

INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros
VMware
Rex Fernando
Ali Sajassi
Cisco Systems

Kitty Pang
Alibaba

Tapraj Singh
Juniper

Expires: April 24, 2017

October 21, 2016

EVPN auto provisioning using a controller
draft-boutros-bess-evpn-auto-provisoning-02

Abstract

In some datacenter use cases, priori knowledge of what PE/NVE to be configured for a given L2 or L3 service may not be available. This document describes how EVPN can be extended to discover what L2 or L3 services to be enabled on a given PE/NVE, based on first sign of life FSOL packets received on the PE/NVE ports. An EVPN route based on the FSOL packets will be sent to a controller to trigger a push of the related L2/L3 or subscriber service configuration to be provisioned on the PE/NVE and on the switch ports.

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) 2016 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.	Requirements	3
2.1	Auto-Provisioning	3
2.2	Scalability	3
2.3	Redundancy	4
2.4	Multi-homing	4
2.5	Fast Convergence	4
3.	Benefits	4
4.	Solution Overview	4
5	Ethernet Segment identifier encoding	6
6	Acknowledgements	6
7	Security Considerations	6
8	IANA Considerations	6