpnMrouteInIfIndex               InterfaceIndexOrZero,
    mvpnMrouteExpiryTime              TimeTicks,
    mvpnMrouteProtocol                IANAipMRouteProtocol,
    mvpnMrouteRtProtocol              IANAipRouteProtocol,
    mvpnMrouteRtAddrType              InetAddressType,
    mvpnMrouteRtAddr                  InetAddress,
    mvpnMrouteRtPrefixLength           InetAddressPrefixLength,
    mvpnMrouteRtType                   INTEGER,
    mvpnMrouteOctets                   Counter64,
    mvpnMroutePkts                     Counter64,
    mvpnMrouteTtlDroppedOctets         Counter64,
    mvpnMrouteTtlDroppedPackets        Counter64,
    mvpnMrouteDroppedInOctets          Counter64,
    mvpnMrouteDroppedInPackets         Counter64,
    mvpnMroutePmsiPointer              RowPointer,
    mvpnMrouteNumberOfLocalReplication Unsigned32,
    mvpnMrouteNumberOfRemoteReplication Unsigned32,
    mvpnMrouteCounterDiscontinuityTime TimeStamp
}

mvpnMrouteCmcastGroupAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The InetAddressType of the mvpnMrouteCmcastGroupAddr object
         that follows.
        "
    ::= { mvpnMrouteEntry 1 }

mvpnMrouteCmcastGroupAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The IP multicast group address which, along with
         the corresponding mvpnMrouteCmcastGroupPrefixLength object,
         identifies destinations for which this entry contains
         multicast routing information.

        This address object is only significant up to
        mvpnMrouteCmcastGroupPrefixLength bits. The remaining address
        bits MUST be set to zero."

```

For addresses of type 'ipv4z' or 'ipv6z', the appended zone index is significant even though it lies beyond the prefix length. The use of these address types indicate that this forwarding state applies only within the given zone. Zone index zero is not valid in this table.

"

::= { mvpnMrouteEntry 2 }

mvpnMrouteCmcastGroupPrefixLength OBJECT-TYPE

SYNTAX InetAddressPrefixLength

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The length in bits of the mask which, along with the corresponding mvpnMrouteCmcastGroupAddr object, identifies destinations for which this entry contains multicast routing information.

If the corresponding InetAddressType is 'ipv4' or 'ipv4z', this object must be in the range 4..32.

If the corresponding InetAddressType is 'ipv6' or 'ipv6z', this object must be in the range 8..128.

"

::= { mvpnMrouteEntry 3 }

mvpnMrouteCmcastSourceAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The InetAddressType of the mvpnMrouteCmcastSourceAddrs object that follows.

A value of unknown(0) indicates a non-source-specific entry, corresponding to all sources in the group. Otherwise, the value MUST be the same as the value of mvpnMrouteCmcastGroupAddrType.

"

::= { mvpnMrouteEntry 4 }

mvpnMrouteCmcastSourceAddrs OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The network address which, along with the corresponding mvpnMrouteCmcastSourcePrefixLength object, identifies the sources for which this entry contains

multicast routing information.

This address object is only significant up to mvpnMrouteCmcastSourcePrefixLength bits.
The remaining address bits MUST be set to zero.

For addresses of type 'ipv4z' or 'ipv6z', the appended zone index is significant even though it lies beyond the prefix length. The use of these address types indicate that this source address applies only within the given zone. Zone index zero is not valid in this table.

"

::= { mvpnMrouteEntry 5 }

mvpnMrouteCmcastSourcePrefixLength OBJECT-TYPE

SYNTAX InetAddressPrefixLength

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The length in bits of the mask which, along with the corresponding mvpnMrouteCmcastSourceAddr object, identifies the sources for which this entry contains multicast routing information.

If the corresponding InetAddressType is 'ipv4' or 'ipv4z', this object must be in the range 4..32.

If the corresponding InetAddressType is 'ipv6' or 'ipv6z', this object must be in the range 8..128.

If the corresponding InetAddressType is 'unknown', this object must be zero.

"

::= { mvpnMrouteEntry 6 }

mvpnMrouteUpstreamNeighborAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The InetAddressType of the mvpnMrouteUpstreamNeighborAddr object that follows.

A value of unknown(0) indicates that the upstream neighbor is unknown, for example in BIDIR-PIM."

REFERENCE

"RFC 5015"

::= { mvpnMrouteEntry 7 }

mvpnMrouteUpstreamNeighborAddr OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The address of the upstream neighbor (for example, Reverse Path Forwarding (RPF) neighbor) from which IP datagrams from these sources represented by this entry to this multicast address are received.

"

::= { mvpnMrouteEntry 8 }

mvpnMrouteInIfIndex OBJECT-TYPE

SYNTAX InterfaceIndexOrZero

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of ifIndex for the interface on which IP datagrams sent by these sources represented by this entry to this multicast address are received.

A value 0 indicates that datagrams are not subject to an incoming interface check, but may be accepted on multiple interfaces (for example, in BIDIR-PIM).

"

REFERENCE

"RFC 5015"

::= { mvpnMrouteEntry 9 }

mvpnMrouteExpiryTime OBJECT-TYPE

SYNTAX TimeTicks

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The minimum amount of time remaining before this entry will be aged out. The value 0 indicates that the entry is not subject to aging. If the corresponding mvpnMrouteNextHopState object is pruned(1), this object represents the remaining time for the prune to expire after which the state will return to forwarding(2).

If the corresponding mvpnMrouteNextHopState object is forwarding(2), this object indicates the time after which this entry will be removed from the table.

"

::= { mvpnMrouteEntry 10 }

mvpnMrouteProtocol OBJECT-TYPE

SYNTAX IANAipMRouteProtocol

MAX-ACCESS read-only

```
STATUS      current
DESCRIPTION
    "The multicast routing protocol via which this multicast
      forwarding entry was learned.
    "
 ::= { mvpnMrouteEntry 11 }

mvpnMrouteRtProtocol OBJECT-TYPE
SYNTAX      IANAipRouteProtocol
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The routing protocol via which the route used to find the
      upstream or parent interface for this multicast forwarding
      entry was learned.
    "
 ::= { mvpnMrouteEntry 12 }

mvpnMrouteRtAddrType OBJECT-TYPE
SYNTAX      InetAddressType
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The InetAddressType of the mvpnMrouteRtAddr object
      that follows.
    "
 ::= { mvpnMrouteEntry 13 }

mvpnMrouteRtAddr OBJECT-TYPE
SYNTAX      InetAddress
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The address portion of the route used to find the upstream
      or parent interface for this multicast forwarding entry.

    This address object is only significant up to
    mvpnMrouteRtPrefixLength bits.  The remaining address bits
    MUST be set to zero.

    For addresses of type 'ipv4z' or 'ipv6z', the appended zone
    index is significant even though it lies beyond the prefix
    length.  The use of these address types indicate that this
    forwarding state applies only within the given zone.  Zone
    index zero is not valid in this table.
    "
 ::= { mvpnMrouteEntry 14 }
```



```
mvpnMrouteRtPrefixLength OBJECT-TYPE
    SYNTAX      InetAddressPrefixLength
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The length in bits of the mask associated with the route
        used to find the upstream or parent interface for this
        multicast forwarding entry.

        If the corresponding InetAddressType is 'ipv4' or 'ipv4z',
        this object must be in the range 4..32.
        If the corresponding InetAddressType is 'ipv6' or 'ipv6z',
        this object must be in the range 8..128.
        "
    ::= { mvpnMrouteEntry 15 }

mvpnMrouteRtType OBJECT-TYPE
    SYNTAX      INTEGER {
                                unicast    (1),
                                multicast  (2)
                            }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The reason for placing the route in the (logical)
        multicast Routing Information Base (RIB).

        The enumerated reasons and the corresponding
        descriptions are as follows:

        unicast:
            The route would normally be placed only in
            the unicast RIB, but was placed in the multicast RIB
            by local configuration, such as when running PIM over
            RIP.

        multicast:
            The route was explicitly added to the multicast RIB by
            the routing protocol, such as the Distance Vector
            Multicast Routing Protocol (DVMRP) or Multiprotocol BGP.
        "
    ::= { mvpnMrouteEntry 16 }

mvpnMrouteOctets OBJECT-TYPE
    SYNTAX      Counter64
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
```

"The number of octets contained in IP datagrams that were received from sources represented by this entry and addressed to this multicast group address, and which were forwarded by this router.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnMrouteCounterDiscontinuityTime object.

"

::= { mvpnMrouteEntry 17 }

mvpnMroutePkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets routed using this multicast route entry.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnMrouteCounterDiscontinuityTime object.

"

::= { mvpnMrouteEntry 18 }

mvpnMrouteTtlDroppedOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of octets contained in IP datagrams that this router has received from sources represented by this entry and addressed to this multicast group address, which were dropped due to Time To Live (TTL) issues. TTL issues occur when the TTL (IPv4) or Hop Limit (IPv6) of the incoming packet was decremented to zero, or to a value less than ipMcastInterfaceTtl of the corresponding interface.

The ipMcastInterfaceTtl object is defined in IPMCAST-MIB [RFC5132] and represents the datagram TTL threshold for the interface. Any IP multicast datagrams with a TTL (IPv4) or Hop Limit (IPv6) less than this threshold will not be forwarded out of the interface. The default value of 0 means all multicast packets are forwarded out of the interface. A value of 256 means that

no multicast packets are forwarded out of the interface.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnMrouteCounterDiscontinuityTime object.

"

REFERENCE

"RFC5132, Sec. 6

"

::= { mvpnMrouteEntry 19 }

mvpnMrouteTtlDroppedPackets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets that this router has received from the sources represented by this entry and addressed to this multicast group address, which were dropped due to Time To Live (TTL) issues. TTL issues occur when the TTL (IPv4) or Hop Limit (IPv6) of the incoming packet was decremented to zero, or to a value less than ipMcastInterfaceTtl of the corresponding interface.

The ipMcastInterfaceTtl object is defined in IPMCAST-MIB [RFC5132] and represents the datagram TTL threshold for the interface. Any IP multicast datagrams with a TTL (IPv4) or Hop Limit (IPv6) less than this threshold will not be forwarded out of the interface. The default value of 0 means all multicast packets are forwarded out of the interface. A value of 256 means that no multicast packets are forwarded out of the interface.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnMrouteCounterDiscontinuityTime object.

"

REFERENCE

"RFC5132, Sec. 6

"

::= { mvpnMrouteEntry 20 }

mvpnMrouteDroppedInOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of octets contained in IP datagrams that this router has received from sources represented by this entry and addressed to this multicast group address, which were dropped due to error(s).

The value of this object includes the octets counted in the corresponding mvpnMrouteTtlDroppedOctets object.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnMrouteCounterDiscontinuityTime object.

"

::= { mvpnMrouteEntry 21 }

mvpnMrouteDroppedInPackets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets which this router has received from sources represented by this entry and addressed to this multicast group address, which were dropped due to error(s).

The value of this object includes the number of octets counted in the corresponding mvpnMrouteTtlDroppedPackets object.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnMrouteCounterDiscontinuityTime object.

"

::= { mvpnMrouteEntry 22 }

mvpnMroutePmsiPointer OBJECT-TYPE

SYNTAX RowPointer

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A pointer to a conceptual row representing the corresponding I-PMSI in mvpnPmsiTable or S-PMSI in mvpnSpmsiTable, that this C-multicast route is using.

"

::= { mvpnMrouteEntry 23 }

mvpnMrouteNumberOfLocalReplication OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of replications for local receivers.

For example, if an ingress PE needs to send traffic out of N PE-CE interfaces, then mvpnMrouteNumberOfLocalReplication is N.

"

::= { mvpnMrouteEntry 24 }

mvpnMrouteNumberOfRemoteReplication OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of local replications for remote PEs. For example, if the number of remote PEs that need to receive traffic is N, then mvpnMrouteNumberOfRemoteReplication is N in case of Ingress Replication, but may be less than N in case of RSVP-TE or mLDP P2MP tunnels, depending on the actual number of replications the PE needs to do.

"

::= { mvpnMrouteEntry 25 }

mvpnMrouteCounterDiscontinuityTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime on the most recent occasion at which any one or more of this application's counters, viz., counters with OID prefix 'mvpnMrouteOctets' or 'mvpnMroutePkts' or 'mvpnMrouteTtlDroppedOctets' or 'mvpnMrouteTtlDroppedPackets' or 'mvpnMrouteDroppedInOctets' or 'mvpnMrouteDroppedInPackets' suffered a discontinuity.

If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this object will have a zero value.

"

::= { mvpnMrouteEntry 26 }

-- Table of next hops for multicast routes in an MVPN

mvpnMrouteNextHopTable OBJECT-TYPE

SYNTAX SEQUENCE OF MvpnMrouteNextHopEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual table containing information on the next-hops for routing IP multicast datagrams. Each entry is one of a list of next-hops for a set of sources sending to a multicast group address."

```
::= { mvpnObjects 8 }
```

mvpnMrouteNextHopEntry OBJECT-TYPE

SYNTAX MvpnMrouteNextHopEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A conceptual row corresponding to a next-hop to which IP multicast datagrams from a set of sources to an IP multicast group address are routed."

Implementers need to be aware that if the total number of octets in mplsL3VpnVrfName, mvpnMrouteNextHopGroupAddr, mvpnMrouteNextHopSourceAddrs, and mvpnMrouteNextHopAddr exceeds 111, the OIDs of column instances in this row will have more than 128 sub-identifiers and cannot be accessed using SNMPv1, SNMPv2c, or SNMPv3.

```
INDEX {
    mplsL3VpnVrfName,
    mvpnMrouteNextHopGroupAddrType,
    mvpnMrouteNextHopGroupAddr,
    mvpnMrouteNextHopGroupPrefixLength,
    mvpnMrouteNextHopSourceAddrType,
    mvpnMrouteNextHopSourceAddrs,
    mvpnMrouteNextHopSourcePrefixLength,
    mvpnMrouteNextHopIfIndex,
    mvpnMrouteNextHopAddrType,
    mvpnMrouteNextHopAddr
}
::= { mvpnMrouteNextHopTable 1 }
```

MvpnMrouteNextHopEntry ::= SEQUENCE {

mvpnMrouteNextHopGroupAddrType	InetAddressType,
mvpnMrouteNextHopGroupAddr	InetAddress,
mvpnMrouteNextHopGroupPrefixLength	InetAddressPrefixLength,
mvpnMrouteNextHopSourceAddrType	InetAddressType,
mvpnMrouteNextHopSourceAddrs	InetAddress,
mvpnMrouteNextHopSourcePrefixLength	InetAddressPrefixLength,
mvpnMrouteNextHopIfIndex	InterfaceIndex,
mvpnMrouteNextHopAddrType	InetAddressType,

```

    mvpnMrouteNextHopAddr          InetAddress,
    mvpnMrouteNextHopState         INTEGER,
    mvpnMrouteNextHopExpiryTime    TimeTicks,
    mvpnMrouteNextHopClosestMemberHops Unsigned32,
    mvpnMrouteNextHopProtocol      IANAipMrouteProtocol,
    mvpnMrouteNextHopOctets        Counter64,
    mvpnMrouteNextHopPkts          Counter64,
    mvpnMrouteNextHopCounterDiscontinuityTime TimeStamp
}

mvpnMrouteNextHopGroupAddrType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The InetAddressType of the mvpnMrouteNextHopGroupAddr object
        that follows.
        "
    ::= { mvpnMrouteNextHopEntry 1 }

mvpnMrouteNextHopGroupAddr OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The IP multicast group address which, along with
        the corresponding mvpnMrouteNextHopGroupPrefixLength object,
        identifies destinations for which this entry contains
        multicast forwarding information.

        This address object is only significant up to
        mvpnMrouteNextHopGroupPrefixLength bits. The remaining
        address bits MUST be set to zero.

        For addresses of type 'ipv4z' or 'ipv6z', the appended zone
        index is significant even though it lies beyond the prefix
        length. The use of these address types indicate that this
        forwarding state applies only within the given zone. Zone
        index zero is not valid in this table.
        "
    ::= { mvpnMrouteNextHopEntry 2 }

mvpnMrouteNextHopGroupPrefixLength OBJECT-TYPE
    SYNTAX      InetAddressPrefixLength
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The length in bits of the mask which, along with

```

the corresponding mvpnMrouteGroupAddr object, identifies destinations for which this entry contains multicast routing information.

If the corresponding InetAddressType is 'ipv4' or 'ipv4z', this object must be in the range 4..32.
If the corresponding InetAddressType is 'ipv6' or 'ipv6z', this object must be in the range 8..128.

"

::= { mvpnMrouteNextHopEntry 3 }

mvpnMrouteNextHopSourceAddrType OBJECT-TYPE

SYNTAX InetAddressType

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The InetAddressType of mvpnMrouteNextHopSourceAddrs object that follows.

A value of unknown(0) indicates a non-source-specific entry, corresponding to all sources in the group. Otherwise, the value MUST be the same as the value of mvpnMrouteNextHopGroupAddrType."

::= { mvpnMrouteNextHopEntry 4 }

mvpnMrouteNextHopSourceAddrs OBJECT-TYPE

SYNTAX InetAddress

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The network address which, along with the corresponding mvpnMrouteNextHopSourcePrefixLength object, identifies the sources for which this entry specifies a next-hop.

This address object is only significant up to mvpnMrouteNextHopSourcePrefixLength bits. The remaining address bits MUST be set to zero.

For addresses of type 'ipv4z' or 'ipv6z', the appended zone index is significant even though it lies beyond the prefix length. The use of these address types indicate that this source address applies only within the given zone. Zone index zero is not valid in this table.

"

::= { mvpnMrouteNextHopEntry 5 }

mvpnMrouteNextHopSourcePrefixLength OBJECT-TYPE

SYNTAX InetAddressPrefixLength
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "The length in bits of the mask which, along with
 the corresponding mvpnMrouteNextHopSourceAddrs object,
 identifies the sources for which this entry specifies
 a next-hop.

 If the corresponding InetAddressType is 'ipv4' or 'ipv4z',
 this object must be in the range 4..32.
 If the corresponding InetAddressType is 'ipv6' or 'ipv6z',
 this object must be in the range 8..128.
 If the corresponding InetAddressType is 'unknown',
 this object must be zero.
 "
::= { mvpnMrouteNextHopEntry 6 }

mvpnMrouteNextHopIfIndex OBJECT-TYPE
SYNTAX InterfaceIndex
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "The ifIndex value of the outgoing interface
 for this next-hop.
 "
::= { mvpnMrouteNextHopEntry 7 }

mvpnMrouteNextHopAddrType OBJECT-TYPE
SYNTAX InetAddressType
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "The InetAddressType of the mvpnMrouteNextHopAddr object
 that follows.
 "
::= { mvpnMrouteNextHopEntry 8 }

mvpnMrouteNextHopAddr OBJECT-TYPE
SYNTAX InetAddress
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "The address of the next-hop specific to this entry. For
 most interfaces, this is identical to
 mvpnMrouteNextHopGroupAddr. Non-Broadcast Multi-Access
 (NBMA) interfaces, however, may have multiple next-hop
 addresses out of a single outgoing interface.

```

"
 ::= { mvpnMrouteNextHopEntry 9 }

mvpnMrouteNextHopState OBJECT-TYPE
    SYNTAX      INTEGER {
                                pruned(1),
                                forwarding(2)
                            }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "An indication of whether the outgoing interface and next-
        hop represented by this entry is currently being used to
        forward IP datagrams.

        The enumerated states and the corresponding
        descriptions are as follows:

            pruned      : this entry is not currently being used.
            forwarding  : this entry is currently being used.
        "
 ::= { mvpnMrouteNextHopEntry 10 }

mvpnMrouteNextHopExpiryTime OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The minimum amount of time remaining before this entry will
        be aged out. If mvpnMrouteNextHopState is pruned(1),
        this object represents the remaining time for the prune
        to expire after which the state will return to forwarding(2).
        If mvpnMrouteNextHopState is forwarding(2),
        this object indicates the time after which this
        entry will be removed from the table.

        The value of 0 indicates that the entry is not subject to
        aging.
        "
 ::= { mvpnMrouteNextHopEntry 11 }

mvpnMrouteNextHopClosestMemberHops OBJECT-TYPE
    SYNTAX      Unsigned32 (0..256)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The minimum number of hops between this router and any
        member of this IP multicast group reached via this next-hop

```

on the corresponding outgoing interface. Any IP multicast datagram for the group that has a TTL (IPv4) or Hop Count (IPv6) less than mvpnMrouteNextHopClosestMemberHops will not be forwarded through this interface.

A value of 0 means all multicast datagrams are forwarded out of the interface. A value of 256 means that no multicast datagrams are forwarded out of the interface.

This is an optimization applied by multicast routing protocols that explicitly track hop counts to downstream listeners. Multicast protocols that are not aware of hop counts to downstream listeners set this object to 0.

"

::= { mvpnMrouteNextHopEntry 12 }

mvpnMrouteNextHopProtocol OBJECT-TYPE

SYNTAX IANAipMRouteProtocol

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The routing protocol via which this next-hop was learned."

::= { mvpnMrouteNextHopEntry 13 }

mvpnMrouteNextHopOctets OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of octets of multicast packets that have been forwarded using this route.

Discontinuities in the value of this counter can occur at re-initialization of the management system, and at other times as indicated by the corresponding mvpnMrouteNextHopCounterDiscontinuityTime object.

"

::= { mvpnMrouteNextHopEntry 14 }

mvpnMrouteNextHopPkts OBJECT-TYPE

SYNTAX Counter64

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The number of packets which have been forwarded using this route.

Discontinuities in the value of this counter can

```
        occur at re-initialization of the management system,
        and at other times as indicated by the corresponding
        mvpnMrouteNextHopCounterDiscontinuityTime object.
    "
 ::= { mvpnMrouteNextHopEntry 15 }

mvpnMrouteNextHopCounterDiscontinuityTime OBJECT-TYPE
    SYNTAX          TimeStamp
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of sysUpTime on the most recent occasion
        at which any one or more of this application's
        counters, viz., counters with OID prefix
        'mvpnMrouteNextHopOctets' or 'mvpnMrouteNextHopPackets'
        suffered a discontinuity.
        If no such discontinuities have occurred since the
        last re-initialization of the local management
        subsystem, then this object will have a zero value.
    "
 ::= { mvpnMrouteNextHopEntry 16 }

-- MVPN Notifications

mvpnMvrfActionTaken NOTIFICATION-TYPE
    OBJECTS          {
        mvpnGenMvrfCreationTime,
        mvpnGenMvrfLastAction,
        mvpnGenMvrfLastActionTime,
        mvpnGenMvrfCreationTime,
        mvpnGenCmcastRouteProtocol,
        mvpnGenUmhSelection,
        mvpnGenCustomerSiteType
    }
    STATUS          current
    DESCRIPTION
        "mvpnMvrfActionTaken notifies about a change
        in a MVRF on the PE. The change itself will be given by
        mvpnGenMvrfLastAction.
    "
 ::= { mvpnNotifications 1 }

-- MVPN MIB Conformance Information

mvpnGroups          OBJECT IDENTIFIER ::= { mvpnConformance 1 }
mvpnCompliances     OBJECT IDENTIFIER ::= { mvpnConformance 2 }

-- Compliance Statements
```

```
mvpnModuleFullCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance statement for agents that provide full support
    for the BGP-MPLS-LAYER3-VPN-MULTICAST-MIB
    "
  MODULE -- this module
  MANDATORY-GROUPS {
    mvpnScalarGroup,
    mvpnGenericGroup,
    mvpnPmsiGroup,
    mvpnAdvtStatsGroup,
    mvpnMrouteGroup,
    mvpnMrouteNextHopGroup,
    mvpnNotificationGroup
  }

  GROUP mvpnBgpScalarGroup
  DESCRIPTION
    "This group is mandatory for systems that support
    BGP-MVPN.
    "

  GROUP mvpnBgpGroup
  DESCRIPTION
    "This group is mandatory for systems that support
    BGP-MVPN.
    "

  ::= { mvpnCompliances 1 }

mvpnModuleReadOnlyCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION "Compliance requirement for implementations that
    only provide read-only support for
    BGP-MPLS-LAYER3-VPN-MULTICAST-MIB. Such devices
    can then be monitored but cannot be configured
    using this MIB module.
    "
  MODULE -- this module
  MANDATORY-GROUPS {
    mvpnScalarGroup,
    mvpnGenericGroup,
    mvpnPmsiGroup,
    mvpnAdvtStatsGroup,
    mvpnMrouteGroup,
    mvpnMrouteNextHopGroup,
    mvpnNotificationGroup
  }
```

```
}

GROUP mvpnBgpScalarGroup
  DESCRIPTION
    "This group is mandatory for systems that support
    BGP-MVPN.
    "

GROUP mvpnBgpGroup
  DESCRIPTION
    "This group is mandatory for systems that support
    BGP-MVPN.
    "

OBJECT      mvpnSPTunnelLimit
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required."

OBJECT      mvpnBgpCmcastRouteWithdrawalTimer
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required."

OBJECT      mvpnBgpSrcSharedTreeJoinTimer
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required."

OBJECT      mvpnBgpMsgRateLimit
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required."

OBJECT      mvpnBgpMaxSpmsiAdRoutes
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required."

OBJECT      mvpnBgpMaxSpmsiAdRouteFreq
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required."

OBJECT      mvpnBgpMaxSrcActiveAdRoutes
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required."

OBJECT      mvpnBgpMaxSrcActiveAdRouteFreq
MIN-ACCESS  read-only
DESCRIPTION "Write access is not required."

::= { mvpnCompliances 2 }
```

```
mvpnModuleAdvtStatsCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "Compliance statement for agents that support
    monitoring of the statistics pertaining to
    advertisements sent/received by a PE.
    "
  MODULE -- this module
  MANDATORY-GROUPS {
    mvpnAdvtStatsGroup
  }

  ::= { mvpnCompliances 3 }

-- units of conformance

mvpnScalarGroup OBJECT-GROUP
  OBJECTS {
    mvpnMvrfs,
    mvpnV4Mvrfs,
    mvpnV6Mvrfs,
    mvpnPimV4Mvrfs,
    mvpnPimV6Mvrfs,
    mvpnSPTunnelLimit
  }
  STATUS current
  DESCRIPTION
    "These objects are used to monitor/manage
    global statistics and parameters.
    "
  ::= { mvpnGroups 1 }

mvpnBgpScalarGroup OBJECT-GROUP
  OBJECTS {
    mvpnMldpMvrfs,
    mvpnBgpV4Mvrfs,
    mvpnBgpV6Mvrfs,
    mvpnBgpCmcastRouteWithdrawalTimer,
    mvpnBgpSrcSharedTreeJoinTimer
  }
  STATUS current
  DESCRIPTION
    "These objects are used to monitor/manage
    BGP-MVPN specific global parameters.
    "
  ::= { mvpnGroups 2 }

mvpnGenericGroup OBJECT-GROUP
```

```
OBJECTS {
    mvpnGenMvrfLastAction,
    mvpnGenMvrfLastActionTime,
    mvpnGenMvrfCreationTime,
    mvpnGenCmcastRouteProtocol,
    mvpnGenIpmsiInfo,
    mvpnGenInterAsPmsiInfo,
    mvpnGenUmhSelection,
    mvpnGenCustomerSiteType
}
STATUS      current
DESCRIPTION
    "These objects are used to monitor MVPNs on a PE.
    "
 ::= { mvpnGroups 3 }

mvpnBgpGroup      OBJECT-GROUP
OBJECTS {
    mvpnBgpMode,
    mvpnBgpVrfRouteImportExtendedCommunity,
    mvpnBgpSrcASExtendedCommunity,
    mvpnBgpMsgRateLimit,
    mvpnBgpMaxSpmsiAdRoutes,
    mvpnBgpMaxSpmsiAdRouteFreq,
    mvpnBgpMaxSrcActiveAdRoutes,
    mvpnBgpMaxSrcActiveAdRouteFreq
}
STATUS      current
DESCRIPTION
    "These objects are used to monitor/manage
    the MVPN-wise BGP specific parameters.
    "
 ::= { mvpnGroups 4 }

mvpnPmsiGroup      OBJECT-GROUP
OBJECTS {
    mvpnPmsiRD,
    mvpnPmsiTunnelType,
    mvpnPmsiTunnelAttribute,
    mvpnPmsiTunnelPimGroupAddrType,
    mvpnPmsiTunnelPimGroupAddr,
    mvpnPmsiEncapsulationType,
    mvpnSpmsiPmsiPointer
}
STATUS      current
DESCRIPTION
    "These objects are used to monitor
    I-PMSIs and S-PMSIs tunnel on a PE.
```



```
"
 ::= { mvpnGroups 5 }

mvpnAdvtStatsGroup      OBJECT-GROUP
  OBJECTS {
    mvpnAdvtSent,
    mvpnAdvtReceived,
    mvpnAdvtReceivedError,
    mvpnAdvtReceivedMalformedTunnelType,
    mvpnAdvtReceivedMalformedTunnelId,
    mvpnAdvtLastSentTime,
    mvpnAdvtLastReceivedTime,
    mvpnAdvtCounterDiscontinuityTime
  }
  STATUS      current
  DESCRIPTION
    "These objects are used to monitor
    the statistics pertaining to I-PMSI and S-PMSI
    advertisements sent/received by a PE.
    "
 ::= { mvpnGroups 6 }

mvpnMrouteGroup         OBJECT-GROUP
  OBJECTS {
    mvpnMrouteUpstreamNeighborAddrType,
    mvpnMrouteUpstreamNeighborAddr,
    mvpnMrouteInIfIndex,
    mvpnMrouteExpiryTime,
    mvpnMrouteProtocol,
    mvpnMrouteRtProtocol,
    mvpnMrouteRtAddrType,
    mvpnMrouteRtAddr,
    mvpnMrouteRtPrefixLength,
    mvpnMrouteRtType,
    mvpnMrouteOctets,
    mvpnMroutePkts,
    mvpnMrouteTtlDroppedOctets,
    mvpnMrouteTtlDroppedPackets,
    mvpnMrouteDroppedInOctets,
    mvpnMrouteDroppedInPackets,
    mvpnMroutePmsiPointer,
    mvpnMrouteNumberOfLocalReplication,
    mvpnMrouteNumberOfRemoteReplication,
    mvpnMrouteCounterDiscontinuityTime
  }
  STATUS      current
  DESCRIPTION
    "These objects are used to monitor multicast routing
```

```

        information corresponding to the MVRFs on a PE.
    "
    ::= { mvpnGroups 7 }

mvpnMrouteNextHopGroup OBJECT-GROUP
    OBJECTS {
        mvpnMrouteNextHopState,
        mvpnMrouteNextHopExpiryTime,
        mvpnMrouteNextHopClosestMemberHops,
        mvpnMrouteNextHopProtocol,
        mvpnMrouteNextHopOctets,
        mvpnMrouteNextHopPkts,
        mvpnMrouteNextHopCounterDiscontinuityTime
    }
    STATUS current
    DESCRIPTION
        "These objects are used to monitor the information on
        next-hops for routing datagrams to MVPNs on a PE.
        "
    ::= { mvpnGroups 8 }

mvpnNotificationGroup NOTIFICATION-GROUP
    NOTIFICATIONS {
        mvpnMvrfActionTaken
    }
    STATUS current
    DESCRIPTION
        "Objects required for MVPN notifications."
    ::= { mvpnGroups 9 }

END

```

4. Security Considerations

This MIB module contains some read-only objects that may be deemed sensitive. It also contains some read-write objects, whose setting will change the device's MVPN related behavior. Appropriate security procedures related to SNMP in general but not specific to this MIB module need to be implemented by concerned operators.

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection opens devices to attack. These are the tables and objects and their sensitivity/vulnerability:

- o mvpnSPTunnelLimit

The value of this object is used to control the maximum number of selective provider tunnels that a PE allows for a particular MVPN. Access to this object may be abused to impact the performance of the PE or prevent the PE from having new selective provider tunnels.

- o mvpnBgpCmcastRouteWithdrawalTimer

The value of this object is used to control the delay for the advertisement of withdrawals of C-multicast routes. Access to this object may be abused to impact the performance of a PE.

- o mvpnBgpSrcSharedTreeJoinTimer

The value of this object is used to control the delay for the advertisement of Source/Shared Tree Join C-multicast routes. Access to this object may be abused to impact the propagation of C-multicast routing information.

- o mvpnBgpMsgRateLimit

The value of this object is used to control the upper bound for the rate of BGP C-multicast routing information message exchange among PEs. Access to this object may be abused to impact the performance of the PE or disrupt the C-multicast routing information message exchange using BGP.

- o mvpnBgpMaxSpmsiAdRoutes

The value of this object is used to control the upper bound for the number of S-PMSI A-D routes. Access to this object may be abused to impact the performance of the PE or prevent the PE from receiving S-PMSI A-D routes.

- o mvpnBgpMaxSpmsiAdRouteFreq

The value of this object is used to control the upper bound for the frequency of S-PMSI A-D route generation. Access to this object may be abused to impact the performance of the PE or prevent the PE from generating new S-PMSI A-D routes.

- o mvpnBgpMaxSrcActiveAdRoutes

The value of this object is used to control the upper bound for the number of Source Active A-D routes. Access to this object may be abused to impact the performance of the PE or prevent the PE from receiving Source Active A-D routes.

- o mvpnBgpMaxSrcActiveAdRouteFreq

The value of this object is used to control the upper bound for the frequency of Source Active A-D route generation. Access to this object may be abused to impact the performance of the PE or prevent the PE from generating new Source Active A-D routes.

Some of the readable objects in this MIB module (e.g., objects with a MAX-ACCESS other than not-accessible) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- o The address-related objects in this MIB module may have impact on privacy and security. These objects may reveal the locations of senders and recipients.

- * mvpnPmsiTunnelPimGroupAddr
- * mvpnSpmsiCmcastGroupAddr
- * mvpnSpmsiCmcastSourceAddr
- * mvpnAdvtPeerAddr
- * mvpnMrouteCmcastGroupAddr
- * mvpnMrouteCmcastSourceAddrs
- * mvpnMrouteUpstreamNeighborAddr
- * mvpnMrouteRtAddr
- * mvpnMrouteNextHopGroupAddr
- * mvpnMrouteNextHopSourceAddrs
- * mvpnMrouteNextHopAddr

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Implementations SHOULD provide the security features described by the SNMPv3 framework (see [RFC3410]), and implementations claiming compliance to the SNMPv3 standard MUST include full support for authentication and privacy via the User-based Security Model (USM) [RFC3414] with the AES cipher algorithm [RFC3826]. Implementations MAY also provide support for the Transport Security Model (TSM) [RFC5591] in combination with a secure transport such as SSH [RFC5592] or TLS/DTLS [RFC6353].

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

5. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

Name	Description	OBJECT IDENTIFIER value
-----	-----	-----
mvpnMIB	BGP-MPLS-LAYER3-VPN-MULTICAST-MIB	{ mib-2 AAAA }

Editor's Note (to be removed prior to publication): the IANA is requested to assign a value for "AAAA" under the 'mib-2' subtree and to record the assignment in the SMI Numbers registry. When the assignment has been made, the RFC Editor is asked to replace "AAAA" (here and in the MIB module) with the assigned value and to remove this note.

6. Acknowledgement

An earlier draft version of this document was coauthored by Zhaohui (Jeffrey) Zhang, Saud Asif, Andy Green, Sameer Gulrajani, and Pradeep G. Jain, based on an earlier draft written by Susheela Vaidya, Thomas D. Nadeau, and Harmen Van der Linde.

This document also borrows heavily from the design and descriptions of ipMcastRouteTable and ipMcastRouteNextHopTable from IPMCAST-MIB[RFC5132].

Glenn Mansfield Keeni did the MIB Doctor review and provided valuable comments.

7. References

7.1. Normative References

- [RFC2003] Perkins, C., "IP Encapsulation within IP", RFC 2003, DOI 10.17487/RFC2003, October 1996, <<https://www.rfc-editor.org/info/rfc2003>>.
- [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>>.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <<https://www.rfc-editor.org/info/rfc2578>>.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, DOI 10.17487/RFC2579, April 1999, <<https://www.rfc-editor.org/info/rfc2579>>.
- [RFC2580] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Conformance Statements for SMIv2", STD 58, RFC 2580, DOI 10.17487/RFC2580, April 1999, <<https://www.rfc-editor.org/info/rfc2580>>.
- [RFC2784] Farinacci, D., Li, T., Hanks, S., Meyer, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 2784, DOI 10.17487/RFC2784, March 2000, <<https://www.rfc-editor.org/info/rfc2784>>.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<https://www.rfc-editor.org/info/rfc2863>>.
- [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>>.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, DOI 10.17487/RFC3414, December 2002, <<https://www.rfc-editor.org/info/rfc3414>>.

- [RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, DOI 10.17487/RFC3826, June 2004, <<https://www.rfc-editor.org/info/rfc3826>>.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, DOI 10.17487/RFC4001, February 2005, <<https://www.rfc-editor.org/info/rfc4001>>.
- [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>>.
- [RFC4382] Nadeau, T., Ed. and H. van der Linde, Ed., "MPLS/BGP Layer 3 Virtual Private Network (VPN) Management Information Base", RFC 4382, DOI 10.17487/RFC4382, February 2006, <<https://www.rfc-editor.org/info/rfc4382>>.
- [RFC5132] McWalter, D., Thaler, D., and A. Kessler, "IP Multicast MIB", RFC 5132, DOI 10.17487/RFC5132, December 2007, <<https://www.rfc-editor.org/info/rfc5132>>.
- [RFC5591] Harrington, D. and W. Hardaker, "Transport Security Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 5591, DOI 10.17487/RFC5591, June 2009, <<https://www.rfc-editor.org/info/rfc5591>>.
- [RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure Shell Transport Model for the Simple Network Management Protocol (SNMP)", RFC 5592, DOI 10.17487/RFC5592, June 2009, <<https://www.rfc-editor.org/info/rfc5592>>.
- [RFC6353] Hardaker, W., "Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)", STD 78, RFC 6353, DOI 10.17487/RFC6353, July 2011, <<https://www.rfc-editor.org/info/rfc6353>>.
- [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>>.

- [RFC6625] Rosen, E., Ed., Rekhter, Y., Ed., Hendrickx, W., and R. Qiu, "Wildcards in Multicast VPN Auto-Discovery Routes", RFC 6625, DOI 10.17487/RFC6625, May 2012, <<https://www.rfc-editor.org/info/rfc6625>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [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>>.
- [RFCXXXX] Zhang, Z. and H. Tsunoda, "L2L3 VPN Multicast MIB", draft-ietf-bess-l2l3-vpn-mcast-mib-16 (work in progress), September 2018.
- [RTPROTO] IANA, "IP Route Protocol MIB", 2016, <<http://www.iana.org/assignments/ianaiprouteprotocol-mib>>.

7.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, DOI 10.17487/RFC3410, December 2002, <<https://www.rfc-editor.org/info/rfc3410>>.

Author's Address

Hiroshi Tsunoda
Tohoku Institute of Technology
35-1, Yagiyama Kasumi-cho, Taihaku-ku
Sendai 982-8577
Japan

Phone: +81-22-305-3411
Email: tsuno@m.ieice.org

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 20, 2018

L. Berger
LabN Consulting, L.L.C.
C. Hopps
Deutsche Telekom
A. Lindem
Cisco Systems
D. Bogdanovic

X. Liu
Jabil
March 19, 2018

YANG Model for Network Instances
draft-ietf-rtgwg-ni-model-12

Abstract

This document defines a network instance module. This module can be used to manage the virtual resource partitioning that may be present on a network device. Examples of common industry terms for virtual resource partitioning are Virtual Routing and Forwarding (VRF) instances and Virtual Switch Instances (VSIs).

The YANG model in this document conforms to the Network Management Datastore Architecture defined in I-D.ietf-netmod-revised-datastores.

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 September 20, 2018.

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
1.1. Terminology	3
2. Overview	4
3. Network Instances	5
3.1. NI Types and Mount Points	6
3.1.1. Well Known Mount Points	7
3.1.2. NI Type Example	8
3.2. NIs and Interfaces	9
3.3. Network Instance Management	10
3.4. Network Instance Instantiation	12
4. Security Considerations	13
5. IANA Considerations	14
6. Network Instance Model	14
7. References	20
7.1. Normative References	20
7.2. Informative References	22
Appendix A. Acknowledgments	23
Appendix B. Example NI usage	23
B.1. Configuration Data	23
B.2. State Data - Non-NMDA Version	27
B.3. State Data - NMDA Version	33
Authors' Addresses	42

1. Introduction

This document defines the second of two new modules that are defined to support the configuration and operation of network-devices that allow for the partitioning of resources from both, or either, management and networking perspectives. Both leverage the YANG functionality enabled by YANG Schema Mount [I-D.ietf-netmod-schema-mount].

The YANG model in this document conforms to the Network Management Datastore Architecture defined in the [I-D.ietf-netmod-revised-datastores].

The first form of resource partitioning provides a logical partitioning of a network device where each partition is separately managed as essentially an independent network element which is 'hosted' by the base network device. These hosted network elements are referred to as logical network elements, or LNEs, and are supported by the logical-network-element module defined in [I-D.ietf-rtgwg-lne-model]. That module is used to identify LNEs and associate resources from the network-device with each LNE. LNEs themselves are represented in YANG as independent network devices; each accessed independently. Examples of vendor terminology for an LNE include logical system or logical router, and virtual switch, chassis, or fabric.

The second form, which is defined in this document, provides support for what is commonly referred to as Virtual Routing and Forwarding (VRF) instances as well as Virtual Switch Instances (VSI), see [RFC4026] and [RFC4664]. In this form of resource partitioning, multiple control plane and forwarding/bridging instances are provided by and managed via a single (physical or logical) network device. This form of resource partitioning is referred to as a Network Instance and is supported by the network-instance module defined below. Configuration and operation of each network-instance is always via the network device and the network-instance module.

One notable difference between the LNE model and the NI model is that the NI model provides a framework for VRF and VSI management. This document envisions the separate definition of VRF and VSI, i.e., L3 and L2 VPN, technology specific models. An example of such can be found in the emerging L3VPN model defined in [I-D.ietf-bess-l3vpn-yang] and the examples discussed below.

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.

Readers are expected to be familiar with terms and concepts of YANG [RFC7950] and YANG Schema Mount [I-D.ietf-netmod-schema-mount].

This document uses the graphical representation of data models defined in [I-D.ietf-netmod-yang-tree-diagrams].

2. Overview

In this document, we consider network devices that support protocols and functions defined within the IETF, e.g, routers, firewalls, and hosts. Such devices may be physical or virtual, e.g., a classic router with custom hardware or one residing within a server-based virtual machine implementing a virtual network function (VNF). Each device may sub-divide their resources into logical network elements (LNEs) each of which provides a managed logical device. Examples of vendor terminology for an LNE include logical system or logical router, and virtual switch, chassis, or fabric. Each LNE may also support virtual routing and forwarding (VRF) and virtual switching instance (VSI) functions, which are referred to below as a network instances (NIs). This breakdown is represented in Figure 1.

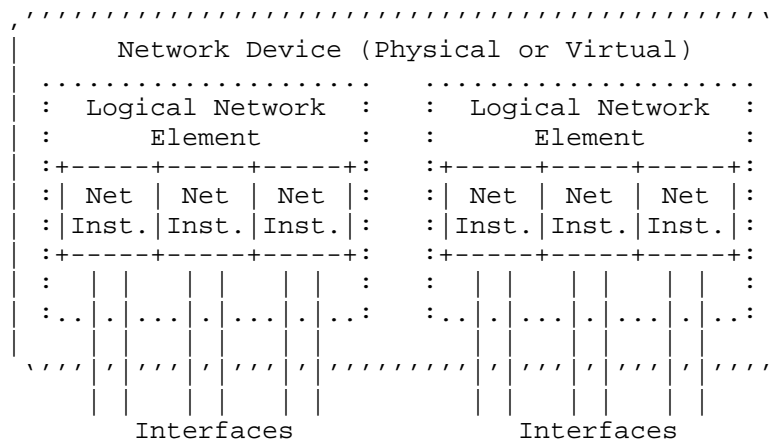


Figure 1: Module Element Relationships

A model for LNEs is described in [I-D.ietf-rtgwg-lne-model] and the model for NIs is covered in this document in Section 3.

The current interface management model [I-D.ietf-netmod-rfc7223bis] is impacted by the definition of LNEs and NIs. This document and [I-D.ietf-rtgwg-lne-model] define augmentations to the interface module to support LNEs and NIs.

The network instance model supports the configuration of VRFs and VSIs. Each instance is supported by information that relates to the device, for example the route target used when advertising VRF routes via the mechanisms defined in [RFC4364], and information that relates to the internal operation of the NI, for example for routing protocols [I-D.ietf-netmod-rfc8022bis] and OSPF [I-D.ietf-ospf-yang].

This document defines the network-instance module that provides a basis for the management of both types of information.

NI information that relates to the device, including the assignment of interfaces to NIs, is defined as part of this document. The defined module also provides a placeholder for the definition of NI-technology specific information both at the device level and for NI internal operation. Information related to NI internal operation is supported via schema mount [I-D.ietf-netmod-schema-mount] and mounting appropriate modules under the mount point. Well known mount points are defined for L3VPN, L2VPN, and L2+L3VPN NI types.

3. Network Instances

The network instance container is used to represent virtual routing and forwarding instances (VRFs) and virtual switching instances (VSIs). VRFs and VSIs are commonly used to isolate routing and switching domains, for example to create virtual private networks, each with their own active protocols and routing/switching policies. The model supports both core/provider and virtual instances. Core/provider instance information is accessible at the top level of the server, while virtual instance information is accessible under the root schema mount points.

```

module: ietf-network-instance
  +--rw network-instances
    +--rw network-instance* [name]
      +--rw name                string
      +--rw enabled?            boolean
      +--rw description?        string
      +--rw (ni-type)?
      +--rw (root-type)
        +--:(vrf-root)
          | +--mp vrf-root
        +--:(vsi-root)
          | +--mp vsi-root
        +--:(vv-root)
          +--mp vv-root
  augment /if:interfaces/if:interface:
    +--rw bind-ni-name? -> /network-instances/network-instance/name
  augment /if:interfaces/if:interface/ip:ipv4:
    +--rw bind-ni-name? -> /network-instances/network-instance/name
  augment /if:interfaces/if:interface/ip:ipv6:
    +--rw bind-ni-name? -> /network-instances/network-instance/name

notifications:
  +---n bind-ni-name-failed
    +--ro name                -> /if:interfaces/interface/name
    +--ro interface
      | +--ro bind-ni-name?
      |                               -> /if:interfaces/interface/ni:bind-ni-name
    +--ro ipv4
      | +--ro bind-ni-name?
      |                               -> /if:interfaces/interface/ip:ipv4/ni:bind-ni-name
    +--ro ipv6
      | +--ro bind-ni-name?
      |                               -> /if:interfaces/interface/ip:ipv6/ni:bind-ni-name
    +--ro error-info?         string

```

A network instance is identified by a 'name' string. This string is used both as an index within the network-instance module and to associate resources with a network instance as shown above in the interface augmentation. The ni-type and root-type choice statements are used to support different types of L2 and L3 VPN technologies. The bind-ni-name-failed notification is used in certain failure cases.

3.1. NI Types and Mount Points

The network-instance module is structured to facilitate the definition of information models for specific types of VRFs and VSIs using augmentations. For example, the information needed to support

VPLS, VxLAN and EVPN based L2VPNs are likely to be quite different. Example models under development that could be restructured to take advantage on NIs include, for L3VPNs [I-D.ietf-bess-l3vpn-yang] and for L2VPNs [I-D.ietf-bess-l2vpn-yang].

Documents defining new YANG models for the support of specific types of network instances should augment the network instance module. The basic structure that should be used for such augmentations include a case statement, with containers for configuration and state data and finally, when needed, a type specific mount point. Generally ni types, are expected to not need to define type specific mount points, but rather reuse one of the well known mount point, as defined in the next section. The following is an example type specific augmentation:

```
augment "/ni:network-instances/ni:network-instance/ni:ni-type" {
  case l3vpn {
    container l3vpn {
      ...
    }
    container l3vpn-state {
      ...
    }
  }
}
```

3.1.1.1. Well Known Mount Points

YANG Schema Mount, [I-D.ietf-netmod-schema-mount], identifies mount points by name within a module. This definition allows for the definition of mount points whose schema can be shared across ni-types. As discussed above, ni-types largely differ in the configuration information needed in the core/top level instance to support the NI, rather than in the information represented within an NI. This allows the use of shared mount points across certain NI types.

The expectation is that there are actually very few different schema that need to be defined to support NIs on an implementation. In particular, it is expected that the following three forms of NI schema are needed, and each can be defined with a well known mount point that can be reused by future modules defining ni-types.

The three well known mount points are:

vrf-root
vrf-root is intended for use with L3VPN type ni-types.

vsi-root

vsi-root is intended for use with L2VPN type ni-types.

vv-root

vv-root is intended for use with ni-types that simultaneously support L2VPN bridging and L3VPN routing capabilities.

Future model definitions should use the above mount points whenever possible. When a well known mount point isn't appropriate, a model may define a type specific mount point via augmentation.

3.1.2. NI Type Example

The following is an example of an L3VPN VRF using a hypothetical augmentation to the networking instance schema defined in [I-D.ietf-bess-l3vpn-yang]. More detailed examples can be found in Appendix B.

```

module: ietf-network-instance
  +--rw network-instances
    +--rw network-instance* [name]
      +--rw name string
      +--rw enabled? boolean
      +--rw description? string
      +--rw (ni-type)?
        |   +--:(l3vpn)
        |     +--rw l3vpn:l3vpn
        |         |   ... // config data
        |         +--ro l3vpn:l3vpn-state
        |             |   ... // state data
        +--rw (root-type)
          +--:(vrf-root)
            +--mp vrf-root
              +--rw rt:routing/
                +--rw router-id? yang:dotted-quad
                +--rw control-plane-protocols
                  +--rw control-plane-protocol* [type name]
                  +--rw ospf:ospf/
                    +--rw area* [area-id]
                    +--rw interfaces
                      +--rw interface* [name]
                        +--rw name if:interface-ref
                        +--rw cost? uint16
              +--ro if:interfaces@
                |   ...

```

This shows YANG Routing Management [I-D.ietf-netmod-rfc8022bis] and YANG OSPF [I-D.ietf-ospf-yang] as mounted modules. The mounted

modules can reference interface information via a parent-reference to the containers defined in [I-D.ietf-netmod-rfc7223bis].

3.2. NIs and Interfaces

Interfaces are a crucial part of any network device's configuration and operational state. They generally include a combination of raw physical interfaces, link-layer interfaces, addressing configuration, and logical interfaces that may not be tied to any physical interface. Several system services, and layer 2 and layer 3 protocols may also associate configuration or operational state data with different types of interfaces (these relationships are not shown for simplicity). The interface management model is defined by [I-D.ietf-netmod-rfc7223bis].

As shown below, the network-instance module augments the existing interface management model by adding a name which is used on interface or sub-interface types to identify an associated network instance. Similarly, this name is also added for IPv4 and IPv6 types, as defined in [I-D.ietf-netmod-rfc7277bis].

The following is an example of envisioned usage. The interfaces container includes a number of commonly used components as examples:

```
module: ietf-interfaces
  +--rw interfaces
  |   +--rw interface* [name]
  |   |   +--rw name                                string
  |   |   +--rw ip:ipv4!
  |   |   |   +--rw ip:enabled?                    boolean
  |   |   |   +--rw ip:forwarding?                 boolean
  |   |   |   +--rw ip:mtu?                         uint16
  |   |   |   +--rw ip:address* [ip]
  |   |   |   |   +--rw ip:ip                      inet:ipv4-address-no-zone
  |   |   |   |   +--rw (ip:subnet)
  |   |   |   |   |   +--:(ip:prefix-length)
  |   |   |   |   |   |   +--rw ip:prefix-length?  uint8
  |   |   |   |   |   +--:(ip:netmask)
  |   |   |   |   |   |   +--rw ip:netmask?        yang:dotted-quad
  |   |   |   +--rw ip:neighbor* [ip]
  |   |   |   |   +--rw ip:ip                      inet:ipv4-address-no-zone
  |   |   |   |   +--rw ip:link-layer-address      yang:phys-address
  |   |   |   +--rw ni:bind-network-instance-name? string
  |   +--rw ni:bind-network-instance-name?  string
```

The [I-D.ietf-netmod-rfc7223bis] defined interface model is structured to include all interfaces in a flat list, without regard to virtual instances (e.g., VRFs) supported on the device. The bind-

network-instance-name leaf provides the association between an interface and its associated NI (e.g., VRF or VSI). Note that as currently defined, to assign an interface to both an LNE and NI, the interface would first be assigned to the LNE using the mechanisms defined in [I-D.ietf-rtgwg-lne-model] and then within that LNE's interface module, the LNE's representation of that interface would be assigned to an NI.

3.3. Network Instance Management

Modules that may be used to represent network instance information will be available under the ni-type specific 'root' mount point. The "shared-schema" method defined in the "ietf-yang-schema-mount" module [I-D.ietf-netmod-schema-mount] MUST be used to identify accessible modules. A future version of this document could relax this requirement. Mounted modules SHOULD be defined with access, via the appropriate schema mount parent-references [I-D.ietf-netmod-schema-mount], to device resources such as interfaces. An implementation MAY choose to restrict parent referenced information to information related to a specific instance, e.g., only allowing references to interfaces that have a "bind-network-instance-name" which is identical to the instance's "name".

All modules that represent control-plane and data-plane information may be present at the 'root' mount point, and be accessible via paths modified per [I-D.ietf-netmod-schema-mount]. The list of available modules is expected to be implementation dependent, as is the method used by an implementation to support NIs.

For example, the following could be used to define the data organization of the example NI shown in Section 3.1.2:

```
"ietf-yang-schema-mount:schema-mounts": {
  "mount-point": [
    {
      "module": "ietf-network-instance",
      "label": "vrf-root",
      "shared-schema": {
        "parent-reference": [
          "/*[namespace-uri() = 'urn:ietf:...:ietf-interfaces']"
        ]
      }
    ]
  }
}
```

Module data identified according to the ietf-yang-schema-mount module will be instantiated under the mount point identified under "mount-

point". These modules will be able to reference information for nodes belonging to top-level modules that are identified under "parent-reference". Parent referenced information is available to clients via their top level paths only, and not under the associated mount point.

To allow a client to understand the previously mentioned instance restrictions on parent referenced information, an implementation MAY represent such restrictions in the "parent-reference" leaf-list. For example:

```
"namespace": [
  {
    "prefix": "if",
    "uri": "urn:ietf:params:xml:ns:yang:ietf-interfaces"
  },
  {
    "prefix": "ni",
    "uri": "urn:ietf:params:xml:ns:yang:ietf-network-instance"
  }
],
"mount-point": [
  {
    "module": "ietf-network-instance",
    "label": "vrf-root",
    "shared-schema": {
      "parent-reference": [
        "/if:interfaces/if:interface
          [ni:bind-network-instance-name = current()/../ni:name]",
        "/if:interfaces/if:interface/ip:ipv4
          [ni:bind-network-instance-name = current()/../ni:name]",
        "/if:interfaces/if:interface/ip:ipv6
          [ni:bind-network-instance-name = current()/../ni:name]"
      ]
    }
  }
],
```

The same such "parent-reference" restrictions for non-NMDA implementations can be represented based on the [RFC7223] and [RFC7277] as:

```

"namespace": [
  {
    "prefix": "if",
    "uri": "urn:ietf:params:xml:ns:yang:ietf-interfaces"
  },
  {
    "prefix": "ni",
    "uri": "urn:ietf:params:xml:ns:yang:ietf-network-instance"
  }
],
"mount-point": [
  {
    "module": "ietf-network-instance",
    "label": "vrf-root",
    "shared-schema": {
      "parent-reference": [
        "/if:interfaces/if:interface
          [ni:bind-network-instance-name = current()/../ni:name]",
        "/if:interfaces-state/if:interface
          [if:name = /if:interfaces/if:interface
            [ni:bind-ni-name = current()/../ni:name]/if:name]",
        "/if:interfaces/if:interface/ip:ipv4
          [ni:bind-network-instance-name = current()/../ni:name]",
        "/if:interfaces-state/if:interface/ip:ipv4
          [if:name = /if:interfaces/if:interface/ip:ipv4
            [ni:bind-ni-name = current()/../ni:name]/if:name]",
        "/if:interfaces/if:interface/ip:ipv6
          [ni:bind-network-instance-name = current()/../ni:name]",
        "/if:interfaces-state/if:interface/ip:ipv6
          [if:name = /if:interfaces/if:interface/ip:ipv4
            [ni:bind-ni-name = current()/../ni:name]/if:name]"
      ]
    }
  }
],

```

3.4. Network Instance Instantiation

Network instances may be controlled by clients using existing list operations. When a list entry is created, a new instance is instantiated. The models mounted under an NI root are expected to be dependent on the server implementation. When a list entry is deleted, an existing network instance is destroyed. For more information, see [RFC7950] Section 7.8.6.

Once instantiated, host network device resources can be associated with the new NI. As previously mentioned, this document augments ietf-interfaces with the bind-ni-name leaf to support such

associations for interfaces. When a bind-ni-name is set to a valid NI name, an implementation MUST take whatever steps are internally necessary to assign the interface to the NI or provide an error message (defined below) with an indication of why the assignment failed. It is possible for the assignment to fail while processing the set operation, or after asynchronous processing. Error notification in the latter case is supported via a notification.

4. Security Considerations

The YANG modules specified in this document define 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 [RFC5246].

The NETCONF access control model [RFC6536] 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 two different sets of security considerations to consider in the context of this document. One set is security related to information contained within mounted modules. The security considerations for mounted modules are not substantively changed based on the information being accessible within the context of an NI. For example, when considering the modules defined in [I-D.ietf-netmod-rfc8022bis], the security considerations identified in that document are equally applicable, whether those modules are accessed at a server's root or under an NI instance's root node.

The second area for consideration is information contained in the NI module itself. NI information represents network configuration and route distribution policy information. As such, the security of this information is important, but it is fundamentally no different than any other interface or routing configuration information that has already been covered in [I-D.ietf-netmod-rfc7223bis] and [I-D.ietf-netmod-rfc8022bis].

The vulnerable "config true" parameters and subtrees are the following:

/network-instances/network-instance: This list specifies the network instances and the related control plane protocols configured on a device.

/if:interfaces/if:interface/*/bind-network-instance-name: This leaf indicates the NI instance to which an interface is assigned.

Unauthorized access to any of these lists can adversely affect the routing subsystem of both the local device and the network. This may lead to network malfunctions, delivery of packets to inappropriate destinations and other problems.

5. IANA Considerations

This document registers a URI in the IETF XML registry [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made.

URI: urn:ietf:params:xml:ns:yang:ietf-network-instance

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers a YANG module in the YANG Module Names registry [RFC6020].

name: ietf-network-instance
namespace: urn:ietf:params:xml:ns:yang:ietf-network-instance
prefix: ni
reference: RFC XXXX

6. Network Instance Model

The structure of the model defined in this document is described by the YANG module below.

```
<CODE BEGINS> file "ietf-network-instance@2018-03-20.yang"
module ietf-network-instance {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-network-instance";
  prefix ni;

  // import some basic types

  import ietf-interfaces {
    prefix if;
    reference "draft-ietf-netmod-rfc7223bis: A YANG Data Model
              for Interface Management";
  }
  import ietf-ip {
    prefix ip;
```

```
reference "draft-ietf-netmod-rfc7277bis: A YANG Data Model
        for IP Management";
}
import ietf-yang-schema-mount {
  prefix yangmnt;
  reference "draft-ietf-netmod-schema-mount: YANG Schema Mount";
  // RFC Ed.: Please replace this draft name with the
  // corresponding RFC number
}

organization
  "IETF Routing Area (rtgwg) Working Group";
contact
  "WG Web:    <http://tools.ietf.org/wg/rtgwg/>
   WG List:   <mailto:rtgwg@ietf.org>

   Author:    Lou Berger
               <mailto:lberger@labn.net>
   Author:    Christan Hopps
               <mailto:chopps@chopps.org>
   Author:    Acee Lindem
               <mailto:acee@cisco.com>
   Author:    Dean Bogdanovic
               <mailto:ivandean@gmail.com>";

description
  "This module is used to support multiple network instances
   within a single physical or virtual device.  Network
   instances are commonly known as VRFs (virtual routing
   and forwarding) and VSIs (virtual switching instances).

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

   Redistribution and use in source and binary forms, with or
   without modification, is permitted pursuant to, and subject
   to the license terms contained in, the Simplified BSD License
   set forth in Section 4.c of the IETF Trust's Legal Provisions
   Relating to IETF Documents
   (http://trustee.ietf.org/license-info).

   This version of this YANG module is part of RFC XXXX; see
   the RFC itself for full legal notices.";

// RFC Ed.: replace XXXX with actual RFC number and remove
// this note
// RFC Ed.: please update TBD

revision 2018-03-20 {
```

```
    description
      "Initial revision.";
    reference "RFC TBD";
  }

  // top level device definition statements

  container network-instances {
    description
      "Network instances each of which consists of a
       VRFs (virtual routing and forwarding) and/or
       VSIs (virtual switching instances).";
    reference "draft-ietf-rtgwg-rfc8022bis - A YANG Data Model
              for Routing Management";
    list network-instance {
      key "name";
      description
        "List of network-instances.";
      leaf name {
        type string;
        mandatory true;
        description
          "device scoped identifier for the network
           instance.";
      }
      leaf enabled {
        type boolean;
        default "true";
        description
          "Flag indicating whether or not the network
           instance is enabled.";
      }
      leaf description {
        type string;
        description
          "Description of the network instance
           and its intended purpose.";
      }
      choice ni-type {
        description
          "This node serves as an anchor point for different types
           of network instances. Each 'case' is expected to
           differ in terms of the information needed in the
           parent/core to support the NI, and may differ in their
           mounted schema definition. When the mounted schema is
           not expected to be the same for a specific type of NI
           a mount point should be defined.";
      }
    }
  }
```



```
choice root-type {
  mandatory true;
  description
    "Well known mount points.";
  container vrf-root {
    description
      "Container for mount point.";
    yangmnt:mount-point "vrf-root" {
      description
        "Root for L3VPN type models. This will typically
        not be an inline type mount point.";
    }
  }
  container vsi-root {
    description
      "Container for mount point.";
    yangmnt:mount-point "vsi-root" {
      description
        "Root for L2VPN type models. This will typically
        not be an inline type mount point.";
    }
  }
  container vv-root {
    description
      "Container for mount point.";
    yangmnt:mount-point "vv-root" {
      description
        "Root models that support both L2VPN type bridging
        and L3VPN type routing. This will typically
        not be an inline type mount point.";
    }
  }
}

// augment statements

augment "/if:interfaces/if:interface" {
  description
    "Add a node for the identification of the network
    instance associated with the information configured
    on a interface.

    Note that a standard error will be returned if the
    identified leafref isn't present. If an interfaces cannot
    be assigned for any other reason, the operation SHALL fail
    with an error-tag of 'operation-failed' and an
```

```
    error-app-tag of 'ni-assignment-failed'. A meaningful
    error-info that indicates the source of the assignment
    failure SHOULD also be provided.";
  leaf bind-ni-name {
    type leafref {
      path "/network-instances/network-instance/name";
    }
    description
      "Network Instance to which an interface is bound.";
  }
}
augment "/if:interfaces/if:interface/ip:ipv4" {
  description
    "Add a node for the identification of the network
    instance associated with the information configured
    on an IPv4 interface.

    Note that a standard error will be returned if the
    identified leafref isn't present. If an interfaces cannot
    be assigned for any other reason, the operation SHALL fail
    with an error-tag of 'operation-failed' and an
    error-app-tag of 'ni-assignment-failed'. A meaningful
    error-info that indicates the source of the assignment
    failure SHOULD also be provided.";
  leaf bind-ni-name {
    type leafref {
      path "/network-instances/network-instance/name";
    }
    description
      "Network Instance to which IPv4 interface is bound.";
  }
}
augment "/if:interfaces/if:interface/ip:ipv6" {
  description
    "Add a node for the identification of the network
    instance associated with the information configured
    on an IPv6 interface.

    Note that a standard error will be returned if the
    identified leafref isn't present. If an interfaces cannot
    be assigned for any other reason, the operation SHALL fail
    with an error-tag of 'operation-failed' and an
    error-app-tag of 'ni-assignment-failed'. A meaningful
    error-info that indicates the source of the assignment
    failure SHOULD also be provided.";
  leaf bind-ni-name {
    type leafref {
      path "/network-instances/network-instance/name";
    }
  }
}
```

```
    }
    description
      "Network Instance to which IPv6 interface is bound.";
  }
}

// notification statements

notification bind-ni-name-failed {
  description
    "Indicates an error in the association of an interface to an
    NI. Only generated after success is initially returned when
    bind-ni-name is set.

    Note: some errors may need to be reported for multiple
    associations, e.g., a single error may need to be reported
    for an IPv4 and an IPv6 bind-ni-name.

    At least one container with a bind-ni-name leaf MUST be
    included in this notification.";
  leaf name {
    type leafref {
      path "/if:interfaces/if:interface/if:name";
    }
    mandatory true;
    description
      "Contains the interface name associated with the
      failure.";
  }
  container interface {
    description
      "Generic interface type.";
    leaf bind-ni-name {
      type leafref {
        path "/if:interfaces/if:interface/ni:bind-ni-name";
      }
      description
        "Contains the bind-ni-name associated with the
        failure.";
    }
  }
}
container ipv4 {
  description
    "IPv4 interface type.";
  leaf bind-ni-name {
    type leafref {
      path "/if:interfaces/if:interface"
        + "/ip:ipv4/ni:bind-ni-name";
    }
  }
}
```

```
    }
    description
      "Contains the bind-ni-name associated with the
       failure.";
  }
}
container ipv6 {
  description
    "IPv6 interface type.";
  leaf bind-ni-name {
    type leafref {
      path "/if:interfaces/if:interface"
        + "/ip:ipv6/ni:bind-ni-name";
    }
    description
      "Contains the bind-ni-name associated with the
       failure.";
  }
}
leaf error-info {
  type string;
  description
    "Optionally, indicates the source of the assignment
     failure.";
}
}
}
<CODE ENDS>
```

7. References

7.1. Normative References

- [I-D.ietf-netmod-revised-datastores]
Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K.,
and R. Wilton, "Network Management Datastore
Architecture", draft-ietf-netmod-revised-datastores-10
(work in progress), January 2018.
- [I-D.ietf-netmod-rfc7223bis]
Bjorklund, M., "A YANG Data Model for Interface
Management", draft-ietf-netmod-rfc7223bis-03 (work in
progress), January 2018.
- [I-D.ietf-netmod-rfc7277bis]
Bjorklund, M., "A YANG Data Model for IP Management",
draft-ietf-netmod-rfc7277bis-03 (work in progress),
January 2018.

- [I-D.ietf-netmod-schema-mount]
Bjorklund, M. and L. Lhotka, "YANG Schema Mount", draft-ietf-netmod-schema-mount-08 (work in progress), October 2017.
- [I-D.ietf-netmod-yang-tree-diagrams]
Bjorklund, M. and L. Berger, "YANG Tree Diagrams", draft-ietf-netmod-yang-tree-diagrams-06 (work in progress), February 2018.
- [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>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <<https://www.rfc-editor.org/info/rfc5246>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <<https://www.rfc-editor.org/info/rfc8040>>.

- [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

- [I-D.ietf-bess-l2vpn-yang]
Shah, H., Brissette, P., Chen, I., Hussain, I., Wen, B., and K. Tiruveedhula, "YANG Data Model for MPLS-based L2VPN", draft-ietf-bess-l2vpn-yang-08 (work in progress), February 2018.
- [I-D.ietf-bess-l3vpn-yang]
Jain, D., Patel, K., Brissette, P., Li, Z., Zhuang, S., Liu, X., Haas, J., Esale, S., and B. Wen, "Yang Data Model for BGP/MPLS L3 VPNs", draft-ietf-bess-l3vpn-yang-02 (work in progress), October 2017.
- [I-D.ietf-netmod-rfc8022bis]
Lhotka, L., Lindem, A., and Y. Qu, "A YANG Data Model for Routing Management (NMDA Version)", draft-ietf-netmod-rfc8022bis-11 (work in progress), January 2018.
- [I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, Z., Chen, I., and A. Lindem, "Yang Data Model for OSPF Protocol", draft-ietf-ospf-yang-10 (work in progress), March 2018.
- [I-D.ietf-rtgwg-device-model]
Lindem, A., Berger, L., Bogdanovic, D., and C. Hopps, "Network Device YANG Logical Organization", draft-ietf-rtgwg-device-model-02 (work in progress), March 2017.
- [I-D.ietf-rtgwg-lne-model]
Berger, L., Hopps, C., Lindem, A., Bogdanovic, D., and X. Liu, "YANG Model for Logical Network Elements", draft-ietf-rtgwg-lne-model-09 (work in progress), March 2018.
- [RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual Private Network (VPN) Terminology", RFC 4026, DOI 10.17487/RFC4026, March 2005, <<https://www.rfc-editor.org/info/rfc4026>>.
- [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>>.

- [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>>.
- [RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <<https://www.rfc-editor.org/info/rfc7223>>.
- [RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <<https://www.rfc-editor.org/info/rfc7277>>.
- [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>>.
- [RFC8022] Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", RFC 8022, DOI 10.17487/RFC8022, November 2016, <<https://www.rfc-editor.org/info/rfc8022>>.

Appendix A. Acknowledgments

The Routing Area Yang Architecture design team members included Acee Lindem, Anees Shaikh, Christian Hopps, Dean Bogdanovic, Lou Berger, Qin Wu, Rob Shakir, Stephane Litkowski, and Yan Gang. Useful review comments were also received by Martin Bjorklund and John Scudder.

This document was motivated by, and derived from, [I-D.ietf-rtgwg-device-model].

Thanks for AD and IETF last call comments from Alia Atlas, Liang Xia, Benoit Claise, and Adam Roach.

The RFC text was produced using Marshall Rose's xml2rfc tool.

Appendix B. Example NI usage

The following subsections provide example uses of NIs.

B.1. Configuration Data

The following shows an example where two customer specific network instances are configured:

```
{
  "ietf-network-instance:network-instances": {
    "network-instance": [
```

```

{
  "name": "vrf-red",
  "vrf-root": {
    "ietf-routing:routing": {
      "router-id": "192.0.2.1",
      "control-plane-protocols": {
        "control-plane-protocol": [
          {
            "type": "ietf-routing:ospf",
            "name": "1",
            "ietf-ospf:ospf": {
              "af": "ipv4",
              "areas": {
                "area": [
                  {
                    "area-id": "203.0.113.1",
                    "interfaces": {
                      "interface": [
                        {
                          "name": "eth1",
                          "cost": 10
                        }
                      ]
                    }
                  }
                ]
              }
            }
          ]
        }
      }
    }
  },
  {
    "name": "vrf-blue",
    "vrf-root": {
      "ietf-routing:routing": {
        "router-id": "192.0.2.2",
        "control-plane-protocols": {
          "control-plane-protocol": [
            {
              "type": "ietf-routing:ospf",
              "name": "1",
              "ietf-ospf:ospf": {
                "af": "ipv4",
                "areas": {
                  "area": [

```



```

    {
      "area-id": "203.0.113.1",
      "interfaces": {
        "interface": [
          {
            "name": "eth2",
            "cost": 10
          }
        ]
      }
    }
  ]
},
{
  "ietf-interfaces:interfaces": {
    "interfaces": {
      "interface": [
        {
          "name": "eth0",
          "ip:ipv4": {
            "address": [
              {
                "ip": "192.0.2.10",
                "prefix-length": 24,
              }
            ]
          },
          "ip:ipv6": {
            "address": [
              {
                "ip": "2001:db8:0:2::10",
                "prefix-length": 64,
              }
            ]
          }
        }
      ]
    },
    {
      "name": "eth1",
      "ip:ipv4": {

```

```

        "address": [
            {
                "ip": "192.0.2.11",
                "prefix-length": 24,
            }
        ],
    },
    "ip:ipv6": {
        "address": [
            {
                "ip": "2001:db8:0:2::11",
                "prefix-length": 64,
            }
        ]
    },
    "ni:bind-network-instance-name": "vrf-red"
},
{
    "name": "eth2",
    "ip:ipv4": {
        "address": [
            {
                "ip": "192.0.2.11",
                "prefix-length": 24,
            }
        ]
    },
    "ip:ipv6": {
        "address": [
            {
                "ip": "2001:db8:0:2::11",
                "prefix-length": 64,
            }
        ]
    },
    "ni:bind-network-instance-name": "vrf-blue"
}
]
}
},
"ietf-system:system": {
    "authentication": {
        "user": [
            {
                "name": "john",
                "password": "$0$password"
            }
        ]
    }
}

```

```

    }
  }
}

```

B.2. State Data - Non-NMDA Version

The following shows state data for the configuration example above based on [RFC7223], [RFC7277], and [RFC8022].

```

{
  "ietf-network-instance:network-instances": {
    "network-instance": [
      {
        "name": "vrf-red",
        "vrf-root": {
          "ietf-yang-library:modules-state": {
            "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
            "module": [
              {
                "name": "ietf-yang-library",
                "revision": "2016-06-21",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-yang-library",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-ospf",
                "revision": "2018-03-03",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-ospf",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-routing",
                "revision": "2018-03-13",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-routing",
                "conformance-type": "implement"
              }
            ]
          },
          "ietf-routing:routing-state": {
            "router-id": "192.0.2.1",
            "control-plane-protocols": {
              "control-plane-protocol": [
                {
                  "type": "ietf-routing:ospf",

```

```

        "name": "1",
        "ietf-ospf:ospf": {
            "af": "ipv4",
            "areas": {
                "area": [
                    {
                        "area-id": "203.0.113.1",
                        "interfaces": {
                            "interface": [
                                {
                                    "name": "eth1",
                                    "cost": 10
                                }
                            ]
                        }
                    }
                ]
            }
        }
    },
    {
        "name": "vrf-blue",
        "vrf-root": {
            "ietf-yang-library:modules-state": {
                "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
                "module": [
                    {
                        "name": "ietf-yang-library",
                        "revision": "2016-06-21",
                        "namespace":
                            "urn:ietf:params:xml:ns:yang:ietf-yang-library",
                        "conformance-type": "implement"
                    },
                    {
                        "name": "ietf-ospf",
                        "revision": "2018-03-03",
                        "namespace":
                            "urn:ietf:params:xml:ns:yang:ietf-ospf",
                        "conformance-type": "implement"
                    },
                    {
                        "name": "ietf-routing",
                        "revision": "2018-03-13",

```

```

        "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-routing",
        "conformance-type": "implement"
    }
]
},
"ietf-routing:routing-state": {
    "router-id": "192.0.2.2",
    "control-plane-protocols": {
        "control-plane-protocol": [
            {
                "type": "ietf-routing:ospf",
                "name": "1",
                "ietf-ospf:ospf": {
                    "af": "ipv4",
                    "areas": {
                        "area": [
                            {
                                "area-id": "203.0.113.1",
                                "interfaces": {
                                    "interface": [
                                        {
                                            "name": "eth2",
                                            "cost": 10
                                        }
                                    ]
                                }
                            }
                        ]
                    }
                }
            }
        ]
    }
}
},
"ietf-interfaces:interfaces-state": {
    "interfaces": {
        "interface": [
            {
                "name": "eth0",
                "type": "iana-if-type:ethernetCsmacd",
                "oper-status": "up",
                "phys-address": "00:01:02:A1:B1:C0",

```

```

    "statistics": {
      "discontinuity-time": "2017-06-26T12:34:56-05:00"
    },
    "ip:ipv4": {
      "address": [
        {
          "ip": "192.0.2.10",
          "prefix-length": 24,
        }
      ]
    }
    "ip:ipv6": {
      "address": [
        {
          "ip": "2001:db8:0:2::10",
          "prefix-length": 64,
        }
      ]
    }
  },
  {
    "name": "eth1",
    "type": "iana-if-type:ethernetCsmacd",
    "oper-status": "up",
    "phys-address": "00:01:02:A1:B1:C1",
    "statistics": {
      "discontinuity-time": "2017-06-26T12:34:56-05:00"
    },
    "ip:ipv4": {
      "address": [
        {
          "ip": "192.0.2.11",
          "prefix-length": 24,
        }
      ]
    }
    "ip:ipv6": {
      "address": [
        {
          "ip": "2001:db8:0:2::11",
          "prefix-length": 64,
        }
      ]
    }
  }
},
{
  "name": "eth2",
  "type": "iana-if-type:ethernetCsmacd",

```

```
    "oper-status": "up",
    "phys-address": "00:01:02:A1:B1:C2",
    "statistics": {
      "discontinuity-time": "2017-06-26T12:34:56-05:00"
    },
    "ip:ipv4": {
      "address": [
        {
          "ip": "192.0.2.11",
          "prefix-length": 24,
        }
      ]
    }
    "ip:ipv6": {
      "address": [
        {
          "ip": "2001:db8:0:2::11",
          "prefix-length": 64,
        }
      ]
    }
  }
}
},
"ietf-system:system-state": {
  "platform": {
    "os-name": "NetworkOS"
  }
}
"ietf-yang-library:modules-state": {
  "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
  "module": [
    {
      "name": "iana-if-type",
      "revision": "2014-05-08",
      "namespace":
        "urn:ietf:params:xml:ns:yang:iana-if-type",
      "conformance-type": "import"
    },
    {
      "name": "ietf-inet-types",
      "revision": "2013-07-15",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-inet-types",
      "conformance-type": "import"
    }
  ]
}
```

```
    },
    {
      "name": "ietf-interfaces",
      "revision": "2014-05-08",
      "feature": [
        "arbitrary-names",
        "pre-provisioning"
      ],
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-interfaces",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-ip",
      "revision": "2014-06-16",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-ip",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-network-instance",
      "revision": "2018-02-03",
      "feature": [
        "bind-network-instance-name"
      ],
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-network-instance",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-ospf",
      "revision": "2018-03-03",
      "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-routing",
      "revision": "2018-03-13",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-routing",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-system",
      "revision": "2014-08-06",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-system",
      "conformance-type": "implement"
    }
  ],
  "namespace":
    "urn:ietf:params:xml:ns:yang:ietf-system",
  "conformance-type": "implement"
}
```



```

    },
    {
      "name": "ietf-yang-library",
      "revision": "2016-06-21",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-yang-library",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-yang-schema-mount",
      "revision": "2017-05-16",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-yang-types",
      "revision": "2013-07-15",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-yang-types",
      "conformance-type": "import"
    }
  ]
},
"ietf-yang-schema-mount:schema-mounts": {
  "mount-point": [
    {
      "module": "ietf-network-instance",
      "label": "vrf-root",
      "shared-schema": {
        "parent-reference": [
          "/*[namespace-uri() = 'urn:ietf:...:ietf-interfaces']"
        ]
      }
    }
  ]
}
}

```

B.3. State Data - NMDA Version

The following shows state data for the configuration example above based on [I-D.ietf-netmod-rfc7223bis], [I-D.ietf-netmod-rfc7277bis], and [I-D.ietf-netmod-rfc8022bis].

```

{
  "ietf-network-instance:network-instances": {

```

```
"network-instance": [
  {
    "name": "vrf-red",
    "vrf-root": {
      "ietf-yang-library:yang-library": {
        "checksum": "41e2ab5dc325f6d86f743e8da3de323f1a61a801",
        "module-set": [
          {
            "name": "ni-modules",
            "module": [
              {
                "name": "ietf-yang-library",
                "revision": "2016-06-21",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-yang-library",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-ospf",
                "revision": "2018-03-03",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-ospf",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-routing",
                "revision": "2018-03-13",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-routing",
                "conformance-type": "implement"
              }
            ]
          }
        ],
        "import-only-module": [
          {
            "name": "ietf-inet-types",
            "revision": "2013-07-15",
            "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types"
          },
          {
            "name": "ietf-yang-types",
            "revision": "2013-07-15",
            "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types"
          },
          {
            "name": "ietf-datastores",
            "revision": "2018-02-14",
            "namespace": "urn:ietf:params:xml:ns:yang:ietf-datastores"
          }
        ]
      }
    }
  ]
}
```

```

    ]
  }
],
"schema": [
  {
    "name": "ni-schema",
    "module-set": [ "ni-modules" ]
  }
],
"datastore": [
  {
    "name": "ietf-datastores:running",
    "schema": "ni-schema"
  },
  {
    "name": "ietf-datastores:operational",
    "schema": "ni-schema"
  }
]
},
"ietf-routing:routing": {
  "router-id": "192.0.2.1",
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-routing:ospf",
        "name": "1",
        "ietf-ospf:ospf": {
          "af": "ipv4",
          "areas": {
            "area": [
              {
                "area-id": "203.0.113.1",
                "interfaces": {
                  "interface": [
                    {
                      "name": "eth1",
                      "cost": 10
                    }
                  ]
                }
              }
            ]
          }
        }
      }
    ]
  }
}

```

```

    }
  },
  {
    "name": "vrf-blue",
    "vrf-root": {
      "ietf-yang-library:yang-library": {
        "checksum": "41e2ab5dc325f6d86f743e8da3de323f1a61a801",
        "module-set": [
          {
            "name": "ni-modules",
            "module": [
              {
                "name": "ietf-yang-library",
                "revision": "2016-06-21",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-yang-library",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-ospf",
                "revision": "2018-03-03",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-ospf",
                "conformance-type": "implement"
              },
              {
                "name": "ietf-routing",
                "revision": "2018-03-13",
                "namespace":
                  "urn:ietf:params:xml:ns:yang:ietf-routing",
                "conformance-type": "implement"
              }
            ],
            "import-only-module": [
              {
                "name": "ietf-inet-types",
                "revision": "2013-07-15",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-inet-types"
              },
              {
                "name": "ietf-yang-types",
                "revision": "2013-07-15",
                "namespace": "urn:ietf:params:xml:ns:yang:ietf-yang-types"
              },
              {
                "name": "ietf-datastores",
                "revision": "2018-02-14",

```

```

        "namespace": "urn:ietf:params:xml:ns:yang:ietf-datastores"
      }
    ]
  },
  "schema": [
    {
      "name": "ni-schema",
      "module-set": [ "ni-modules" ]
    }
  ],
  "datastore": [
    {
      "name": "ietf-datastores:running",
      "schema": "ni-schema"
    },
    {
      "name": "ietf-datastores:operational",
      "schema": "ni-schema"
    }
  ]
},
"ietf-routing:routing": {
  "router-id": "192.0.2.2",
  "control-plane-protocols": {
    "control-plane-protocol": [
      {
        "type": "ietf-routing:ospf",
        "name": "1",
        "ietf-ospf:ospf": {
          "af": "ipv4",
          "areas": {
            "area": [
              {
                "area-id": "203.0.113.1",
                "interfaces": {
                  "interface": [
                    {
                      "name": "eth2",
                      "cost": 10
                    }
                  ]
                }
              }
            ]
          }
        }
      }
    ]
  }
}

```

```

    ]
  }
}
}
],
},
"ietf-interfaces:interfaces": {
  "interfaces": {
    "interface": [
      {
        "name": "eth0",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "phys-address": "00:01:02:A1:B1:C0",
        "statistics": {
          "discontinuity-time": "2017-06-26T12:34:56-05:00"
        },
        "ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.10",
              "prefix-length": 24,
            }
          ]
        },
        "ip:ipv6": {
          "address": [
            {
              "ip": "2001:db8:0:2::10",
              "prefix-length": 64,
            }
          ]
        }
      },
      {
        "name": "eth1",
        "type": "iana-if-type:ethernetCsmacd",
        "oper-status": "up",
        "phys-address": "00:01:02:A1:B1:C1",
        "statistics": {
          "discontinuity-time": "2017-06-26T12:34:56-05:00"
        },
        "ip:ipv4": {
          "address": [
            {
              "ip": "192.0.2.11",

```

```

        "prefix-length": 24,
      }
    ]
  }
  "ip:ipv6": {
    "address": [
      {
        "ip": "2001:db8:0:2::11",
        "prefix-length": 64,
      }
    ]
  }
},
{
  "name": "eth2",
  "type": "iana-if-type:ethernetCsmacd",
  "oper-status": "up",
  "phys-address": "00:01:02:A1:B1:C2",
  "statistics": {
    "discontinuity-time": "2017-06-26T12:34:56-05:00"
  },
  "ip:ipv4": {
    "address": [
      {
        "ip": "192.0.2.11",
        "prefix-length": 24,
      }
    ]
  }
  "ip:ipv6": {
    "address": [
      {
        "ip": "2001:db8:0:2::11",
        "prefix-length": 64,
      }
    ]
  }
}
]
},
{
  "ietf-system:system-state": {
    "platform": {
      "os-name": "NetworkOS"
    }
  }
}

```

```
"ietf-yang-library:modules-state": {
  "module-set-id": "123e4567-e89b-12d3-a456-426655440000",
  "module": [
    {
      "name": "iana-if-type",
      "revision": "2014-05-08",
      "namespace":
        "urn:ietf:params:xml:ns:yang:iana-if-type",
      "conformance-type": "import"
    },
    {
      "name": "ietf-inet-types",
      "revision": "2013-07-15",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-inet-types",
      "conformance-type": "import"
    },
    {
      "name": "ietf-interfaces",
      "revision": "2018-01-09",
      "feature": [
        "arbitrary-names",
        "pre-provisioning"
      ],
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-interfaces",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-ip",
      "revision": "2018-01-09",
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-ip",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-network-instance",
      "revision": "2018-02-03",
      "feature": [
        "bind-network-instance-name"
      ],
      "namespace":
        "urn:ietf:params:xml:ns:yang:ietf-network-instance",
      "conformance-type": "implement"
    },
    {
      "name": "ietf-ospf",
      "revision": "2017-10-30",
```



```
    "namespace": "urn:ietf:params:xml:ns:yang:ietf-ospf",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-routing",
    "revision": "2018-01-25",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-routing",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-system",
    "revision": "2014-08-06",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-system",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-yang-library",
    "revision": "2016-06-21",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-yang-library",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-yang-schema-mount",
    "revision": "2017-05-16",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-yang-schema-mount",
    "conformance-type": "implement"
  },
  {
    "name": "ietf-yang-types",
    "revision": "2013-07-15",
    "namespace":
      "urn:ietf:params:xml:ns:yang:ietf-yang-types",
    "conformance-type": "import"
  }
]
},
"ietf-yang-schema-mount:schema-mounts": {
  "mount-point": [
    {
      "module": "ietf-network-instance",
      "label": "vrf-root",
      "shared-schema": {
        "parent-reference": [
```

```
        "/*[namespace-uri() = 'urn:ietf:...:ietf-interfaces']"  
      ]  
    }  
  ]  
}
```

Authors' Addresses

Lou Berger
LabN Consulting, L.L.C.

Email: lberger@labn.net

Christan Hopps
Deutsche Telekom

Email: chopps@chopps.org

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC 27513
USA

Email: acee@cisco.com

Dean Bogdanovic

Email: ivandean@gmail.com

Xufeng Liu
Jabil

Email: Xufeng_Liu@jabil.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.

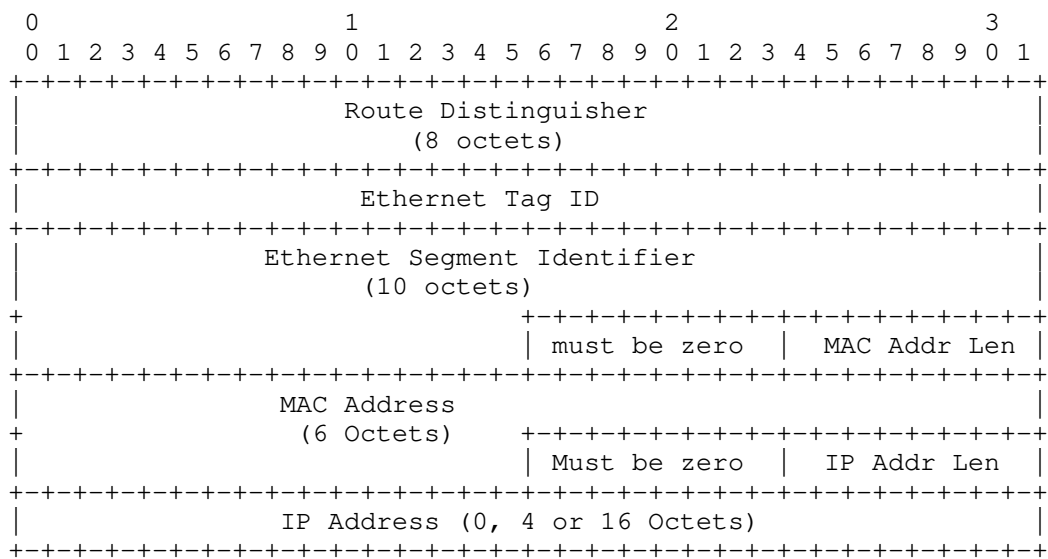


Figure 1: EVPN MAC sub-TLV format

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.

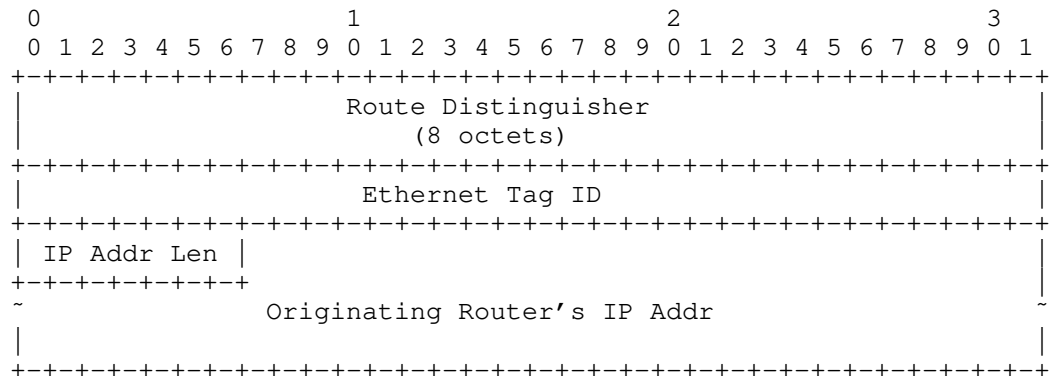


Figure 2: EVPN Inclusive Multicast sub-TLV format

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.

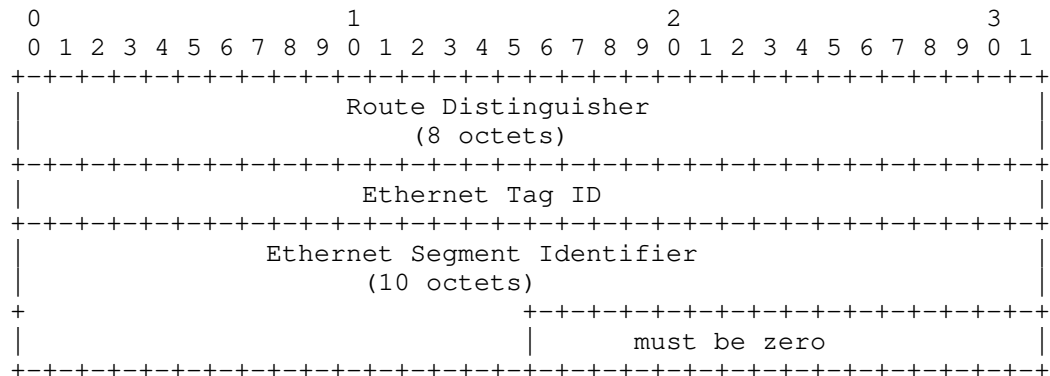


Figure 3: EVPN Auto-Discovery sub-TLV format

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.

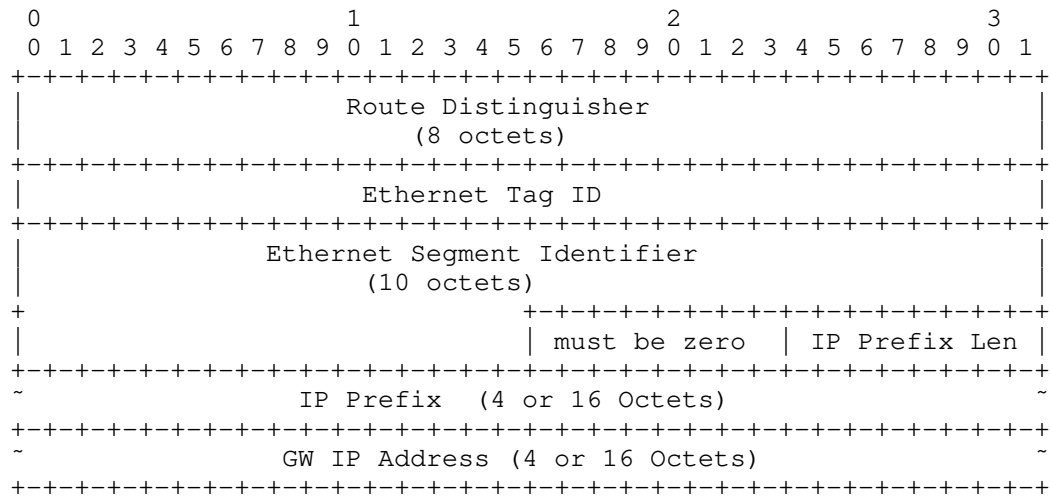


Figure 4: EVPN IP Prefix sub-TLV format

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.

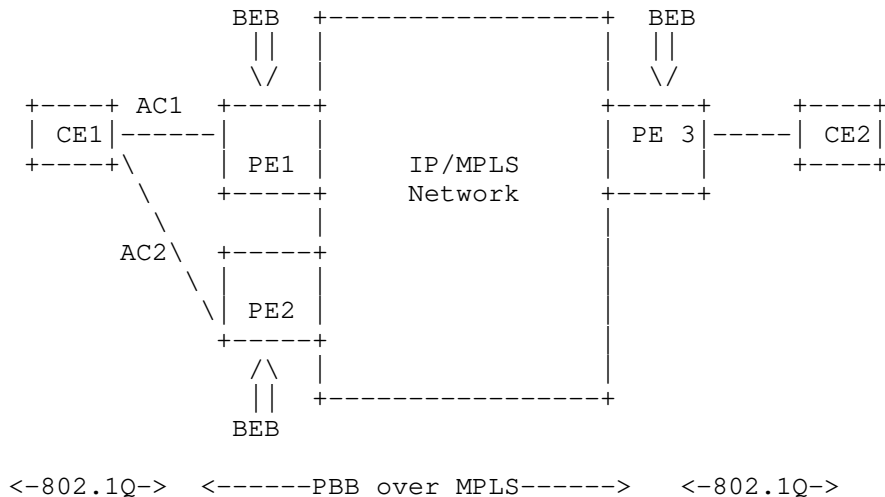


Figure 5: PBB EVPN network

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>

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: December 24, 2017

K. Vairavakkalai
M. Jeyananth
Juniper Networks, Inc.
June 22, 2017

BGP signalled private MPLS-labels
draft-kaliraj-bess-bgp-sig-private-mpls-labels-00

Abstract

The MPLS-forwarding-layer in a core network is a shared resource. The MPLS FIB at nodes in this layer contains labels that are dynamically allocated and locally significant at that node.

For some usecases like upstream-label-allocation, it is useful to be able to create virtual private MPLS-forwarding-layers over this shared MPLS-forwarding-layer. This allows installing deterministic private label-values in the private-FIBs created at nodes participating in this private MPLS forwarding-layer, while preserving the "locally significant" nature of the underlying shared 'public' MPLS-forwarding-layer.

This specification describes the procedures to create such virtual private MPLS-forwarding layers (private MPLS-planes) using a new BGP family. And gives a few example use-cases on how this private forwarding-layers can be used.

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].

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 <http://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 December 24, 2017.

Copyright 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	3
2. Motivation	3
3. Constructs and building blocks	4
3.1. Context Protocol Nexthop Address	4
3.2. MPLS context FIB	4
3.3. Context Label	5
3.4. Roles of nodes in a MPLS-plane	5
3.4.1. Edge-nodes (PLER)	5
3.4.2. Transit-nodes (PLSR)	5
3.5. Sending traffic into the MPLS plane	5
4. Terminology	6
5. BGP families, routes and encoding	7
5.1. New address-families	7
5.1.1. AFI: MPLS, SAFI: 128	7
5.1.2. AFI: MPLS, SAFI: 1	8
5.2. Routes and Operational procedures	8
5.2.1. "Context-Nexthop" discovery route	8
5.2.2. "Private Label" routes	10
6. Example of Usecases	12
6.1. Mezanine transport layer in a Seamless-MPLS network	12
6.2. Service Forwarding Helper usecase	12
6.3. Standard BGP API to a MPLS network's forwarding-plane	13
6.4. Traffic engineering and Security advantages	13
7. IANA Considerations	13
8. Security Considerations	14
9. Acknowledgements	14
10. References	14
11. Normative References	14
Authors' Addresses	14

1. Introduction

The MPLS-forwarding-layer in a core network is a shared resource. The MPLS FIB at nodes in this layer contains labels that are dynamically allocated and locally significant at that node.

For some usecases like upstream-label-allocation, it is useful to be able to create virtual private MPLS-forwarding-layers over this shared MPLS-forwarding-layer. This allows installing deterministic private label-values in the private-FIBs in this private forwarding-layer, while preserving the "locally significant" nature of the underlying shared 'public' MPLS-forwarding-layer.

It can be noted that, mechanism described in this document is nothing but a [RFC-4364] style BGP VPN where the FEC is MPLS-Label, instead of IP-prefix. This document defines new address-families (AFI: MPLS, SAFI: VPN-Unicast, Unicast) and associated signaling mechanisms to create and use MPLS forwarding-contexts in a network. The concepts of MPLS-Context-tables and upstream allocation are described in [RFC-5331].

BGP speakers participating in the private MPLS FIB layer create instances of "MPLS forwarding-context" FIBs, which are identified using a "Context-Protocol-Nexthop (CPNH)". A Context-label MAY be advertised in conjunction with the Context Protocol Nexthop (CPNH) using new BGP address-family to other speakers.

2. Motivation

A provider's core network consists of a global-domain (default forwarding-tables in P and PE nodes) that is shared by all tenants in the network and may also contain multiple private user-domains (e.g. VRF route tables).

The global MPLS forwarding-layer can be viewed as the collection of all default MPLS forwarding-tables. This global MPLS Fib layer contains labels locally significant to each node. The "local-significance of labels" gives the nodes freedom to participate in MPLS-forwarding with whatever label-ranges they can support in forwarding hardware.

In emerging usecases some applications using the MPLS-network may benefit from a "static labels" view of the MPLS-network. In some other usecases, a standard mechanism to do Upstream label-allocation is beneficial.

It is desirable to leave the global MPLS FIB layer intact, and build private MPLS FIB-layers on top of it to achieve these requirements.

The private-MPLS-FIBs can then be used by the applications as desired. The private MPLS-FIBs need to be created only at the nodes in the network where predictable label-values (external label allocation) is desired. E.g. P-routers that need to act as a "Detour-nodes" or "Service-Forwarding-Helpers" that need to mirror service-labels.

In other words, provisioning of these private MPLS-FIBs can be gradual and can co-exist with nodes not supporting the feature described in this document. These private-MPLS-FIBs can be stitched together using either the Context-labels over the existing shared MPLS-network tunnels, or 'private' context-interfaces - to form the "private MPLS-FIB layer".

An application can then install the routes with desired label-values in the private forwarding-contexts with desired forwarding-semantics.

3. Constructs and building blocks

The building-blocks that construct a private MPLS plane are described in this section.

3.1. Context Protocol Nexthop Address

A private MPLS plane (just "MPLS plane" here-after) is identified by an IP-address called Context Protocol Nexthop (CPNH). This address is unique in the core-network, like any other loopback address.

A loopback-address uniquely identifies a specific node in the network, and we call it Global Protocol Nexthop (GPNH) in this document. The CPNH address uniquely identifies a "MPLS-plane".

Each node that has forwarding-context for a MPLS-plane MUST be configured with the same CPNH but a different RD, such that the RD:CPNH will uniquely identify that node in the MPLS-plane.

3.2. MPLS context FIB

An instance of a MPLS forwarding-table at a node in the private MPLS-plane. This Private MPLS FIB contains the private-label routes.

A node can have context-FIB for multiple MPLS-planes. The same label-value can have a different forwarding-semantic in each MPLS-plane. Thus the applications using that MPLS-plane get a deterministic label-value independent of other applications using other MPLS-planes.

The terms "private MPLS FIB-layer" and "private MPLS-plane" are used interchangeably in this document.

3.3. Context Label

A context-label is a non-reserved dynamically allocated label, that is installed in the global MPLS FIB, and points to a MPLS-Context-FIB. The Context-Label have forwarding semantics as follows in the global MPLS-FIB:

Context-Label -> Pop and Lookup in MPLS-Context-Fib

Advertising the "Context-Label in conjunction with the GPNH" tells the network how to reach a "RD:CPNH".

3.4. Roles of nodes in a MPLS-plane

The node roles in a MPLS-plane can be classified into "edge nodes" (call them PLER) or "transit-nodes" (call them PLSR).

3.4.1. Edge-nodes (PLER)

Private Label Edge-routers (PLER) have MPLS context-FIB that belong to the MPLS-plane. They advertise the presence of this context-FIB, and private-label routes from this FIB, using new BGP AFI/SAFI described in this document.

3.4.2. Transit-nodes (PLSR)

Private Label Transit-nodes do label-swap forwarding for the Context-Labels they see in the Context-Protocol-Nexthop advertisement routes going thru them. They basically stitch/extend the label switched path to a RD:CPNH when they re-advertise the CPNH routes with nexthop-self.

PLSRs dont have context-FIBs. PLSRs dont have Context Protocol-Nexthop. Because they dont have Private label routes to originate.

However a node in the network can play both roles, of PLER and PLSR.

3.5. Sending traffic into the MPLS plane

MPLS-traffic arriving with private-labels hits the correct private MPLS-FIB by virtue of either arriving on a "private network-interface" that is attached to the FIB, or arriving on a shared network-interface with a "Context-label".

To send data traffic into this private MPLS FIB-layer, the application MUST use as handle either a "Context-label" advertised by a node or a "Private-interface" owned by the application at the node.

The Context-Label is the only label-value the application needs to learn from the network (PLER node it is connected to), to be able to use the private MPLS-plane. The application can decide the value of the labels to be programmed in the private MPLS-FIBs.

Once the packet enters the private MPLS plane at an edge-node (PLER), the node will forward the packet to the next node (PLSR or PLER), by pushing the Context-label advertised by that next-node, and the transport-label to reach that node's GPNH. This will repeat until the packet reaches the private MPLS-FIB that originated that private MPLS-label.

At each PLER in the MPLS-plane, the private-label value remains the same, and points towards the same resource attached to the MPLS-plane. This allows the applications using the MPLS-network a static-labels view of the resources attached to the private MPLS-plane.

At each PLSR in the MPLS-plane, the context-label value will change (be swapped in forwarding), but is transparent to the application.

4. Terminology

P-router : A Provider core router, also called a LSR

LSR : Label Switch Router (pure transport node speaking LDP, RSVP etc)

PLSR: a transit node in a private MPLS-plane. It has a forwarding-context for private-labels.

PLER: an edge node in a private MPLS-plane. It has a forwarding-context for private-labels.

Detour-router : A P-router that is used as a loose-hop in a traffic-engineered path

PE-router : Provider Edge router, that hosts a service (Internet, L3VPN etc)

SE-router : Service Edge router. Same as PE.

SFH-router : Service Forwarding Helper. A node helping an SE-router with service-traffic forwarding, using Service-routes mirrored by the SE.

MPLS FIB : MPLS Forwarding table

Global MPLS FIB : Global MPLS Forwarding table, to which shared-interfaces are connected

Private MPLS FIB : Private MPLS Forwarding table, to which private-interfaces are connected

Private MPLS FIB Layer : The group of Private MPLS FIBs in the network, connected together via Context-Labels

Context-Label : Locally-significant Non-reserved label pointing to a private MPLS FIB

Context nexthop IP-address (CPNH) : An IP-address that identifies the "Private MPLS FIB Layer". RD:CPNH identifies a Private MPLS FIB at a node.

Global nexthop IP-address (GPNH) : Global Protocol Nexthop address. E.g. a loopback address used as transport tunnel end-point.

5. BGP families, routes and encoding

This section describes the new constructs defined by this document.

5.1. New address-families

This document defines a new AFI: "MPLS". And two new address-families.

5.1.1. AFI: MPLS, SAFI: 128

This address-family is used to exchange private label-routes into private MPLS-FIBs at routers that are connected using a common network-interface.

Routes in this family contain Route-Target extended-community identifying the private-FIB-Layer (VPN) the route belongs to. This address-family also advertises the Context-Label that the receiving router uses to access the private MPLS-FIB. The Context-Label is required when the connecting-interface is a shared common interface that terminates into the global MPLS FIB. The Context-Label installed in the global MPLS-FIB points to the private MPLS-FIB.

5.1.2. AFI: MPLS, SAFI: 1

This address-family is used to exchange private label-routes in private MPLS-FIBs to routers that are connected using a private network-interface.

Because the interface is private, and terminates directly into the private MPLS-FIB, a Context-Label is not required to access the private MPLS-FIB.

5.2. Routes and Operational procedures

5.2.1. "Context-Nexthop" discovery route

NLRI prefix

```

+-----+
| Route Type = 1 (2 octets) |
+-----+
| Route Distinguisher (RD) (8 octets) |
+-----+
| NH-Len in bits (1 octets) |
+-----+
| Context-Nexthop IP-address |
+-----+

```

The Context-NH discovery route contains the following path-attributes:

- o The BGP MultiNexthop-attribute [BGP_MULTI_NH] with forwarding-semantic:
 - * Push <Context-Label> to GPNH (for AFI, SAFI: "MPLS, VpnUni"),
OR
 - * Forward to GPNH (for AFI, SAFI: "MPLS, Uni")
- o Route-Target extended community, identifying the private FIB-layer

MultiNexthop BGP-attribute

```

+-----+
| MultiNH.NumNexthops = 1 |
+-----+
| FwdSemanticsTLV.FwdAction = Push |
+-----+
| NhopDescrType = Labeled-IP-Nhop |
+-----+
| Nexthop-Leg = (Context-Label, GPNH) |
+-----+

```

The "Context-Nexthop discovery route" is originated by each speaker who acts as a PLER. The "RD:Context-nexthop" uniquely identifies the private-FIB at the speaker. The "Context-nexthop address" uniquely identifies the private-FIB-layer.

A speaker (re)advertising a Context-Nexthop discovery-route with "next-hop self" MUST allocate a new Context-Label with a forwarding semantic of "Swap Received-Context-Label, Forward to Received-GPNH". This new Context-label along with self-GPNH is advertised in the Multinexthop-attribute [MULTI_NH] attached to the re-advertised Context-nexthop discovery route.

5.2.1.1. Crossing Tunneled domain boundary

"Nexthop-attributes" include BGP Nexthop attribute (code 3), Nexthop-field inside MP_Reach attribute (code 14) or the Multi-Nexthop BGP attribute (code TBD). Two nodes are deemed to be in same tunneled-domain-boundary if they have some sort of transport-tunnel reachability between them (LDP, RSVP, BGP-LU).

A node receiving a "Context-nexthop discovery route" MAY re-advertise it to other BGP speakers who have negotiated the address-family carrying the route. While doing so, the node SHOULD NOT reset the RD:GPNH next-hop address carried in the "Nexthop-attributes" if the re-advertisement does not cross tunneled-domain boundaries.

If a Context-nexthop discovery route is re-advertised across tunneled-domain-boundaries, the re-advertising node MUST set nexthop-address carried in the "Nexthop-attributes" to Self's GPNH, and allocate a new non-reserved label. The route advertised further MUST carry a Multi-nexthop attribute with a forwarding semantic of:

- o "SWAP <Received Context-Label> and Forward to Received-GPNH".

This new-context-label is installed in the global MPLS FIB at the advertising node. And is used as the Context-Label in the re-advertised RD:CPNH route's Multi-Nexthop attribute, with a forwarding-semantic of:

- o "Push <New-Context-Label> and Forward to Advertising-GPNH"

.

5.2.2. "Private Label" routes

NLRI prefix (Private Label route)

```

+-----+
| Route Type = 2 (2 octets) |
+-----+
| Route Distinguisher (RD) (8 octets) |
+-----+
| 3107 Private Label value |
+-----+

```

Private-Label-Value: The (upstream assigned) label value

Attributes on this route:

- o The Multi-nexthop attribute with forwarding-semantic:
 - * "Forward to RD:CPNH"
- o Route-Target extended-community, identifying the private FIB-layer

MultiNexthop BGP-attribute (Private Label route)

```

+-----+
| MultiNH.Num-Nexthops = 1 |
+-----+
| FwdSemanticsTLV.FwdAction = Forward |
+-----+
| NHDescrTLV.NhopDescrType = RD-IP-Nhop |
+-----+
| "RD:CPNH" advertised in Typel route |
+-----+

```

A speaker MAY readvertise a private-label-route without changing the Nexthop (RD:CPNH) carried in it, if the speaker is a pure PLSR.

If it does alter the nexthop to SelfRD:CPNH, it SHOULD act as a PLER, and for e.g. originate a "Context-Nexthop discovery route" for prefix "SelfRD:CPNH".

Even if the speaker sets nexthop-address to Self because of regular BGP readvertisement-rules, new label MUST NOT be allocated, and the received NLRI "RD:Private-Label1" MUST be re-advertised as-is. Such that value of label "Private-Label1" doesn't change while the packet traverses multiple nodes in the private-MPLS-FIB-layer.

The Route-target attached to the route is the one identifying the private MPLS FIB layer (VPN). The Private-label routes resolve over the Context-nexthop route that belong to the same VPN.

A node receiving a "Private-Label route" RD:L1 MUST install the label L1 in the private MPLS Forwarding-context identified by the Route-Target attached to the route.

The label route MUST be installed with forwarding-semantic as specified in the received Multi-nexthop attribute. As an example, a Detour node MAY receive the private-label-route with a forwarding-semantic of "Forward to RD:CPNH" operation. And an Egress node MAY receive a private-label-route with a forwarding-semantic pointing to a resource it houses. Note that such a Private-label BGP-route MAY be received from external-application also.

5.2.2.1. Resolving received Private Label-routes

A node receiving a "Context-nexthop discovery route" MUST be capable of using either the CPNH or the RD:CPNH carried in the NLRI, to resolve other routes received with this CPNH address or RD:CPNH in the "Nexthop-attributes".

The receiver of a private-label route MUST recursively resolve the received nexthop (RD:CPNH) over the Context-Nexthop discovery-route for prefix "RD:CPNH" to determine the label stack "Context-Label, Transport-Label" to push, so that the MPLS packet with private-label reaches the private MPLS FIB originating the route.

If a node receives multiple "Context-nexthop discovery route" for a CPNH, it SHOULD run path-selection after stripping the RD, to find the closest ingress to the private-MPLS-plane identified by the CPNH. This best path SHOULD be used to resolve a received private-label-route.

6. Example of Usecases

6.1. Mezanine transport layer in a Seamless-MPLS network

Typically service-routes in a MPLS network bind to the following entities that identify point-of-presence of a service:

- o Protocol Nexthop - PE loopback address (GPNH)
- o Service Label - PE advertised locally significant label that identifies the service

In this model, whenever a PE is taken out of service the GPNH changes, and Service-Label changes - which causes maintenance a heavy convergence event. Because the service-routes with massive-scale need to be readvertised with new service-label or PE-address.

An alternate model could be: to advertise the Service-routes with a protocol-nexthop of CPNH (without RD), with a forwarding-semantic of:

- o "Push <Private-Label>, and Forward to CPNH"

This model fully decouples the service-layer from the transport-layer identifiers, by making the Service-routes refer to the CPNH and Private-Labels. Thus the underlying transport-layer can change (nodes representing a Private-label can be added or removed) without any changes to the service-routes. Which present good scaling properties for the network.

This model also allows anycast traffic forwarding to any resource in the network. Multiple PEs can advertise the same Private-Label to identify a specific service (e.g. peering with an AS) they are offering.

Once the service-route traffic enters the private-FIB-layer, at the closest entry-point determined by path-selection of CPNH auto-discovery routes; then the Private-Labels (with pre-determined values) pushed will determine the loose hop path taken by the traffic and also the destination-resource.

6.2. Service Forwarding Helper usecase

In a virtualized environment a Service-PE node (that comprises of a vCP and multiple vFPs) can mirror MPLS labels (GL1) in its global MPLS-FIB to a private forwarding context at an upstream node (SFH) with information on which vFPs are optimal exit-points for that label. Such that the SFH can optimally forward traffic to GL1 to the right vFPs, thus avoiding intra fabric traffic hops.

To do this, the service-PE advertises a private-label route with RD:GL1 to the SFH node. The route is advertised with a Multi-nexthop attribute with one or more legs that have a "Forward to SEPx" semantics. Where SEPx is one of many exit-points at the Service-PE node.

6.3. Standard BGP API to a MPLS network's forwarding-plane

This mechanism facilitates predictable (external-allocator determined) label-values, using a standard BGP-family as the API. It gives the external applications a separate MPLS-FIB to play with, totally separate from other applications.

This also avoids vendor specific-API dependencies for external-allocators (controller softwares), and vice-versa.

This mechanism also increases the overall MPLS label-space available in the network, because it creates per-app label-forwarding-contexts (namespaces), instead of reserving/splitting the global MPLS FIB among various applications.

6.4. Traffic engineering and Security advantages

- o Ability of ingress to steer mpls-traffic thru specific detour loose-hop nodes using predictable-labels' stack.
- o Provide label-spoofing protection at edge-nodes - by virtue of using separate mpls-forwarding-contexts
- o Allow private-MPLS label usage to spread across multiple-domains/ AS and work seamlessly with existing technologies like Inter-AS VPN option C.

7. IANA Considerations

This document makes following requests of IANA.

New BGP AFI code:

- o <TBD> for "MPLS"

Which will be used to create new BGP AFI-SAFI pairs:

- o MPLS Uni(SAFI:1),
- o MPLS VpnUni(SAFI:128)

.

New NLRI Route-types for these AFI SAFIs:

- o Type 1: Context-Nexthop-Discovery-route.
- o Type 2: Private-Label route

Note to RFC Editor: this section may be removed on publication as an RFC.

8. Security Considerations

Using separate mpls-forwarding-contexts for separate applications and stitching them into separate MPLS-planes increases the security attributes of the MPLS network.

9. Acknowledgements

The authors thank Jeffrey (Zhaohui) Zhang, Ron Bonica, Jeff Haas and John Scudder for the valuable discussions.

10. References

[MULTI_NH] <https://www.ietf.org/id/draft-kaliraj-idr-multinexthop-attribute-00.txt>

[RFC-4364] BGP/MPLS IP Virtual Private Networks (VPNs)

[RFC-5331] MPLS Upstream Label Assignment and Context-Specific Label Space

11. 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>>.

Authors' Addresses

Kaliraj Vairavakkalai
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: kaliraj@juniper.net

Minto Jeyananth
Juniper Networks, Inc.
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: kaliraj@juniper.net

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 1 July 2022

K. Vairavakkalai
M. Jeyanthan
Juniper Networks, Inc.
28 December 2021

BGP signalled MPLS-namespaces
draft-kaliraj-bess-bgp-sig-private-mpls-labels-04

Abstract

The MPLS-forwarding-layer in a core network is a shared resource. The MPLS FIB at nodes in this layer contains labels that are dynamically allocated and locally significant at that node.

For some usecases like upstream-label-allocation, it is useful to be able to create virtual private MPLS-forwarding-layers over this shared MPLS-forwarding-layer. This allows installing deterministic private label-values in the private-FIBs created at nodes participating in this private MPLS forwarding-layer, while preserving the "locally significant" nature of the underlying shared 'public' MPLS-forwarding-layer.

This specification describes the procedures to create such virtual private MPLS-forwarding layers (private MPLS-planes) using a new BGP family. And gives a few example use-cases on how this private forwarding-layers can be used.

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].

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 1 July 2022.

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 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	3
2. Motivation	3
3. Constructs and building blocks	4
3.1. Context Protocol Nexthop Address	4
3.2. MPLS context FIB	4
3.3. Context Label	5
3.4. Roles of nodes in a MPLS-plane	5
3.4.1. Edge-nodes (PLER)	5
3.4.2. Transit-nodes (PLSR)	5
3.5. Sending traffic into the MPLS plane	6
4. Terminology	6
5. BGP families, routes and encoding	7
5.1. New address-families for "MPLS namespace signaling" . . .	8
5.1.1. AFI: MPLS, SAFI: 128	8
5.1.2. AFI: MPLS, SAFI: 1	8
5.2. Routes and Operational procedures	9
5.2.1. "Context-Nexthop" discovery route	9
5.2.2. MPLS namespace "Private Label" routes	10
6. Example of Usecases	13
6.1. Mezanine transport layer in a Seamless-MPLS network . . .	13
6.2. Service Forwarding Helper usecase	14
6.3. Standard BGP API to a MPLS network's forwarding-plane . .	14
6.4. Traffic engineering and Security advantages	14
7. IANA Considerations	15
8. Security Considerations	15
9. Acknowledgements	15
10. Normative References	15
Authors' Addresses	16

1. Introduction

The MPLS-forwarding-layer in a core network is a shared resource. The MPLS FIB at nodes in this layer contains labels that are dynamically allocated and locally significant at that node.

For some usecases like upstream-label-allocation, it is useful to be able to create virtual private MPLS-forwarding-layers over this shared MPLS-forwarding-layer. This allows installing deterministic private label-values in the private-FIBs in this private forwarding-layer, while preserving the "locally significant" nature of the underlying shared 'public' MPLS-forwarding-layer.

It can be noted that, mechanism described in this document is nothing but a [RFC4364] style BGP VPN where the FEC is MPLS-Label, instead of IP-prefix. This document defines new address-families (AFI: MPLS, SAFI: VPN-Unicast, Unicast) and associated signaling mechanisms to create and use MPLS forwarding-contexts in a network. The concepts of MPLS-Context-tables and upstream allocation are described in [RFC5331].

BGP speakers participating in the private MPLS FIB layer create instances of "MPLS forwarding-context" FIBs, which are identified using a "Context-Protocol-Nexthop (CPNH)". A Context-label MAY be advertised in conjunction with the Context Protocol Nexthop (CPNH) using new BGP address-family to other speakers.

2. Motivation

A provider's core network consists of a global-domain (default forwarding-tables in P and PE nodes) that is shared by all tenants in the network and may also contain multiple private user-domains (e.g. VRF route tables).

The global MPLS forwarding-layer can be viewed as the collection of all default MPLS forwarding-tables. This global MPLS Fib layer contains labels locally significant to each node. The "local-significance of labels" gives the nodes freedom to participate in MPLS-forwarding with whatever label-ranges they can support in forwarding hardware.

In emerging usecases some applications using the MPLS-network may benefit from a "static labels" view of the MPLS-network. In some other usecases, a standard mechanism to do Upstream label-allocation is beneficial.

It is desirable to leave the global MPLS FIB layer intact, and build private MPLS FIB-layers on top of it to achieve these requirements. The private-MPLS-FIBs can then be used by the applications as desired. The private MPLS-FIBs need to be created only at the nodes in the network where predictable label-values (external label allocation) is desired. E.g. P-routers that need to act as a "Detour-nodes" or "Service-Forwarding-Helpers" that need to mirror service-labels.

In other words, provisioning of these private MPLS-FIBs can be gradual and can co-exist with nodes not supporting the feature described in this document. These private-MPLS-FIBs can be stitched together using either the Context-labels over the existing shared MPLS-network tunnels, or 'private' context-interfaces - to form the "private MPLS-FIB layer".

An application can then install the routes with desired label-values in the private forwarding-contexts with desired forwarding-semantics.

3. Constructs and building blocks

The building-blocks that construct a private MPLS plane are described in this section.

3.1. Context Protocol Nexthop Address

A private MPLS plane (just "MPLS plane" here-after) is identified by an IP-address called Context Protocol Nexthop (CPNH). This address is unique in the core-network, like any other loopback address.

A loopback-address uniquely identifies a specific node in the network, and we call it Global Protocol Nexthop (GPNH) in this document. The CPNH address uniquely identifies a "MPLS-plane".

Each node that has forwarding-context for a MPLS-plane MUST be configured with the same CPNH but a different RD, such that the RD:CPNH will uniquely identify that node in the MPLS-plane.

3.2. MPLS context FIB

An instance of a MPLS forwarding-table at a node in the private MPLS-plane. This Private MPLS FIB contains the private-label routes.

A node can have context-FIB for multiple MPLS-planes. The same label-value can have a different forwarding-semantic in each MPLS-plane. Thus the applications using that MPLS-plane get a deterministic label-value independent of other applications using other MPLS-planes.

The terms "private MPLS FIB-layer" and "private MPLS-plane" are used interchangeably in this document.

3.3. Context Label

A context-label is a non-reserved dynamically allocated label, that is installed in the global MPLS FIB, and points to a MPLS-Context-FIB. The Context-Label have forwarding semantics as follows in the global MPLS-FIB:

Context-Label -> Pop and Lookup in MPLS-Context-Fib

Advertising the "Context-Label in conjunction with the GPNH" tells the network how to reach a "RD:CPNH".

3.4. Roles of nodes in a MPLS-plane

The node roles in a MPLS-plane can be classified into "edge nodes" (call them PLER) or "transit-nodes" (call them PLSR).

3.4.1. Edge-nodes (PLER)

Private Label Edge-routers (PLER) have MPLS context-FIB that belong to the MPLS-plane. They advertise the presence of this context-FIB using transport layer address families like BGP-CT [BGP-CT] or BGP-LU, and private-label routes from this FIB are advertised using new BGP AFI/SAFI described in this document.

3.4.2. Transit-nodes (PLSR)

These are just Border-nodes that do label-swap forwarding for the Context-Labels they see in the Context-Protocol-Nexthop advertisement routes (BGP-CT or BGP-LU) going thru them. They basically stitch/extend the label switched path to a PLER's CPNH when they re-advertise the CPNH routes with nexthop-self.

PLSRs don't have MPLS context-FIBs. PLSRs don't have Context Protocol-Nexthop. Because they don't have Private label routes to originate.

However a node in the network can play both roles, of PLER and PLSR.

3.5. Sending traffic into the MPLS plane

At a PLER, MPLS-traffic arriving with private-label hits the correct private MPLS-FIB by virtue of either arriving on a "private network-interface" that is attached to the MPLS context-FIB, or arriving with a "Context-label" on a network-interface attached to the global MPLS-FIB.

To send data traffic into this private MPLS plane, the sender MUST use as handle either a "Context-label" advertised by a node or a "Private-interface" owned by the MPLS context-FIB at the node. The MPLS context-FIB is created for an application that needs a private MPLS-plane.

The Context-Label is the only dynamic label-value the application needs to learn from the network (PLER node it is connected to), to be able to use the private MPLS-plane. The application can chose predictable value for the labels to be programmed in the private MPLS-FIBs.

Once the packet enters the private MPLS plane at an edge-node (PLER), the node will forward the packet to the next node (PLSR or PLER), by pushing the Context-label advertised by that next-node, and the transport-label to reach that node's GPNH. This will repeat until the packet reaches the PLER's private MPLS-FIB that originated that private MPLS-label.

At each PLER in the MPLS-plane, the private-label value remains the same, and points towards the same resource attached to the MPLS-plane. This allows the applications using the MPLS-network a static-labels view of the resources attached to the private MPLS-plane.

At each PLSR in the MPLS-plane, the context-label value will change (be swapped in forwarding), but is transparent to the application.

4. Terminology

P-router : A Provider core router, also called a LSR

LSR : Label Switch Router (pure transport node speaking LDP, RSVP etc)

PLSR: a BGP-CT or BGP-LU transit node in a private MPLS-plane, that does label-swap forwarding for Context-Label.

PLER: an edge node in a private MPLS-plane. It has a forwarding-context for private-labels.

Detour-router : A BGP border node that is used as a loose-hop in a traffic-engineered path

PE-router : Provider Edge router, that hosts a service (Internet, L3VPN etc)

SE-router : Service Edge router. Same as PE.

SFH-router : Service Forwarding Helper. A node helping an SE-router with service-traffic forwarding, using Service-routes mirrored by the SE.

MPLS FIB : MPLS Forwarding table

Global MPLS FIB : Global MPLS Forwarding table, to which shared-interfaces are connected

Private MPLS FIB : Private MPLS Forwarding table, to which private-interfaces are connected

Private MPLS FIB Layer (Private MPLS plane): The group of Private MPLS FIBs in the network, connected together via Context-Labels

Context-Label : Locally-significant Non-reserved label pointing to a private MPLS FIB

Context nexthop IP-address (CPNH) : An IP-address that identifies the "Private MPLS FIB Layer". RD:CPNH identifies a Private MPLS FIB at a specific BGP node.

Global nexthop IP-address (GPNH) : Global Protocol Nexthop address. E.g. a loopback address used as transport tunnel end-point.

5. BGP families, routes and encoding

This section describes the new constructs defined by this document.

5.1. New address-families for "MPLS namespace signaling"

This document defines a new AFI: "MPLS" (IANA code TBD). And two new address-families, using SAFIs 128 and 1. These address families are used to signal "MPLS namespaces" in BGP. To send or receive routes of these address families, these AFI, SAFI pair of values MUST be negotiated in Multiprotocol Extensions capability described in RFC4760 [RFC4760]

5.1.1. AFI: MPLS, SAFI: 128

This address-family is used to exchange private label-routes in private MPLS-FIBs at routers that are connected using a common network interface. The private label route has NLRI prefix format "RD:PrivateLabel" and contains Route-Target extended-community identifying the private-FIB-Layer (VPN) the route belongs to. The nexthop of these routes is set to either the GPNH or the CPNH of the BGP-speaker advertising the RFC-8277 label.

Any transport layer protocol is used to advertise the Context-Label that the receiving router uses to send traffic into the private MPLS-FIB. The Context-Label installed in the global MPLS-FIB points to the private MPLS-FIB. The Context-Label is required when the connecting-interface is a shared common interface that terminates into the global MPLS FIB.

Routes of this address-family can be sent with either IPv4 or IPv6 nexthop. The type of nexthop is inferred from the length of the nexthop.

When the length of Next Hop Address field is 24 (or 48) the nexthop address is of type VPN-IPv6 with 8-octet RD set to zero (potentially followed by the link-local VPN-IPv6 address of the next hop with an 8-octet RD).

When the length of Next Hop Address field is 12 the nexthop address is of type VPN-IPv4 with 8-octet RD.

5.1.2. AFI: MPLS, SAFI: 1

This address-family is used to exchange private label-routes in private MPLS-FIBs to routers that are connected using a private network-interface.

Because the interface is private, and terminates directly into the private MPLS-FIB, a Context-Label is not required to access the private MPLS-FIB and NLRI prefix format is just "PrivateLabel/24", without the RD.

Routes of this address-family can be sent with either IPv4 or IPv6 nexthop. The type of nexthop is inferred from the length of the nexthop.

When the length of Next Hop Address field is 16 (or 32) the nexthop address is of type IPv6 (potentially followed by the link-local IPv6 address of the next hop).

When the length of Next Hop Address field is 4 the nexthop address is a 4 octet IPv4 address.

5.2. Routes and Operational procedures

5.2.1. "Context-Nexthop" discovery route

The Context-NH discovery route may be a BGP-LU or [BGP-CT] family route that carries CPNH in the "Prefix" portion of the NLRI. And the Context-Label is carried in the "Label" field in the [RFC8277] format NLRI.

This route is advertised with the following path-attributes:

- * BGP Nexthop attribute (code 14, MP_REACH) carrying GPNH address.
- * Route-Target extended community, identifying the Transport class, if applicable.

The "Context-Nexthop discovery route" is originated by each speaker who acts as a PLER. The "RD:Context-nexthop" uniquely identifies the private-MPLS-FIB at the speaker. The "Context-nexthop address" uniquely identifies the private-MPLS-plane in the network. The Context-Label advertised in this route has a local forwarding semantic of "Pop, Lookup in Private-MPLS-FIB".

A BGP speaker readvertising a BGP-CT Context-Nexthop for RD:CPNH discovery-route MUST follow the mechanisms described in [BGP-CT]. Specifically when re-advertising with "next-hop self" MUST allocate a new Label with a forwarding semantic of "Swap Received-Context-Label, Forward to Received-GPNH". This extends reachability to the CPNH across tunnel domains.

5.2.2. MPLS namespace "Private Label" routes

The Private Label routes are carried in the new address-family "MPLS VpnUnicast" (AFI:MPLS, SAFI:128) aka "MPLS-namespace signaling", defined in this document.

The NLRI format follows the specifications in [RFC8277], with the "Prefix" portion of the NLRI comprising of the RD and "Private MPLS Label" encoded as shown below.

In a MP_REACH_NLRI attribute whose AFI/SAFI is MPLS/128, the "Length" field will be 112 bits or less, comprising of the Label, RD and "Private MPLS Label".

In a MP_REACH_NLRI attribute whose AFI/SAFI is MPLS/1, the "Length" field will be 48 bits or less, comprising of the Label, and "Private MPLS Label".

NLRI Prefix (Private Label route, AFI:MPLS, SAFI:128)

This picture shows NLRI format when the RFC-8277 Multiple Labels Capability is not used:

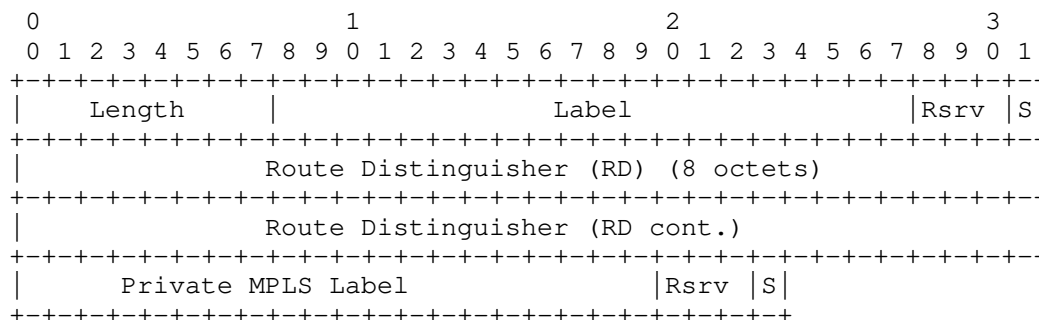


Fig 1: RFC-8277 NLRI with one Label.

- Length:

The Length field consists of a single octet. It specifies the length in bits of the remainder of the NLRI field.

In a MP_REACH_NLRI attribute whose AFI/SAFI is MPLS/128, the "Length" field will be 112 bits or less, comprising of the Label, RD and "Private MPLS Label".

As specified in [RFC4760], the actual length of the NLRI field will be the number of bits specified in the Length field,

rounded up to the nearest integral number of octets.

- Label:
The Label field is a 20-bit field containing an MPLS label value (see [RFC3032]). This label is locally significant, downstream allocated at the speaker identified in the BGP Nexthop field in MP_REACH_NLRI (code 14). This label is pushed in nexthop of the route installed in MPLS context FIB at receiving router.
- Route Distinguisher (RD):
The 8 byte Route Distinguisher as specified in [RFC4760].
- Private MPLS Label:
The "Private MPLS Label" field is a 20-bit field containing an MPLS label value (see [RFC3032]). This is an upstream assigned MPLS label, used as destination of route installed in MPLS context FIB at the receiving router.
- Rsrv:
This 3-bit field SHOULD be set to zero on transmission and MUST be ignored on reception.
- S:
This 1-bit field MUST be set to one on transmission and MUST be ignored on reception.

Attributes on this route:

- * BGP Nexthop attribute (code 14, MP_REACH) carrying a GPNH address.
(OR)
- * The Multi-nexthop attribute [MULTI-NH] with forwarding-semantic:
 - "Forward to RD:CPNH"
- * Route-Target extended-community, identifying the private FIB-layer

MultiNexthop BGP-attribute (Private Label route)

MultiNH.Num-Nexthops = 1
FwdSemanticsTLV.FwdAction = Forward
NHDescrTLV.NhopDescrType = RD:CPNH or GPNH

Fig 2: MultiNexthop attr of Private Label route

A speaker MAY readvertise a private-label-route without changing the Nexthop (RD:CPNH) carried in it, if the speaker is a pure PLSR.

If it does alter the nexthop to SelfRD:CPNH, it SHOULD act as a PLER, and for e.g. originate a "Context-Nexthop discovery route" for prefix "SelfRD:CPNH".

Even if the speaker sets nexthop-address to Self because of regular BGP readvertisement-rules, Label Prefix MUST NOT be altered, and the received NLRI "RD:Private-Label1" MUST be re-advertised as-is. Such that value of label "Private-Label1" doesn't change while the packet traverses multiple nodes in the private-MPLS-FIB-layer.

The Route-target attached to the route is the one identifying the private MPLS FIB layer (VPN). The Private-label routes resolve over the Context-nexthop route that belong to the same VPN.

A node receiving a "Private-Label route" RD:L1 MUST install the label L1 in the private MPLS Forwarding-context identified by the Route-Target attached to the route.

The label route MUST be installed with forwarding-semantic as specified in the received Multi-nexthop attribute. As an example, a Detour node MAY receive the private-label-route with a forwarding-semantic of "Forward to RD:CPNH" operation. And an Egress node MAY receive a private-label-route with a forwarding-semantic pointing to a resource it houses. Note that such a Private-label BGP-route MAY be received from external-application also.

5.2.2.1. Resolving received Private Label-routes

A node receiving a "Context-nexthop discovery route" MUST be capable of using either the CPNH or the RD:CPNH carried in the NLRI, to resolve other routes received with this CPNH address or RD:CPNH in the "Nexthop-attributes".

The receiver of a private-label route MUST recursively resolve the received nexthop (RD:CPNH) over the Context-Nexthop discovery-route for prefix "RD:CPNH" to determine the label stack "Context-Label, Transport-Label" to push, so that the MPLS packet with private-label reaches the private MPLS FIB originating the route.

If a node receives multiple "Context-nexthop discovery route" for a CPNH, it SHOULD run path-selection after stripping the RD, to find the closest ingress to the private-MPLS-plane identified by the CPNH. This best path SHOULD be used to resolve a received private-label-route.

6. Example of Usecases

6.1. Mezanine transport layer in a Seamless-MPLS network

Typically service-routes in a MPLS network bind to the following entities that identify point-of-presence of a service:

- * Protocol Nexthop - PE loopback address (GPNH)
- * Service Label - PE advertised locally significant label that identifies the service

In this model, whenever a PE is taken out of service the GPNH changes, and Service-Label changes - which causes maintenance a heavy convergence event. Because the service-routes with massive-scale need to be readvertised with new service-label or PE-address.

An alternate model could be: to advertise the Service-routes with a protocol-nexthop of CPNH (without RD), with a forwarding-semantic of:

- * "Push <Private-Label>, and Forward to CPNH"

This model fully decouples the service-layer from the transport-layer identifiers, by making the Service-routes refer to the CPNH and Private-Labels. Thus the underlying transport-layer can change (nodes representing a Private-label can be added or removed) without any changes to the service-routes. Which present good scaling properties for the network.

This model also allows anycast traffic forwarding to any resource in the network. Multiple PEs can advertise the same Private-Label to identify a specific service (e.g. peering with an AS) they are offering.

Once the service-route traffic enters the private-FIB-layer, at the closest entry-point determined by path-selection of CPNH auto-discovery routes; then the Private-Labels (with pre-determined values) pushed will determine the loose hop path taken by the traffic and also the destination-resource.

6.2. Service Forwarding Helper usecase

In a virtualized environment a Service-PE node (that comprises of a vCP and multiple vFPs) can mirror MPLS labels (GL1) in its global MPLS-FIB to a private forwarding context at an upstream node (SFH) with information on which vFPs are optimal exit-points for that label. Such that the SFH can optimally forward traffic to GL1 to the right vFPs, thus avoiding intra fabric traffic hops.

To do this, the service-PE advertises a private-label route with RD:GL1 to the SFH node. The route is advertised with a Multi-nexthop attribute with one or more legs that have a "Forward to SEPx" semantics. Where SEPx is one of many exit-points at the Service-PE node.

6.3. Standard BGP API to a MPLS network's forwarding-plane

This mechanism facilitates predictable (external-allocator determined) label-values, using a standard BGP-family as the API. It gives the external applications a separate MPLS-FIB to play with, totally separate from other applications.

This also avoids vendor specific-API dependencies for external-allocators (controller softwares), and vice-versa.

This mechanism also increases the overall MPLS label-space available in the network, because it creates per-app label-forwarding-contexts (namespaces), instead of reserving/splitting the global MPLS FIB among various applications.

6.4. Traffic engineering and Security advantages

- * Ability of ingress to steer mpls-traffic thru specific detour loose-hop nodes using predictable-labels' stack.
- * Provide label-spoofing protection at edge-nodes - by virtue of using separate mpls-forwarding-contexts
- * Allow private-MPLS label usage to spread across multiple-domains/ AS and work seamlessly with existing technologies like Inter-AS VPN option C.

7. IANA Considerations

This document makes following requests of IANA.

New BGP AFI code ("Address Family Numbers" registry):

* 16399 for "MPLS Namespaces"

Note to RFC Editor: this section may be removed on publication as an RFC.

8. Security Considerations

Using separate mpls-forwarding-contexts for separate applications and stitching them into separate MPLS-planes increases the security attributes of the MPLS network.

9. Acknowledgements

The authors thank Jeffrey (Zhaohui) Zhang, Ron Bonica, Jeff Haas and John Scudder for the valuable discussions.

10. Normative References

- [BGP-CT] Vairavakkalai, K., "BGP Classful Transport Planes", 25 August 2021, <<https://tools.ietf.org/html/draft-kaliraj-idr-bgp-classful-transport-planes-12#section-11.3>>.
- [MULTI-NH] Vairavakkalai, K., "BGP MultiNexthop attribute", 28 December 2021, <<https://tools.ietf.org/html/draft-kaliraj-idr-multinexthop-attribute-04>>.
- [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>>.
- [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>>.
- [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>>.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, DOI 10.17487/RFC5331, August 2008, <<https://www.rfc-editor.org/info/rfc5331>>.

[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>>.

Authors' Addresses

Kaliraj Vairavakkalai
Juniper Networks, Inc.
1133 Innovation Way,
Sunnyvale, CA 94089
United States of America

Email: kaliraj@juniper.net

Minto Jeyananth
Juniper Networks, Inc.
1133 Innovation Way,
Sunnyvale, CA 94089
United States of America

Email: minto@juniper.net

INTERNET-DRAFT

Intended Status: Proposed Standard

Expires: December 29, 2017

N. Malhotra
A. Sajassi
A. Pattekar
(Cisco)
A. Lingala
(AT&T)
June 27, 2017

Extended Mobility Procedures for EVPN-IRB
draft-malhotra-bess-evpn-irb-extended-mobility-00

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-LAG based multi-homing further introduces scenarios that require additional consideration from mobility perspective. Intent of this draft is to enumerate a set of design considerations applicable to mobility across EVPN IRB use cases and define 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	5
3.	Mobility Use Cases	5
3.1	VM MAC+IP Move	5
3.2	VM IP Move to new MAC	6
3.2.1	VM Reload	6
3.2.2	MAC Sharing	6
3.2.3	Problem	6
3.3	VM MAC move to new IP	7
3.3.1	Problem	7
4.	Multi-homed hosts via EVLAG	9
5.	Design Considerations	10
6.	Solution Components	11
6.1	Sequence Number Inheritance	11
6.2	MAC Sharing	12
6.3	Multi-homing Mobility Synchronization	13
7.	Requirements for Sequence Number Assignment	13
7.1	LOCAL MAC-IP learning	13
7.2	LOCAL MAC learning	14
7.3	Remote MAC OR MAC-IP Update	14
7.4	REMOTE (SYNC) MAC update	14
7.5	REMOTE (SYNC) MAC-IP update	15

7.6	Inter-op	15
8.	Duplicate Host Detection	15
8.1	Duplicate IP detection Procedure	16
8.2	Duplicate Host Recovery	17
8.2.1	Route Un-freezing CLI	17
8.2.2	Route clearing CLI	17
9.	Security Considerations	18
10.	IANA Considerations	18
11.	References	18
11.1	Normative References	18
11.2	Informative References	18
12.	Acknowledgements	18
	Authors' Addresses	18
	Appendix A	18

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. 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 VXLAN. It is also largely independent of the EVPN IRB solution being based on symmetric OR asymmetric IRB design. In other words, ideas presented in this

draft should apply equally to symmetric and asymmetric EVPN IRB designs, as well as MPLS and VXLAN based overlays.

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].

ARP is widely referred to in this document. This is simply for ease of reading, and as such, these references are equally applicable to ND (neighbor discovery) as well.

Term GW used widely in the document refers to an IRB GW that is doing routing and bridging between an access network and an EVPN enabled overlay network.

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 IRB GW. 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 goes over IRB mobility use cases taken into consideration while defining mobility procedures proposed in later sections:

- 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

3.1 VM MAC+IP Move

This is the baseline case, wherein a VM move results in both VM MAC and IP moving together with no change in MAC-IP binding across a move. Existing MAC mobility defined in RFC 7432 may be extrapolated

to apply to corresponding MAC+IP route to support this mobility scenario.

3.2 VM IP Move to new MAC

This is the case, where a VM move results in VM IP moving to a new MAC binding.

3.2.1 VM Reload

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

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

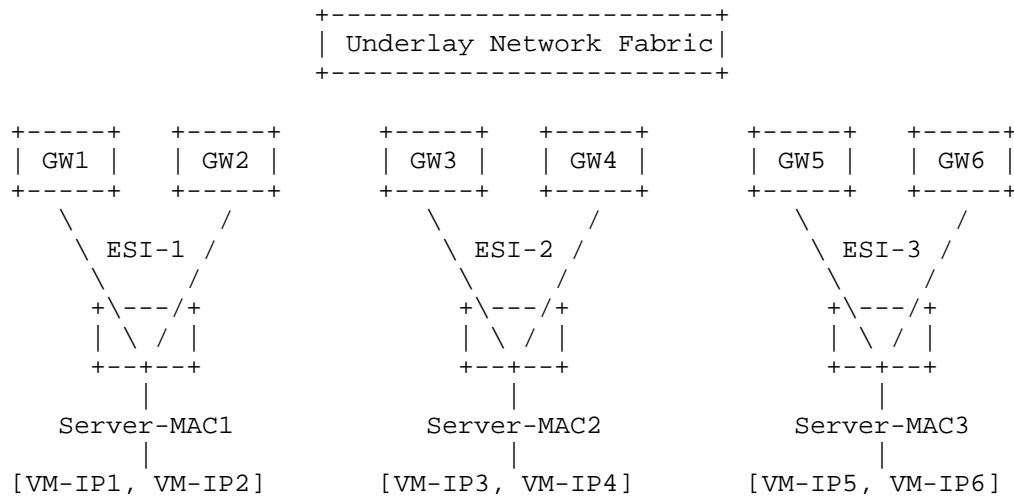
3.2.2 MAC Sharing

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, host VMs 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 VM IP move, in such scenarios, could result in new MAC association for the VM 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.



As an example, consider the above topology with host VMs sharing the physical server MAC. In steady state, [IP1, MAC1] route is learnt at [GW1, GW2] and advertised to remote GWs with a sequence number N. Now, VM-IP1 is moved to Server-MAC2. ARP or ND based local learning at [GW3, GW4] 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 clear 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.

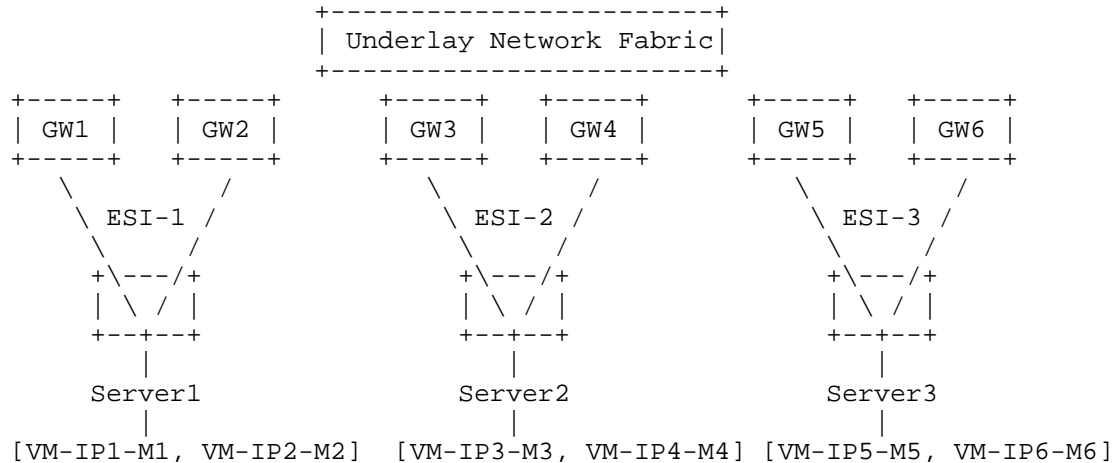
3.3 VM MAC move to new IP

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

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 opposed to the older existing MAC reachability.

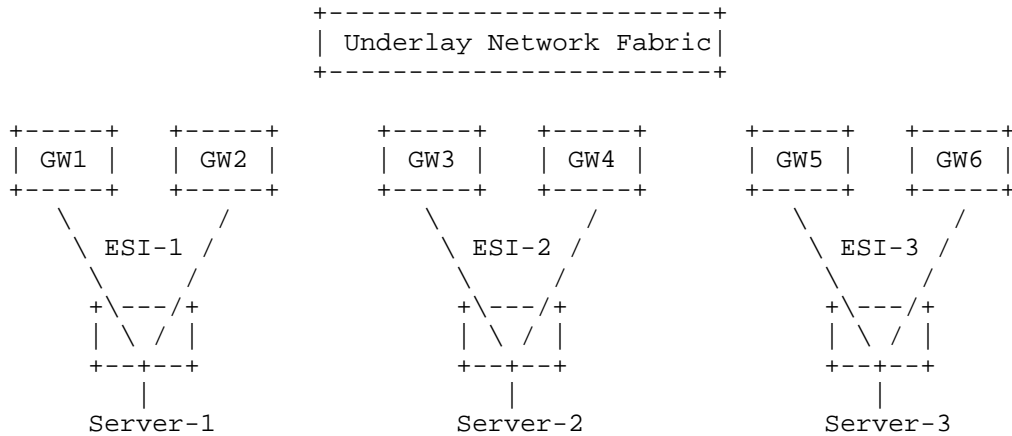


As an example, IP1-M1 is learnt locally at [GW1, GW2] and currently advertised to remote hosts with a sequence number N. Consider a scenario where a VM with MAC M1 is re-provisioned at server 2, however, as part of this re-provisioning, assigned a different IP address say IP7. [IP7, M1] is learnt as a new route at [GW3, GW4] and advertised to remote GWs 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 RFC-7432. 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, GW1/GW2, on learning new reachability for [IP7, M1] via GW3/GW4 MUST probe and delete any local IPs associated with MAC M1, such as [IP1, M1] in the above example.

4. Multi-homed hosts via EVLAG



Consider an EVPN-IRB overlay network with hosts multi-homed to two or more leaf GW devices via an MC-LAG ESI. MAC and ARP entries learnt on a local ESI may also be synchronized across the multi-homing GW devices sharing this ESI. 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 Ethernet segment (ESI) may be learnt via local learning and / or sync from another GW device sharing the same ESI.

For a host that is multi-homed to multiple GW devices via an MC-LAG interface, local learning of host MAC and MAC-IP at each GW 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 GW on the MC-LAG interface. As a result, sequence number mobility attribute value assigned to a locally learnt MAC or MAC-IP route (as per RFC 7432) 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, GW1 following deletion of the remote route from [GW3, GW4], while being learnt on GW2 prior to deletion of remote route from [GW3, GW4]. If so, GW1 would process local host route learning as a new route and assign a sequence number of 0, while GW2 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 [GW3, GW4]. Inconsistent sequence numbers advertised from multi-homing devices

leads to various issues such as:

- o Ambiguity with respect to how the remote ToRs should handle paths with same ESI and different sequence numbers. A remote ToR may not program ECMP paths if it receives routes with different sequence numbers from a set of multi-homing GWs 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, GW2 would drop a remote route received for the same host with sequence number N (as its local sequence number is N+1), while GW1 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 MUST be advertised with the same sequence number by all GW devices that the ESI is multi-homed to. In other words, 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.

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 ESI multi-homed to multiple GW devices, hence requires sequence numbers to be synchronized across multi-homing GW 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 is always accompanied by a LOCAL MAC learn, however, learning 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. It is therefore goal of solution presented here to address 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)

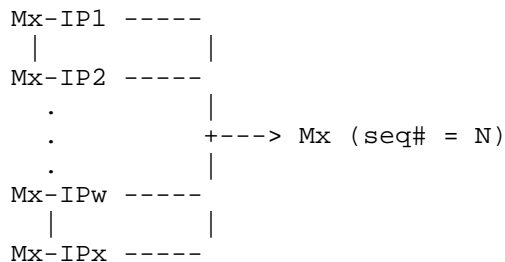
As a result, both parent MAC and child MAC-IP routes share one common sequence number associated with the parent MAC route. Doing so ensures that a single sequence number attribute carried in a combined MAC+IP route represents sequence number for both a MAC only route as well as a MAC+IP route, and hence 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.

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 GW1. IPx moving to a new physical server behind GW2 results in IPx being associated with MAC Mz. A new local Mz-IPx route resulting from this move at GW2 must now be advertised with a sequence number higher than N. This is so that GW devices, including GW1, GW2, and other remote GW devices that are part of the overlay can clearly determine and program the most recent MAC binding and reachability for the IP. GW1 on receiving this new Mz-IPx route with sequence number say, N+1, for symmetric IRB case, would update IPx reachability via GW2 in forwarding, for asymmetric IRB case, would update IPx's ARP binding to Mz, 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 GW2 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

it's 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 the shared ESI MUST be advertised with the same sequence number by all GW devices that the ESI 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 GW to achieve local switching. Local and SYNC route learning can happen in any order. Local MAC-IP routes advertised by all multi-homing GW devices sharing the ESI 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 the 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 a LOCAL route for MAC Mx:

- o GW MUST trigger probe and deletion procedure for all LOCAL IPs associated with MAC Mx
- o GW 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 MUST be at least equal to corresponding SYNC MAC sequence number that is received.
- 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 ESI, 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 MUST be at least equal to corresponding SYNC MAC sequence number that is received.
- 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 GW nodes in the overlay network follow the above sequence number assignment procedure, and the GW 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 GW implementation non-compliant with this proposal 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 GW, 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 GW would now result in the MAC (and hence all MAC-IP) sequence numbers incremented from the above computed remote MAC sequence number.

8. Duplicate Host Detection

This section specifies additional considerations and requirements, incremental to the baseline duplicate MAC detection procedure described in RFC 7432.

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.

However, 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.

8.1 Duplicate IP detection Procedure

- o If the IP is also associated with a different remote MAC "Mz", MUST be higher than "Mz" sequence number
- o On learning a LOCAL MAC-IP route Mx-IPx, check if there is an existing REMOTE route for IPx with a different MAC association, say, Mz-IPx
- o 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
- o If so, count this as an "IP move" count for IPx, independent of the MAC
- o Duplicate detection procedure described in RFC 7432 can now be applied to IPx independent of MAC

- o If "N" IP moves are detected within "M" seconds for IPx, treat the corresponding LOCAL MAC-IP route for IPx as DUPLICATE
- o Apply duplicate handling on local MAC-IP route by alerting the operator and freezing EVPN RT-2 advertisements for it

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

8.2 Duplicate Host Recovery

Once a MAC or IP is marked as DUPLICATE, corrective action must be taken to un-provision one of the duplicate MAC or IP. Once one of the duplicate hosts is un-provisioned, normal operation would not resume until the duplicate MAC 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 fast recovery to normal operation.

8.2.1 Route Un-freezing CLI

Unfreezing the DUPLICATE MAC or IP via a CLI can be leveraged to automatically clear the duplicate MAC / IP entries following un-provisioning of the duplicate host:

- o If the duplicate host is un-provisioned at the location where it was NOT marked as DUPLICATE, unfreezing the route from the other location will result in automatic clearing of local routes on receiving higher sequence number routes following the un-freeze.
- o 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 and no re-learning would happen.

8.2.2 Route clearing CLI

Alternatively a CLI can be provided to clear the local route, to be executed AFTER the duplicate host is un-provisioned.

9. Security Considerations

10. IANA Considerations

11. References

11.1 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>>.

11.2 Informative References

12. Acknowledgements

Authors would like to thank Vibov Bhan and Patrice Brisset for feedback and comments through the process.

Authors' Addresses

Neeraj Malhotra
Cisco
EMail: nmalhotr@cisco.com

Ali Sajassi
Cisco
EMail: sajassi@cisco.com

Aparna Pattekar
Cisco
Email: apjoshi@cisco.com

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

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.

INTERNET-DRAFT

Intended Status: Proposed Standard

N. Malhotra, Ed.
(Arrcus)
A. Sajassi
A. Pattekar
(Cisco)
A. Lingala
(AT&T)
J. Rabadan
(Nokia)
J. Drake
(Juniper Networks)

Expires: Jul 19, 2019

Jan 15, 2019

Extended Mobility Procedures for EVPN-IRB
draft-malhotra-bess-evpn-irb-extended-mobility-04

Abstract

The 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. Intent of this draft is to enumerate a set of design considerations applicable to mobility across EVPN IRB use cases and define 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	5
3.	Mobility Use Cases	6
3.1	VM MAC+IP Move	6
3.2	VM IP Move to new MAC	6
3.2.1	VM Reload	6
3.2.2	MAC Sharing	6
3.2.3	Problem	7
3.3	VM MAC move to new IP	8
3.3.1	Problem	8
4.	EVPN All Active multi-homed ES	10
5.	Design Considerations	11
6.	Solution Components	12
6.1	Sequence Number Inheritance	12
6.2	MAC Sharing	13
6.3	Multi-homing Mobility Synchronization	14

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

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-INTER-SUBNET] 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 ARP is widely referred to in this document. This is simply for ease of reading, and as such, these references are equally applicable to ND (neighbor discovery) as well.
- o GW: used widely in the document refers to an IRB GW that is doing routing and bridging between an access network and an EVPN enabled overlay network.
- 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 ES: EVPN Ethernet Segment
- 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 IRB GW. 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 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

3.1 VM MAC+IP Move

This is the baseline case, wherein a VM move results in both VM 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 VM IP Move to new MAC

This is the case, where a VM move results in VM IP moving to a new MAC binding.

3.2.1 VM Reload

A VM reload or an orchestrated VM move that results in VM being re-spawned at a new location may result in VM getting a new MAC assignment, while maintaining existing IP address. This results in a VM 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, host VMs 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 VM IP move, in such scenarios (for e.g., to a new physical server), could result in new MAC association for the VM 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.

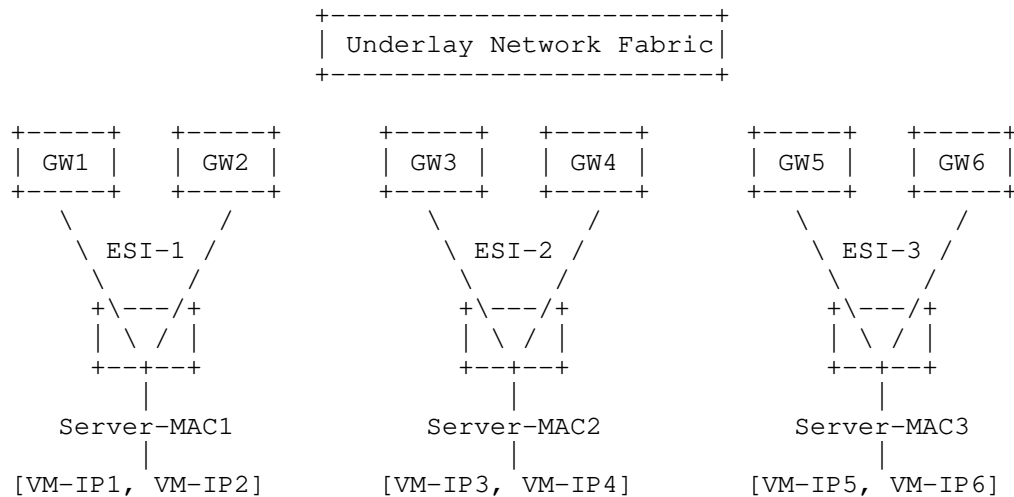


Figure 1

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 [GW1, GW2] and advertised to remote GWs with a sequence number N. Now, VM-IP1 is moved to Server-MAC2. ARP or ND based local learning at [GW3, GW4] 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.

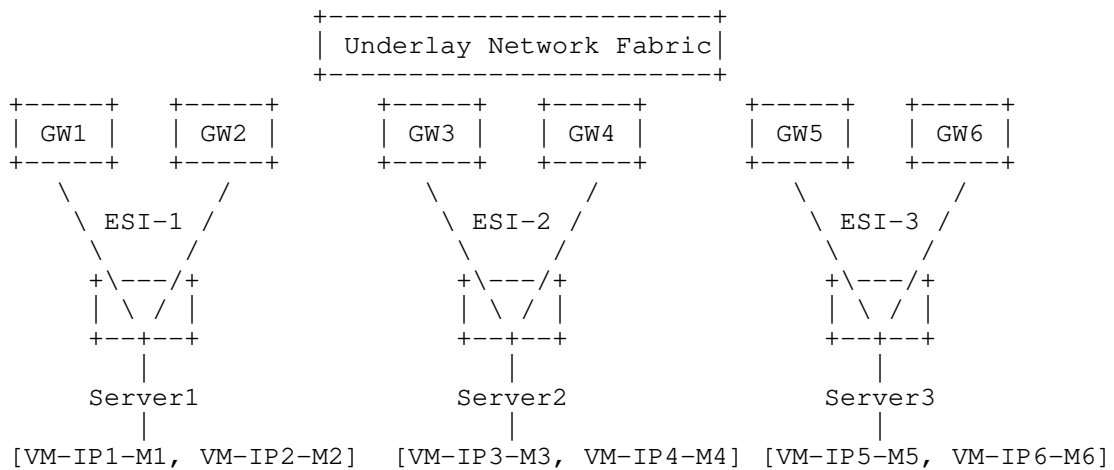
A clear sequence number assignment procedure needs to be defined to unambiguously determine the most recent IP reachability, IP to MAC binding, and MAC reachability for such a MAC sharing scenario.

3.3 VM MAC move to new IP

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

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 opposed to the older existing MAC reachability.



As an example, IP1-M1 is learnt locally at [GW1, GW2] and currently advertised to remote hosts with a sequence number N. Consider a scenario where a 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 [GW3, GW4] and advertised to remote GWs 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, GW1/GW2, on learning new reachability for [IP7, M1] via GW3/GW4 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

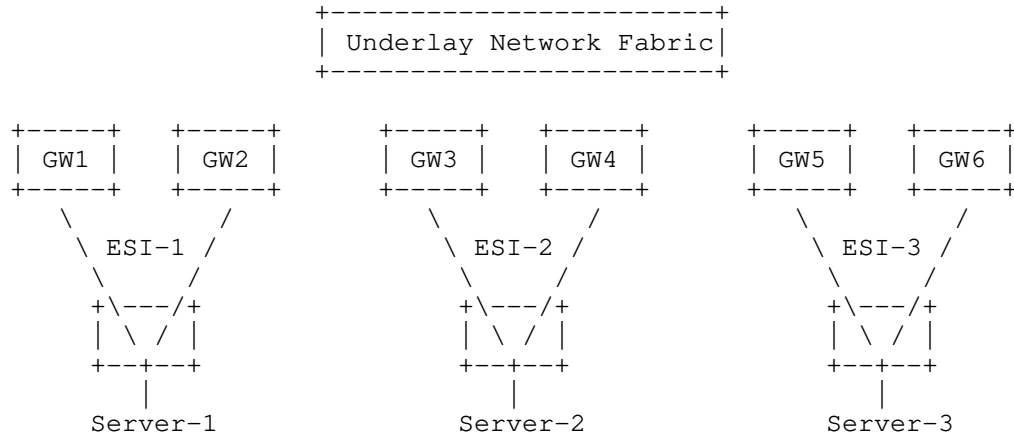


Figure 2

Consider an EVPN-IRB overlay network shown in Figure 2, with hosts multi-homed to two or more leaf GW devices via an all-active multi-homed ES. MAC and ARP entries learnt on a local ESI may also be synchronized across the multi-homing GW devices sharing this ESI. 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 Ethernet segment (ES) may be learnt via local learning and / or sync from another GW device sharing the same ES.

For a host that is multi-homed to multiple GW devices via an all-active ES interface, local learning of host MAC and MAC-IP at each GW 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 GW on the MC-LAG interface. As a result, sequence number mobility attribute value assigned to a locally learnt MAC or MAC-IP route (as per RFC 7432) 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, GW1 following deletion of the remote route from [GW3, GW4], while being learnt on GW2 prior to deletion of remote route from [GW3, GW4]. If so, GW1 would process local host route learning as a new route and assign a sequence number of 0, while GW2 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 [GW3, GW4]. Inconsistent sequence numbers advertised from multi-homing devices introduces ambiguity with respect to sequence number based mobility procedures across the overlay.

- o Ambiguity with respect to how the remote ToRs should handle paths with same ESI and different sequence numbers. A remote ToR may not program ECMP paths if it receives routes with different sequence numbers from a set of multi-homing GWs 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, GW2 would drop a remote route received for the same host with sequence number N (as its local sequence number is $N+1$), while GW1 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 MUST be advertised with the same sequence number by all GW devices that the ESI is multi-homed to. In other words, 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.

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 ESI multi-homed to multiple GW devices, hence requires sequence numbers to be synchronized across multi-homing GW 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. It is therefore goal of solution presented here to address 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

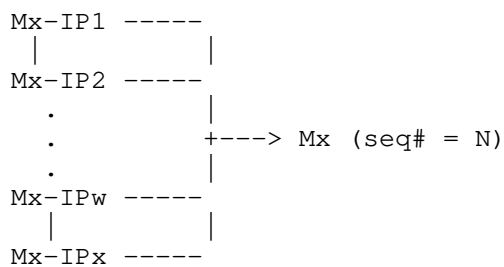
sequence number from Mx.

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 GW1. IPx moving to a new physical server behind GW2 results in IPx being associated with MAC Mz. A new local Mz-IPx route resulting from this move at GW2 must now be advertised with a sequence number higher than N. This is so that GW devices, including GW1, GW2, and other remote GW devices that are part of the overlay can clearly determine and program the most recent MAC binding and reachability for the IP. GW1, on receiving this new Mz-IPx route with sequence number say, N+1, for symmetric IRB case, would update IPx reachability via GW2 in forwarding, for asymmetric IRB case, would update IPx's ARP binding to Mz. In addition, GW1 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 GW2 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 it's 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 the shared ESI MUST be advertised with the same sequence number by all GW devices that the ESI 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 GW to achieve local switching. Local and SYNC route learning can happen in any order. Local MAC-IP routes advertised by all multi-homing GW devices sharing the ESI 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 the 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 a LOCAL route for MAC Mx:

- o GW MUST trigger probe and deletion procedure for all LOCAL IPs associated with MAC Mx
- o GW 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 ESI, 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 GW nodes in the overlay network follow the above sequence number assignment procedure, and the GW 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 GW 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 GW, 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 GW would now result in the MAC (and hence all MAC-IP) sequence numbers incremented from the above computed remote MAC sequence number.

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 GW.

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 GW 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 GW 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-02, work in progress, April 2017, <<https://tools.ietf.org/html/draft-ietf-bess-evpn-proxy-arp-nd-02>>.

[EVPN-INTER-SUBNET] Sajassi et al., "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-03,

work in progress, Feb 2017,
<<https://tools.ietf.org/html/draft-ietf-bess-evpn-inter-subnet-forwarding-03>>.

[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 and comments through the process.

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 WorkGroup
Internet-Draft
Intended status: Standards Track
Expires: September 11, 2017

S. Mohanty
A. Sreekantiah
D. Rao
Cisco Systems
K. Patel
Arrcus, Inc
March 10, 2017

BGP Multipath in Inter-AS Option-B
draft-mohanty-bess-multipath-interas-00

Abstract

By default, The Border Gateway Protocol, BGP only installs the best-path to the IP Routing Table. BGP multi-path is a well known feature that enables installation of multiple paths to the IP Routing Table. This is done to achieve load balancing while forwarding traffic. For a path to be eligible as a multi-path, certain criteria need to be fulfilled. Inter-AS VPNs are commonly deployed to span organizations across Service Provider boundaries. In this draft, we describe an issue relating to multi-path load balancing that can arise in an Option B Inter-AS Deployment. With the help of a representative topology, we illustrate the problem and then present two simple schemes as the solution to the problem. We also note as a matter of independent interest that the same underlying issue is applicable to deployments that employ next-hop-self behavior (implicit or explicit) downstream and the multi-path feature upstream.

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 <http://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 September 11, 2017.

Copyright 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	2
2. Requirements Language	3
3. Topology notation	3
4. Problem Description	4
5. BGP ADDpath with the non-unique RD case	4
6. BGP Labeled unicast with Add-Path	5
7. BGP Multi-path Inter-As Solution 1	5
8. BGP Multi-path Inter-As Solution 2	5
9. Protocol Considerations	6
10. Operational Considerations	6
11. Security Considerations	6
12. Acknowledgements	6
13. References	6
13.1. Normative References	6
13.2. Informative References	7
Authors' Addresses	7

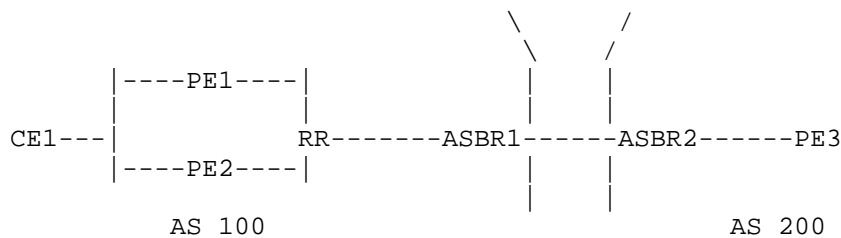
1. Introduction

By Default BGP [RFC4271] only advertises the best-path to a peer and also installs the best-path to the IP Routing Table (RIB) and thereby to the Forwarding Information Base (FIB). BGP multi-path is a feature where more than one received BGP route, rather than only the one corresponding to the BGP best-path, are installed in the IP Routing Table and the Forwarding Information Base. This offers benefits of load balancing, efficient utilization of system resources network-wide, and enabling high throughput for traffic flows which would be lacking otherwise. It also has the added benefit of providing redundancy in case one of the BGP paths are withdrawn due to a link going down or some other event. Often vendors have a

configurable knob which dictates how many paths to a given destination can be installed in the forwarding.

BGP Multi-path is widely deployed in practice and when augmented with the Demilitarized Link Bandwidth (DMZ LB) [I-D.ietf-idr-link-bandwidth] can be used to provide unequal cost load balancing as per user control.

The BGP best-path algorithm proceeds through a well-known and deterministic selection mechanism in determining the best-path. Typically, a path is deemed eligible as a multi-path, if it encounters a tie with the best-path, when it is determined that the IGP cost (metric) to the BGP next-hop is the same, as per the BGP best-path algorithm [RFC4271]. In addition, two paths, which match all criteria until the IGP metric but have the same next-hop IP address cannot both be considered as multi-paths. This is regardless of EBGP or IBGP rules. In this draft we point out an issue that limits the benefits of multi-path deployments arising out of above restrictions when the BGP path is propagated across Inter-AS Option B [RFC4364] Autonomous System Boundary Routers (ASBRs).



Inter-AS Option B.

Figure 1

2. 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 [RFC2119].

3. Topology notation

In the Figure 1. above, we consider a typical Inter-AS Option B topology, ASBR1 peering with ASBR2 over the inter-AS eBGP link. A VPN, vpn has a presence in both the Autonomous Systems, on all the PE routers shown; i.e. a Virtual routing Forwarding (VRF) tables associated with the VPN vpn exists at each of the Provider Routers

shown. A dual-homed CE, CE1 is peering with PE1 and PE2 respectively in the context of vrf VRF1.

Denote the Route-Distinguisher (RD) of the vrf VRF1 configured in PE1 by RD1. Denote the Route-Distinguisher of the vrf VRF2 configured in PE2 by RD2. Assume that CE1 advertises an ipv4 prefix p, at ASBR1, the received VPN route prefix will be RD1:p and RD2:p, with next-hops PE1 and PE2 respectively, with the vpn (service) label as L1 and L2 respectively.

4. Problem Description

As per EBGp rules at the advertising ASBR, ASBR1, the next-hop will be reset to the ASBR1 itself. This causes the two routes RD1:p and RD2:p to be advertised to the receiving AS, AS2, with the mandatory attribute, the next-hop which points to ASBR1.

Let's say the swapped label for RD1:p and RD2:p at ASBR1 is L1 and L2 respectively. If ASBR2 does not reset the next-hop (usual behavior), then the two paths will be received at PE3 with the same next-hop, i.e. ASBR1. If ASBR2 does reset the next-hop, then the two paths will be received at PE3 with the next-hop set to ASBR2.

In either case above, the two paths received at PE3 have the same next-hop, even though the labels are different. As explained earlier, if two received BGP paths have the same next-hop, then both of them cannot be eligible for multi-paths at the same time. This means that at the PE3, only one of the routes will be installed in the forwarding.

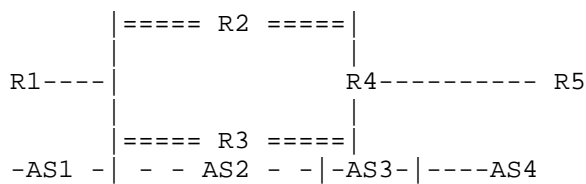
In the Figure 1 above, even though the advertising AS (AS 100) has path redundancy, this is not visible to AS 200, and therefore load balancing cannot be done at ASBR1. Note that this is different from the classic same RD problem which one often encounters in the Route-Reflector context.

5. BGP ADDpath with the non-unique RD case

The above scenario is described in the context of the unique-RD case. Now consider the case when one has non-unique RDs configured for the vpn VRF at PE1 and PE2, and BGP Add-Path [RFC7911] is used to propagate the paths to AS200 via RR, ASBR1 and ASBR2 respectively. In this case, the ASBR1 resets the next-hop to itself in both of the add-paths thus ensuring that the two add-paths cannot be installed as primary and backup in the FIB at PE3 in AS200.

6. BGP Labeled unicast with Add-Path

A similar situation exists for non-VPN labeled traffic. Figure 2 shows a simple ebgp topology, in which R1 is in AS 1, R2 and R3 are in AS 2, R4 is in AS 3, and R5 is in AS 4. A labeled unicast [RFC3107] prefix, p, is being advertised from R1 to R5. Add-Path is configured at R4 and R5 and the capability is negotiated. Both R2 and R3, will set the next-hop to themselves. When R4 receives the prefix p from R2 and R3, the situation is similar to the add-path scenario for the VPN case as described in the earlier section. As a result only one of the paths will be advertised to R5.



Inter-AS Option B.

Figure 2

7. BGP Multi-path Inter-As Solution 1

The first solution is to consider the uniqueness of the label and the next-hop by considering the tuple (next-hop, label). This translates to (ASBR1, L1) and (ASBR2, L2) and therefore they can be distinguished. However many existing deployments today consider only the next-hop as the key. Therefore this solution requires upgrade to existing deployment software. An independent issue is that there should be no implications on hashing the weights assigned to the paths in the FIB due to the dependency on the label.

8. BGP Multi-path Inter-As Solution 2

The second solution is to inject two loopback ip addresses at ASBR1 into the IBGP of the receiving AS corresponding to the PE1 and PE2's configured ip address or loopbacks that are in the next-hop attribute of the vpn routes RD1:p and RD2:p. These loopback addresses need to be injected into the IGP of the receiving AS. Also ASBR2 needs to be configured with a static route pointing to ASBR1 for this purpose. Alternatively, ASBR1 can redistribute these loopbacks into EBGP. This is also equivalent to doing next-hop-self. The above solution won't require any software upgrade. However it will require the

implementation to support policy and may have security implications since routes need to be leaked from one AS to the other.

9. Protocol Considerations

No Protocol Changes are necessary

10. Operational Considerations

Any of the two methods above can be adopted. A note may be made that these solutions also are applicable to EVPN [RFC7432]

11. Security Considerations

This document raises no new security issues for L3VPN.

12. Acknowledgements

The authors would like to thank Yuri Tsier for his feedback and useful discussions

13. References

13.1. Normative References

- [I-D.ietf-idr-extcomm-iana]
Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", draft-ietf-idr-extcomm-iana-02 (work in progress), December 2013.
- [I-D.ietf-idr-link-bandwidth]
Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", draft-ietf-idr-link-bandwidth-06 (work in progress), January 2013.
- [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>>.
- [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, <<http://www.rfc-editor.org/info/rfc4271>>.
- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <<http://www.rfc-editor.org/info/rfc4360>>.

- [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>>.

13.2. Informative References

- [RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <<http://www.rfc-editor.org/info/rfc3107>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<http://www.rfc-editor.org/info/rfc4364>>.
- [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, <<http://www.rfc-editor.org/info/rfc6624>>.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<http://www.rfc-editor.org/info/rfc7911>>.

Authors' Addresses

Satya Ranjan Mohanty
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: satyamoh@cisco.com

Arjun Sreekantiah
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: asreekan@cisco.com

Dhananjaya Rao
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dhrao@cisco.com

Keyur Patel
Arrcus, Inc

Email: keyur@arrcus.com

BESS WorkGroup
Internet-Draft
Intended status: Informational
Expires: March 14, 2018

S. Mohanty
A. Sreekantiah
D. Rao
Cisco Systems
K. Patel
Arrcus, Inc
September 10, 2017

BGP Multipath in Inter-AS Option-B
draft-mohanty-bess-multipath-interas-01

Abstract

By default, The Border Gateway Protocol, BGP only installs the best-path to the IP Routing Table. BGP multi-path is a well known feature that enables installation of multiple paths to the IP Routing Table. This is done to achieve load balancing while forwarding traffic. For a path to be eligible as a multi-path, certain criteria need to be fulfilled. Inter-AS VPNs are commonly deployed to span organizations across Service Provider boundaries. In this draft, we describe an issue relating to multi-path load balancing that can arise in an Option B Inter-AS Deployment. With the help of a representative topology, we illustrate the problem and then present two simple schemes as the solution to the problem. We also note as a matter of independent interest that the same underlying issue is applicable to deployments that employ next-hop-self behavior (implicit or explicit) downstream and the multi-path feature upstream.

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 March 14, 2018.

Copyright 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 (<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. Topology notation	3
4. Problem Description	4
5. BGP ADDpath with the non-unique RD case	4
6. BGP Labeled unicast with Add-Path	5
7. BGP Multi-path Inter-As Solution 1	5
8. BGP Multi-path Inter-As Solution 2	5
9. Protocol Considerations	6
10. Operational Considerations	6
11. Security Considerations	6
12. Acknowledgements	6
13. References	6
13.1. Normative References	6
13.2. Informative References	7
Authors' Addresses	7

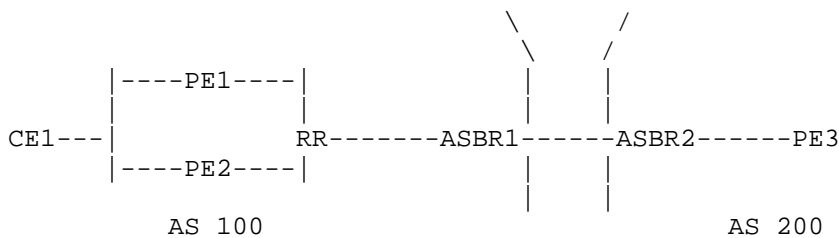
1. Introduction

By Default BGP [RFC4271] only advertises the best-path to a peer and also installs the best-path to the IP Routing Table (RIB) and thereby to the Forwarding Information Base (FIB). BGP multi-path is a feature where more than one received BGP route, rather than only the one corresponding to the BGP best-path, are installed in the IP Routing Table and the Forwarding Information Base. This offers benefits of load balancing, efficient utilization of system resources network-wide, and enabling high throughput for traffic flows which would be lacking otherwise. It also has the added benefit of providing redundancy in case one of the BGP paths are withdrawn due to a link going down or some other event. Often vendors have a

configurable knob which dictates how many paths to a given destination can be installed in the forwarding.

BGP Multi-path is widely deployed in practice and when augmented with the Demilitarized Link Bandwidth (DMZ LB) [I-D.ietf-idr-link-bandwidth] can be used to provide unequal cost load balancing as per user control.

The BGP best-path algorithm proceeds through a well-known and deterministic selection mechanism in determining the best-path. Typically, a path is deemed eligible as a multi-path, if it encounters a tie with the best-path, when it is determined that the IGP cost (metric) to the BGP next-hop is the same, as per the BGP best-path algorithm [RFC4271]. In addition, two paths, which match all criteria until the IGP metric but have the same next-hop IP address cannot both be considered as multi-paths. This is regardless of EBGP or IBGP rules. In this draft we point out an issue that limits the benefits of multi-path deployments arising out of above restrictions when the BGP path is propagated across Inter-AS Option B [RFC4364] Autonomous System Boundary Routers (ASBRs).



Inter-AS Option B.

Figure 1

2. 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 [RFC2119].

3. Topology notation

In the Figure 1. above, we consider a typical Inter-AS Option B topology, ASBR1 peering with ASBR2 over the inter-AS eBGP link. A VPN, vpn has a presence in both the Autonomous Systems, on all the PE routers shown; i.e. a Virtual routing Forwarding (VRF) tables associated with the VPN vpn exists at each of the Provider Routers

shown. A dual-homed CE, CE1 is peering with PE1 and PE2 respectively in the context of vrf VRF1.

Denote the Route-Distinguisher (RD) of the vrf VRF1 configured in PE1 by RD1. Denote the Route-Distinguisher of the vrf VRF2 configured in PE2 by RD2. Assume that CE1 advertises an ipv4 prefix p, at ASBR1, the received VPN route prefix will be RD1:p and RD2:p, with next-hops PE1 and PE2 respectively, with the vpn (service) label as L1 and L2 respectively.

4. Problem Description

As per EBGp rules at the advertising ASBR, ASBR1, the next-hop will be reset to the ASBR1 itself. This causes the two routes RD1:p and RD2:p to be advertised to the receiving AS, AS2, with the mandatory attribute, the next-hop which points to ASBR1.

Let's say the swapped label for RD1:p and RD2:p at ASBR1 is L1 and L2 respectively. If ASBR2 does not reset the next-hop (usual behavior), then the two paths will be received at PE3 with the same next-hop, i.e. ASBR1. If ASBR2 does reset the next-hop, then the two paths will be received at PE3 with the next-hop set to ASBR2.

In either case above, the two paths received at PE3 have the same next-hop, even though the labels are different. As explained earlier, if two received BGP paths have the same next-hop, then both of them cannot be eligible for multi-paths at the same time. This means that at the PE3, only one of the routes will be installed in the forwarding.

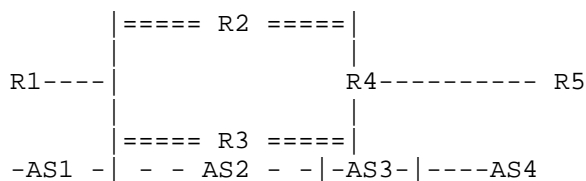
In the Figure 1 above, even though the advertising AS (AS 100) has path redundancy, this is not visible to AS 200, and therefore load balancing cannot be done at ASBR1. Note that this is different from the classic same RD problem which one often encounters in the Route-Reflector context.

5. BGP ADDpath with the non-unique RD case

The above scenario is described in the context of the unique-RD case. Now consider the case when one has non-unique RDs configured for the vpn VRF at PE1 and PE2, and BGP Add-Path [RFC7911] is used to propagate the paths to AS200 via RR, ASBR1 and ASBR2 respectively. In this case, the ASBR1 resets the next-hop to itself in both of the add-paths thus ensuring that the two add-paths cannot be installed as primary and backup in the FIB at PE3 in AS200.

6. BGP Labeled unicast with Add-Path

A similar situation exists for non-VPN labeled traffic. Figure 2 shows a simple ebgp topology, in which R1 is in AS 1, R2 and R3 are in AS 2, R4 is in AS 3, and R5 is in AS 4. A labeled unicast [RFC3107] prefix, p, is being advertised from R1 to R5. Add-Path is configured at R4 and R5 and the capability is negotiated. Both R2 and R3, will set the next-hop to themselves. When R4 receives the prefix p from R2 and R3, the situation is similar to the add-path scenario for the VPN case as described in the earlier section. As a result only one of the paths will be advertised to R5.



Inter-AS Option B.

Figure 2

7. BGP Multi-path Inter-As Solution 1

The first solution is to consider the uniqueness of the label and the next-hop by considering the tuple (next-hop, label). This translates to (ASBR1, L1) and (ASBR2, L2) and therefore they can be distinguished. However many existing deployments today consider only the next-hop as the key. Therefore this solution requires upgrade to existing deployment software. An independent issue is that there should be no implications on hashing the weights assigned to the paths in the FIB due to the dependency on the label.

8. BGP Multi-path Inter-As Solution 2

The second solution is to inject two loopback ip addresses at ASBR1 into the IBGP of the receiving AS corresponding to the PE1 and PE2's configured ip address or loopbacks that are in the next-hop attribute of the vpn routes RD1:p and RD2:p. These loopback addresses need to be injected into the IGP of the receiving AS. Also ASBR2 needs to be configured with a static route pointing to ASBR1 for this purpose. Alternatively, ASBR1 can redistribute these loopbacks into EBGP. This is also equivalent to doing next-hop-self. The above solution won't require any software upgrade. However it will require the

implementation to support policy and may have security implications since routes need to be leaked from one AS to the other.

9. Protocol Considerations

No Protocol Changes are necessary

10. Operational Considerations

Any of the two methods above can be adopted. A note may be made that these solutions also are applicable to EVPN [RFC7432]

11. Security Considerations

This document raises no new security issues for L3VPN.

12. Acknowledgements

The authors would like to thank Yuri Tsier for his feedback and useful discussions

13. References

13.1. Normative References

- [I-D.ietf-idr-extcomm-iana]
Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended Communities", draft-ietf-idr-extcomm-iana-02 (work in progress), December 2013.
- [I-D.ietf-idr-link-bandwidth]
Mohapatra, P. and R. Fernando, "BGP Link Bandwidth Extended Community", draft-ietf-idr-link-bandwidth-06 (work in progress), January 2013.
- [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>>.
- [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>>.
- [RFC4360] Sangli, S., Tappan, D., and Y. Rekhter, "BGP Extended Communities Attribute", RFC 4360, DOI 10.17487/RFC4360, February 2006, <<https://www.rfc-editor.org/info/rfc4360>>.

- [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>>.

13.2. Informative References

- [RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <<https://www.rfc-editor.org/info/rfc3107>>.
- [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>>.
- [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>>.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<https://www.rfc-editor.org/info/rfc7911>>.

Authors' Addresses

Satya Ranjan Mohanty
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: satyamoh@cisco.com

Arjun Sreekantiah
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: asreekan@cisco.com

Dhananjaya Rao
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: dhrao@cisco.com

Keyur Patel
Arrcus, Inc

Email: keyur@arrcus.com

BESS Working Group
Internet Draft
Category: Standard Track

Ali Sajassi
Gaurav Badoni
Priyanka Warade
Suresh Pasupula
Cisco Systems

Expires: January 2, 2017

July 2, 2017

L3 Aliasing and Mass Withdrawal Support for EVPN
draft-sajassi-bess-evpn-ip-aliasing-00.txt

Abstract

This draft proposes an extension to [RFC7432] to do Aliasing for Layer 3 routes that is needed for symmetric IRB to build a complete IP ECMP.

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	3
1.1	Terminology	3
2	IP Aliasing and Backup Path	4
2.1	Constructing Ethernet A-D per EVPN Instance Route	5
3	Fast Convergence for Routed Traffic	6
3.1	Constructing Ethernet A-D per Ethernet Segment Route	7
3.1.1	Ethernet A-D Route Targets	7
3.2	Avoiding convergence issues by syncing IP prefixes	7
3.3	Handling Silent Host	8
3.4	MAC Aging	8
4	Determining Reach-ability to Unicast IP Addresses	9
4.1	Local Learning	9
4.2	Remote Learning	9
4.2.1	Constructing MAC/IP Address Advertisement	9
4.2.2	Route Resolution	9
5	Forwarding Unicast Packets	9
6	Load Balancing of Unicast Packets	10
7	Security Considerations	10
8	IANA Considerations	10
9	References	10
9.1	Normative References	10
9.2	Informative References	10
	Authors' Addresses	10

1 Introduction

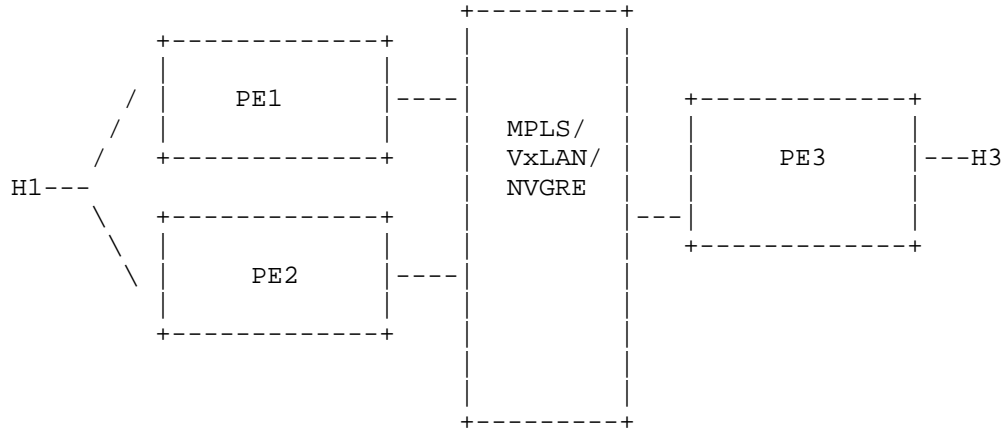


Figure 1: Inter-subnet traffic between Multihoming PEs and Remote PE

Consider a pair of multi-homing TORs PE1 and PE2. Let there be a host H1 attached to them. Consider another TOR PE3 and a host H3 attached to it.

With Asymmetric IRB, if H3 sends inter-subnet traffic to H1, routing will happen at PE3. PE3 will have the destination SVI and will trigger ARP if it does not have an ARP adjacency to H1. Finally routing lookup will resolve destination MAC to H1's MAC address. Furthermore, H1's MAC will point to a VxLAN ECMP to T1 and T2, either due to host route advertisement or MAC Aliasing as detailed in [RFC 7432].

With Symmetric IRB, if H3 sends inter-subnet traffic to H1, routing lookup will happen at PE3. PE3 will do a routing lookup in the L3VNI-VRF context and is not expected to have the destination SVI. Therefore at PE3, we need an IP ECMP list (PE1/PE2) to be built for H1's IP address for proper load balancing. If H1 is locally learnt only at one of the PEs, PE1 or PE2 due to port-channel hashing, we will not be able to build IP ECMP at PE3 as we do not do Aliasing for Layer 3 addresses.

This draft proposes an extension to do Aliasing for Layer 3 routes that is needed for symmetric IRB to build a complete IP ECMP.

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].

IRB: Integrated Routing and Bridging

IRB Interface: A virtual interface that connects the bridging module and the routing module on an NVE.

Broadcast Domain: 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: An instantiation of a broadcast domain on a MAC-VRF.

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

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.

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'.

LACP: Link Aggregation Control Protocol.

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.

2 IP Aliasing and Backup Path

Host IP and MAC routes are learnt by PEs on the access side via a control plane protocol like ARP. In case where a CE is multihomed to multiple PE nodes using a LAG and is running in All-Active Redundancy Mode, the Host IP will be learnt and advertised in the MAC/IP Advertisement only by the PE that receives the ARP packet. As a result, the remote PE sees only one next-hop for the Host IP and forwards traffic to that advertising PE. Hence, the remote PE is not be able to effectively load balance the traffic towards the multihomed Ethernet Segment.

To address this issue, concept of Aliasing that was introduced in RFC 7432 [RFC7432], can be extended for Layer 3 routes as well. The PE SHOULD advertise reachability to an L3 VRF instance on a given ES for IP addresses using the existing EAD/EVI route. In this case, the EVPN instance is the VRF table to which the host IP address belongs. This will henceforth be referred to as the IP-EAD/EVI route.

A remote PE that receives an IP route with a non reserved ESI SHOULD consider it reachable by all PEs that have advertised the IP-EAD/EVI advertisement route and the EAD/ES advertisement route containing the VRF Route-Targets for that ES. The EAD/ES route must have the Single-Active bit in the flags of the ESI Label extended community set to 0 for Aliasing to take effect.

The IP-EAD/EVI route cannot be used for route forwarding until the associated Ethernet A-D per ES route is received.

In case of Single-Active redundancy mode, the remote PE SHOULD use the IP-EAD/EVI route EVPN Layer 2 attribute extended community as mentioned in draft-ietf-bess-evpn-vpws-07 in combination with the EAD/ES route to determine the Backup Path for the IP addresses for the given IP VRF context. This alternate path SHOULD be installed as a backup path for the IP address.

2.1 Constructing Ethernet A-D per EVPN Instance Route

This draft proposes the advertisement of per EVI Ethernet A-D route for IP VRFs to enable Aliasing for IP addresses. The usage/construction of this route remains similar to that described in RFC 7432 with a few notable exceptions as below.

- * The Route-Distinguisher should be set to the corresponding L3VPN context.
- * The Ethernet Tag should be set to 0.
- * The L3 EAD/EVI SHOULD carry one or more IP VRF Route-Target (RT)

attributes.

- * The L3 EAD/EVI SHOULD carry the RMAC Extended Community attribute.
- * The MPLS Label usage should be as described in RFC 7432.

It is important to note that the prefix for a IP-EAD/EVI and L2-EAD/EVI may be identical. However, since the RD of the IP-EAD/EVI is set to the corresponding L3VPN context and the RD of the L2-EAD/EVI is set to the corresponding MAC-VRF context, the import will happen in the respective IP-VRFs and MAC-VRFs and hence, the prefix will not be overwritten.

3 Fast Convergence for Routed Traffic

In EVPN, Host IP reachability is learned via the BGP control plane over the MPLS network. All the hosts that are dually connected behind an ES are advertised by the PEs belonging to the redundancy group. A remote TOR receiving these host routes can lose reachability from any of the PEs either due to box reload or core failure or access failure for that PE.

BGP PIC functionality is the existing mechanism for fast convergence as described in <https://tools.ietf.org/html/draft-rtgwg-bgp-pic-02>. PIC feature doesn't solve the convergence issue for the access failure cases as the PEs are still reachable from the remote TOR.

To alleviate this, EVPN defines a mechanism to efficiently and quickly signal, to remote PE nodes, the need to update their forwarding tables upon the occurrence of a failure in connectivity to an Ethernet segment. This is done by having each PE advertise a set of one or more Ethernet A-D per ES routes for each locally attached Ethernet segment (refer to Section 3.1 below for details on how these routes are constructed). A PE may need to advertise more than one Ethernet A-D per ES route for a given ES because the ES may be in a multiplicity of EVIs and the RTs for all of these EVIs may not fit into a single route. Advertising a set of Ethernet A-D per ES routes for the ES allows each route to contain a subset of the complete set of RTs. Each Ethernet A-D per ES route is differentiated from the other routes in the set by a different Route Distinguisher (RD).

Upon failure in connectivity to the attached ES, the PE withdraws the corresponding set of Ethernet A-D per ES routes. This triggers all PEs that receive the withdrawal to update their next-hop adjacencies for all IP addresses across IP VRFs associated with the Ethernet segment in question. If no other PE has advertised an Ethernet A-D route for the same segment, then the PE that received the withdrawal simply invalidates the IP entries for that segment. Otherwise, the

PE updates its next-hop adjacencies accordingly.

These routes should be processed with higher priority than other MAC or MAC-IP withdrawals upon failure. Similar priority processing is needed even on the intermittent RRs.

This draft is addressing the mass withdrawal behavior for routed traffic. For Layer-2, please refer to Section 8.2 of RFC 7432.

3.1 Constructing Ethernet A-D per Ethernet Segment Route

This section describes the procedures used to construct the Ethernet A-D per ES route, which is used for fast convergence (as discussed above). The usage/construction of this route remains similar to that described in section 8.2.1. of RFC 7432 with a few notable exceptions as explained in following sections.

3.1.1 Ethernet A-D Route Targets

Each Ethernet A-D per ES route MUST carry one or more Route Target (RT attributes). The set of Ethernet A-D routes per ES MUST carry the entire set of IP VRF RTs for all the IP VRFs in addition to MAC VRF RTS for all the EVPN instance to which the Ethernet segment belongs.

3.2 Avoiding convergence issues by syncing IP prefixes

Consider a pair of multi-homing TORs PE1 and PE2. Let there be a host H1 attached to them. Consider another TOR PE3 and a host H3 attached to it.

If the host H1 is learnt on both the PEs, ECMP path list is formed on PE3 pointing to (PE1/PE2). Traffic from H3 to H1 is not impacted even if one of the TORs becomes unreachable as the path list gets corrected upon receiving the mass withdrawal route (Ethernet A-D segment).

Let us consider a case where H1 is locally learnt only on PE1 due to port-channel hashing. At PE3, H1 has ECMP path list (PE1/PE2) using Aliasing as described in section 2 of this draft. Traffic from H3 can reach either of the TORs PE1 or PE2.

On PE2, all the remote MAC-IP routes belonging to the same Ethernet Segment that are advertised by it's respective peers (PE1 in our example) should be synced and installed locally on PE2 but not advertised as local routes by BGP. When the traffic from H3 reaches PE2, it will be able forward the traffic to H1 without any convergence delay caused by triggering ARP/ND. In a scaled setup, the convergence can be significant as the ARP and ND resolution can take

a lot of time. So syncing the IPv4/6 prefixes that belong to same Ethernet Segment helps in solving convergence issues.

3.3 Handling Silent Host

In continuation with the discussion above, if the reachability of PE1 is lost, PE3 will update the ECMP list for H1 to PE2, upon receiving mass withdrawal from PE1. If host H1 is also withdrawn from PE1, then the same route is withdrawn from PE2 and PE3. Hence traffic from H3 to H1 is black-holed till H1 is re-learnt on PE2.

This black-holing can be much worse if the H1 behaves like a silent host. IP address of H1 will not be re-learnt on PE2 till H1 re-ARPs or some traffic triggers ARP for H1.

PE2 can detect the failure of PE1's reachability in following ways

- a) When core failure or box reload happens on PE1, next hop reachability to PE1 can be detected by the underlay routing protocols.
- b) Upon access failure, PE1 sends withdraws the EAD/ES Route and PE2 can use this as a trigger to detect failure.

Thus to avoid the black-holing, when PE2 detects loss of reachability to PE1, it should trigger ARP/ND for all remote IP prefixes received from it's ES peers (i.e. PE1) belonging to same Ethernet Segment across IP-VRF contexts. This will force host H1 to reply to the solicited ARP/ND from PE2 and refresh both MAC and IP for the corresponding host in its tables.

Even in core failure scenario on PE1, PE1 must withdraw all its local L2 connectivity, as L2 traffic should not be received by PE1. So when ARP/ND is triggered from PE2 the replies from host H1 can only be received by PE2. Thus H1 will be learnt as local route and also advertised from PE2.

It is recommended to have a staggered or delayed deletion of the IP routes from PE1, so that ARP/ND refresh can happen on PE2 before the deletion.

3.4 MAC Aging

PE1 would do ARP/ND refresh for H1 before it ages out. During this process, H1 can age out genuinely or due to the ARP/ND reply landing on PE2. PE1 must withdraw the local entry from BGP when H1 entry ages out. PE1 deletes the entry from the local forwarding only

when there are no remote synced entries.

4 Determining Reach-ability to Unicast IP Addresses

4.1 Local Learning

The procedures for local learning do not change from [RFC7432].

4.2 Remote Learning

The procedures for remote learning do not change from [RFC7432].

4.2.1 Constructing MAC/IP Address Advertisement

The procedures for constructing MAC/IP Address Advertisement do not change from RFC 7432

4.2.2 Route Resolution

If the ESI field is set to reserved values of 0 or MAX-ESI, the the IP route resolution MUST be based on the MAC-IP route alone.

If the ESI field is set to a non-reserved ESI, the IP route resolution MUST happen only when both the MAC-IP route and the associated set of Ethernet AD per ES routes have been received. To illustrate this with an example, consider a pair of multi-homed TORs PE1 and PE2 connected to an Ethernet Segment. ES1 in an all-active redundancy mode. A given host with IP address H1 is learnt by PE1 but not by PE2. When the MAC-IP advertisement route from PE1 and a set of EAD/ES and Layer 3 EAD/EVI routes from PE1 and PE2 are received, PE3 can forward traffic destined to H1 to both PE1 and PE2.

If after (1) PE1 withdraws EAD/ES, then PE3 will forward the said traffic to PE2 only.

If after (1) PE2 withdraws EAD/ES, then PE3 will forward the said traffic to PE1 only.

If after (1) PE1 withdraws the MAC-IP route, then PE3 will do delayed deletion of H1, as described in section 3.3.

If after (1) PE2 advertised the MAC-IP route, but PE1 withdraws it, PE3 will continue forwarding to both PE1 and PE2 as long as it has the EAD/ES and the Layer 3 EAD/EVI route from both.

5 Forwarding Unicast Packets

Please refer to Section 5 in the draft-ietf-bess-evpn-inter-subnet-forwarding-01

6 Load Balancing of Unicast Packets

The procedures for load balancing of Unicast Packets do not change from [RFC7432]

7 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 [ietf-evpn-overlay] are equally applicable.

8 IANA Considerations

9 References

9.1 Normative References

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC1776] Crocker, S., "The Address is the Message", RFC 1776, April 1 1995.
- [TRUTHS] Callon, R., "The Twelve Networking Truths", RFC 1925, April 1 1996.

9.2 Informative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Suresh Pasupula
Cisco
Email: spasupula@cisco.com

Gaurav Badoni
Cisco
Email: gbadoni@cisco.com

Priyanka Warade
Cisco
Email: pwarade@cisco.com

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

A. Sajassi, Ed.
G. Badoni
P. Warade
S. Pasupula
L. Krattiger
Cisco Systems
J. Drake, Ed.
Juniper
J. Rabadan, Ed.
Nokia
7 March 2022

EVPN Support for L3 Fast Convergence and Aliasing/Backup Path
draft-sajassi-bess-evpn-ip-aliasing-04

Abstract

This document proposes an EVPN extension to allow several of its multihoming functions, fast convergence and aliasing/backup path, to be used in conjunction with inter-subnet forwarding.

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	3
1.1. Ethernet Segments for Host Routes in Symmetric IRB . . .	3
1.2. Inter-subnet Forwarding for Prefix Routes in the Interface-less IP-VRF-to-IP-VRF Model	4
1.3. Ethernet Segments for Prefix routes in IP-VRF-to-IP-VRF use-cases	5
1.3.1. IP Aliasing for EVPN IP Prefix routes	5
1.3.2. Centralized Routing Model	6
1.4. Terminology and Conventions	8
2. Ethernet Segments for L3 Aliasing/Backup Path and Fast Convergence	9
3. IP Aliasing and Backup Path	10
3.1. Constructing the IP A-D per EVI Route	11
4. Fast Convergence for Routed Traffic	12
4.1. Constructing IP A-D per Ethernet Segment Route	13
4.1.1. IP A-D per ES Route Targets	13
4.2. Avoiding convergence issues by synchronizing IP prefixes	13
4.3. Handling Silent Host MAC/IP route for IP Aliasing	14
4.4. MAC Aging	14
5. Determining Reachability to Unicast IP Addresses	14
5.1. Local Learning	15
5.2. Remote Learning	15
5.3. Constructing the EVPN IP Routes	15
5.3.1. Route Resolution	15
6. Forwarding Unicast Packets	15
7. Load Balancing of Unicast Packets	16
8. IP Aliasing and Unequal ECMP for IP Prefix Routes	16
9. Security Considerations	17
10. IANA Considerations	17
11. Contributors	17
12. Acknowledgments	17
13. References	17
13.1. Normative References	17
13.2. Informative References	18
Authors' Addresses	18

1. Introduction

This document proposes an EVPN extension to allow several of its multihoming functions, fast convergence and aliasing/backup path, to be used in conjunction with inter-subnet forwarding. It re-uses the existing EVPN routes, the Ethernet A-D per ES and the Ethernet A-D per EVI routes, which are used for these multihoming functions. In particular, there are three use-cases that could benefit from the use of these multihoming functions:

- a. Inter-subnet forwarding for host routes in symmetric IRB [RFC9135].
- b. Inter-subnet forwarding for prefix routes in the interface-less IP-VRF-to-IP-VRF model [RFC9136].
- c. Inter-subnet forwarding for prefix routes when the ESI is used exclusively as an L3 construct [RFC9136].

1.1. Ethernet Segments for Host Routes in Symmetric IRB

Consider a pair of multi-homing PEs, PE1 and PE2, as illustrated in Figure 1. Let there be a host H1 attached to them. Consider PE3 and a host H3 attached to it.

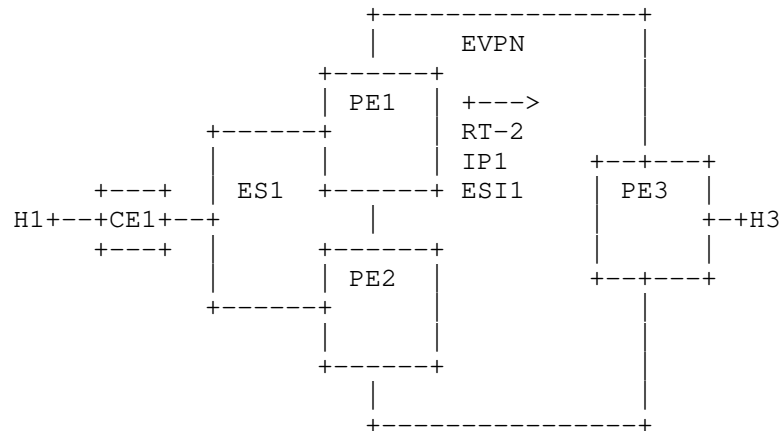


Figure 1: Inter-subnet traffic between Multihoming PEs and Remote PE

With Asymmetric IRB [RFC9135], if H3 sends inter-subnet traffic to H1, routing will happen at PE3. PE3 will be attached to the destination IRB interface and will trigger ARP/ND requests if it does not have an ARP/ND adjacency to H1. A subsequent routing lookup will resolve the destination MAC to H1's MAC address. Furthermore, H1's

MAC will point to an ECMP EVPN destination on PE1 and PE2, either due to host route advertisement from both PE1 and PE2, or due to Ethernet Segment MAC Aliasing as detailed in [RFC7432].

With Symmetric IRB [RFC9135], if H3 sends inter-subnet traffic to H1, a routing lookup will happen at PE3's IP-VRF and this routing lookup will not yield the destination IRB interface and therefore MAC Aliasing is not possible. In order to have per-flow load balancing for H3's routed traffic to H1, an IP ECMP list (to PE1/PE2) needs to be associated to H1's host route in the IP-VRF route-table. If H1 is locally learned only at one of the multi-homing PEs, PE1 or PE2, due to LAG hashing, PE3 will not be able to build an IP ECMP list for the H1 host route.

With the extension described in this document, PE3's IP-VRF becomes Ethernet-Segment-aware and builds an IP ECMP list for H1 based on the advertisement of ES1 along with H1 in a MAC/IP route and the availability of ES1 on PE1 and PE2.

1.2. Inter-subnet Forwarding for Prefix Routes in the Interface-less IP-VRF-to-IP-VRF Model

In the Interface-less IP-VRF-to-IP-VRF model described in [RFC9136] there is no Overlay Index and hence no recursive resolution of the IP Prefix route to either a MAC/IP Advertisement or an Ethernet A-D per ES/EVI route, which means that the fast convergence and aliasing/backup path functions are disabled. The recursive resolution of an IP Prefix route to an Ethernet A-D per ES/EVI route is already described in [RFC9136].

The scenario illustrated in Figure 2 will be used to explain the procedures.

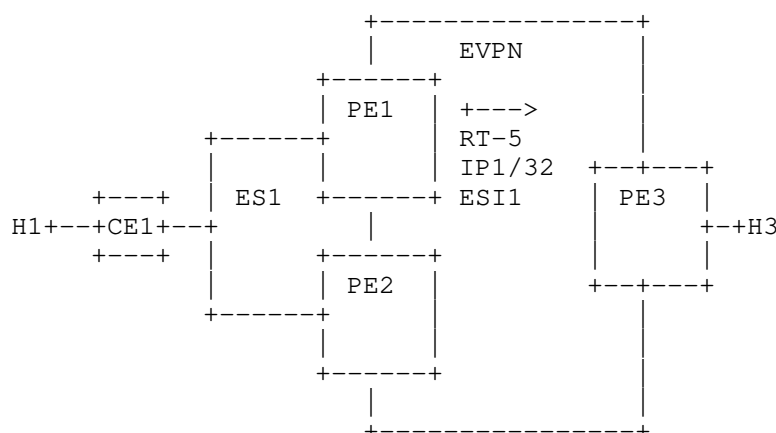


Figure 2: Inter-subnet example with IP Prefix routes

Consider PE1 and PE2 are multi-homed to CE1 (in an All-Active Ethernet Segment ES1), and PE1, PE2 and PE3 are attached to an IP-VRF of the same tenant. Suppose H1's host route is learned (via ARP or ND snooping) on PE1 only, and PE1 advertises an EVPN IP Prefix route for H1's host route. If H3 sends inter-subnet traffic to H1, a routing lookup on PE3 would normally yield a single next-hop, i.e., PE1.

This document proposes the use of the ESI in the IP Prefix route and the recursive resolution to A-D per ES/EVI routes advertised from PE1 and PE2, so that H1's host route in PE3 can be associated to an IP ECMP list (to PE1/PE2) for aliasing purposes.

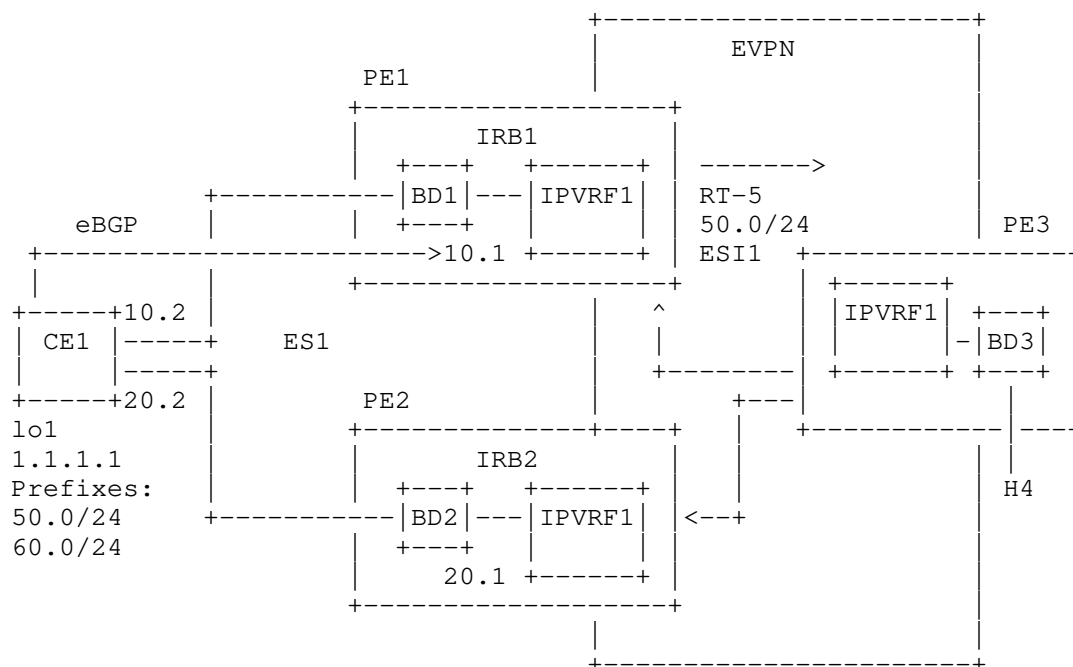
1.3. Ethernet Segments for Prefix routes in IP-VRF-to-IP-VRF use-cases

This document also enables fast convergence and aliasing/backup path to be used even when the ESI is used exclusively as an L3 construct, in an Interface-less IP-VRF-to-IP-VRF scenario [RFC9136]. There are two use cases analyzed and supported by this document:

- * IP Aliasing for EVPN IP Prefix routes
- * Centralized Routing Model

1.3.1. IP Aliasing for EVPN IP Prefix routes

As an example, consider the scenario in Figure 3 in which PE1 and PE2 are multi-homed to CE1. However, and contrary to CE1 in Figure 2, in this case the links between CE1 and PE1/PE2 are used exclusively for L3 protocols and L3 forwarding in different BDs, and a BGP session established between CE1's loopback address and PE1's IRB address.



Note:

IP addresses expanded by adding 0s

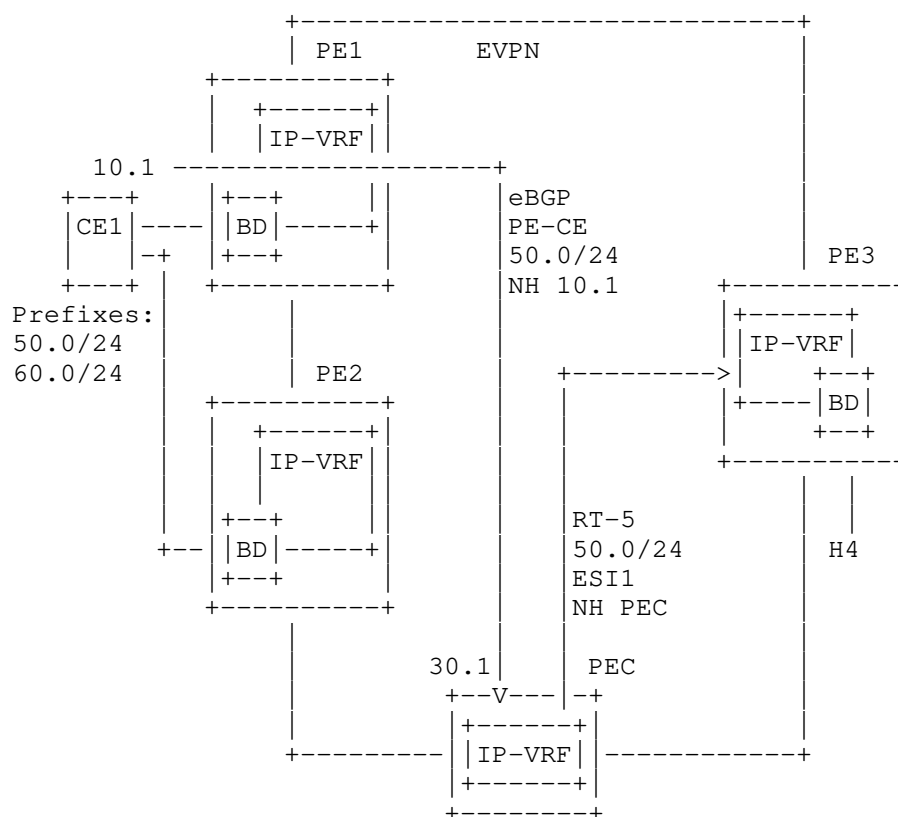
E.g., 50.0 expands to 50.0.0.0

Figure 3: Layer-3 Multihoming PEs

In these use-cases, sometimes the CE supports a single BGP session to one of the PEs (through which it advertises a number of IP Prefixes seating behind itself) and yet, it is desired that remote PEs can build an IP ECMP list or backup IP list including all the PEs multi-homed to the same CE. For example, in Figure 3, CE1 has a single eBGP neighbor, i.e., PE1. Load-balancing for traffic from CE1 to H4 can be accomplished by a default route with next-hops PE1 and PE2, however, load-balancing from H4 to any of the prefixes attached to CE1 would not be possible since only PE1 would advertise EVPN IP Prefix routes for CE1's prefixes. This document provides a solution so that PE3 considers PE2 as a next-hop in the IP ECMP list for CE1's prefixes, even if PE2 did not advertise the IP Prefix routes for those prefixes in the first place.

1.3.2. Centralized Routing Model

Figure 4 illustrates a model in which multiple CEs establish an eBGP PE-CE session with a Centralized PE.



Note:

IP addresses expanded by adding 0s

E.g., 50.0 expands to 50.0.0.0

Figure 4: Centralized Routing Model

The CEs in this case are usually VNFs (Virtual Network Function entities) or CNFs (Containerized Network Function entities) and by provisioning the same network parameters on all of them, the operation gets significantly simplified. The configuration on the PEs also gets simplified, since the PE-CE eBGP sessions to the CEs are only configured on a centralized PE. In the diagram, CE1 is one of these VNF/CNFs that sets up a multi-hop eBGP session to the centralized PEC. As an example, CE1 advertises prefix 50.0.0.0/24 with Next Hop 10.0.0.1 (to PEC) via the multi-hop eBGP session. PEC then exports the prefix into a RT-5 route, following the Interface-less IP-VRF-to-IP-VRF model [RFC9136], with Next Hop PEC. When H4 sends traffic to an IP address of the subnet 50.0.0.0/24, the traffic will be forwarded to PEC first, and PEC will then forward to PE1 (or PE2). In other words, this model simplifies the configuration and

operation of the CEs, however, it introduces an inefficiency since traffic needs to go through the Centralized PE (PEC) instead of going directly to the PE(s) attached to the destination CE. The IP Aliasing solution specified in this document overcomes this inefficiency and allows traffic from PE3 to be forwarded directly to PE1 or PE2, without going through PEC.

1.4. Terminology and Conventions

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.

- * IRB: Integrated Routing and Bridging
- * IRB Interface: Integrated Bridging and Routing Interface. A virtual interface that connects the Bridge Table and the IP-VRF on an NVE.
- * BD: Broadcast Domain. An EVI may be comprised of one BD (VLAN-based or VLAN Bundle services) or multiple BDs (VLAN-aware Bundle services).
- * Bridge Table: An instantiation of a broadcast domain on a MAC-VRF.
- * CE: Customer Edge device, e.g., a host, router, or switch.s
- * 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.
- * 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'.
- * IP-VRF: A VPN Routing and Forwarding table for IP routes on an NVE/PE. The IP routes could be populated by any routing protocol, E.g., EVPN, IP-VPN and BGP PE-CE IP address families. An IP-VRF is also an instantiation of a layer 3 VPN in an NVE/PE.

- * EVPN IP route: An EVPN IP Prefix route or an EVPN MAC/IP Advertisement route.
- * LACP: Link Aggregation Control Protocol.
- * 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.
- * RT-2: EVPN MAC/IP Advertisement route, as specified in [RFC7432].
- * RT-4: EVPN Ethernet Segment route, as specified in [RFC7432].
- * RT-5: EVPN IP Prefix route, as specified in [RFC9136].

2. Ethernet Segments for L3 Aliasing/Backup Path and Fast Convergence

The first two use cases described in Section 1 do not require any extensions to the Ethernet Segment definition and both cases support Ethernet Segments as a set of Ethernet links and specified in [RFC7432], or virtual Ethernet Segments as a set of logical links specified in [I-D.ietf-bess-evpn-virtual-eth-segment].

The third use case in Section 1 requires an extension to the way Ethernet Segments are defined and associated. In this case, the Ethernet Segment is a Layer-3 construct characterized as follows:

1. The ES is defined as a set of Layer-3 links to the multi-homed CE and its state MUST be linked to the layer-3 reachability from each multi-homed PE to the CE's loopback address via a non-EVPN route in the PE's IP-VRF.
2. The ESI SHOULD be of type 4 [RFC7432] and set to the router ID of the multi-homed CE.
3. All-active or single-active multi-homing redundancy modes are supported, however, the redundancy mode only affects the procedures in Section 3.

4. PEs attached to the same Layer-3 ES discover each other through the exchange of RT-4 routes (Ethernet Segment routes). DF Election procedures [RFC8584] MAY be used for single-active multi-homing mode.
5. The routes advertised from the multi-homed CE's and installed in the PE's IP-VRF table with the CE's loopback as the next-hop MUST be re-advertised by the PE in EVPN IP Prefix routes with the ESI of the CE. The rest of the EVPN IP Prefix routes fields are set as per the Interface-less model in [RFC9136]. Note that the BGP PE-CE routes advertised by the multi-homed CE are installed in the IP-VRF normally irrespective of the Next Hop being resolved to an EVPN or a non-EVPN route, and they are exported as a RT-5 with the ESI.

In the example depicted in Figure 3, ES1 is defined as the set of layer-3 links that connects PE1 and PE2 to CE1. Its ESI, e.g., ESI-1, is derived as a type 4 ESI using the CE's router ID. ES-1 will be operationally up in the PE as long as CE1's loopback route is installed in the PE's IP-VRF and learned via any routing protocol except for an EVPN route. E.g., an active static route to 1.1.1.1 via next-hop 10.0.0.2 would make the ES operationally up in PE1, and the eBGP routes received from CE1 with next-hop 1.1.1.1 will be re-advertised as RT-5 routes with ESI-1.

In the example illustrated in Figure 4, ES1 is a set of layer-3 links connecting PE1, PE2 and PEC to CE1. ESI-1 is derived as a type 4 ESI using the CE's router ID, as in the previous example. CE1's loopback route (which is associated to ES1) is installed in PE1 and PE2 via non-EVPN route, hence ES1 is operationally up in PE1 and PE2. On PE-C though, CE1's loopback is installed via EVPN IP Prefix route, therefore, as per point 1 in the current section, ES1 is operationally down in PEC. As per point 5, this does not prevent PEC from exporting CE1's prefixes into RT-5 routes with ESI-1. However, since ES-1 is operationally down in PEC, no IP A-D per EVI routes (Section 3) and no IP A-D per ES routes Section 4 for ESI-1 will be advertised from PEC, preventing PEC from attracting traffic destined to CE1.

3. IP Aliasing and Backup Path

In order to address the use-cases described in Section 1, above, this document proposes that:

1. A PE that is attached to a given ES will advertise a set of one or more Ethernet A-D per ES routes for that ES. Each is termed an 'IP A-D per ES' route and is tagged with the route targets (RTs) for one or more of the IP-VRFs defined on it for that ES; the complete set of IP A-D per ES routes contains the RTs for all of the IP-VRFs defined on it for that ES.

A remote PE imports an IP A-D per ES route into the IP-VRFs corresponding to the RTs with which the route is tagged. When the complete set of IP A-D per ES routes has been processed, a remote PE will have imported an IP A-D per ES route into each of the IP-VRFs defined on it for that ES; this enables fast convergence for each of these IP-VRFs.

2. A PE advertises for this ES, an Ethernet A-D Per EVI route for each of the IP-VRFs defined on it. Each is termed an 'IP A-D per EVI' route and is tagged with the RT for a given IP-VRF, and conveys a label that identifies that IP-VRF.

A remote PE imports an IP A-D per EVI route into the IP-VRF corresponding to the RT with which the route is tagged. The label contained in the route enables aliasing/backup path for the routes in that IP-VRF.

To address the third use-case described in Section 1, where the links between a CE and its multihomed PEs are used exclusively for L3 protocols and L3 forwarding, a PE uses the procedures described in 1) and 2), above.

The processing of the IP A-D per ES and the IP A-D per EVI routes is as defined in [RFC7432] and [RFC8365] except that the fast convergence and aliasing/backup path functions apply to the routes contained in an IP-VRF. In particular, a remote PE that receives an EVPN MAC/IP Advertisement route or an IP Prefix route with a non-reserved ESI and the RT of a particular IP-VRF SHOULD consider it reachable by every PE that has advertised an IP A-D per ES and IP A-D per EVI route for that ESI and IP-VRF.

3.1. Constructing the IP A-D per EVI Route

The construction of the IP A-D per EVI route is the same as that of the Ethernet A-D per EVI route, as described in [RFC7432], with the following exceptions:

- * The Route-Distinguisher is for the corresponding IP-VRF.
- * The Ethernet Tag should be set to 0.

- * The route SHOULD carry the Route Target of the corresponding IP-VRF.
- * The route MUST carry the MPLS label, VNI (VXLAN Network Identifier [RFC8365]) or Segment Routing IPv6 SID (Segment Identifier [I-D.ietf-bess-srv6-services]) that identifies the corresponding IP-VRF.
- * The route MUST carry the PE's MAC Extended Community if the encapsulation used between the PEs for inter-subnet forwarding is an Ethernet NVO tunnel [RFC9136].
- * The route SHOULD carry the EVPN Layer 2 Extended Community [I-D.ietf-bess-rfc7432bis]. For all-active multihoming, all PEs attached to the specified ES will advertise P=1. For backup path, the Primary PE will advertise P=1 and the Backup PE will advertise P=0, B=1.
 - The Primary PE SHOULD be a PE with a routing adjacency to the attached CE.
 - The Primary PE MAY be determined by policy or MAY be elected by a DF Election as in [RFC8584] as described in Section 2.

4. Fast Convergence for Routed Traffic

Host or Prefix reachability is learned via the BGP-EVPN control plane over the MPLS/NVO network. EVPN IP routes for a given ES are advertised by one or more of the PEs attached to that ES. When one of these PEs fails, a remote PE needs to quickly invalidate the EVPN IP routes received from it.

To accomplish this, EVPN defined the fast convergence function specified in [RFC7432]. This document extends fast convergence to inter-subnet forwarding by having each PE advertise a set of one or more IP A-D per ES routes for each locally attached Ethernet segment (refer to Section 4.1 below for details on how these routes are constructed). A PE may need to advertise more than one IP A-D per ES route for a given ES because the ES may be in a multiplicity of IP-VRFs and the Route Targets for all of these IP-VRFs may not fit into a single route. Advertising a set of IP A-D per ES routes for the ES allows each route to contain a subset of the complete set of Route Targets. Each IP A-D per ES route is differentiated from the other routes in the set by a different Route Distinguisher (RD).

Upon failure in connectivity to the attached ES, the PE withdraws the corresponding set of IP A-D per ES routes. This triggers all PEs that receive the withdrawal to update their next-hop adjacencies for

all IP addresses associated with the Ethernet Segment in question, across IP-VRFs. If no other PE has advertised an IP A-D per ES route for the same Ethernet Segment, then the PE that received the withdrawal simply invalidates the IP entries for that segment. Otherwise, the PE updates its next-hop adjacencies accordingly.

These routes should be processed with higher priority than EVPN IP route withdrawals upon failure. Similar priority processing is needed even on the intermediate Route Reflectors.

4.1. Constructing IP A-D per Ethernet Segment Route

This section describes the procedures used to construct the IP A-D per ES route, which is used for fast convergence (as discussed in Section 4). The usage/construction of this route remains similar to that described in section 8.2.1. of [RFC7432] with a few notable exceptions as explained in following sections.

4.1.1. IP A-D per ES Route Targets

Each IP A-D per ES route MUST carry one or more Route Targets. The set of IP A-D per ES routes MUST carry the entire set of IP-VRF Route Targets for all the IP-VRFs defined on that ES.

4.2. Avoiding convergence issues by synchronizing IP prefixes

Consider a pair of multi-homing PEs, PE1 and PE2. Let there be a host H1 attached to them. Consider PE3 and a host H3 attached to it.

If the host H1 is learned on both the PEs, the ECMP path list is formed on PE3 pointing to (PE1/PE2). Traffic from H3 to H1 is not impacted even if one of the PEs fails as the path list gets corrected upon receiving the withdrawal of the fast convergence route(s) (IP A-D per ES routes).

In a case where H1 is locally learned only on PE1 due to LAG hashing or a single routing protocol adjacency to PE1, at PE3, H1 has ECMP path list (PE1/PE2) using Aliasing as described in this document. Traffic from H3 can reach H1 via either PE1 or PE2.

PE2 should install local forwarding state for EVPN IP routes advertised by other PEs attached to the same ES (i.e., PE1) but not advertise them as local routes. When the traffic from H3 reaches PE2, PE2 will be able forward the traffic to H1 without any convergence delay (caused by triggering ARP/ND to H1 or to the next-hop to reach H1). The synchronization of the EVPN IP routes across all PEs of the same Ethernet Segment is important to solve convergence issues.

4.3. Handling Silent Host MAC/IP route for IP Aliasing

Consider the example of Figure 1 for IP aliasing. If PE1 fails, PE3 will receive the withdrawal of the fast convergence route(s) and update the ECMP list for H1 to be just PE2. When the EVPN IP route for H1 is also withdrawn, neither PE2 nor PE3 will have a route to H1, and traffic from H3 to H1 is blackholed until PE2 learns H1 and advertises an EVPN IP route for it.

This blackholing can be much worse if the H1 behaves like a silent host. IP address of H1 will not be re-learned on PE2 till H1 ARP/ND messages or some traffic triggers ARP/ND for H1.

PE2 can detect the failure of PE1's reachability in different ways:

- a. When PE1 fails, the next hop tracking to PE1 in the underlay routing protocols can help detect the failure.
- b. Upon the failure of its link to CE1, PE1 will withdraw its IP A-D route(s) and PE2 can use this as a trigger to detect failure.

Thus to avoid blackholing, when PE2 detects loss of reachability to PE1, it should trigger ARP/ND requests for all remote IP prefixes received from PE1 across all affected IP-VRFs. This will force host H1 to reply to the solicited ARP/ND messages from PE2 and refresh both MAC and IP for the corresponding host in its tables.

Even in core failure scenario on PE1, PE1 must withdraw all its local layer-2 connectivity, as Layer-2 traffic should not be received by PE1. So when ARP/ND is triggered from PE2 the replies from host H1 can only be received by PE2. Thus H1 will be learned as local route and also advertised from PE2.

It is recommended to have a staggered or delayed deletion of the EVPN IP routes from PE1, so that ARP/ND refresh can happen on PE2 before the deletion.

4.4. MAC Aging

In the same example as in Section 4.3, PE1 would do ARP/ND refresh for H1 before it ages out. During this process, H1 can age out genuinely or due to the ARP/ND reply landing on PE2. PE1 must withdraw the local entry from BGP when H1 entry ages out. PE1 deletes the entry from the local forwarding only when there are no remote synced entries.

5. Determining Reachability to Unicast IP Addresses

5.1. Local Learning

The procedures for local learning do not change from [RFC7432] or [RFC9136].

5.2. Remote Learning

The procedures for remote learning do not change from [RFC7432] or [RFC9136].

5.3. Constructing the EVPN IP Routes

The procedures for constructing MAC/IP Address or IP Prefix Advertisements do not change from [RFC7432] or [RFC9136].

5.3.1. Route Resolution

If the ESI field is set to reserved values of 0 or MAX-ESI, the EVPN IP route resolution MUST be based on the EVPN IP route alone.

If the ESI field is set to a non-reserved ESI, the EVPN IP route resolution MUST happen only when both the EVPN IP route and the associated set of IP A-D per ES routes have been received. To illustrate this with an example, consider a pair of multi-homed PEs, PE1 and PE2, connected to an all-active Ethernet Segment. A given host with IP address H1 is learned by PE1 but not by PE2. When the EVPN IP route from PE1 and a set of IP A-D per ES and IP A-D per EVI routes from PE1 and PE2 are received, then (1) PE3 can forward traffic destined to H1 to both PE1 and PE2.

If after (1) PE1 withdraws the IP A-D per ES route, then PE3 will forward the traffic to PE2 only.

If after (1) PE2 withdraws the IP A-D per ES route, then PE3 will forward the traffic to PE1 only.

If after (1) PE1 withdraws the EVPN IP route, then PE3 will do delayed deletion of H1, as described in Section 4.3.

If after (1) PE2 advertised the EVPN IP route, but PE1 withdraws it, PE3 will continue forwarding to both PE1 and PE2 as long as it has the IP A-D per ES and the IP A-D per EVI route from both.

6. Forwarding Unicast Packets

Refer to Section 5 in [RFC9135] and [RFC9136].

7. Load Balancing of Unicast Packets

The procedures for load balancing of Unicast Packets do not change from [RFC7432]

8. IP Aliasing and Unequal ECMP for IP Prefix Routes

[I-D.ietf-bess-evpn-unequal-lb] specifies the use of the EVPN Link bandwidth extended community to achieve weighted load balancing to an ES or Virtual ES for unicast traffic. The procedures in [I-D.ietf-bess-evpn-unequal-lb] MAY be used along with the procedures described in this document for any of the three cases described in Section 1, with the following considerations:

- * The ES weight is signaled by the multi-homed PEs in the IP A-D per ES routes.
- * The remote ingress PE learning an EVPN IP Route to prefix/host P that is associated to a weighted load balancing ES, will follow the procedures in [I-D.ietf-bess-evpn-unequal-lb] to influence the load balancing for traffic to P.
- * [I-D.ietf-bess-evpn-unequal-lb] also allows the use of the EVPN Link Bandwidth Extended Community along with RT-5s. If the ingress PE learns a prefix P via a non-reserved ESI RT-5 route with a weight (for which IP A-D per ES routes also signal a weight) and a zero ESI RT-5 that includes a weight, the ingress PE will consider all the PEs attached to the ES as a single PE when normalizing weights.

As an example, consider PE1 and PE2 are attached to ES-1 and PE1 advertises an RT-5 for prefix P with ESI-1 (and EVPN Link Bandwidth of 1). Consider PE3 advertises an RT-5 for P with ESI=0 and EVPN Link Bandwidth of 2. If PE1 and PE2 advertise an EVPN Link Bandwidth of 1 and 2, respectively, in the IP A-D per ES routes for ES-1, an ingress PE4 SHOULD assign a normalized weight of 1 to ES-1 and a normalized weight of 2 to PE3. When PE4 sprays the flows to P, it will send twice as many flows to PE3. For the flows sent to ES-1, the individual PE EVPN Link Bandwidths advertised in the IP A-D per ES routes will be considered.

9. 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.

10. IANA Considerations

No IANA considerations.

11. Contributors

12. Acknowledgments

13. References

13.1. 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>>.
- [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>>.
- [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>>.
- [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>>.

- [RFC9135] Sajassi, A., Salam, S., Thoria, S., Drake, J., and J. Rabadan, "Integrated Routing and Bridging in Ethernet VPN (EVPN)", RFC 9135, DOI 10.17487/RFC9135, October 2021, <<https://www.rfc-editor.org/info/rfc9135>>.
- [RFC9136] Rabadan, J., Ed., Henderickx, W., Drake, J., Lin, W., and A. Sajassi, "IP Prefix Advertisement in Ethernet VPN (EVPN)", RFC 9136, DOI 10.17487/RFC9136, October 2021, <<https://www.rfc-editor.org/info/rfc9136>>.
- [I-D.ietf-bess-rfc7432bis]
Sajassi, A., Burdet, L. A., Drake, J., and J. Rabadan, "BGP MPLS-Based Ethernet VPN", Work in Progress, Internet-Draft, draft-ietf-bess-rfc7432bis-03, 28 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-bess-rfc7432bis-03.txt>>.

13.2. Informative References

- [I-D.ietf-bess-evpn-virtual-eth-segment]
Sajassi, A., Brissette, P., Schell, R., Drake, J. E., and J. Rabadan, "EVPN Virtual Ethernet Segment", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-virtual-eth-segment-07, 6 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-bess-evpn-virtual-eth-segment-07.txt>>.
- [I-D.ietf-bess-evpn-unequal-lb]
Malhotra, N., Sajassi, A., Rabadan, J., Drake, J., Lingala, A., and S. Thoria, "Weighted Multi-Path Procedures for EVPN Multi-Homing", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-unequal-lb-15, 17 November 2021, <<https://www.ietf.org/archive/id/draft-ietf-bess-evpn-unequal-lb-15.txt>>.
- [I-D.ietf-bess-srv6-services]
Dawra, G., Filsfils, C., Talaulikar, K., Raszuk, R., Decraene, B., Zhuang, S., and J. Rabadan, "SRv6 BGP based Overlay Services", Work in Progress, Internet-Draft, draft-ietf-bess-srv6-services-12, 5 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-bess-srv6-services-12.txt>>.

Authors' Addresses

A. Sajassi (editor)
Cisco Systems
Email: sajassi@cisco.com

G. Badoni
Cisco Systems
Email: gbadoni@cisco.com

P. Warade
Cisco Systems
Email: pwarade@cisco.com

S. Pasupula
Cisco Systems
Email: surpasup@cisco.com

L. Krattiger
Cisco Systems
Email: lkrattig@cisco.com

J. Drake (editor)
Juniper
Email: jdrake@juniper.net

J. Rabadan (editor)
Nokia
520 Almanor Avenue
Sunnyvale, CA 94085
United States of America
Email: jorge.rabadan@nokia.com

BESS Working Group
Internet Draft
Category: Standard Track

A. Sajassi
S. Thoria
N. Fazlollahi
Cisco
A. Gupta
Avi Networks

Expires: January 2, 2017

July 2, 2017

Seamless Multicast Interoperability between EVPN and MVPN PEs
draft-sajassi-bess-evpn-mvpn-seamless-interop-00.txt

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 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 RFC 6513 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 EVPN-IRB PEs per [EVPN-IRB].

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	5
4.1. Optimum Forwarding	6
4.2. Optimum Replication	6
4.3. All-Active and Single-Active Multi-Homing	6
4.4. Inter-AS Tree Stitching	6
4.5. EVPN Service Interfaces	7
4.6. Distributed Anycast Gateway	7
4.7. Selective & Aggregate Selective Tunnels	7
4.8. Tenants' (S,G) or (*,G) states	7
5. Solution	7
5.1. Operational Model for Homogenous EVPN IRB NVEs	8
5.1.1 Control Plane Operation	10
5.1.2 Data Plane Operation	12
5.1.2.1 Sender and Receiver in same MAC-VRF	12
5.1.2.2 Sender and Receiver in different MAC-VRF	13
5.2. Operational Model for Heterogeneous EVPN IRB PEs	13
5.3. All-Active Multi-Homing	13
5.3.1. Source and receivers in same ES but on different subnets	14
5.3.2. Source and some receivers in same ES and on same	

subnet	14
5.4. Mobility for Tenant's sources and receivers	15
5.5. Single-Active Multi-Homing	15
6. DCs with only EVPN NVEs	15
6.1 Setup of overlay multicast delivery	16
6.3 Data plane considerations	17
7 Handling of different encapsulations	17
7.1 MPLS Encapsulation	18
7.2 VxLAN Encapsulation	18
7.3 Other Encapsulation	18
8. DCI with MPLS in WAN and VxLAN in DCs	18
8.1 Control plane inter-connect	18
8.2 Data plane inter-connect	20
8.3 Multi-homing among DCI gateways	20
9. Inter-AS Operation	20
10. Use Cases	20
10.1 DCs with only IGMP/MLD hosts w/o tenant router	20
10.2 DCs with mixed of IGMP/MLD hosts & multicast routers running PIM-SSM	21
10.3 DCs with mixed of IGMP/MLD hosts & multicast routers running PIM-ASM	21
10.4 DCs with mixed of IGMP/MLD hosts & multicast routers running PIM-Bidir	22
11. IANA Considerations	22
12. Security Considerations	22
13. Acknowledgements	22
14. References	22
14.1. Normative References	22
15.2. Informative References	23
15. Authors' Addresses	23

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 millions). For the same speed and feed, these high scale gateway boxes are relatively much more expensive than their TOR devices 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 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 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 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 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. 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 for EVPN-only applications in

data centers (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

ARP: Address Resolution Protocol
BEB: Backbone Edge Bridge
B-MAC: Backbone MAC Address
CE: Customer Edge
C-MAC: Customer/Client MAC Address
ES: Ethernet Segment
ESI: Ethernet Segment Identifier
IRB: Integrated Routing and Bridging
LSP: Label Switched Path
MP2MP: Multipoint to Multipoint
MP2P: Multipoint to Point
ND: Neighbor Discovery
NA: Neighbor Advertisement
P2MP: Point to Multipoint
P2P: Point to Point
PE: Provider Edge
EVPN: Ethernet VPN
EVI: EVPN Instance
RT: Route Target

Single-Active Redundancy Mode: When only a single PE, among a group of PEs attached to an Ethernet segment, is allowed to forward traffic to/from that Ethernet Segment, 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 traffic to/from that Ethernet Segment, then the Ethernet segment is defined to be operating in All-Active redundancy mode.

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. 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 [OVERLAY]
- 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 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 [OVERLAY] 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.

5. Solution

[EVPN-IRB] describes the operation for EVPN PEs in IRB mode for unicast traffic. The same EVPN PE model, where an IP-VRF is attached to one or more MAC-VRF via virtual IRB interfaces, is also applicable here. However, there are some noticeable differences between the IRB mode operation for unicast traffic described in [EVPN-IRB] versus for multicast traffic described here. 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. For multicast traffic, the intra-subnet traffic is bridged for non-IP traffic and it is Layer-2

switched for IP traffic. The differentiation between bridging and L2-switching for multicast traffic is that the former uses MAC-DA lookup for forwarding the traffic; whereas, the latter uses IP-DA lookup for forwarding the multicast traffic where the forwarding states are built using IGMP/MLD snooping. The inter-subnet multicast traffic is always routed in the corresponding IP-VRF.

This section describes a multicast VPN solution based on [MVPN] for EVPN PEs operating in IRB mode that want to perform seamless interoperability with their counterparts MVPN PEs.

5.1. Operational Model for Homogenous EVPN IRB NVEs

In this section, we consider the scenario where 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). In this scenario, 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 and the multicast routing protocol on the PE-CE link is presumed to be PIM-SM with both the ASM and the SSM service models per [RFC6513]. Furthermore, the PE uses MVPN BGP protocol and procedures per [RFC6513] and [RFC6514]. With respect to tenant PIM protocol, PIM-SM with Any Source Multicast (ASM) mode, PIM-SM with Source Specific Multicast (SSM) mode, and PIM Bidirectional (BIDIR) mode are all supported per [RFC6513]. Support of PIM-DM (Dense Mode) is excluded in this document per [RFC6513].

The EVPN PEs use MVPN BGP routes [RFC 6514] 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. The leaves and roots of these overlay trees are composed of Provider Multicast Service Interface (PMSI) and it can be Inclusive-PMSI (I-PMSI) or Selective-PMSI (S-PMSI) per [RFC6513]. A given PMSI is associated with a single IP-VRF of an EVPN PE and/or a MVPN PE for that MVPN - e.g., a MVPN PMSI is never associated with a MAC-VRF of an EVPN PE. Overlay-trees are instantiated by underlay provider tunnels (P-tunnels) - e.g., P2MP, MP2MP, or unicast tunnels per [RFC 6513]. When there are many-to-one mapping of PMSIs to a P-tunnel (e.g. mapping many S-PMSIs or many I-PMSI to a single P-tunnel), the tunnel is referred to as 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 IRB capability. An EVPN PE (with IRB capability) can be modeled as a MVPN PE where the virtual IRB interface of an EVPN PE (virtual interface between MAC-

VRF and IP-VRF) can be considered as an attachment circuit (AC) for the MVPN PE. In other words, an EVPN PE can be modeled as a PE that consists of a MVPN PE whose ACs are replaced with IRB interfaces connecting each IP-VRF of the MVPN PE to a set of MAC-VRFs. 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].

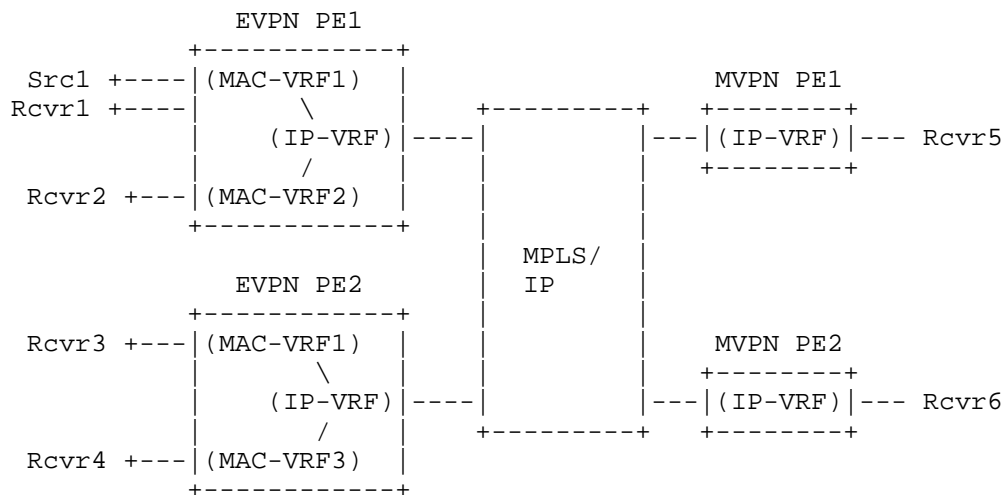


Figure-1: Homogenous EVPN NVEs

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 are impacted and needs to be factored in the solution:

- 1) All-Active multi-homing of IP multicast sources and receivers
- 2) Mobility for Tenant's sources and receivers
- 3) Unicast route advertisements for IP multicast source
- 4) non-IP multicast traffic handling

The first bullet, All-Active multi-homing of IP multicast source and receivers, is described in section 5.3. The second bullet is described in section 5.4. Third and fourth bullets are described next.

When an IP multicast source is attached to an EVPN PE, the unicast route for that IP multicast source needs to be advertised. This unicast route is advertised 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 MVPN PEs. The EVPN PE advertises this unicast route using EVPN route type 5 or IPVPN unicast route or both along with VRF Route Import extended community. When unicast routes are advertised by MVPN PEs, they are advertised using IPVPN unicast route along with VRF Route Import extended community per [RFC6514].

Link local multicast traffic (e.g. addressed to 224.0.0.x in case of IPv4) as well as IP protocols such as OSPF, and 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 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 5.1.1 for data plane operation).

5.1.1 Control Plane Operation

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, (C-*, C-G) or (C-S, C-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].

The following NLRIs from [RFC6514] SHOULD be used for forming Underlay/Core tunnels inside a data center.

Intra-AS I-PMSI A-D route is used to form default tunnel (also called inclusive tunnel) for a tenant VRF. The tunnel attributes are indicated using PMSI attribute with this route.

S-PMSI A-D route is used to form 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.

Source Active A-D route is used by source connected PE in order to announce active multicast source. This enables PEs having active receivers for the flow to join the tunnels and switch to Shortest Path tree.

Each EVPN PE supporting a specific MVPN 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 PE also sets up a multipoint-to-multipoint (MP2MP) tree per EVI using Inclusive Multicast Ethernet Tag route (route type 3) of EVPN NLRI per [RFC7432]. This MP2MP tree can be instantiated using unicast tunnels or P2MP tunnels. In [RFC7432], this tree is used for transmission of all BUM traffic including IP multicast traffic. However, for multicast traffic handling in EVPN-IRB PEs, this tree is used for all broadcast, unknown-unicast and non-IP multicast traffic - i.e., it is used for all BUM traffic except IP multicast user traffic. Therefore, an EVPN-IRB PE sends a customer IP multicast flow only on a single tunnel that is instantiated for MVPN I-PMSI or S-PMSI. In other words, IP multicast traffic sent over MPLS/IP network are not sent off of MAC-VRF but rather IP-VRF.

If a tenant host device 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). There is no need to use EVPN Selective Multicast Tag route (SMET route) because the IGMP messages are terminated by the EVPN-IRB PE and tenant (S,G) or (*,G) join messages are sent via MVPN Source/Shared Tree Join messages.

5.1.2 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 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 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 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 interfaces associated with that tenant (S,G) or (*,G) are removed.

When an EVPN-IRB PE receives IP multicast traffic, if it has any attached receivers for that subnet, it does L2 switching for such intra-subnet traffic. It then sends the multicast traffic over the corresponding IRB interface. The multicast traffic then gets routed over IRB interfaces that are included in the OIF list for that multicast traffic (and TTL gets decremented). When the multicast traffic is received on an IRB interface by the MAC-VRF corresponding to that interface, it gets L2 switched and sent over ACs that belong to the L2 OIF list. Furthermore, the multicast traffic gets sent over I-PMSI or S-PMSI associated with that multicast flow to other PE devices that are participating in that MVPN.

5.1.2.1 Sender and Receiver in same MAC-VRF

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. On PE2, since C-S falls in the MAC-VRF1 subnet, MAC-VRF1 Outgoing interface is treated as Ingress MAC-VRF bridging. Hence no rewrite is performed on the received customer data packet while forwarding towards Rcvr3.

5.1.2.2 Sender and Receiver in different MAC-VRF

Rcvr2 in Figure 1 is connected to PE1 in MAC-VRF2 and hence PE2 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). Rcvr3 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

5.2. Operational Model for Heterogeneous EVPN IRB PEs

5.3. All-Active Multi-Homing

EVPN solution [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. In MVPN, there is no concept of ESI label and split-horizon filtering because there is no support for All-Active multi-homing; however, EVPN NVEs rely on this function to prevent loop for an access Ethernet Segment. Figure-2 depicts a source sitting behind an All-Active dual-homing Ethernet Segment. The following scenarios needs special considerations:

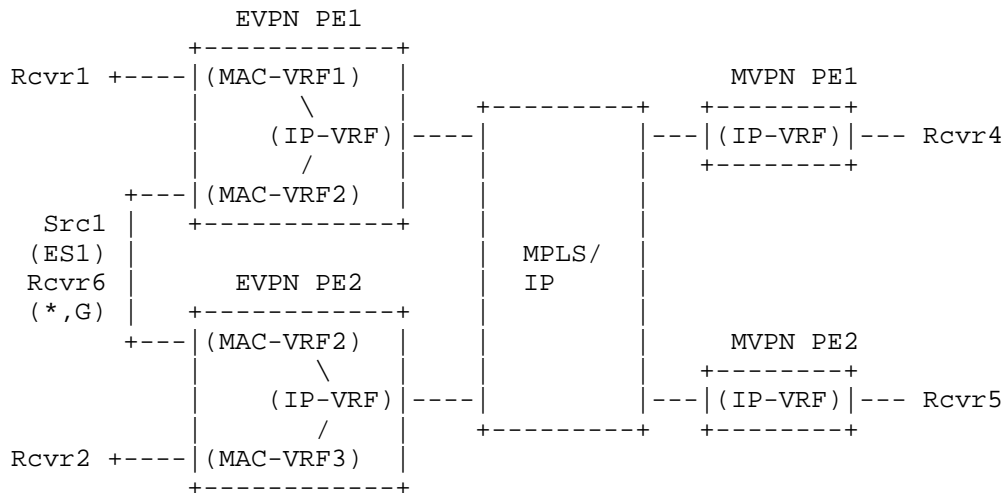


Figure-2: Multi-homing

5.3.1. Source and receivers in same ES but on different subnets

If the tenant multicast source sits on a different subnet than its receivers, then EVPN DF election procedure for multi-homing ES is sufficient and there will be no need to do any split-horizon filtering for that Ethernet Segment because with IGMP/MLD snooping enabled on VLANs for the multi-homing ES, only the VLANs for which IGMP/MLD join have been received are placed in OIF list for that (S,G) or (*,G) on that ES. Therefore, multicast traffic will not be loop backed on the source subnet (because there is no receiver on that subnet) and for other subnets that the multicast traffic is loop backed, the DF election ensures only a single copy of the multicast traffic is sent on that subnet.

5.3.2. Source and some receivers in same ES and on same subnet

If the tenant multicast source sits on the same subnet and the same ES as some of its receivers and those receivers have interest in (*,G), then Besides DF election mechanism, there needs to be split-horizon filtering to ensure that the multicast traffic originated from that <ES, EVI, BD> is not loop backed to itself. The existing split-horizon filtering as specified in [RFC7432] cannot be used because the received VPN label identifies the multicast IP-VRF and not MAC-VRF. Therefore, egress PE doesn't know for which EVI/BD it needs to perform split-horizon filtering and for which EVI/BDs

belonging to the the same ES, it needs not to perform split-horizon filtering. This issue is resolved by extending the local-bias solution per [OVERLAY] to MPLS tunnels. There are two cases to consider here: a) Ingress-replication tunnels used for the multicast traffic and b) P2MP tunnels used for the multicast traffic.

If ingress-replication tunnels are used, then each PE in the multi-homing group instead of advertising an ESI label, it advertises to each PE in the multi-homing group a downstream assigned label identifying that PE, so that when it receives a packet with this label, it know who the originating PE is. Once the egress PE can identify the originating PE for a packet, then it can execute local-bias procedure per [OVERLAY] for each of its EVI/BDs corresponding to that IP-VRF.

If P2MP tunnels are used (e.g., mLDP, RSVP-TE, or BIER), the tunnel label identifies the tunnel and thus the originating PE. Since the originating PE can be identified, the local-bias procedure per [OVERLAY] is applied to prevent multicast data to be sent on the Ethernet Segments in common with the originating PE. The difference between the local-bias procedure in here versus the one described in [OVERLAY] is that the multicast traffic in [OVERLAY] is only intended for one subnet (and thus one BD) whereas the multicast traffic in Figure-2 can span across multiple subnets (and thus multiple BDs). Therefore, local-bias procedure in [OVERLAY] is expanded to perform local bias across all the BDs of that tenant. In other words, the same local-bias procedure is applied to all BDs of that tenant in both the originating EVPN NVE as well as all other EVPN NVEs that share the Ethernet Segment with the originating EVPN NVE.

5.4. Mobility for Tenant's sources and receivers

5.5. Single-Active Multi-Homing

6. DCs with only EVPN NVEs

As mentioned earlier, the proposed solution can be used as a routed multicast solution for EVPN-only applications in data centers (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-IRB PE. In other words, the IP multicast traffic that needs to be forwarded from one PE to another is always routed (L3 forwarded) 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. Based on past experiences with MVPN over last dozen years for supported IP multicast applications, layer-3 forwarding of intra-subnet multicast traffic should be fine. However, if there are applications that require intra-subnet multicast traffic to be L2 forwarded (e.g., without decrementing TTL value), then [EVPN-IRB-MCAST] proposes a solution to accommodate such applications.

6.1 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 IP Prefix route (route type 5) of EVPN NLRI per [RFC7432].

The VRF Route Import (VRI) extended community that is carried with the IP-VPN routes in [RFC6514] MUST be carried via the EVPN unicast routes instead. 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).
- 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.

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 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 core VRF 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 join or a local 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 the active multicast source to all PEs in the Overlay so they can potentially switch from RP-Shared-Tree to Shortest-Path-Tree. The above procedure is optional per [RFC6514], and user SHALL enable an auto-discovery mode where the temporary RP-Shared-Tree is not involved. In this mode, the source PE MUST advertise the MCAST-VPN Source Active A-D route (type 5) as soon as it detects data traffic from the local tenant multicast source. Hence the PEs at different sites of the same MVPN will directly join the Shortest-Path-Tree once they receive the MCAST-VPN Source Active A-D route.

6.3 Data plane considerations

Data-center fabrics are implemented using variety of core technologies but predominant ones are IP/VXLAN Ingress Replication, IP/VXLAN PIM and MPLS LSM. IP and MPLS have been predominant choice for MVPN core as well hence all existing procedures for forming tunnels for these technologies are applicable in EVPN as well. Also as described in earlier section, each PE acts as PIM DR in its locally connected Bridge Domain, we MUST NOT forward post-routed traffic out of IRB interfaces towards the core.

7 Handling of different encapsulations

Just as in [RFC6514] the 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.

7.1 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.

7.2 VxLAN Encapsulation

In order to signal VXLAN, the corresponding BGP encapsulation extended community [TUNNEL-ENCAP] SHOULD be appended to the 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 [OVERLAY].

In this case, a gateway is needed for inter-operation between the EVPN-IRB 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.

7.3 Other Encapsulation

In order to signal a different tunneling encapsulation such as NVGRE, VXLAN-GPE or MPLSoGRE the corresponding BGP encapsulation extended community [TUNNEL-ENCAP] SHOULD be appended to the 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.

8. DCI with MPLS in WAN and VxLAN in DCs

This section describes the inter-operation between MVPN MPLS WAN with MVPN-EVPN in a data-center which runs on VxLAN. 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. Some aspects of the multi-homing between VxLAN DC networks and MPLS WAN is in common with [INTERCON-EVPN]. Herein, only the differences are described.

8.1 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.

Other point to notice is that if there are multiple gateways in an ESI which peer with each other, each one will receive two sets of the local MCAST-VPN routes from the other gateway: 1) the WAN set 2) the DC set. Following the same procedure as in [INTERCON-EVPN], the WAN set SHALL be given a higher priority.

8.2 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 or bridge 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 hosts on each gateway flows to the other gateway with the preferred encapsulation (WAN encapsulation is preferred as described in previous section).

8.3 Multi-homing among DCI gateways

Just as in [INTERCON-EVPN] every set of multi-homed gateways between the WAN and a given DC are assigned a unique ESI.

9. Inter-AS Operation

10. Use Cases

10.1 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 since there are no other routers RP-PE would be the PE configured as RP 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 RFC6513. To achieve this, all PE's can 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. One 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.

10.2 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.

10.3 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.

10.4 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.

11. IANA Considerations

There is no additional IANA considerations for PBB-EVPN beyond what is already described in [RFC7432].

12. Security Considerations

All the security considerations in [RFC7432] apply directly to this document because this document leverages [RFC7432] control plane and their associated procedures.

13. Acknowledgements

The authors would like to thank Samir Thoria, Ashutosh Gupta, Niloofar Fazlollahi, and Aamod Vyavaharkar for their discussions and contributions.

14. References

14.1. Normative References

[RFC7024] Jeng, H., Uttaro, J., Jalil, L., Decraene, B., Rekhter, Y., and R. Aggarwal, "Virtual Hub-and-Spoke in BGP/MPLS VPNs", RFC 7024, October 2013.

- [RFC7432] A. Sajassi, et al., "BGP MPLS Based Ethernet VPN", RFC 7432, February 2015.

15.2. 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.
- [OVERLAY] A. Sajassi, et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-01, work in progress, February 2015.
- [RFC6514] R. Aggarwal, et al., "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC6514, February 2012.
- [RFC6513] E. Rosen, et al., "Multicast in MPLS/BGP IP VPNs", RFC6513, February 2012.
- [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.

15. Authors' Addresses

Ali Sajassi
Cisco
170 West Tasman Drive
San Jose, CA 95134, US
Email: sajassi@cisco.com

Samir Thoria
Cisco
170 West Tasman Drive
San Jose, CA 95134, US
Email: sthoria@cisco.com

Niloofar Fazlollahi
Cisco
170 West Tasman Drive
San Jose, CA 95134, US
Email: nifazlol@cisco.com

Ashutosh Gupta
Avi Networks
Email: ashutosh@avinetworks.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.

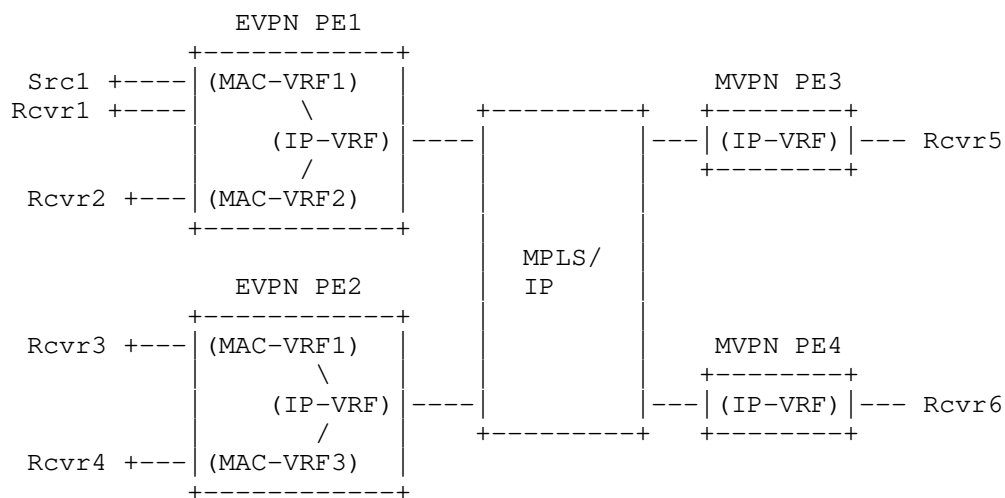


Figure-1: EVPN & MVPN PEs Seamless Interop

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.

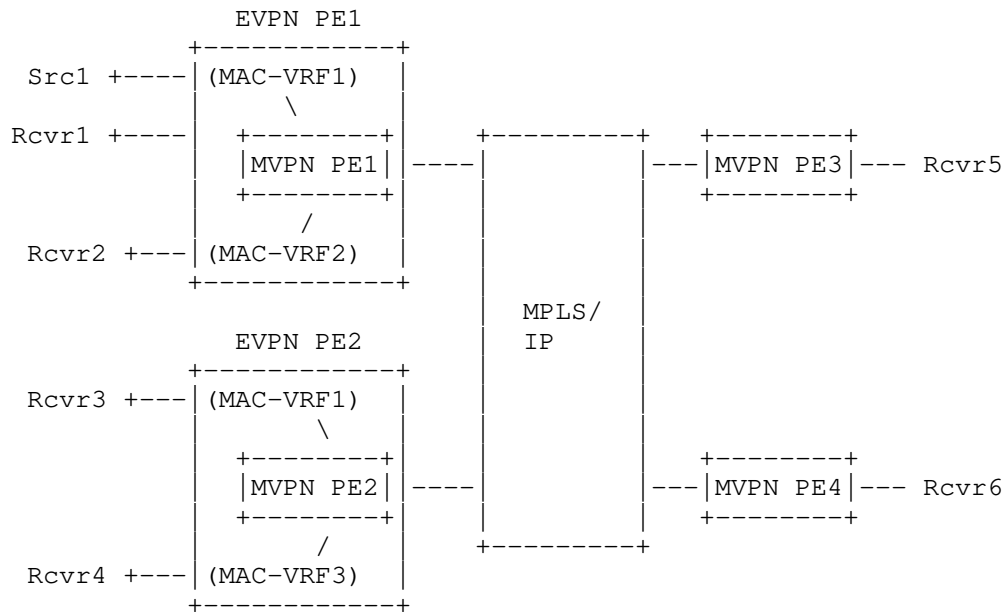


Figure-2: Modeling EVPN PEs as MVPN PEs

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.

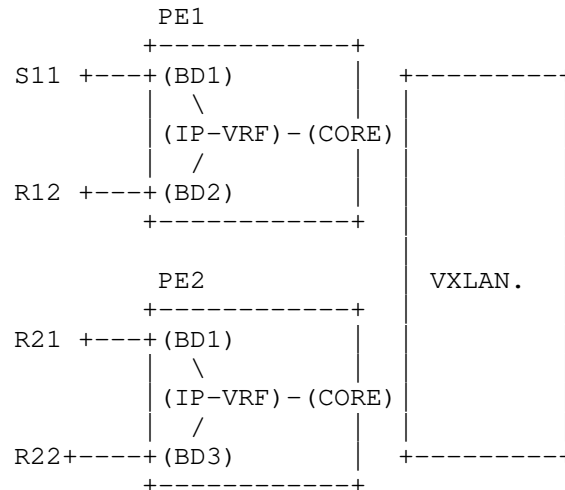


Figure 3 Intra-subnet bridging

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
CSCO
170 West Tasman Drive
San Jose, CA 95134, US
Email: sajassi@cisco.com

Kesavan Thiruvengatasamy
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 Working Group
Internet Draft
Category: Standard Track

A. Sajassi
S. Salam
P. Brissette
Cisco

L. Jalil
Verizon

Expires: January 2, 2018

July 2, 2017

(PBB-)EVPN Integration with (PBB-)VPLS in All-Active Mode
draft-sajassi-bess-evpn-vpls-all-active-00

Abstract

This draft discusses the backward compatibility of the (PBB-)EVPN solution with (PBB-)VPLS in all-active redundancy mode.

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	3
1.1	Terminology	3
2	Limitations	3
3	Solution for MAC Flip-Flopping	4
3.1	Load-Balancing	5
4	Changes on EVPN PEs	5
4.1	Control Plane Changes	5
4.2	Data Plane Changes	6
4.2.1	Known Unicast Traffic	6
4.2.2	BUM Traffic	6
5	Failure Handling	7
6	EVPN-VPWS termination onto multi-homing EVPN PEs	7
7	Security Considerations	8
8	IANA Considerations	8
9	References	8
9.1	Normative References	8
9.2	Informative References	8
	Authors' Addresses	8

1 Introduction

VPLS and PBB-VPLS are widely-deployed L2VPN technologies. Many SPs who are looking at adopting EVPN and PBB-EVPN want to preserve their investment in the (PBB-)VPLS networks. Hence, it is required to provide mechanisms by which (PBB-)EVPN technology can be introduced into existing L2VPN networks without requiring a fork-lift upgrade. [EVPN-VPLS] discusses mechanisms for the seamless integration of the two technologies in the same MPLS/IP network, however, operation is limited to single-active redundancy mode. In this document, we extend the solution to support all-active redundancy.

Section 2 provides the limitations in the current (PBB-)EVPN/(PBB-)VPLS interoperability solution. Section 3 discusses the solution for addressing those limitations. Section 4 describes the required control and data plane changes to support all-active redundancy. Section 5 covers the failure handling.

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].

2 Limitations

[EVPN-VPLS] defines mechanisms for (PBB-)EVPN seamless interoperability with (PBB-)VPLS. The solution defined in [EVPN-VPLS] suffers from a major limitation that hinders brown-field deployment of EVPN solution: It provides support for all-active redundancy only for VPN instances confined to (PBB-)EVPN PEs. For VPN instances that span both (PBB-)EVPN as well as (PBB-)VPLS PEs only single-active redundancy mode is supported. This eliminates one of the key value propositions of inserting EVPN solution in existing networks.

The reason why this capability is not currently supported is due to the issue of MAC address flip-flopping on the VPLS PEs. This is best explained with an example: Consider the example network of Figure 1 below. Assume that CE1 is connected over an all-active link aggregation group (LAG) to EVPN-capable PEs (PE2 and PE3). For traffic destined from CE1 to CE2, different flows from the same source MAC address MAC-A will be load-balanced over the LAG to PE2 and PE3. PE2 will forward the traffic over its own pseudowire (call it PW-Blue) to PE5, whereas PE3 will forward the traffic over its own pseudowire (call it PW-Red) to PE5. As such, VPLS PE (PE5) will learn the same MAC address (MAC-A) over both PW-Red and PW-Blue, depending on the load-balancing. This MAC flip-flopping will continue

indefinitely depending on traffic patterns.

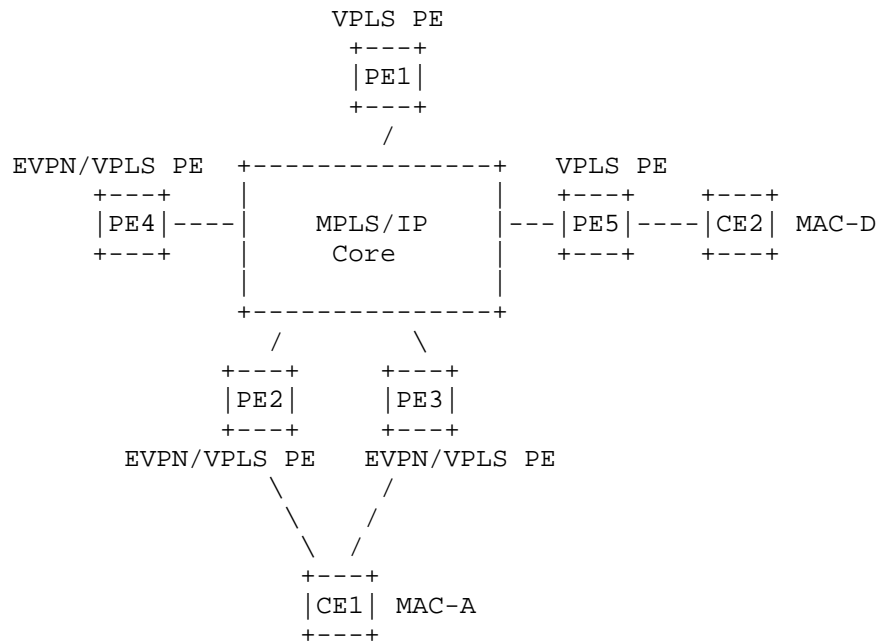


Figure 1: Seamless Integration of (PBB-)EVPN PEs & (PBB-)VPLS

The focus of this draft is on providing a solution that addresses the above limitation, thereby enabling the support of all-active redundancy in mixed (PBB-)EVPN/(PBB-)VPLS deployments.

3 Solution for MAC Flip-Flopping

In order to address the MAC flip-flopping problem on the VPLS PEs, these PEs must learn the traffic originating from a given source MAC address over the same pseudowire consistently, regardless of which remote EVPN-capable PE forwarded the traffic in a given multi-homed setup. To that end, every multi-homed EVPN-capable PE must maintain, in addition to its own pseudowires, a set of shadow or "alias" pseudowires for each of its peers in a given Redundancy Group (RG). For instance, in the example network of Figure 1, PE2 maintains its own pseudowire towards PE5 in addition to an "alias" pseudowire corresponding to the pseudowire between PE3 and PE5.

When traffic arrives from a multi-homed CE over a multi-chassis LAG, the EVPN-capable PE then examines whether or not it is the Designated Forwarder (DF) for the Ethernet Segment (ES) in question. In the case

where the PE is the DF for the ES, it would use its own pseudowire label to forward traffic towards a remote VPLS PE. However, in the case where the PE is not the DF for the ES, it would then use the "alias" pseudowire label associated with the DF PE in order to forward traffic towards the remote VPLS PE. To illustrate this using the example of Figure 1, consider that PE3 is the DF for the ES associated with CE1. Furthermore, assume that the pseudowire labels from PE2 and PE3 to PE5 are Label-Blue and Label-Red, respectively. When CE1 load-balances traffic destined to CE2 towards PE3, the latter will use its own pseudowire label (Label-Red) to forward traffic to PE5. Whereas, when CE1 forwards traffic destined to CE2 towards PE2, it will use the alias pseudowire label (Label-Red) instead of its own pseudowire label to forward traffic towards PE5. This is because PE2 is not the DF for the Ethernet Segment associated with CE1.

3.1 Load-Balancing

For traffic flowing from the EVPN-capable PEs towards the MPLS network, the load-balancing is on a per-flow granularity, regardless of whether the traffic is destined towards remote EVPN or VPLS PEs.

For traffic flowing from the VPLS PEs towards the EVPN-capable PEs, the load-balancing is on a per-VLAN per destination site granularity. That is, the traffic for a given VLAN in a destination site is sent to only one of the multi-homed EVPN-capable PEs. This is because all the EVPN-capable PEs in a given redundancy group will use the pseudowire label associated with the DF to forward traffic towards remote VPLS PEs (recall, also, that EVPN DF election is per VLAN per ES).

4 Changes on EVPN PEs

The changes to support the mechanisms of this draft are confined to the EVPN-capable PEs. In the following two sub-sections we cover both the control plane as well as data plane changes required.

4.1 Control Plane Changes

In order for the EVPN-capable PEs to maintain the alias pseudowires, it is required to synchronize the VPLS pseudowire labels among the PEs in the same Redundancy Group. For VPLS-BGP [RFC4761], this is straight-forward to achieve because the VE-IDs and label blocks associated with all PEs are advertised in BGP. Hence, a PE in an EVPN RG can easily extract the alias pseudowire labels associated with its peers in the same RG. For VPLS-LDP [RFC4762], protocol message extensions are required but are outside the scope of the current document.

Another control plane extension that is required is to synchronize the MAC addresses learnt over the active pseudowire at DF EVPN PEs to the non-DF EVPN PEs with alias pseudowire using BGP. This can be done using the existing EVPN MAC Advertisement route. The identity of the pseudowire over which the address was learnt is encoded in the ESI field. This can be done using a Type 4 ESI, where the Router ID holds the IP address of the remote pseudowire endpoint IP address (i.e. VPLS PE address) and the high-order 2 octets of the Local Discriminator encode the VE-ID of the remote pseudowire endpoint (i.e. EVPN-capable PE that is the DF).

4.2 Data Plane Changes

4.2.1 Known Unicast Traffic

After DF election is complete, the EVPN-capable PE programs its data plane based on the outcome of DF election as follows:

If known unicast traffic is received by the PE from an Ethernet Segment for which it is the DF, then it uses its own pseudowire label in the label stack when forwarding traffic to remote VPLS PEs.

If known unicast traffic is received by the PE from an Ethernet Segment for which is non-DF, then it uses the alias pseudowire label (associated with the DF) instead of its own pseudowire label in the label stack when forwarding traffic to remote VPLS PEs.

In other words, the EVPN-capable PE must use the DF/non-DF status of the incoming attachment circuit interface in order to choose the correct label stack for VPLS forwarding.

4.2.2 BUM Traffic

The EVPN-capable PEs must maintain two replication lists: one that uses their own pseudowires, and another that uses the alias pseudowires. When BUM traffic is received from the attachment circuit, the PE examines the DF status of the incoming interface to identify which of the two replication lists to use: If the PE is the DF, then it uses the replication list which encompasses its own pseudowires. Whereas, if the PE is non-DF, then it uses the replication list encompassing the alias pseudowires.

BUM traffic received over a VPLS pseudowire is handled as follows:

Broadcast and multicast traffic is identified as such by inspecting the destination MAC address, and is handled as usual per EVPN MPLS ingress flooding mechanisms. At egress to the attachment circuit, all broadcast and multicast VPLS traffic is subjected to DF filtering

procedures per existing EVPN procedures.

Unknown unicast traffic cannot be identified as such by the disposition PE on egress from the pseudowire, since nothing in the Ethernet frame or the MPLS label stack (unlike EVPN) distinguishes this traffic from known unicast. Furthermore, the disposition PE cannot rely on its own MAC forwarding table to infer whether the frame was flooded or not - i.e., an unknown MAC address on the imposition PE cannot be known to the disposition PE. Due to this, the egress (disposition) PE will treat unicast MAC addresses based on its own local forwarding state - i.e., if the MAC address is known locally, then it is treated as such and if the MAC address is unknown locally, then it is treated as BUM traffic and will apply DF filtering. This can lead to a side-effect for a very specific scenario where the MAC-DA is unknown at the ingress PE but it is known to the egress multi-homing PEs (i.e., there is no issue when MAC-DA is known at the ingress and unknown at the egress, or MAC-DA is unknown at both the ingress and egress PEs). In such a specific scenario, a multi-homed CE will experience duplicate packets for an interim period of time until the remote VPLS PE learns the MAC address from reverse traffic. The CE's application layer will handle the discard of transient duplicate frames. While it is acknowledged that this behavior deviates from classical Ethernet, which guarantees the absence of packet duplication, the side-effect occurs in very specific scenario and it is both short-lived and confined in scope to the PE/CE links. Hence, it is a reasonable trade-off to accept in favor of enabling all-active redundancy in the solution.

5 Failure Handling

Failure handling follows standard EVPN and VPLS procedures:

For link failure on DF EVPN-capable PE, the PE sends a mass withdraw indication using per ES Ethernet A-D route to other EVPN PEs, causing them to update their forwarding entries to point to only the non-DF PE. The DF PE also sends VPLS MAC address flush message to remote VPLS PEs, causing them to flush their entries. The non-DF EVPN PE takes over and assumes the DF role. It uses its own VPLS pseudowire labels for sending traffic towards the VPLS PEs.

For link failure on non-DF EVPN PE, the PE sends mass withdraw per ES Ethernet A-D route to other EVPN PEs, causing them to update their forwarding entries to point to only the DF PE. Nothing is done with respect to the VPLS PEs, as this failure is transparent to them.

6 EVPN-VPWS termination onto multi-homing EVPN PEs This section will be added in the future revision to describe how the MAC synchroniation

mechanism over PW described above can be used for this scenario.

7 Security Considerations

No new security considerations beyond those for VPLS and EVPN.

8 IANA Considerations

This document has no actions for IANA.

9 References

9.1 Normative References

- [KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4761] Kompella, K., Ed., and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.
- [RFC4762] Lasserre, M., Ed., and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, January 2007.
- [EVPN-VPLS] Sajassi, A., Salam, S., Del Regno, N., and Rabadan, J., "(PBB-)EVPN Seamless Integration with (PBB-)VPLS", draft-ietf-bess-evpn-vpls-seamless-integ-00, work in progress, February 2015, <<https://datatracker.ietf.org/doc/html/draft-sajassi-bess-evpn-vpls-seamless-integ>>.

9.2 Informative References

Authors' Addresses

Ali Sajassi
Cisco
170 West Tasman Drive

San Jose, CA 95134, US
Email: sajassi@cisco.com

Samer Salam
Cisco
595 Burrard Street, Suite 2123
Vancouver, BC V7X 1J1, Canada
Email: ssalam@cisco.com

Patrice Brissette
Cisco
Email: pbrisset@cisco.com

Luay Jalil
Cisco
Email: luay.jalil@verizon.com

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
J. Kotalwar
S. Sathappan
Nokia
Z. Zhang
Juniper
A. Sajassi
Cisco

Expires: January 4, 2018

July 3, 2017

PIM Proxy in EVPN Networks
draft-skr-bess-evpn-pim-proxy-00

Abstract

Ethernet Virtual Private Networks [RFC7432] are becoming prevalent in Data Centers, Data Center Interconnect (DCI) and Service Provider VPN applications. One of the goals that EVPN pursues is the reduction of flooding and the efficiency of CE-based control plane procedures in Broadcast Domains. Examples of this are [EVPN-PROXY-ARP-ND] for improving the efficiency of CE's ARP/ND protocols, and [EVPN-IGMP-MLD-PROXY] for IGMP/MLD protocols. This document complements the latter, describing the procedures required to minimize the flooding of PIM messages in EVPN Broadcast Domains, and optimize the IP Multicast delivery between PIM routers.

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 4, 2018.

Copyright 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	3
2. PIM Proxy Operation in EVPN Broadcast Domains	4
2.1. Multicast Router Discovery Procedures in EVPN	5
2.1.1. Discovering PIM Routers	5
2.1.2. Discovering IGMP Queriers	7
2.2. PIM Join/Prune Proxy Procedures	7
2.3. PIM Assert Optimization	10
2.3.1 Assert Optimization Procedures in Downstream PEs	11
2.3.2 Assert Optimization Procedures in Upstream PEs	12
2.4. EVPN Multi-Homing and State Synchronization	12
2.5. PIM Bootstrap and RP Discovery	13
2.6. PIM-DM (Dense Mode) Proxy Procedures	13
3. Interaction with IGMP-snooping and Sources	13
4. BGP Information Model	14
4.1 Multicast Router Discovery (MRD) Route	15
4.2 Selective Multicast Ethernet Tag Route for PIM Proxy	16
4.3 PIM RPT-Prune Route	18
4.4 IGMP/PIM Join Synch Route for PIM Proxy	19
4.5 IGMP/PIM RPT-Prune Synch Route for PIM Proxy	20
5. Conclusions	21
6. Conventions used in this document	21
7. Security Considerations	21
8. IANA Considerations	21
9. Terminology	22
10. References	22

10.1 Normative References	22
10.2 Informative References	23
11. Acknowledgments	23
12. Contributors	23
13. Authors' Addresses	23

1. Introduction

Ethernet Virtual Private Networks [RFC7432] are becoming prevalent in Data Centers, Data Center Interconnect (DCI) and Service Provider VPN applications. One of the goals that EVPN pursues is the reduction of flooding and the efficiency of CE-based control plane procedures in Broadcast Domains. Examples of this are [EVPN-PROXY-ARP-ND] for improving the efficiency of CE's ARP/ND protocols, and [EVPN-IGMP-MLD-PROXY] for IGMP/MLD protocols.

This document focuses on optimizing the behavior of PIM in EVPN Broadcast Domains and re-uses some procedures of [EVPN-IGMP-MLD-PROXY]. The reader is also advised to check out [VPLS-PIM-PROXY] to understand certain aspects of the procedures of PIM Join/Prune messages received on Attachment Circuits (ACs).

Section 2 describes the PIM Proxy procedures that the implementation should follow, including:

- o The use of EVPN to suppress the flooding of PIM Hello messages in shared Broadcast Domains. The benefit of this is twofold:
 - PIM Hello messages will be ONLY flooded to Attachment Circuits that are connected to PIM routers, as opposed to all the CEs and hosts in the Broadcast Domain.
 - Soft-state PIM Hello messages will be replaced by hard-state BGP messages that don't need to be refreshed periodically.
- o The use of EVPN to discover IGMP Queriers, while avoiding the flooding of IGMP Queries in the core.
- o The procedures to proxy PIM Join/Prune messages and replace them by hard-state EVPN routes that don't need to be refreshed periodically. By using BGP EVPN to propagate both, Hello and Join/Prune messages, we also avoid out-of-order delivery between both types of PIM messages.
- o This document also describes an EVPN based procedure so that the PIM routers connected to the shared Broadcast Domain don't need to run any PIM Assert procedure. PIM Assert procedures may be

expensive for PIM routers in terms of resource consumption. With this procedure, there is no PIM Assert needed on PIM routers.

- o The use of procedures similar to the ones defined in [EVPN-IGMP-MLD-PROXY] to synchronize multicast states among the PEs in the same Ethernet Segment.

Section 3 describes the interaction of PIM Proxy with IGMP Proxy PEs and Multicast Sources connected to the same EVPN Broadcast Domain.

Section 4 defines the BGP Information Model that this document requires to address the PIM Proxy procedures.

This document assumes the reader is familiar with PIM and IGMP protocols.

2. PIM Proxy Operation in EVPN Broadcast Domains

This section describes the operation of PIM Proxy in EVPN Broadcast Domains (BDs). Figure 1 depicts an EVPN Broadcast Domain defined in four PEs that are connected to PIM routers. This example will be used throughout this section and assumes both R4 and R5 are PIM Upstream Neighbors for PIM routers R1, R2 and R3 and multicast group G1. In this situation, the PIM multicast traffic flows from R4 or R5 to R1, R2 and R3. The PIM Join/Prune signaling will flow in the opposite direction. From a terminology perspective, we consider PE1 and PE2 as egress or downstream PEs, whereas PE3 and PE4 are ingress or upstream PEs.

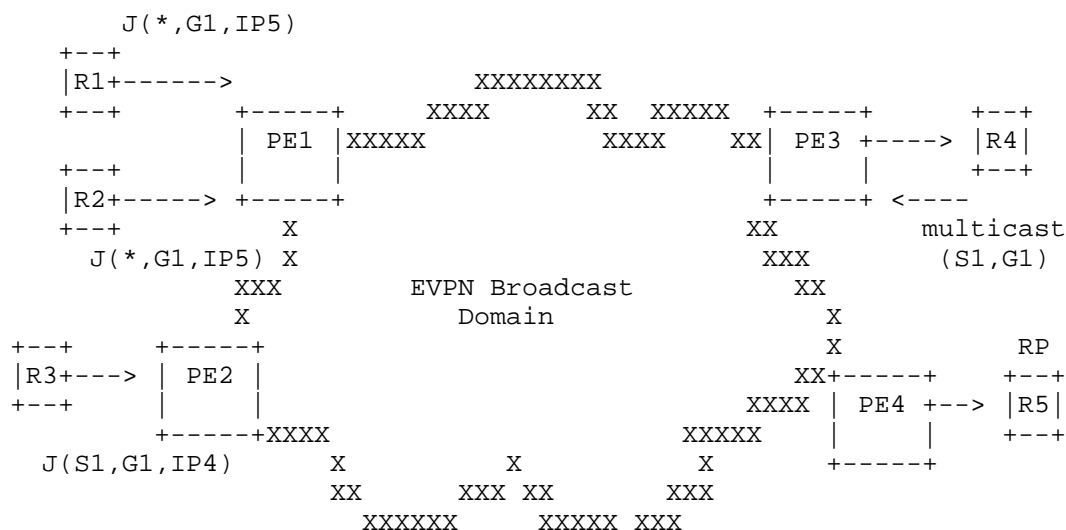


Figure 1 - PIM Routers connected by an EVPN Broadcast Domain

It is important to note that any Router's PIM message not explicitly specified in this document will be forwarded by the PEs normally, in the data path, as a unicast or multicast packet.

2.1. Multicast Router Discovery Procedures in EVPN

The procedures defined in this section make use of the Multicast Router Discovery (MRD) route described in section 4 and are OPTIONAL. An EVPN router not implementing this specification will transparently flood PIM Hello messages and IGMP Queries to remote PEs.

2.1.1. Discovering PIM Routers

As described in [RFC4601] for shared LANs, an EVPN Broadcast Domain may have multiple PIM routers connected to it and a single one of these routers, the DR, will act on behalf of directly connected hosts with respect to the PIM-SM protocol. The DR election, as well as discovery and negotiation of options in PIM, is performed using Hello messages. PIM Hello messages are periodically exchanged and flooded in EVPN Broadcast Domains that don't follow this specification.

When PIM Proxy is enabled, an EVPN PE will snoop PIM Hello messages and forward them only to local ACs where PIM routers have been detected. This document assumes that all the procedures defined in

[VPLS-PIM-PROXY] to snoop PIM Hellos on local ACs and build the PIM Neighbor DB on the PEs are followed. PIM Hello messages MUST NOT be forwarded to remote EVPN PEs though.

Using Figure 1 as an example, the PIM Proxy operation for Hello messages is as follows:

- 1) The arrival of a new PIM Hello message at e.g. PE1 will trigger an MRD route advertisement including:
 - o The IP address and length of the multicast router that issued the Hello message. E.g. R1's IP address and length.
 - o The DR Priority copied from the Hello DR Priority TLV.
 - o Q flag set (if the multicast router is a Querier).
 - o P flag set that indicates the router is PIM capable.
- 2) All other PEs import the MRD route and do the following:
 - o Add the multicast router address to the PIM Neighbor Database (PIM Nbr DB) associated to the Originator Router Address.
 - o Generate a PIM hello where the IP Source Address is the Multicast Router IP and the DR Priority is copied from the route. This PIM hello is sent to all the local ACs connected to a PIM router. For example, PE3 will send the generated hello message to R4.
- 3) Each PE will build its PIM Nbr DB out of the local PIM hello messages and/or remote MRD routes. The PIM hello timers and other hello parameters are not propagated in the MRD routes.
 - o The timers are handled locally by the PE and as per [RFC4601]. This is valid for the hold_time (when a PIM router or PE receives a hello message, resets the neighbor-expiry timer), and other timers.
 - o The Generation ID option is also processed locally on the PE, as well as the Generation ID changes for a given multicast router. It is not propagated in the MRD route.
 - o Procedures described in [RFC4601] are used to remove a local AC PIM router from the PIM Nbr DB. When a local router is removed from the DB, the MRD route is withdrawn. If the local router is still sending Queries, the route is updated with flags P=0 and Q=1. Upon receiving the update, the other PEs will remove the router from the PIM Nbr DB but not from the list of queriers.
- 4) Based on regular PIM DR election procedures (highest DR Priority or highest IP), each PE is aware of who the DR is for the BD. For more information, refer to section "3. Interaction with IGMP-snooping and Sources".

2.1.2. Discovering IGMP Queriers

In (EVPN) Broadcast Domains that are shared among not only PIM routers but also IGMP hosts, one or more PIM routers will also be configured as IGMP Queriers. The proxy Querier mechanism described in [EVPN-IGMP-MLD-PROXY] suppresses the flooding of queries on the Broadcast Domain, by using PE generated Queries from an anycast IP address.

While the proxy Querier mechanism works in most of the use-cases, sometimes it is desired to have a more transparent behavior and propagate existing multicast router IGMP Queries as opposed to "blindly" querying all the hosts from the PEs. The MRD route defined in section 4 can be used for that purpose.

When the discovered local PIM router is also sending IGMP Queries, the PE will issue an MRD route for the multicast router with both Q (IGMP Querier) and P (PIM router) flags set. Note that the PE may set both flags or only one of them, depending on the capabilities of the local router.

A PE receiving an MRD route with Q=1 will generate IGMP Query messages, using the multicast router IP address encoded in the received MRD route. If more than one IGMP Queriers exist in the EVI, the PE receiving the MRD routes with Q=1 will select the lower IP address, as per [RFC2236]. Note that, upon receiving the MRD routes with Q=1, the PE must generate IGMP Queries and forward them to all the local ACs. Other Queriers listening to these received Query messages will stop sending Queries if they are no longer the selected Querier, as per [RFC2236].

This procedure allows the EVPN PEs to act as proxy Queriers, but using the IP address of the best existing IGMP Querier in the EVPN Broadcast Domain. This can help IGMP hosts troubleshoot any issues on the IGMP routers and check their connectivity to them.

2.2. PIM Join/Prune Proxy Procedures

This section describes the procedures associated to the PIM Proxy function for Join and Prune messages. This document assumes that all the procedures defined in [VPLS-PIM-PROXY] to build multicast states on the PEs' local ACs are followed. Figure 2 illustrates an scenario where PIM Proxy is enabled on the EVPN PEs.

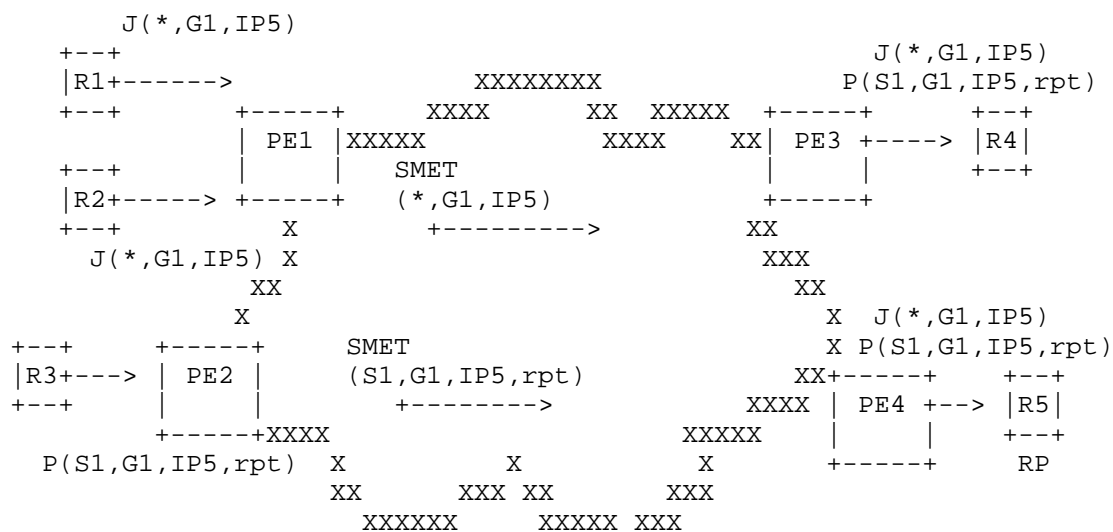


Figure 2 - Proxy PIM Join/Prune in EVPN

PIM J/P messages are sent by the routers towards upstream sources and RPs:

- o (*,G) is used in Join/Prune messages that are sent towards the RP for the specified group.
- o (S,G) used in Join/Prune messages sent towards the specified source.
- o (S,G,rpt) is used in Join/Prune messages sent towards the RP. We refer to this as RPT message and the Prune message always precedes the Join message. The typical sequence of PIM messages (for a group) seen in a BD connecting PIM routers is the following:
 - a) (*,G) Join issued by a downstream router to the RP (to join the RP Tree).
 - b) (S,G) Join issued by a downstream router switching to the SPT.
 - c) (S,G,rpt) Prune issued by a downstream router to the RP to prune a specific source from the RPT.
 - d) (S,G) Prune issued by a downstream router no longer interested in the SPT.
 - e) (S,G,rpt) Join issued by a downstream router interested (again) in the RPT for (S,G).

The Proxy PIM procedures for Join/Prune messages are summarized as follows:

- 1) Downstream PE procedures:

- o A downstream PE will snoop PIM Join/Prune messages and won't forward them to remote PEs.
- o Triggered by the reception of the PIM Join message, a downstream PE will advertise an SMET route, including the source, group and Upstream Neighbor as received from the PIM Join message. A single SMET route is advertised per source, group, with the P flag set. As an example, in Figure 2, PE1 receives two PIM Join messages for the same source, group and Upstream Neighbor, however PE1 advertises a single SMET route.
- o When the last connected router sends a PIM Prune message for a given source, group and Upstream Neighbor and the state is removed, the PE will withdraw the SMET route (note that the state is removed once the prune-pend timer expires).
- o SMET routes must always be generated upon receiving a PIM Join message, irrespective of the location of the Upstream Neighbor and even if the Upstream Neighbor is local to the PE.
- o A downstream PE receiving a PIM Prune (S,G,rpt) message will trigger an RPT-Prune route for the source and group. Subsequently, if the downstream PE receives a PIM Join (S,G,rpt) to cancel the previous Prune (S,G,rpt) and keep pulling the multicast traffic from the RPT, the downstream PE will withdraw the RPT-Prune route.
- o PIM Timers are handled locally. If the holdtime expires for a local Join the PE withdraws the SMET route.

3) Upstream PE procedures:

- o A received SMET route with P=1 will add state for the source and group and will generate a PIM Join message for the source, group that will be forwarded to all the local AC PIM routers.
- o A received SMET route withdrawal will remove the state and generate a PIM Prune message for the source, group and upstream neighbor that will be forwarded to all the local AC PIM routers.
- o A received RPT-Prune route for (S,G) will generate a PIM Prune (S,G,rpt) message that will be forwarded to all the local AC PIM routers.
- o A received RPT-Prune withdrawal for (S,G) will generate a PIM Join (S,G,rpt) message that will be forwarded to all the local AC PIM routers.

It is important to note that, compared to a solution that does not snoop PIM messages and does not use BGP to propagate states in the core, this EVPN PIM Proxy solution will add some latency derived from the procedures described in this document.

2.3. PIM Assert Optimization

The PIM Assert process described in [RFC4601] is intense in terms of resource consumption in the PIM routers, however it is needed in case PIM routers share a multi-access transit LAN. The use of PIM Proxy for EVPN BDs can minimize and even suppress the need for PIM Assert as described in this section.

As a refresher, the PIM Assert procedures are needed to prevent two or more Upstream PIM routers from forwarding the same multicast content to the group of Downstream PIM routers sharing the same (EVPN) Broadcast Domain. This multicast packet duplication may happen in any of the following cases:

- o Two or more Downstream PIM routers on the BD may issue (*,G) Joins to different upstream routers on the BD because they have inconsistent MRIB entries regarding how to reach the RP. Both paths on the RP tree will be set up, causing two copies of all the shared tree traffic to appear on the EVPN Broadcast Domain.
- o Two or more routers on the BD may issue (S,G) Joins to different upstream routers on the BD because they have inconsistent MRIB entries regarding how to reach source S. Both paths on the source-specific tree will be set up, causing two copies of all the traffic from S to appear on the BD.
- o A router on the BD may issue a (*,G) Join to one upstream router on the BD, and another router on the BD may issue an (S,G) Join to a different upstream router on the same BD. Traffic from S may reach the BD over both the RPT and the SPT. If the receiver behind the downstream (*,G) router doesn't issue an (S,G,rpt) prune, then this condition would persist.

PIM does not prevent such duplicate joins from occurring; instead, when duplicate data packets appear on the same BD from different routers, these routers notice this and then elect a single forwarder. This election is performed using the PIM Assert procedure.

The issue is minimized or suppressed in this document by making sure all the Upstream PEs select the same Upstream Neighbor for a given (*,G) or (S,G) in any of the three above situations. If there is only one upstream PIM router selected and the same multicast content is

not allowed to be flooded from more than one Upstream Neighbor, there will not be multicast duplication or need for Assert procedures in the EVPN Broadcast Domain.

Figure 3 illustrates an example of the PIM Assert Optimization in EVPN.

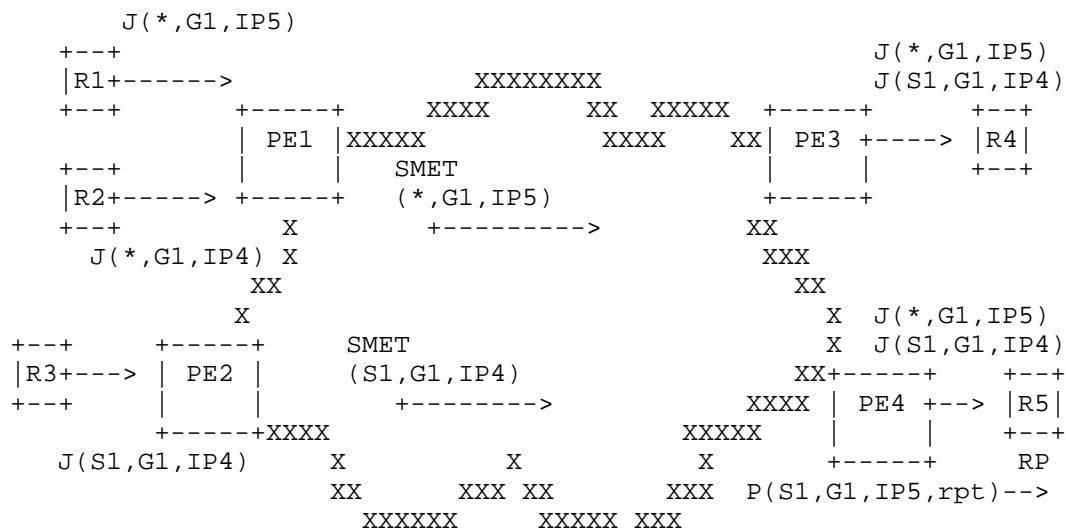


Figure 3 - Proxy PIM Assert Optimization in EVPN

2.3.1 Assert Optimization Procedures in Downstream PEs

The Downstream PES will trigger SMET routes based on the received PIM Join messages. This is their behavior when any of the three situations described in section 2.3 occurs:

- o If the Downstream PE receives two local (*,G) Joins to different Upstream Neighbors, the PE will generate a single SMET route, selecting the highest IP address. In Figure 3, if we assume R1 issues J(*,G1,IP5) and R2 J(*,G1,IP4), PE1 will advertise an SMET route for (*,G,IP5). If PE1 had already advertised (*,G1,IP4), it would have sent an update with (*,G1,IP5). Note that the Upstream Router IP address is not part of the SMET route key, hence there is no need to withdraw the previous (*,G1,IP4).
- o In the same way, if the Downstream PE receives two local (S,G) Joins to different Upstream Neighbors, the PE will generate a single SMET route, selecting the highest IP address.

- o If the Downstream PE receives a local (S,G) and a local (*,G) Joins for the same group but to different Upstream Neighbors, the PE will generate two different SMET routes (since *,G and S,G make two different route keys), keeping the original Upstream Neighbors in the SMET routes.

2.3.2 Assert Optimization Procedures in Upstream PEs

Upon receiving two or more SMET routes for the same group but different Upstream Neighbors, the Upstream PEs will follow this procedure:

- 1) The Upstream PE will select a unique Upstream Neighbor based on the following rules:
 - a) The Upstream Neighbor encoded in a (S,G) SMET route has precedence over the Upstream Neighbor on the (*,G) SMET route for the same group. This is consistent with the Assert winner election in [RFC4601]. In the example of Figure 3, PE3 and PE4 will select IP4 as the Upstream Neighbor for (S1,G1) and (*,G1).
 - b) In case the SMET routes have the same source (* or S), the higher Upstream Neighbor IP Address wins.
- 2) After selecting the Unique Upstream Neighbor, the PE will instruct the data path to discard any ingress multicast stream that is coming from an interface different than the selected Upstream Neighbor for the multicast group. In the example in Figure 3, PE4 will not accept G1 multicast traffic from R5.
- 3) Then the PE will generate the corresponding local PIM messages as usual. In the example, PE3 and PE4 generate PIM Join messages for (S1,G1,IP4) and (*,G1,IP5).
- 4) The PE connected to the non-selected Upstream Neighbor will issue a PIM (S,G)/(*,G) Prune or a PIM (S,G,rpt) Prune to make sure the non-selected Upstream Router does not forward traffic for the group anymore. In the example, PE4 will issue a local (S1,G1,rpt) Prune message to R5, so that R5 does not forward G1 traffic.

In case of any change that impacts on the Upstream Neighbor selection for a given group G1, the upstream PEs will simply update the Upstream Neighbor selection and follow the above procedure. This mechanism prevents the multicast duplication in the EVPN Broadcast Domain and avoids PIM Assert procedures among PIM routers in the BD.

2.4. EVPN Multi-Homing and State Synchronization

PIM Join/Prune States will be synchronized across all the PEs in an Ethernet Segment by using the procedures described in [EVPN-IGMP-MLD-PROXY] and the IGMP/PIM Join Synch Route with the corresponding Flag P set. This document does not require the use of IGMP Leave Synch Routes.

In the same way, RPT-Prune States can be synchronized by using the PIM RPT-Prune Synch route. The generation and process for this route follows similar procedures as for the IGMP/PIM Join Synch Route.

In order to synchronize the PIM Neighbors discovered on an Ethernet Segment, the MRD route and its ESI value will be used. Upon receiving a Hello message on a link that is part of a multi-homed Ethernet Segment, the PE will issue an MRD route that encodes the ESI value of the AC over which the Hello was received. Upon receiving the non-zero ESI MRD route, the PEs in the same ES will add the router to their PIM Neighbor DB, using their AC on the same ES as the PIM Neighbor port. This will allow the DF on the ES to generate Hello messages for the local PIM router.

A PE that is not part of the ESI would normally receive a single non-zero ESI MRD route per multicast router. In certain transient situations the PE may receive more than one non-zero ESI MRD route for the same multicast router. The PE should recognize this and not generate additional PIM Hello messages for the local ACs.

2.5. PIM Bootstrap and RP Discovery

This section will be covered in future revisions of this document.

2.6. PIM-DM (Dense Mode) Proxy Procedures

This section will be covered in future revisions of this document.

3. Interaction with IGMP-snooping and Sources

Figure 4 illustrates an example with a multicast source, an IGMP host and a PIM router in the same EVPN BD.

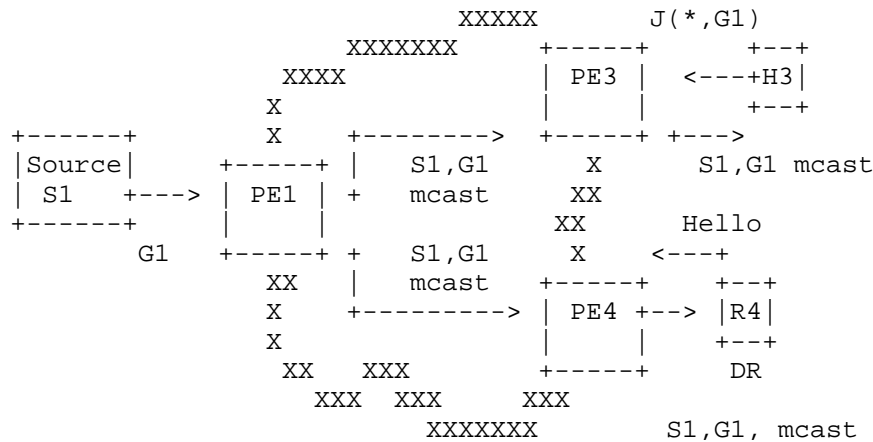


Figure 4 - Proxy PIM interaction with local sources and hosts

When PIM routers, multicast sources and IGMP hosts coexist in the same EVPN Broadcast domain, the PEs supporting both IGMP and PIM proxy will provide the following optimizations in the EVPN BD:

- o If an IGMP host and a PIM router are connected to the same BD on a PE, the PE will advertise a single SMET route per (S,G) or (*,G) irrespective of the received IGMP or PIM message. The IGMP flags can be simultaneously set along with the P flag.
- o In the same way, if IGMP hosts and PIM routers are connected to the same MAC-VRF and Ethernet Segment, the IGMP/PIM Join Synch route can be shared by a host and a router requesting the same multicast source and group.
- o A PE connected to a Source and using Ingress Replication will forward a multicast stream (S1,G1) to all the egress PEs that advertised an SMET route for (S1,G1) and all the egress PEs that advertised an MRD route for the EVPN BD.

4. BGP Information Model

This document defines the following additional routes and requests IANA to allocate a type value in the EVPN route type registry:

- + Type TBD - Multicast Router Discovery (MRD) Route
- + Type TBD - PIM RPT-Prune Route

+ Type TBD - PIM RPT-Prune Join Synch Route

In addition, the following routes defined in [EVPN-IGMP-MLD-PROXY] are re-used and extended in this document's procedures:

+ Type 6 - Selective Multicast Ethernet Tag Route
 + Type 7 - IGMP Join Synch Route

Where Type 7 is requested to be re-named as IGMP/PIM Join Synch Route.

4.1 Multicast Router Discovery (MRD) Route

Figure 5 shows the content of the MRD route:

	RD (8 octets)	
	Ethernet Segment ID (10 octets)	
	Ethernet Tag ID (4 octets)	
	Originator Router Length (1 octet)	
	Originator Router Address (Variable)	
	Mcast Router Length (1 octet)	
	Mcast Router Address 1 (variable)	
	Secondary Address List Length (1 octet)	
	Secondary Mcast Router Address 1 (variable)	
	.	
	Secondary Mcast Router Address n (variable)	
	DR Priority (4 octets)	
	Flags (1 octet)	

Figure 5 Multicast Router Discovery Route

The support for this new route type is OPTIONAL. Since this new route type is OPTIONAL, an implementation not supporting it MUST ignore the route, based on the unknown route type value, as specified by Section 5.4 in [RFC7606].

The encoding of this route is defined as follows:

- o RD, ESI and Ethernet Tag ID are defined as per [RFC7432] for MAC/IP routes.
- o The Originator Router Length and Address encode and IPv4 or IPv6 address that belongs to the advertising PE.
- o The Multicast Router Length and Address field encode the Primary IP address of the PIM neighbor added to the PE's DB.
- o The Secondary Address List Length encodes the number of Secondary IP addresses advertised by the PIM router in the PIM Hello message. If this field is zero, the NLRI will not include any Secondary Multicast Router Address. All the IP addresses will have the same Length, that is, they will all be either IPv4 or IPv6, but not a mix of both.
- o DR Priority is copied from the same field in Hello packets, as per [RFC4601].
- o Flags:
 - Q: Querier flag. Least significant bit. It indicates the encoded multicast router is an IGMP Querier.
 - P: PIM router flag. Second low order bit in the Flags octet. It indicates that the multicast router is a PIM router.
 - Q and P may be set simultaneously.

For BGP processing purposes, only the RD, Ethernet Tag ID, Originator Router Length and Address, and Multicast Router Length and Address are considered part of the route key. The Secondary Multicast Router Addresses and the rest of the fields are not part of the route key.

4.2 Selective Multicast Ethernet Tag Route for PIM Proxy

The extended SMET route used in this document is shown in Figure 6.

NOTE: this route may use the SMET route type, or may be a different route type PIM SMET route. This is TBD.

RD (8 octets)
Ethernet Tag ID (4 octets)
Multicast Source Length (1 octet)
Multicast Source Address (variable)
Multicast Group Length (1 octet)
Multicast Group Address (Variable)
Originator Router Length (1 octet)
Originator Router Address (variable)
Flags (1 octets) (optional)
Upstream Router Length (1B)(optional)
Upstream Router Addr (variable)(opt)

Flags:

0	1	2	3	4	5	6	7
+	-	+	-	+	-	+	-
				P	IE	v3	v2
+	-	+	-	+	-	+	-

Figure 6 Selective Multicast Ethernet Tag Route and Flags

As in the case of the MRD route, this route type is OPTIONAL.

This route will be used as per [EVPN-IGMP-MLD-PROXY], with the following extra and optional fields:

- o Upstream Router Length and Address will contain the same information as received in a PIM Join/Prune message on a local AC. There is only one Upstream Router Address per route.
- o Flags: This field encodes Flags that are now relevant to IGMP and PIM. The following new Flag is defined:
 - Flag P: Indicates the SMET route is generated by a received PIM

Join on a local AC. When P=1, the Upstream Router Length and Address fields are present in the route. Otherwise the two fields will not be present.

Compared to [EVPN-IGMP-MLD-PROXY] there is no change in terms of fields considered part of the route key for BGP processing. The Upstream Router Length and Address are not considered part of the route key.

4.3 PIM RPT-Prune Route

The RPT-Prune route is analogous to the SMET route but for PIM RPT-Prune messages. The SMET routes cannot be used to convey RPT-Prune messages because they are always triggered by IGMP or PIM Join messages. A PIM RPT-Prune message is used to Prune a specific (S,G) from the RP Tree by downstream routers. An RPT-Prune message is typically seen prior to an RPT-Join message for the (S,G), hence it requires its own BGP route.

+-----+	
RD (8 octets)	
+-----+	
Ethernet Tag ID (4 octets)	
+-----+	
Multicast Source Length (1 octet)	
+-----+	
Multicast Source Address (variable)	
+-----+	
Multicast Group Length (1 octet)	
+-----+	
Multicast Group Address (Variable)	
+-----+	
Originator Router Length (1 octet)	
+-----+	
Originator Router Address (variable)	
+-----+	
Upstream Router Length (1B)	
+-----+	
Upstream Router Addr (variable)	
+-----+	

Figure 7 PIM RPT-Prune Route

Fields are defined in the same way as for the SMET route.

4.4 IGMP/PIM Join Synch Route for PIM Proxy

This document renames the IGMP Join Synch Route as IGMP/PIM Join Synch Route and extends it with new fields and Flags as shown in Figure 8:

NOTE: this route may use and extend the IGMP Join Synch Route, or may turn out to be a different route type in future revisions. This is TBD.

+-----+	
	RD (8 octets)
+-----+	
	Ethernet Segment Identifier (10 octets)
+-----+	
	Ethernet Tag ID (4 octets)
+-----+	
	Multicast Source Length (1 octet)
+-----+	
	Multicast Source Address (variable)
+-----+	
	Multicast Group Length (1 octet)
+-----+	
	Multicast Group Address (Variable)
+-----+	
	Originator Router Length (1 octet)
+-----+	
	Originator Router Address (variable)
+-----+	
	Flags (1 octet)
+-----+	
	Upstream Router Length (1B)(optional)
+-----+	
	Upstream Router Addr (variable)(opt)
+-----+	

Flags:

0	1	2	3	4	5	6	7
+---+---+---+---+---+---+---+---+							
					P IE v3 v2 v1		
+---+---+---+---+---+---+---+---+							

Figure 8 IGMP/PIM Join Synch Route and Flags

This route will be used as per [EVPN-IGMP-MLD-PROXY], with the following extra and optional fields:

- o Upstream Router Length and Address will contain the same information as received in a PIM Join/Prune message on a local AC. There is only one Upstream Router Address per route.
- o Flags: This field encodes Flags that are now relevant to IGMP and PIM. The following new Flag is defined:
 - Flag P: Indicates the Join Synch route is generated by a received PIM Join on a local AC. When P=1, the Upstream Router Length and Address fields are present in the route. Otherwise the two fields will not be present.

Compared to [EVPN-IGMP-MLD-PROXY] there is no change in terms of fields considered part of the route key for BGP processing. The Upstream Router Length and Address are not considered part of the route key.

4.5 IGMP/PIM RPT-Prune Synch Route for PIM Proxy

This new route is used to Synch RPT-Prune states among the PEs in the Ethernet Segment.

+-----+		
	RD (8 octets)	
+-----+		
	Ethernet Segment Identifier (10 octets)	
+-----+		
	Ethernet Tag ID (4 octets)	
+-----+		
	Multicast Source Length (1 octet)	
+-----+		
	Multicast Source Address (variable)	
+-----+		
	Multicast Group Length (1 octet)	
+-----+		
	Multicast Group Address (Variable)	
+-----+		
	Originator Router Length (1 octet)	
+-----+		
	Originator Router Address (variable)	
+-----+		
	Upstream Router Length (1B)(optional)	
+-----+		
	Upstream Router Addr (variable)(opt)	
+-----+		

Figure 9 IGMP/PIM RPT-Prune Synch Route

The RD, Ethernet Segment Identifier and other fields are defined as for the IGMP/PIM Join Synch Route. In addition, the Upstream Router Length and Address will contain the same information as received in a PIM RPT-Prune message on a local AC. The Upstream Router points at the RP for the source and group and there is only one Upstream Router Address per route.

The route key for BGP processing is defined as per the IGMP/PIM Join Synch route.

5. Conclusions

This document extends the IGMP Proxy concept of [EVPN-IGMP-MLD-PROXY] to PIM, so that EVPN can also be used to minimize the flooding of PIM control messages and optimize the delivery of IP multicast traffic in EVPN Broadcast Domains that connect PIM routers.

This specification describes procedures to Discover new PIM routers in the BD, as well as propagate PIM Join/Prune messages using EVPN SMET routes and other optimizations.

6. Conventions used in this document

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].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

7. Security Considerations

This section will be added in future versions.

8. IANA Considerations

This document requests IANA to allocate a new EVPN route type in the corresponding registry:

- + Type TBD - Multicast Router Discovery (MRD) Route

- + Type TBD - PIM RPT-Prune Route
- + Type TBD - PIM RPT-Prune Join Synch Route

In addition, the following route defined in [EVPN-IGMP-MLD-PROXY] should be renamed as follows:

- + Type 7 - IGMP/PIM Join Synch Route

9. Terminology

- o EVI: EVPN Instance.
- o EVPN Broadcast Domain: it refers to an EVI in case of VLAN-based and VLAN-bundle interfaces. It refers to a Bridge Domain identified by an Ethernet-Tag (in the control plane) in case of VLAN-Aware Bundle interfaces.
- o AC: Attachment Circuit.
- o PIM-DM: Protocol Independent Multicast - Dense Mode.
- o PIM-SM: Protocol Independent Multicast - Sparse Mode.
- o PIM-SSM: Protocol Independent Multicast - Source Specific Mode.
- o S: IP address of the multicast source.
- o G: IP address of the multicast group.
- o N: Upstream neighbor field in a Join/Prune/Graft message.
- o PIM J/P: PIM Join/Prune messages.
- o RP: PIM Rendezvous Point.
- o MRD route: Multicast Router Discovery.
- o PIM Nbr: PIM Neighbor.

10. References

10.1 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>>.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <<http://www.rfc-editor.org/info/rfc4601>>.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <<http://www.rfc-editor.org/info/rfc2236>>.

[VPLS-PIM-PROXY] Dornon, O. et al, "Protocol Independent Multicast (PIM) over Virtual Private LAN Service (VPLS)", June 2017, work-in-progress, draft-ietf-pals-vpls-pim-snooping-06.

[EVPN-IGMP-MLD-PROXY] Sajassi, A. et al, "IGMP and MLD Proxy for EVPN", March 2017, work-in-progress, draft-ietf-bess-evpn-igmp-mld-proxy-00.

10.2 Informative References

[EVPN-PROXY-ARP-ND] Rabadan, J. et al, "Operational Aspects of Proxy-ARP/ND in EVPN Networks", April 2017, work-in-progress, draft-ietf-bess-evpn-proxy-arp-nd-02.

11. Acknowledgments

12. Contributors

13. 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

Zhaohui Zhang
Juniper Networks
EMail: zzhang@juniper.net

Ali Sajassi
Cisco
Email: sajassi@cisco.com

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
J. Kotalwar
S. Sathappan
Nokia

Z. Zhang
Juniper

A. Sajassi
Cisco

Expires: May 3, 2018

October 30, 2017

PIM Proxy in EVPN Networks
draft-skr-bess-evpn-pim-proxy-01

Abstract

Ethernet Virtual Private Networks [RFC7432] are becoming prevalent in Data Centers, Data Center Interconnect (DCI) and Service Provider VPN applications. One of the goals that EVPN pursues is the reduction of flooding and the efficiency of CE-based control plane procedures in Broadcast Domains. Examples of this are Proxy ARP/ND and IGMP/MLD Proxy. This document complements the latter, describing the procedures required to minimize the flooding of PIM messages in EVPN Broadcast Domains, and optimize the IP Multicast delivery between PIM routers.

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 May 3, 2018.

Copyright 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	3
2. PIM Proxy Operation in EVPN Broadcast Domains	4
2.1. Multicast Router Discovery Procedures in EVPN	5
2.1.1. Discovering PIM Routers	5
2.1.2. Discovering IGMP Queriers	7
2.2. PIM Join/Prune Proxy Procedures	7
2.3. PIM Assert Optimization	10
2.3.1 Assert Optimization Procedures in Downstream PEs	11
2.3.2 Assert Optimization Procedures in Upstream PEs	12
2.4. EVPN Multi-Homing and State Synchronization	13
3. Interaction with IGMP-snooping and Sources	13
4. BGP Information Model	14
4.1 Multicast Router Discovery (MRD) Route	15
4.2 Selective Multicast Ethernet Tag Route for PIM Proxy	16
4.3 PIM RPT-Prune Route	18
4.4 IGMP/PIM Join Synch Route for PIM Proxy	19
4.5 IGMP/PIM RPT-Prune Synch Route for PIM Proxy	20
5. Conclusions	21
6. Conventions used in this document	21
7. Security Considerations	21
8. IANA Considerations	21
9. Terminology	22

10. References	22
10.1 Normative References	22
10.2 Informative References	23
11. Acknowledgments	23
12. Contributors	23
13. Authors' Addresses	23

1. Introduction

Ethernet Virtual Private Networks [RFC7432] are becoming prevalent in Data Centers, Data Center Interconnect (DCI) and Service Provider VPN applications. One of the goals that EVPN pursues is the reduction of flooding and the efficiency of CE-based control plane procedures in Broadcast Domains. Examples of this are [EVPN-PROXY-ARP-ND] for improving the efficiency of CE's ARP/ND protocols, and [EVPN-IGMP-MLD-PROXY] for IGMP/MLD protocols.

This document focuses on optimizing the behavior of PIM in EVPN Broadcast Domains and re-uses some procedures of [EVPN-IGMP-MLD-PROXY]. The reader is also advised to check out [RFC8220] to understand certain aspects of the procedures of PIM Join/Prune messages received on Attachment Circuits (ACs).

Section 2 describes the PIM Proxy procedures that the implementation should follow, including:

- o The use of EVPN to suppress the flooding of PIM Hello messages in shared Broadcast Domains. The benefit of this is twofold:
 - PIM Hello messages will ONLY be flooded to Attachment Circuits that are connected to PIM routers, as opposed to all the CEs and hosts in the Broadcast Domain.
 - Soft-state PIM Hello messages will be replaced by hard-state BGP messages that don't need to be refreshed periodically.
- o The use of EVPN to discover IGMP Queriers, while avoiding the flooding of IGMP Queries in the core.
- o The procedures to proxy PIM Join/Prune messages and replace them by hard-state EVPN routes that don't need to be refreshed periodically. By using BGP EVPN to propagate both, Hello and Join/Prune messages, we also avoid out-of-order delivery between both types of PIM messages.
- o This document also describes an EVPN based procedure so that the PIM routers connected to the shared Broadcast Domain don't need to

run any PIM Assert procedure. PIM Assert procedures may be expensive for PIM routers in terms of resource consumption. With this procedure, there is no PIM Assert needed on PIM routers.

- o The use of procedures similar to the ones defined in [EVPN-IGMP-MLD-PROXY] to synchronize multicast states among the PEs in the same Ethernet Segment.

Section 3 describes the interaction of PIM Proxy with IGMP Proxy PEs and Multicast Sources connected to the same EVPN Broadcast Domain.

Section 4 defines the BGP Information Model that this document requires to address the PIM Proxy procedures.

This document assumes the reader is familiar with PIM and IGMP protocols.

2. PIM Proxy Operation in EVPN Broadcast Domains

This section describes the operation of PIM Proxy in EVPN Broadcast Domains (BDs). Figure 1 depicts an EVPN Broadcast Domain defined in four PEs that are connected to PIM routers. This example will be used throughout this section and assumes both R4 and R5 are PIM Upstream Neighbors for PIM routers R1, R2 and R3 and multicast group G1. In this situation, the PIM multicast traffic flows from R4 or R5 to R1, R2 and R3. The PIM Join/Prune signaling will flow in the opposite direction. From a terminology perspective, we consider PE1 and PE2 as egress or downstream PEs, whereas PE3 and PE4 are ingress or upstream PEs.

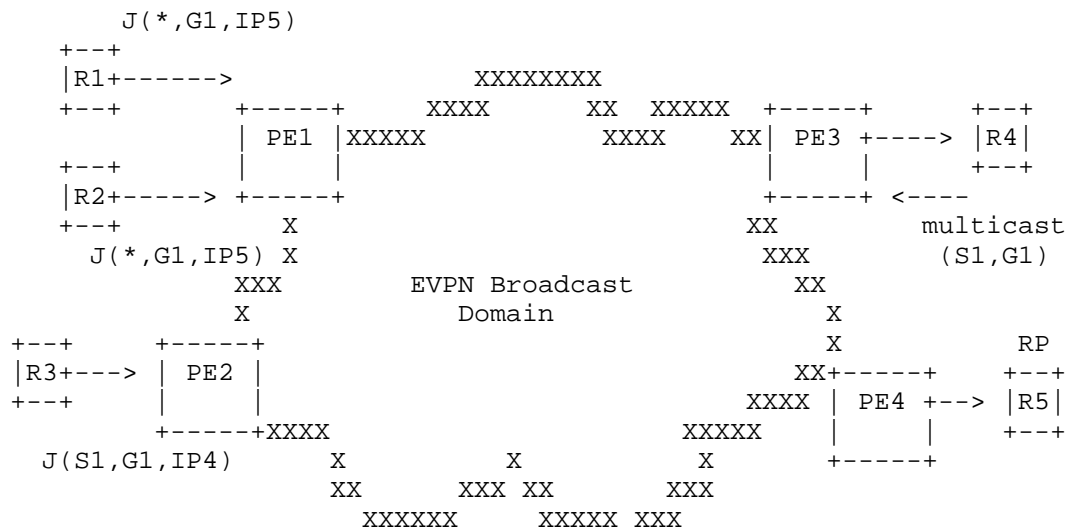


Figure 1 - PIM Routers connected by an EVPN Broadcast Domain

It is important to note that any Router's PIM message not explicitly specified in this document will be forwarded by the PEs normally, in the data path, as a unicast or multicast packet.

2.1. Multicast Router Discovery Procedures in EVPN

The procedures defined in this section make use of the Multicast Router Discovery (MRD) route described in section 4 and are OPTIONAL. An EVPN router not implementing this specification will transparently flood PIM Hello messages and IGMP Queries to remote PEs.

2.1.1. Discovering PIM Routers

As described in [RFC4601] for shared LANs, an EVPN Broadcast Domain may have multiple PIM routers connected to it and a single one of these routers, the DR, will act on behalf of directly connected hosts with respect to the PIM-SM protocol. The DR election, as well as discovery and negotiation of options in PIM, is performed using Hello messages. PIM Hello messages are periodically exchanged and flooded in EVPN Broadcast Domains that don't follow this specification.

When PIM Proxy is enabled, an EVPN PE will snoop PIM Hello messages and forward them only to local ACs where PIM routers have been detected. This document assumes that all the procedures defined in

[RFC8220] to snoop PIM Hellos on local ACs and build the PIM Neighbor DB on the PEs are followed. PIM Hello messages MUST NOT be forwarded to remote EVPN PEs though.

Using Figure 1 as an example, the PIM Proxy operation for Hello messages is as follows:

- 1) The arrival of a new PIM Hello message at e.g. PE1 will trigger an MRD route advertisement including:
 - o The IP address and length of the multicast router that issued the Hello message. E.g. R1's IP address and length.
 - o The DR Priority copied from the Hello DR Priority TLV.
 - o Q flag set (if the multicast router is a Querier).
 - o P flag set that indicates the router is PIM capable.
- 2) All other PEs import the MRD route and do the following:
 - o Add the multicast router address to the PIM Neighbor Database (PIM Nbr DB) associated to the Originator Router Address.
 - o Generate a PIM hello where the IP Source Address is the Multicast Router IP and the DR Priority is copied from the route. This PIM hello is sent to all the local ACs connected to a PIM router. For example, PE3 will send the generated hello message to R4.
- 3) Each PE will build its PIM Nbr DB out of the local PIM hello messages and/or remote MRD routes. The PIM hello timers and other hello parameters are not propagated in the MRD routes.
 - o The timers are handled locally by the PE and as per [RFC4601]. This is valid for the hold_time (when a PIM router or PE receives a hello message, resets the neighbor-expiry timer), and other timers.
 - o The Generation ID option is also processed locally on the PE, as well as the Generation ID changes for a given multicast router. It is not propagated in the MRD route.
 - o Procedures described in [RFC4601] are used to remove a local AC PIM router from the PIM Nbr DB. When a local router is removed from the DB, the MRD route is withdrawn. If the local router is still sending Queries, the route is updated with flags P=0 and Q=1. Upon receiving the update, the other PEs will remove the router from the PIM Nbr DB but not from the list of queriers.
- 4) Based on regular PIM DR election procedures (highest DR Priority or highest IP), each PE is aware of who the DR is for the BD. For more information, refer to section "3. Interaction with IGMP-snooping and Sources".

2.1.2. Discovering IGMP Queriers

In (EVPN) Broadcast Domains that are shared among not only PIM routers but also IGMP hosts, one or more PIM routers will also be configured as IGMP Queriers. The proxy Querier mechanism described in [EVPN-IGMP-MLD-PROXY] suppresses the flooding of queries on the Broadcast Domain, by using PE generated Queries from an anycast IP address.

While the proxy Querier mechanism works in most of the use-cases, sometimes it is desired to have a more transparent behavior and propagate existing multicast router IGMP Queries as opposed to "blindly" querying all the hosts from the PEs. The MRD route defined in section 4 can be used for that purpose.

When the discovered local PIM router is also sending IGMP Queries, the PE will issue an MRD route for the multicast router with both Q (IGMP Querier) and P (PIM router) flags set. Note that the PE may set both flags or only one of them, depending on the capabilities of the local router.

A PE receiving an MRD route with Q=1 will generate IGMP Query messages, using the multicast router IP address encoded in the received MRD route. If more than one IGMP Queriers exist in the EVI, the PE receiving the MRD routes with Q=1 will select the lower IP address, as per [RFC2236]. Note that, upon receiving the MRD routes with Q=1, the PE must generate IGMP Queries and forward them to all the local ACs. Other Queriers listening to these received Query messages will stop sending Queries if they are no longer the selected Querier, as per [RFC2236].

This procedure allows the EVPN PEs to act as proxy Queriers, but using the IP address of the best existing IGMP Querier in the EVPN Broadcast Domain. This can help IGMP hosts troubleshoot any issues on the IGMP routers and check their connectivity to them.

2.2. PIM Join/Prune Proxy Procedures

This section describes the procedures associated to the PIM Proxy function for Join and Prune messages. This document assumes that all the procedures defined in [RFC8220] to build multicast states on the PEs' local ACs are followed. Figure 2 illustrates an scenario where PIM Proxy is enabled on the EVPN PEs.

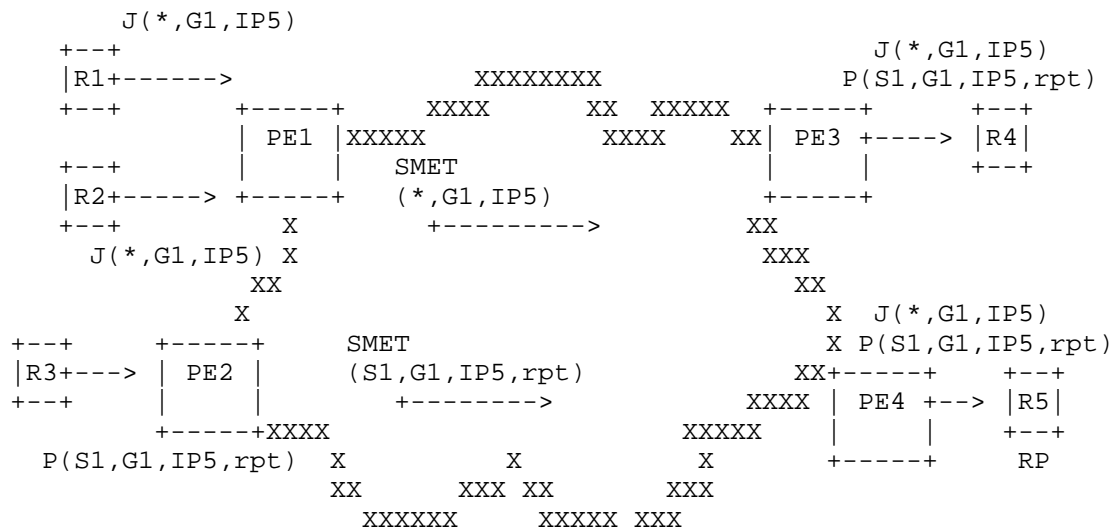


Figure 2 - Proxy PIM Join/Prune in EVPN

PIM J/P messages are sent by the routers towards upstream sources and RPs:

- o (*,G) is used in Join/Prune messages that are sent towards the RP for the specified group.
- o (S,G) used in Join/Prune messages sent towards the specified source.
- o (S,G,rpt) is used in Join/Prune messages sent towards the RP. We refer to this as RPT message and the Prune message always precedes the Join message. The typical sequence of PIM messages (for a group) seen in a BD connecting PIM routers is the following:
 - a) (*,G) Join issued by a downstream router to the RP (to join the RP Tree).
 - b) (S,G) Join issued by a downstream router switching to the SPT.
 - c) (S,G,rpt) Prune issued by a downstream router to the RP to prune a specific source from the RPT.
 - d) (S,G) Prune issued by a downstream router no longer interested in the SPT.
 - e) (S,G,rpt) Join issued by a downstream router interested (again) in the RPT for (S,G).

The Proxy PIM procedures for Join/Prune messages are summarized as follows:

- 1) Downstream PE procedures:

- o A downstream PE will snoop PIM Join/Prune messages and won't forward them to remote PEs.
- o Triggered by the reception of the PIM Join message, a downstream PE will advertise an SMET route, including the source, group and Upstream Neighbor as received from the PIM Join message. A single SMET route is advertised per source, group, with the P flag set. As an example, in Figure 2, PE1 receives two PIM Join messages for the same source, group and Upstream Neighbor, however PE1 advertises a single SMET route.
- o When the last connected router sends a PIM Prune message for a given source, group and Upstream Neighbor and the state is removed, the PE will withdraw the SMET route (note that the state is removed once the prune-pend timer expires).
- o SMET routes must always be generated upon receiving a PIM Join message, irrespective of the location of the Upstream Neighbor and even if the Upstream Neighbor is local to the PE.
- o A downstream PE receiving a PIM Prune (S,G,rpt) message will trigger an RPT-Prune route for the source and group. Subsequently, if the downstream PE receives a PIM Join (S,G,rpt) to cancel the previous Prune (S,G,rpt) and keep pulling the multicast traffic from the RPT, the downstream PE will withdraw the RPT-Prune route.
- o PIM Timers are handled locally. If the holdtime expires for a local Join the PE withdraws the SMET route.

3) Upstream PE procedures:

- o A received SMET route with P=1 will add state for the source and group and will generate a PIM Join message for the source, group that will be forwarded to all the local AC PIM routers.
- o A received SMET route withdrawal will remove the state and generate a PIM Prune message for the source, group and upstream neighbor that will be forwarded to all the local AC PIM routers.
- o A received RPT-Prune route for (S,G) will generate a PIM Prune (S,G,rpt) message that will be forwarded to all the local AC PIM routers.
- o A received RPT-Prune withdrawal for (S,G) will generate a PIM Join (S,G,rpt) message that will be forwarded to all the local AC PIM routers.

It is important to note that, compared to a solution that does not snoop PIM messages and does not use BGP to propagate states in the core, this EVPN PIM Proxy solution will add some latency derived from the procedures described in this document.

2.3. PIM Assert Optimization

The PIM Assert process described in [RFC4601] is intense in terms of resource consumption in the PIM routers, however it is needed in case PIM routers share a multi-access transit LAN. The use of PIM Proxy for EVPN BDs can minimize and even suppress the need for PIM Assert as described in this section.

As a refresher, the PIM Assert procedures are needed to prevent two or more Upstream PIM routers from forwarding the same multicast content to the group of Downstream PIM routers sharing the same (EVPN) Broadcast Domain. This multicast packet duplication may happen in any of the following cases:

- o Two or more Downstream PIM routers on the BD may issue (*,G) Joins to different upstream routers on the BD because they have inconsistent MRIB entries regarding how to reach the RP. Both paths on the RP tree will be set up, causing two copies of all the shared tree traffic to appear on the EVPN Broadcast Domain.
- o Two or more routers on the BD may issue (S,G) Joins to different upstream routers on the BD because they have inconsistent MRIB entries regarding how to reach source S. Both paths on the source-specific tree will be set up, causing two copies of all the traffic from S to appear on the BD.
- o A router on the BD may issue a (*,G) Join to one upstream router on the BD, and another router on the BD may issue an (S,G) Join to a different upstream router on the same BD. Traffic from S may reach the BD over both the RPT and the SPT. If the receiver behind the downstream (*,G) router doesn't issue an (S,G,rpt) prune, then this condition would persist.

PIM does not prevent such duplicate joins from occurring; instead, when duplicate data packets appear on the same BD from different routers, these routers notice this and then elect a single forwarder. This election is performed using the PIM Assert procedure.

The issue is minimized or suppressed in this document by making sure all the Upstream PEs select the same Upstream Neighbor for a given (*,G) or (S,G) in any of the three above situations. If there is only one upstream PIM router selected and the same multicast content is

- o If the Downstream PE receives a local (S,G) and a local (*,G) Joins for the same group but to different Upstream Neighbors, the PE will generate two different SMET routes (since *,G and S,G make two different route keys), keeping the original Upstream Neighbors in the SMET routes.

2.3.2 Assert Optimization Procedures in Upstream PEs

Upon receiving two or more SMET routes for the same group but different Upstream Neighbors, the Upstream PEs will follow this procedure:

- 1) The Upstream PE will select a unique Upstream Neighbor based on the following rules:
 - a) The Upstream Neighbor encoded in a (S,G) SMET route has precedence over the Upstream Neighbor on the (*,G) SMET route for the same group. This is consistent with the Assert winner election in [RFC4601]. In the example of Figure 3, PE3 and PE4 will select IP4 as the Upstream Neighbor for (S1,G1) and (*,G1).
 - b) In case the SMET routes have the same source (* or S), the higher Upstream Neighbor IP Address wins.
- 2) After selecting the Unique Upstream Neighbor, the PE will instruct the data path to discard any ingress multicast stream that is coming from an interface different than the selected Upstream Neighbor for the multicast group. In the example in Figure 3, PE4 will not accept G1 multicast traffic from R5.

NOTE: when the procedure selects an Upstream Neighbor between the (S,G) and (*,G) routes, we assume that the PE's interface that is connected to the non-selected Upstream Neighbor, is not shared with another Source for the same Group. In the example of Figure 3, this means that PE4's AC cannot be shared by R5 and S2 for the same group G. If PE4's AC is connected to a switch where R5 (RP) and S2 are connected, multicast traffic (S2,G) will be dropped by PE4, as per (2).

- 3) Then the PE will generate the corresponding local PIM messages as usual. In the example, PE3 and PE4 generate PIM Join messages for (S1,G1,IP4) and (*,G1,IP5).
- 4) The PE connected to the non-selected Upstream Neighbor will issue a PIM (S,G)/(*,G) Prune or a PIM (S,G,rpt) Prune to make sure the non-selected Upstream Router does not forward traffic for the group anymore. In the example, PE4 will issue a local (S1,G1,rpt) Prune message to R5, so that R5 does not forward G1 traffic.

In case of any change that impacts on the Upstream Neighbor selection for a given group G1, the upstream PEs will simply update the Upstream Neighbor selection and follow the above procedure. This mechanism prevents the multicast duplication in the EVPN Broadcast Domain and avoids PIM Assert procedures among PIM routers in the BD.

2.4. EVPN Multi-Homing and State Synchronization

PIM Join/Prune States will be synchronized across all the PEs in an Ethernet Segment by using the procedures described in [EVPN-IGMP-MLD-PROXY] and the IGMP/PIM Join Synch Route with the corresponding Flag P set. This document does not require the use of IGMP Leave Synch Routes.

In the same way, RPT-Prune States can be synchronized by using the PIM RPT-Prune Synch route. The generation and process for this route follows similar procedures as for the IGMP/PIM Join Synch Route.

In order to synchronize the PIM Neighbors discovered on an Ethernet Segment, the MRD route and its ESI value will be used. Upon receiving a Hello message on a link that is part of a multi-homed Ethernet Segment, the PE will issue an MRD route that encodes the ESI value of the AC over which the Hello was received. Upon receiving the non-zero ESI MRD route, the PEs in the same ES will add the router to their PIM Neighbor DB, using their AC on the same ES as the PIM Neighbor port. This will allow the DF on the ES to generate Hello messages for the local PIM router.

A PE that is not part of the ESI would normally receive a single non-zero ESI MRD route per multicast router. In certain transient situations the PE may receive more than one non-zero ESI MRD route for the same multicast router. The PE should recognize this and not generate additional PIM Hello messages for the local ACs.

3. Interaction with IGMP-snooping and Sources

Figure 4 illustrates an example with a multicast source, an IGMP host and a PIM router in the same EVPN BD.

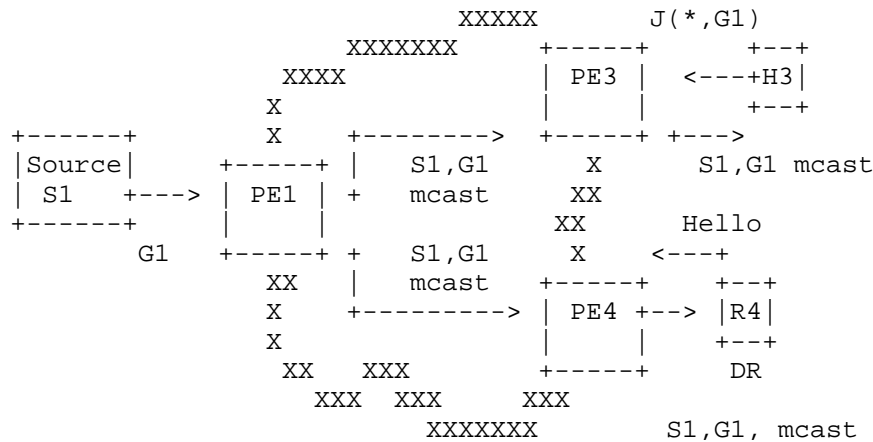


Figure 4 - Proxy PIM interaction with local sources and hosts

When PIM routers, multicast sources and IGMP hosts coexist in the same EVPN Broadcast domain, the PEs supporting both IGMP and PIM proxy will provide the following optimizations in the EVPN BD:

- o If an IGMP host and a PIM router are connected to the same BD on a PE, the PE will advertise a single SMET route per (S,G) or (*,G) irrespective of the received IGMP or PIM message. The IGMP flags can be simultaneously set along with the P flag.
- o In the same way, if IGMP hosts and PIM routers are connected to the same BD and Ethernet Segment, the IGMP/PIM Join Synch route can be shared by a host and a router requesting the same multicast source and group.
- o A PE connected to a Source and using Ingress Replication will forward a multicast stream (S1,G1) to all the egress PEs that advertised an SMET route for (S1,G1) and all the egress PEs that advertised an MRD route for the EVPN BD.

4. BGP Information Model

This document defines the following additional routes and requests IANA to allocate a type value in the EVPN route type registry:

- + Type TBD - Multicast Router Discovery (MRD) Route
- + Type TBD - PIM RPT-Prune Route

+ Type TBD - PIM RPT-Prune Join Synch Route

In addition, the following routes defined in [EVPN-IGMP-MLD-PROXY] are re-used and extended in this document's procedures:

+ Type 6 - Selective Multicast Ethernet Tag Route
 + Type 7 - IGMP Join Synch Route

Where Type 7 is requested to be re-named as IGMP/PIM Join Synch Route.

4.1 Multicast Router Discovery (MRD) Route

Figure 5 shows the content of the MRD route:

	RD (8 octets)	
	Ethernet Segment ID (10 octets)	
	Ethernet Tag ID (4 octets)	
	Originator Router Length (1 octet)	
	Originator Router Address (Variable)	
	Mcast Router Length (1 octet)	
	Mcast Router Address 1 (variable)	
	Secondary Address List Length (1 octet)	
	Secondary Mcast Router Address 1 (variable)	
	.	
	Secondary Mcast Router Address n (variable)	
	DR Priority (4 octets)	
	Flags (1 octet)	

Figure 5 Multicast Router Discovery Route

The support for this new route type is OPTIONAL. Since this new route type is OPTIONAL, an implementation not supporting it MUST ignore the route, based on the unknown route type value, as specified by Section 5.4 in [RFC7606].

The encoding of this route is defined as follows:

- o RD, ESI and Ethernet Tag ID are defined as per [RFC7432] for MAC/IP routes.
- o The Originator Router Length and Address encode and IPv4 or IPv6 address that belongs to the advertising PE.
- o The Multicast Router Length and Address field encode the Primary IP address of the PIM neighbor added to the PE's DB.
- o The Secondary Address List Length encodes the number of Secondary IP addresses advertised by the PIM router in the PIM Hello message. If this field is zero, the NLRI will not include any Secondary Multicast Router Address. All the IP addresses will have the same Length, that is, they will all be either IPv4 or IPv6, but not a mix of both.
- o DR Priority is copied from the same field in Hello packets, as per [RFC4601].
- o Flags:
 - Q: Querier flag. Least significant bit. It indicates the encoded multicast router is an IGMP Querier.
 - P: PIM router flag. Second low order bit in the Flags octet. It indicates that the multicast router is a PIM router.
 - Q and P may be set simultaneously.

For BGP processing purposes, only the RD, Ethernet Tag ID, Originator Router Length and Address, and Multicast Router Length and Address are considered part of the route key. The Secondary Multicast Router Addresses and the rest of the fields are not part of the route key.

4.2 Selective Multicast Ethernet Tag Route for PIM Proxy

This document extends the SMET route defined in [EVPN-IGMP-MLD-PROXY] as shown in Figure 6.

RD (8 octets)
Ethernet Tag ID (4 octets)
Multicast Source Length (1 octet)
Multicast Source Address (variable)
Multicast Group Length (1 octet)
Multicast Group Address (Variable)
Originator Router Length (1 octet)
Originator Router Address (variable)
Flags (1 octets) (optional)
Upstream Router Length (1B)(optional)
Upstream Router Addr (variable)(opt)

Flags:

0	1	2	3	4	5	6	7
+	-	+	-	+	-	+	-
				P	IE	v3	v2
+	-	+	-	+	-	+	-

Figure 6 Selective Multicast Ethernet Tag Route and Flags

As in the case of the MRD route, this route type is OPTIONAL.

This route will be used as per [EVPN-IGMP-MLD-PROXY], with the following extra and optional fields:

- o Upstream Router Length and Address will contain the same information as received in a PIM Join/Prune message on a local AC. There is only one Upstream Router Address per route.
- o Flags: This field encodes Flags that are now relevant to IGMP and PIM. The following new Flag is defined:
 - Flag P: Indicates the SMET route is generated by a received PIM

Join on a local AC. When P=1, the Upstream Router Length and Address fields are present in the route. Otherwise the two fields will not be present.

Compared to [EVPN-IGMP-MLD-PROXY] there is no change in terms of fields considered part of the route key for BGP processing. The Upstream Router Length and Address are not considered part of the route key.

4.3 PIM RPT-Prune Route

The RPT-Prune route is analogous to the SMET route but for PIM RPT-Prune messages. The SMET routes cannot be used to convey RPT-Prune messages because they are always triggered by IGMP or PIM Join messages. A PIM RPT-Prune message is used to Prune a specific (S,G) from the RP Tree by downstream routers. An RPT-Prune message is typically seen prior to an RPT-Join message for the (S,G), hence it requires its own BGP route type (since the SMET route is always advertised based on the received Join messages).

	RD (8 octets)	
+	-----	+
	Ethernet Tag ID (4 octets)	
+	-----	+
	Multicast Source Length (1 octet)	
+	-----	+
	Multicast Source Address (variable)	
+	-----	+
	Multicast Group Length (1 octet)	
+	-----	+
	Multicast Group Address (Variable)	
+	-----	+
	Originator Router Length (1 octet)	
+	-----	+
	Originator Router Address (variable)	
+	-----	+
	Upstream Router Length (1B)	
+	-----	+
	Upstream Router Addr (variable)	
+	-----	+

Figure 7 PIM RPT-Prune Route

Fields are defined in the same way as for the SMET route.

4.4 IGMP/PIM Join Synch Route for PIM Proxy

This document renames the IGMP Join Synch Route defined in [EVPN-IGMP-MLD-PROXY] as IGMP/PIM Join Synch Route and extends it with new fields and Flags as shown in Figure 8:

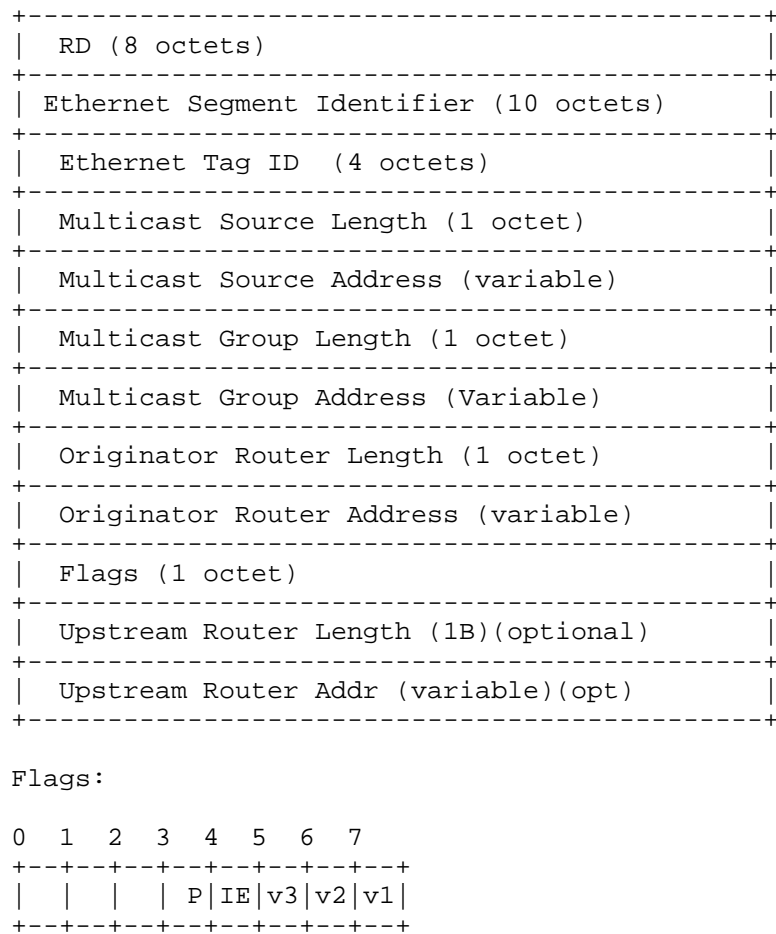


Figure 8 IGMP/PIM Join Synch Route and Flags

This route will be used as per [EVPN-IGMP-MLD-PROXY], with the following extra and optional fields:

- o Upstream Router Length and Address will contain the same information as received in a PIM Join/Prune message on a local AC. There is only one Upstream Router Address per route.
- o Flags: This field encodes Flags that are now relevant to IGMP and PIM. The following new Flag is defined:
 - Flag P: Indicates the Join Synch route is generated by a received PIM Join on a local AC. When P=1, the Upstream Router Length and Address fields are present in the route. Otherwise the two fields will not be present.

Compared to [EVPN-IGMP-MLD-PROXY] there is no change in terms of fields considered part of the route key for BGP processing. The Upstream Router Length and Address are not considered part of the route key.

4.5 IGMP/PIM RPT-Prune Synch Route for PIM Proxy

This new route is used to Synch RPT-Prune states among the PEs in the Ethernet Segment.

+-----+		
	RD (8 octets)	
+-----+		
	Ethernet Segment Identifier (10 octets)	
+-----+		
	Ethernet Tag ID (4 octets)	
+-----+		
	Multicast Source Length (1 octet)	
+-----+		
	Multicast Source Address (variable)	
+-----+		
	Multicast Group Length (1 octet)	
+-----+		
	Multicast Group Address (Variable)	
+-----+		
	Originator Router Length (1 octet)	
+-----+		
	Originator Router Address (variable)	
+-----+		
	Upstream Router Length (1B)(optional)	
+-----+		
	Upstream Router Addr (variable)(opt)	
+-----+		

Figure 9 IGMP/PIM RPT-Prune Synch Route

The RD, Ethernet Segment Identifier and other fields are defined as for the IGMP/PIM Join Synch Route. In addition, the Upstream Router Length and Address will contain the same information as received in a PIM RPT-Prune message on a local AC. The Upstream Router points at the RP for the source and group and there is only one Upstream Router Address per route.

The route key for BGP processing is defined as per the IGMP/PIM Join Synch route.

5. Conclusions

This document extends the IGMP Proxy concept of [EVPN-IGMP-MLD-PROXY] to PIM, so that EVPN can also be used to minimize the flooding of PIM control messages and optimize the delivery of IP multicast traffic in EVPN Broadcast Domains that connect PIM routers.

This specification describes procedures to Discover new PIM routers in the BD, as well as propagate PIM Join/Prune messages using EVPN SMET routes and other optimizations.

6. Conventions used in this document

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].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

7. Security Considerations

This section will be added in future versions.

8. IANA Considerations

This document requests IANA to allocate a new EVPN route type in the corresponding registry:

- + Type TBD - Multicast Router Discovery (MRD) Route

- + Type TBD - PIM RPT-Prune Route
- + Type TBD - PIM RPT-Prune Join Synch Route

In addition, the following route defined in [EVPN-IGMP-MLD-PROXY] should be renamed as follows:

- + Type 7 - IGMP/PIM Join Synch Route

9. Terminology

- o EVI: EVPN Instance.
- o EVPN Broadcast Domain: it refers to an EVI in case of VLAN-based and VLAN-bundle interfaces. It refers to a Bridge Domain identified by an Ethernet-Tag (in the control plane) in case of VLAN-Aware Bundle interfaces.
- o AC: Attachment Circuit.
- o PIM-DM: Protocol Independent Multicast - Dense Mode.
- o PIM-SM: Protocol Independent Multicast - Sparse Mode.
- o PIM-SSM: Protocol Independent Multicast - Source Specific Mode.
- o S: IP address of the multicast source.
- o G: IP address of the multicast group.
- o N: Upstream neighbor field in a Join/Prune/Graft message.
- o PIM J/P: PIM Join/Prune messages.
- o RP: PIM Rendezvous Point.
- o MRD route: Multicast Router Discovery.
- o PIM Nbr: PIM Neighbor.

10. References

10.1 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>>.

[RFC4601] Fenner, B., Handley, M., Holbrook, H., and I. Kouvelas, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", RFC 4601, DOI 10.17487/RFC4601, August 2006, <<http://www.rfc-editor.org/info/rfc4601>>.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <<http://www.rfc-editor.org/info/rfc2236>>.

[RFC8220] Dornon, O. et al, "Protocol Independent Multicast (PIM) over Virtual Private LAN Service (VPLS)", RFC 8220, DOI 10.17487/RFC8220, September 2017, <<http://www.rfc-editor.org/info/rfc8220>>.

[EVPN-IGMP-MLD-PROXY] Sajassi, A. et al, "IGMP and MLD Proxy for EVPN", March 2017, work-in-progress, draft-ietf-bess-evpn-igmp-mld-proxy-00.

10.2 Informative References

[EVPN-PROXY-ARP-ND] Rabadan, J. et al, "Operational Aspects of Proxy-ARP/ND in EVPN Networks", October 2017, work-in-progress, draft-ietf-bess-evpn-proxy-arp-nd-03.

11. Acknowledgments

12. Contributors

13. 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

Zhaohui Zhang
Juniper Networks
EMail: zzhang@juniper.net

Ali Sajassi
Cisco
Email: sajassi@cisco.com

BESS Workgroup
Internet Draft
Intended status: Informational

J. Rabadan, Ed.
S. Sathappan
K. Nagaraj
Nokia

J. Bueno
J. Crespo
Telefonica

Expires: January 4, 2018

July 3, 2017

Loop Protection in EVPN networks
draft-snr-bess-evpn-loop-protect-00

Abstract

Ethernet Virtual Private Networks (EVPN) is becoming the de-facto standard-based control plane solution for Data Center and layer-2 Service Provider applications. The risk of loops caused by backdoor paths accidentally created within the same broadcast domain, is a general common concern, especially among Service Providers in large Layer-2 networks. While other layer-2 Ethernet technologies use Spanning Tree based Protocols (xSTP) to provide a network-wide loop protection, EVPN has the right tools to detect and protect the network against loops in an efficient and effective way. This document describes a mechanism to provide global loop protection in 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), 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 4, 2018.

Copyright 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	3
2. Loop Protection Requirements in EVPN networks	5
3. Loop Protection Solution for EVPN networks	6
3.1 The RFC7432 EVPN MAC Duplication Mechanism and Loop Protection	6
3.2 Loop Protection Solution	7
3.3 The Black-Hole MAC concept for Loop Protection	10
4. Conclusions	11
6. Conventions used in this document	11
7. Security Considerations	11
8. IANA Considerations	12
9. Terminology	12
9. References	12
9.1 Normative References	12
9.2 Informative References	12
10. Acknowledgments	12
11. Contributors	12
17. Authors' Addresses	12

1. Introduction

Ethernet Virtual Private Networks (EVPN) is becoming the de-facto standard-based control plane solution for Data Center and layer-2 Service Provider applications. The risk of loops caused by backdoor paths accidentally created within the same broadcast domain, is a general common concern, especially among Service Providers in large Layer-2 networks. While other layer-2 Ethernet technologies use Spanning Tree based Protocols (xSTP) to provide global loop protection, EVPN has the right tools to detect and protect the network against loops in an efficient and effective way. However, [RFC7432] only addresses the MAC duplication detection and protection at the control plane, and not all the possible loop scenarios.

In this document, backdoor path is defined as a layer-2 connection between two Attachment Circuits (ACs) that, along with the layer-2 connectivity in the EVI, creates a loop. We differentiate between a local and a global loop. A local loop is created by a backdoor path within the same physical port or between two Attachment Circuits (ACs) of the same MAC-VRF. A global loop is created by a backdoor path between two ACs of the same EVI but different PEs. This document addresses global loop protection, since it requires interoperability between PEs. Local loop protection is implementation specific and it is not addressed in this specification.

Figure 1 shows a typical example of a backdoor path that may be created by mistake in a Service Provider network that uses EVPN to provide E-LAN services. A backdoor path is accidentally created between AC4 and AC5.

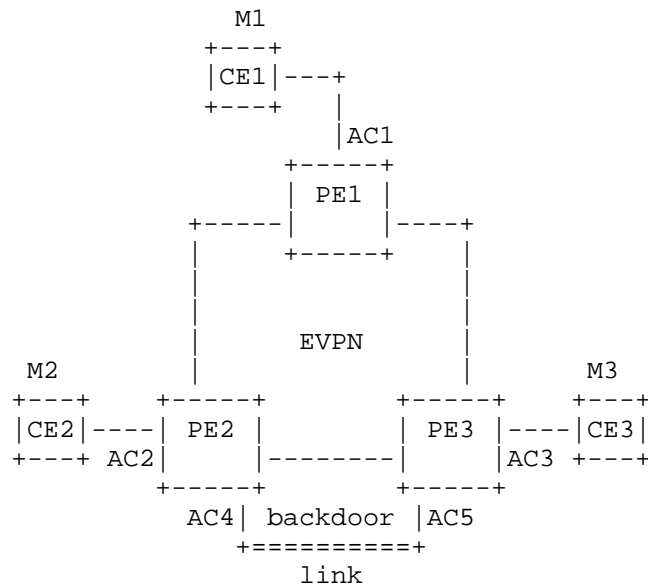


Figure 1 Backdoor link example in Service Provider EVPN networks

When, for instance, CE1 (in Figure 1) sends Broadcast, Unknown unicast or Multicast (BUM) traffic, the frames will be flooded to PE2 and PE3, looped to each other through the backdoor link and flooded back again in the EVPN network, creating an endless loop.

Figure 2 illustrates another example of backdoor path between NVEs in two remote Data Centers.

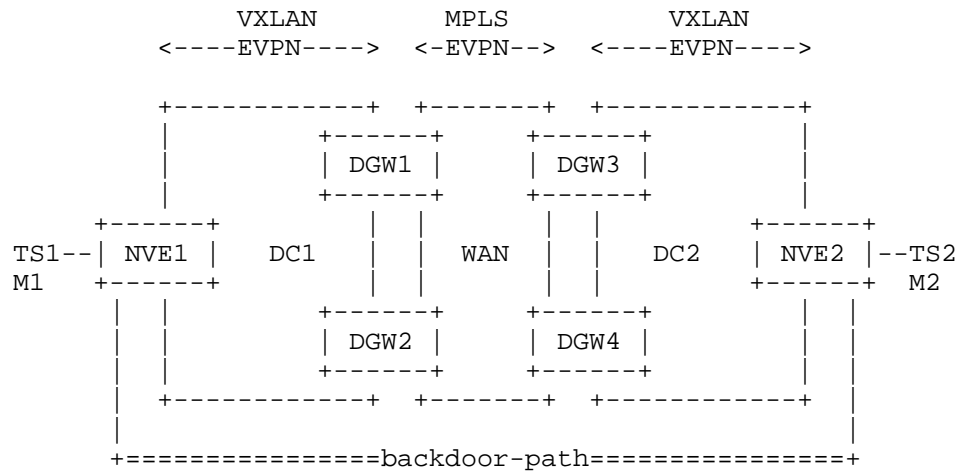


Figure 2 Backdoor path example in DCI EVPN networks

In Figure 2, a backdoor path is accidentally created between NVE1 and NVE2 in two remote Data Centers. BUM traffic generated by TS1 or TS2 will cause a layer-2 loop across DC1 and DC2.

2. Loop Protection Requirements in EVPN networks

The following requirements have been identified for loop protection in EVPN networks:

- 1- The EVPN PEs in a network MUST provide an automatic mechanism for detecting and resolving a loop within the same broadcast domain. In this document 'resolving a loop' refers to an automatic action executed by a PE or group of PEs that stops a frame from being endlessly forwarded back and forth between two PEs.
- 2- The Loop Protection mechanism MUST be compatible with all the procedures described in EVPN [RFC7432], in particular, it must not interfere with regular EVPN Multi-homing, MAC Mobility and MAC Protection procedures.
- 3- The Loop Resolution action SHOULD discard the looped flows without bringing down the Attachment Circuits (ACs) involved in the created loop. For example, when CE2 sends a broadcast frame (in Figure 1) the Loop Resolution action should discard the looped frames that are forwarded between PE2 and PE3 instead of bringing down any AC in the backdoor path.

- 4- The Loop Resolution action MAY bring down the ACs that are involved in the loop for a given flow instead of only discarding the identified looped frames. This action may impact some unicast flows that are not looped in the EVI, but provides an immediate solution to the loop situation. For example, when a loop (for BUM frames sent from CE1) is detected in PE3, the router may bring down the AC corresponding to the backdoor link.
- 5- A PE detecting a loop SHOULD log an event, warning the operator of the existence of a loop.
- 6- The operator SHOULD be able to configure whether the Loop Resolution action is manually or automatically cleared from a given PE, before the Loop Protection mechanism is restarted.
- 7- The solution MUST be compatible with other implementation-specific procedures that protect the PE against local loops.

3. Loop Protection Solution for EVPN networks

This document re-uses and enhances the MAC duplication solution specified in EVPN [RFC7432]. Section 3.1 clarifies this baseline EVPN MAC duplication mechanism and describes the required enhancements so that the EVPN network can protect the EVI user against loops.

3.1 The RFC7432 EVPN MAC Duplication Mechanism and Loop Protection

EVPN [RFC7432] describes a MAC duplication issue and how this anomaly is resolved. In this document, the terms VLAN and broadcast domain are used interchangeably. A VLAN is equivalent to an EVI in case of VLAN-based or VLAN Bundle services, and to a broadcast domain in case of VLAN-Aware Bundle services.

As per RFC7432, if a duplicate MAC situation exists in two or more hosts that are part of two different Ethernet Segments within the same VLAN, the traffic originating from these hosts would trigger continuous MAC moves among the PEs attached to them. If no action was made, the sequence number (in the MAC Mobility extended community attribute) would be incremented by the PEs to infinity.

In order to remedy such a situation, a PE that detects a MAC mobility event via local learning:

- o Starts an M-second timer. M is configurable, with a default value of M = 180.
- o If it detects N MAC moves before the timer expires, it concludes

that a duplicate-MAC situation has occurred and adds the MAC to a duplicate-MAC list. N is configurable with a default value of N = 5.

- o The PE MUST alert the operator and stop sending and processing any BGP MAC/IP Advertisement routes for that MAC address until a corrective action is taken by the operator.
- o While a MAC address is on the duplicate-MAC list for the VLAN, the other PEs in the EVI will forward the traffic for the duplicate-MAC address to one of the PEs that advertised it.

In the example of Figure 1, when CE1 sends BUM traffic to the EVI, the EVPN MAC Duplication Mechanism prevents an endless MAC/IP route exchange for M1 between PE1, PE2 and PE3. For instance, when MAC M1 moves N times in PE2 within the M-second timer period, PE2 will add M1 to the duplicate-MAC list for the broadcast domain and will stop advertising a MAC/IP route for M1. While this helps the control plane settle, Broadcast frames being sent by CE1 are still endlessly looped within the broadcast domain through the backdoor link. This may cause unpredictable issues in the CEs connected to the affected EVI.

3.2 Loop Protection Solution

This document enhances the EVPN MAC Duplication Mechanism by extending it with an optional Loop-protection action that is applied on the duplicate-MAC addresses. This additional mechanism resolves loops created by accidental backdoor links and SHOULD be enabled in all the PEs in the EVI.

Figure 3 outlines the Loop Protection solution when a backdoor link exists between two PEs (PE2 and PE3) in the same EVI and broadcast domain. The following assumptions are made:

- o Loop Protection (this document) is enabled on (at least) PE3.
- o PEs in the EVI are configured with window M-timer = M seconds and number of moves = N.
- o PEs are also configured with a R-timer (retry-timer) = R seconds. This timer is explained later.
- o In this document, a MAC-move refers to a relearn event in the same MAC-VRF, where the same MAC is first learned on an AC and later learned from BGP EVPN. Vice versa is also considered a MAC-move. Relearn events between two ACs in the same PE (i.e. local loops) or between two different EVPN endpoints are not considered. To protect the network against local loops, this procedure should be combined with local loop protection mechanisms.

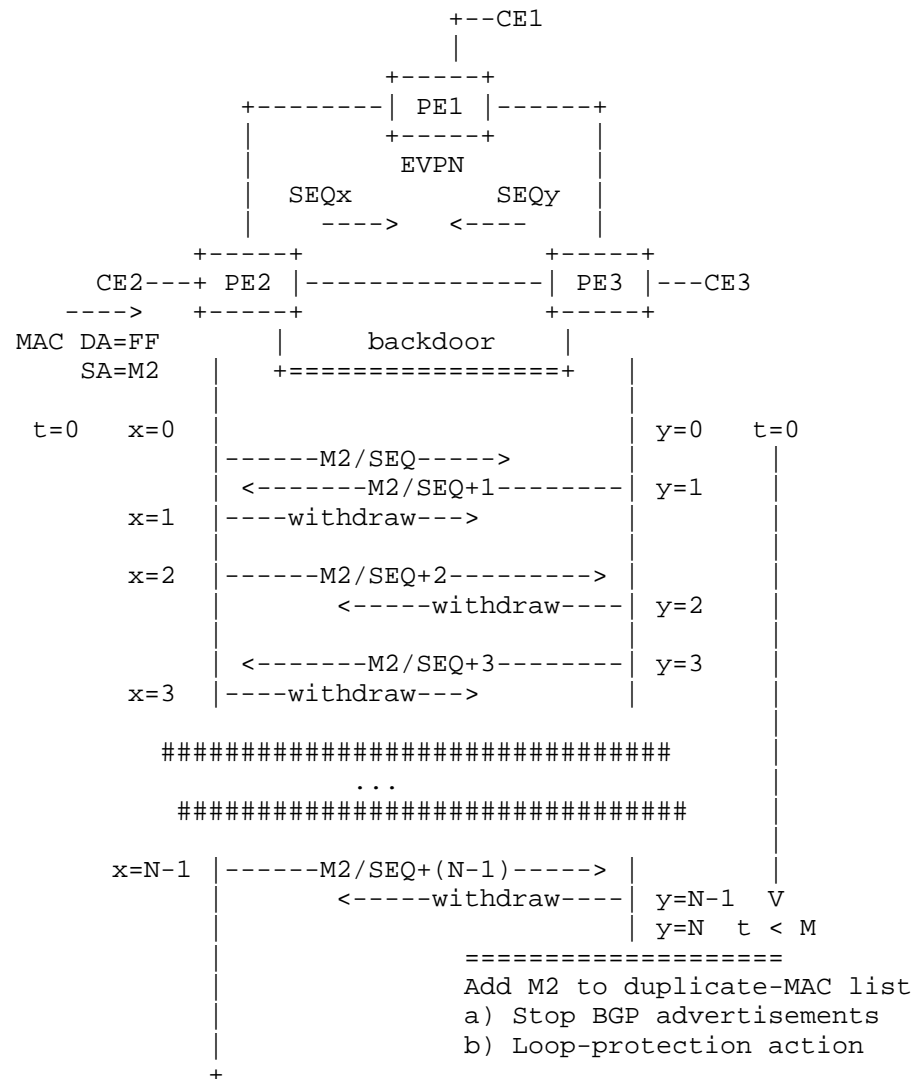


Figure 3 MAC Duplication and Loop Protection process

In the example of Figure 3, we assume CE2 sends a broadcast frame with MAC SA (Source Address) M2. We also assume PE3 learns M2 via BGP first, and via data path later. Although that is unlikely since data path learning is normally faster than BGP-based learning, it helps understand and generalize the procedure. The procedure will work as long as the PE detects N MAC-moves within M seconds for a given MAC.

The following process takes place:

- T0 - PE2 receives the frame, learns M2 (if not learned before) and initializes counter x and timer t. Counter x stores the number of MAC moves, while t stores the delta time since the first MAC move for M2 occurred. PE2 advertises M2 with the currently stored Sequence Number (SEQ). Also, PE2 does a MAC DA (Destination Address) lookup and, since the MAC DA is a broadcast address, it floods the frame to PE1, PE3 and the AC on the backdoor link. This causes a loop between PE2 and PE3.
- T1 - PE3 receives the BGP update and learns M2. Counter y and timer t are initialized. Counter y stores the number of moves for M2 and t stores the delta time since y was initialized. PE3 now advertises M2 with SEQ+1. M2/SEQ+1 route arrives at PE2 and it is installed in the MAC-VRF. The advertisement makes PE2 withdraw the MAC/IP route for M2 and increment x. Immediately after, PE2 receives the frame again through the backdoor link, relearns M2 locally, increments x and advertises M2 with SEQ+2.
- T2 - M2/SEQ+2 route arrives at PE3 and it is installed in the MAC-VRF. The advertisement makes PE3 withdraw the MAC/IP route for M2 and increment y. PE3 receives the frame again through the backdoor link, relearns M2 locally, increments y and advertises M2 with SEQ+3. PE2 receives the route, relearns M2 and increments x. PE2 also withdraws the route for M2. Immediately after, PE2 receives the frame through the backdoor link and repeats the process (updates y and withdraws the route).

Since the frame (with MAC SA=M2) keeps being learned locally on the backdoor link ACs on PE2 and PE3, the above process is repeated until y reaches number of moves = N.

- Tr - When y=N, PE3 compares t against the configured window M, and in case $t < M$, PE3 adds M2 to the duplicate-MAC list for the broadcast domain. Declaring M2 as duplicate triggers three actions:
- PE3 stops advertising M2 and logs a duplicate event.
 - PE3 initializes a retry-timer.
 - Since Loop Protection is enabled in PE3, PE3 executes the Loop Protection action, which we will refer to as "Black-holing" M2. When P3 programs M2 as a Black-Hole MAC in the MAC-VRF, M2 is no longer associated to the backdoor AC, but to a Black-Hole destination.
- Ts - At this point and while M2 is in Black-Hole state:
- If a new frame is received at PE3 (from the EVPN core or the backdoor AC) with MAC SA = M2, PE3 will identify M2 as Black-

- Hole and discard the frame, ending the loop.
- b) Optionally, instead of simply discarding the frame with MAC SA = M2, PE3 MAY bring down the AC on which the offending frame is seen last. In this example, PE3 would bring down the backdoor AC, ending in that way the loop not only for frames from CE2, but for any traffic.
- c) Optionally, any frame that arrives at PE3 with MAC DA = M2 SHOULD be discarded too.

Tt - When the retry-timer for M2 reaches R seconds, PE3 will flush M2 from the MAC-VRF and the process will be restarted.

Section 3.3 provides more details about the Black-Hole MAC in the context of this document.

3.3 The Black-Hole MAC concept for Loop Protection

As discussed in section 3.2, this document enhances the EVPN MAC Duplication mechanism by converting the detected duplicate-MAC addresses into Black-Hole MAC addresses and ending the forwarding plane loop. A Black-hole MAC is modeled as a special MAC-VRF record that has the following characteristics:

- a) A Black-Hole MAC M is automatically installed in the MAC-VRF when M is detected as duplicate-MAC address.
- b) When M is installed as Black-Hole MAC, for any ingress frame and irrespective of the frame arriving at an AC or network port:
 - i) If MAC SA = M the ingress frame MUST be discarded, without any further action.
 - ii) If MAC DA = M the ingress frame SHOULD be discarded, without any further action.
- c) Optionally, any ingress frame with MAC SA = M arriving at an access AC, MAY trigger the PE to bring down the AC. Note that this approach cuts off the backdoor path that created the loop, preventing traffic from other MAC addresses from being forwarded, even if they are not identified as duplicate-MAC addresses yet.
- d) A Black-Hole MAC M can be flushed from the MAC-VRF if any of the following events occur:
 - o Retry-timer R for duplicate-MAC M expires. R is initialized when M is detected as duplicate-MAC. Its value is configurable and SHOULD be at least three times the EVPN MAC Duplication M-timer window. According to EVPN [RFC7432], M's default value is 180 seconds, hence R's default value SHOULD be 540 seconds.
 - o The operator manually flushes a Black-Hole MAC M. This should be done only if the conditions under which M was identified as

- duplicate have been cleared.
 - o The remote PE withdraws the MAC/IP route for M and there are no other remote MAC/IP routes for M.
 - o The remote PE sends a MAC/IP route update for M with the sticky-bit set (in the MAC Mobility extended community).
- e) When a Black-Hole MAC is flushed from the MAC-VRF, the actions described in (b) and (c) are naturally reverted and the EVPN MAC Duplication and Loop Protection process will be restarted.

4. Conclusions

As EVPN is deployed in large layer-2 networks to deliver E-LAN or E-Tree services, it is important that the technology provides a solid protection against loops accidentally created by backdoor links. These backdoors can exist between CEs that can be connected anywhere in the EVI.

The EVPN [RFC7432] MAC Duplication Detection mechanism solves a situation where the same MAC has been accidentally configured on two or more hosts connected to different EVPN Ethernet Segments in the same broadcast domain. However, that mechanism does not provide a solution to resolve loops in those cases where the MAC duplication is caused by backdoor links between CEs.

This document leverages and extends the EVPN [RFC7432] MAC Duplication Detection mechanism by providing additional Loop Protection actions for the duplicate-MAC addresses.

6. Conventions used in this document

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].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

7. Security Considerations

This section will be added in future versions.

8. IANA Considerations

This document does not require new codepoints.

9. Terminology

EVI: EVPN Instance.
E-LAN: MEF-based Ethernet Local Area Network service.
E-Tree: MEF-based Ethernet Tree service.
BUM: Broadcast, Unknown unicast and Multicast traffic.
AC: Attachment Circuit.
MAC-VRF: MAC Virtual Routing and Forwarding instance. Instantiation of an EVI in a PE.
xSTP: Any Spanning Tree based Protocol, e.g. STP, RSTP, MSTP.

9. References

9.1 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>>.

9.2 Informative References

10. Acknowledgments

11. Contributors

17. Authors' Addresses

Jorge Rabadan
Nokia
777 E. Middlefield Road
Mountain View, CA 94043 USA
Email: jorge.rabadan@nokia.com

Senthil Sathappan
Nokia
Email: senthil.sathappan@nokia.com

Kiran Nagaraj
Nokia
Email: kiran.nagaraj@nokia.com

Julio Bueno
Telefonica
Email: julio.buenohernandez@telefonica.com

Jose Manuel Crespo
Telefonica
Email: josemanuel.crespogarcia@telefonica.com

BESS Workgroup
Internet Draft
Intended status: Informational

J. Rabadan, Ed.
S. Sathappan
K. Nagaraj
Nokia

J. Bueno
J. Crespo
Telefonica

Expires: February 6, 2020

August 5, 2019

Loop Protection in EVPN networks
draft-snr-bess-evpn-loop-protect-04

Abstract

Ethernet Virtual Private Networks (EVPN) is becoming the de-facto standard-based control plane solution for Data Center and layer-2 Service Provider applications. The risk of loops caused by backdoor paths accidentally created within the same broadcast domain, is a general common concern, especially among Service Providers in large Layer-2 networks. While other layer-2 Ethernet technologies use Spanning Tree based Protocols (xSTP) to provide a network-wide loop protection, EVPN has the right tools to detect and protect the network against loops in an efficient and effective way. This document describes a mechanism to provide global loop protection in 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), 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 10, 2019.

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. Terminology	5
3. Loop Protection Requirements in EVPN networks	5
4. Loop Protection Solution for EVPN networks	6
4.1 The RFC7432 EVPN MAC Duplication Mechanism and Loop Protection	6
4.2 Loop Protection Solution	7
4.3 The Black-Hole MAC concept for Loop Protection	11
5. Conclusions	12
6. Conventions used in this document	12
7. Security Considerations	12
8. IANA Considerations	12
9. References	13
9.1 Normative References	13
9.2 Informative References	13
10. Acknowledgments	13
11. Contributors	13
17. Authors' Addresses	13

1. Introduction

Ethernet Virtual Private Networks (EVPN) is becoming the de-facto standard-based control plane solution for Data Center and layer-2 Service Provider applications. The risk of loops caused by backdoor paths accidentally created within the same broadcast domain, is a general common concern, especially among Service Providers in large Layer-2 networks. While other layer-2 Ethernet technologies use Spanning Tree based Protocols (xSTP) to provide global loop protection, EVPN has the right tools to detect and protect the network against loops in an efficient and effective way. However, [RFC7432] only addresses the MAC duplication detection and protection at the control plane, and not all the possible loop scenarios.

In this document, backdoor path is defined as a layer-2 connection between two Attachment Circuits (ACs) that, along with the layer-2 connectivity in the EVI, creates a loop. We differentiate between a local and a global loop. A local loop is created by a backdoor path within the same physical port or between two Attachment Circuits (ACs) of the same MAC-VRF. A global loop is created by a backdoor path between two ACs of the same EVI but different PEs. This document addresses global loop protection, since it requires interoperability between PEs. Local loop protection is implementation specific and it is not addressed in this specification.

Figure 1 shows a typical example of a backdoor path that may be created by mistake in a Service Provider network that uses EVPN to provide E-LAN services. A backdoor path is accidentally created between AC4 and AC5.

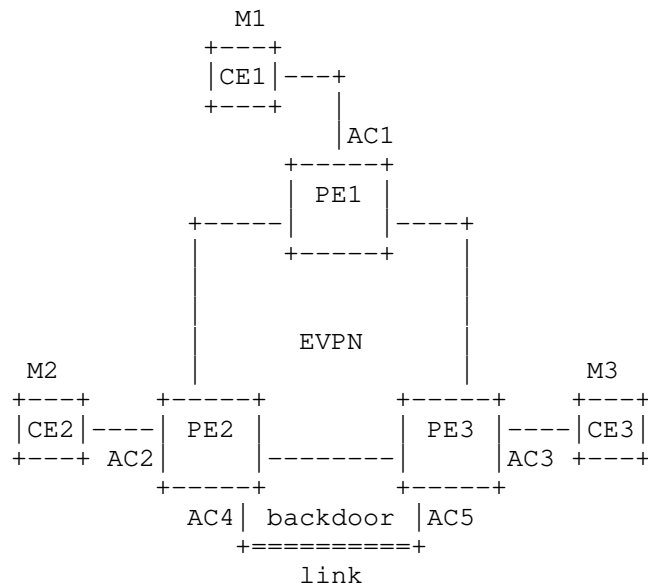


Figure 1 Backdoor link example in Service Provider EVPN networks

When, for instance, CE1 (in Figure 1) sends Broadcast, Unknown unicast or Multicast (BUM) traffic, the frames will be flooded to PE2 and PE3, looped to each other through the backdoor link and flooded back again in the EVPN network, creating an endless loop.

Figure 2 illustrates another example of backdoor path between NVEs in two remote Data Centers.

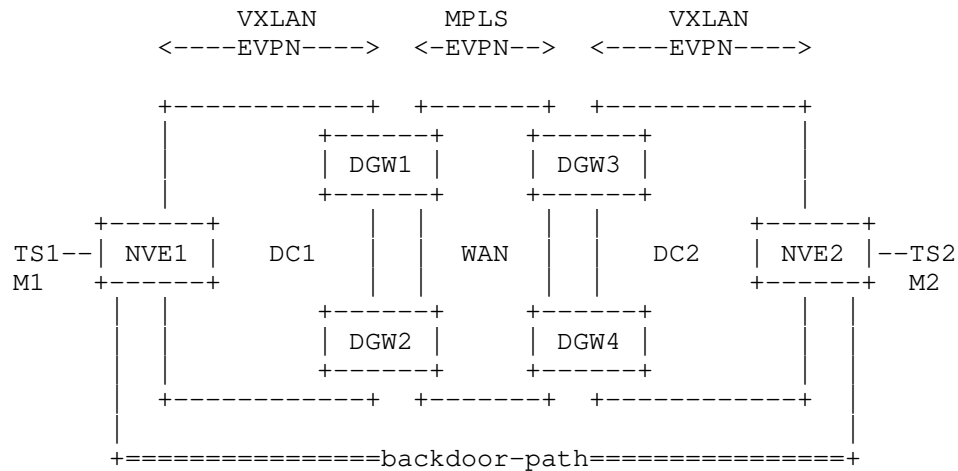


Figure 2 Backdoor path example in DCI EVPN networks

In Figure 2, a backdoor path is accidentally created between NVE1 and NVE2 in two remote Data Centers. BUM traffic generated by TS1 or TS2 will cause a layer-2 loop across DC1 and DC2.

2. Terminology

EVI: EVPN Instance.
 E-LAN: MEF-based Ethernet Local Area Network service.
 E-Tree: MEF-based Ethernet Tree service.
 BUM: Broadcast, Unknown unicast and Multicast traffic.
 AC: Attachment Circuit.
 MAC-VRF: MAC Virtual Routing and Forwarding instance. Instantiation of an EVI in a PE.
 xSTP: Any Spanning Tree based Protocol, e.g. STP, RSTP, MSTP.

3. Loop Protection Requirements in EVPN networks

The following requirements have been identified for loop protection in EVPN networks:

- 1- The EVPN PEs in a network MUST provide an automatic mechanism for detecting and resolving a loop within the same broadcast domain. In this document 'resolving a loop' refers to an automatic action executed by a PE or group of PEs that stops a frame from being endlessly forwarded back and forth between two PEs.

- 2- The Loop Protection mechanism **MUST** be compatible with all the procedures described in EVPN [RFC7432], in particular, it must not interfere with regular EVPN Multi-homing, MAC Mobility and MAC Protection procedures.
- 3- The Loop Resolution action **SHOULD** discard the looped flows without bringing down the Attachment Circuits (ACs) involved in the created loop. For example, when CE2 sends a broadcast frame (in Figure 1) the Loop Resolution action should discard the looped frames that are forwarded between PE2 and PE3 instead of bringing down any AC in the backdoor path.
- 4- The Loop Resolution action **MAY** bring down the ACs that are involved in the loop for a given flow instead of only discarding the identified looped frames. This action may impact some unicast flows that are not looped in the EVI, but provides an immediate solution to the loop situation. For example, when a loop (for BUM frames sent from CE1) is detected in PE3, the router may bring down the AC corresponding to the backdoor link.
- 5- A PE detecting a loop **SHOULD** log an event, warning the operator of the existence of a loop.
- 6- The operator **SHOULD** be able to configure whether the Loop Resolution action is manually or automatically cleared from a given PE, before the Loop Protection mechanism is restarted.
- 7- The solution **MUST** be compatible with other implementation-specific procedures that protect the PE against local loops.

4. Loop Protection Solution for EVPN networks

This document re-uses and enhances the MAC duplication solution specified in EVPN [RFC7432]. Section 4.1 clarifies this baseline EVPN MAC duplication mechanism and describes the required enhancements so that the EVPN network can protect the EVI user against loops.

4.1 The RFC7432 EVPN MAC Duplication Mechanism and Loop Protection

EVPN [RFC7432] describes a MAC duplication issue and how this anomaly is resolved. In this document, the terms VLAN and broadcast domain are used interchangeably. A VLAN is equivalent to an EVI in case of VLAN-based or VLAN Bundle services, and to a broadcast domain in case of VLAN-Aware Bundle services.

As per RFC7432, if a duplicate MAC situation exists in two or more hosts that are part of two different Ethernet Segments within the

same VLAN, the traffic originating from these hosts would trigger continuous MAC moves among the PEs attached to them. If no action was made, the sequence number (in the MAC Mobility extended community attribute) would be incremented by the PEs to infinity.

In order to remedy such a situation, a PE that detects a MAC mobility event via local learning:

- o Starts an M-second timer. M is configurable, with a default value of M = 180.
- o If it detects N MAC moves before the timer expires, it concludes that a duplicate-MAC situation has occurred and adds the MAC to a duplicate-MAC list. N is configurable with a default value of N = 5.
- o The PE MUST alert the operator and stop sending and processing any BGP MAC/IP Advertisement routes for that MAC address until a corrective action is taken by the operator.
- o While a MAC address is on the duplicate-MAC list for the VLAN, the other PEs in the EVI will forward the traffic for the duplicate-MAC address to one of the PEs that advertised it.

In the example of Figure 1, when CE1 sends BUM traffic to the EVI, the EVPN MAC Duplication Mechanism prevents an endless MAC/IP route exchange for M1 between PE1, PE2 and PE3. For instance, when MAC M1 moves N times in PE2 within the M-second timer period, PE2 will add M1 to the duplicate-MAC list for the broadcast domain and will stop advertising a MAC/IP route for M1. While this helps the control plane settle, Broadcast frames being sent by CE1 are still endlessly looped within the broadcast domain through the backdoor link. This may cause unpredictable issues in the CEs connected to the affected EVI.

4.2 Loop Protection Solution

This document enhances the EVPN MAC Duplication Mechanism by extending it with an optional Loop-protection action that is applied on the duplicate-MAC addresses. This additional mechanism resolves loops created by accidental backdoor links and SHOULD be enabled in all the PEs in the EVI.

Figure 3 outlines the Loop Protection solution when a backdoor link exists between two PEs (PE2 and PE3) in the same EVI and broadcast domain. The following assumptions are made:

- o Loop Protection (this document) is enabled on (at least) PE3.

- o PEs in the EVI are configured with window M-timer = M seconds and number of moves = N.
- o PEs are also configured with a R-timer (retry-timer) = R seconds. This timer is explained later.
- o In this document, a MAC-move refers to a relearn event in the same MAC-VRF, where the same MAC is first learned on an AC and later learned from BGP EVPN. Vice versa is also considered a MAC-move. Relearn events between two ACs in the same PE (i.e. local loops) or between two different EVPN endpoints are not considered. To protect the network against local loops, this procedure should be combined with local loop protection mechanisms.

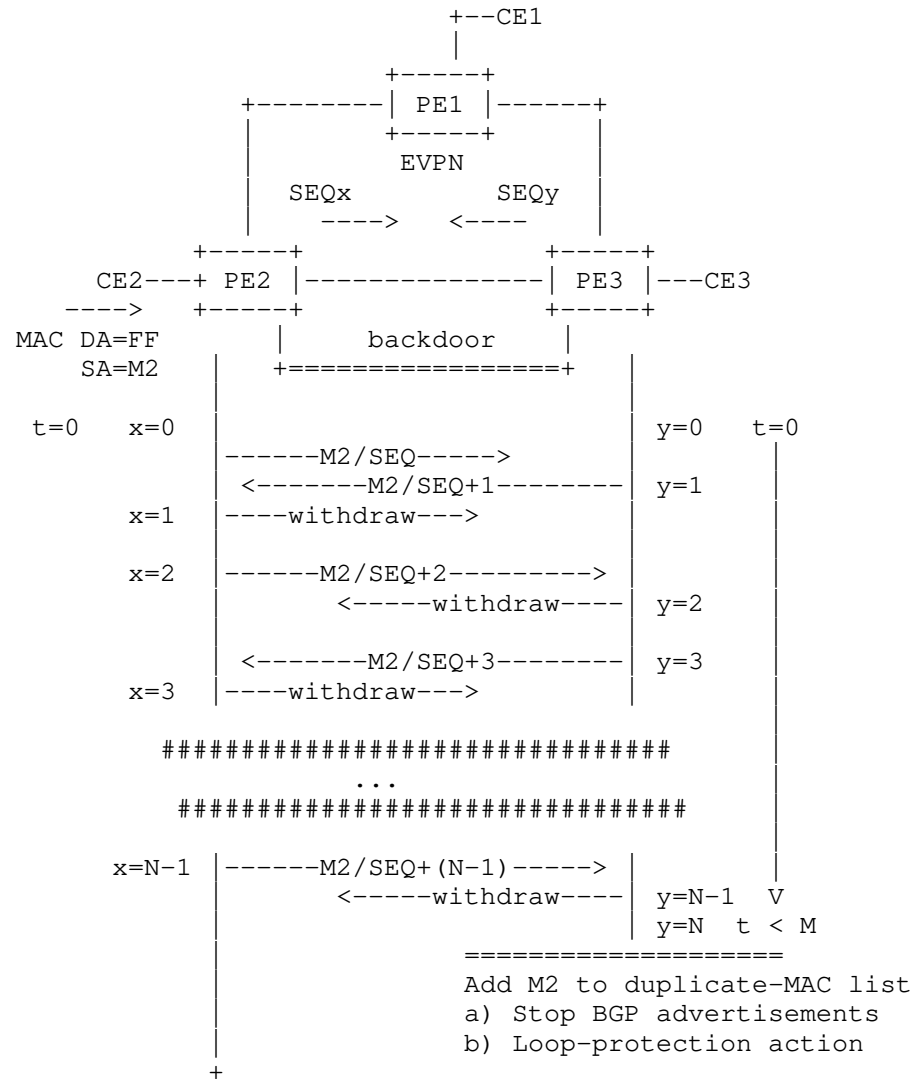


Figure 3 MAC Duplication and Loop Protection process

In the example of Figure 3, we assume CE2 sends a broadcast frame with MAC SA (Source Address) M2. We also assume PE3 learns M2 via BGP first, and via data path later. Although that is unlikely since data path learning is normally faster than BGP-based learning, it helps understand and generalize the procedure. The procedure will work as long as the PE detects N MAC-moves within M seconds for a given MAC.

The following process takes place:

- T0 - PE2 receives the frame, learns M2 (if not learned before) and initializes counter x and timer t. Counter x stores the number of MAC moves, while t stores the delta time since the first MAC move for M2 occurred. PE2 advertises M2 with the currently stored Sequence Number (SEQ). Also, PE2 does a MAC DA (Destination Address) lookup and, since the MAC DA is a broadcast address, it floods the frame to PE1, PE3 and the AC on the backdoor link. This causes a loop between PE2 and PE3.
- T1 - PE3 receives the BGP update and learns M2. Counter y and timer t are initialized. Counter y stores the number of moves for M2 and t stores the delta time since y was initialized. PE3 now advertises M2 with SEQ+1. M2/SEQ+1 route arrives at PE2 and it is installed in the MAC-VRF. The advertisement makes PE2 withdraw the MAC/IP route for M2 and increment x. Immediately after, PE2 receives the frame again through the backdoor link, relearns M2 locally, increments x and advertises M2 with SEQ+2.
- T2 - M2/SEQ+2 route arrives at PE3 and it is installed in the MAC-VRF. The advertisement makes PE3 withdraw the MAC/IP route for M2 and increment y. PE3 receives the frame again through the backdoor link, relearns M2 locally, increments y and advertises M2 with SEQ+3. PE2 receives the route, relearns M2 and increments x. PE2 also withdraws the route for M2. Immediately after, PE2 receives the frame through the backdoor link and repeats the process (updates y and withdraws the route).

Since the frame (with MAC SA=M2) keeps being learned locally on the backdoor link ACs on PE2 and PE3, the above process is repeated until y reaches number of moves = N.

- Tr - When y=N, PE3 compares t against the configured window M, and in case $t < M$, PE3 adds M2 to the duplicate-MAC list for the broadcast domain. Declaring M2 as duplicate triggers three actions:
- PE3 stops advertising M2 and logs a duplicate event.
 - PE3 initializes a retry-timer.
 - Since Loop Protection is enabled in PE3, PE3 executes the Loop Protection action, which we will refer to as "Black-holing" M2. When P3 programs M2 as a Black-Hole MAC in the MAC-VRF, M2 is no longer associated to the backdoor AC, but to a Black-Hole destination.
- Ts - At this point and while M2 is in Black-Hole state:
- If a new frame is received at PE3 (from the EVPN core or the backdoor AC) with MAC SA = M2, PE3 will identify M2 as Black-

- Hole and discard the frame, ending the loop.
- b) Optionally, instead of simply discarding the frame with MAC SA = M2, PE3 MAY bring down the AC on which the offending frame is seen last. In this example, PE3 would bring down the backdoor AC, ending in that way the loop not only for frames from CE2, but for any traffic.
 - c) Optionally, any frame that arrives at PE3 with MAC DA = M2 SHOULD be discarded too.

Tt - When the retry-timer for M2 reaches R seconds, PE3 will flush M2 from the MAC-VRF and the process will be restarted.

Section 4.3 provides more details about the Black-Hole MAC in the context of this document.

4.3 The Black-Hole MAC concept for Loop Protection

As discussed in section 4.2, this document enhances the EVPN MAC Duplication mechanism by converting the detected duplicate-MAC addresses into Black-Hole MAC addresses and ending the forwarding plane loop. A Black-hole MAC is modeled as a special MAC-VRF record that has the following characteristics:

- a) A Black-Hole MAC M is automatically installed in the MAC-VRF when M is detected as duplicate-MAC address.
- b) When M is installed as Black-Hole MAC, for any ingress frame and irrespective of the frame arriving at an AC or network port:
 - i) If MAC SA = M the ingress frame MUST be discarded, without any further action.
 - ii) If MAC DA = M the ingress frame SHOULD be discarded, without any further action.
- c) Optionally, any ingress frame with MAC SA = M arriving at an access AC, MAY trigger the PE to bring down the AC. Note that this approach cuts off the backdoor path that created the loop, preventing traffic from other MAC addresses from being forwarded, even if they are not identified as duplicate-MAC addresses yet.
- d) A Black-Hole MAC M can be flushed from the MAC-VRF if any of the following events occur:
 - o Retry-timer R for duplicate-MAC M expires. R is initialized when M is detected as duplicate-MAC. Its value is configurable and SHOULD be at least three times the EVPN MAC Duplication M-timer window. According to EVPN [RFC7432], M's default value is 180 seconds, hence R's default value SHOULD be 540 seconds.
 - o The operator manually flushes a Black-Hole MAC M. This should be done only if the conditions under which M was identified as

- duplicate have been cleared.
 - o The remote PE withdraws the MAC/IP route for M and there are no other remote MAC/IP routes for M.
 - o The remote PE sends a MAC/IP route update for M with the sticky-bit set (in the MAC Mobility extended community).
- e) When a Black-Hole MAC is flushed from the MAC-VRF, the actions described in (b) and (c) are naturally reverted and the EVPN MAC Duplication and Loop Protection process will be restarted.

5. Conclusions

As EVPN is deployed in large layer-2 networks to deliver E-LAN or E-Tree services, it is important that the technology provides a solid protection against loops accidentally created by backdoor links. These backdoors can exist between CEs that can be connected anywhere in the EVI.

The EVPN [RFC7432] MAC Duplication Detection mechanism solves a situation where the same MAC has been accidentally configured on two or more hosts connected to different EVPN Ethernet Segments in the same broadcast domain. However, that mechanism does not provide a solution to resolve loops in those cases where the MAC duplication is caused by backdoor links between CEs.

This document leverages and extends the EVPN [RFC7432] MAC Duplication Detection mechanism by providing additional Loop Protection actions for the duplicate-MAC addresses.

6. Conventions used in this document

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.

7. Security Considerations

This section will be added in future versions.

8. IANA Considerations

This document does not require new codepoints.

9. References

9.1 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>>.

9.2 Informative References

10. Acknowledgments

11. Contributors

17. Authors' Addresses

Jorge Rabadan
Nokia
777 E. Middlefield Road
Mountain View, CA 94043 USA
Email: jorge.rabadan@nokia.com

Senthil Sathappan
Nokia
Email: senthil.sathappan@nokia.com

Kiran Nagaraj
Nokia
Email: kiran.nagaraj@nokia.com

Julio Bueno
Telefonica
Email: julio.buenohernandez@telefonica.com

Jose Manuel Crespo
Telefonica
Email: josemanuel.crespogarcia@telefonica.com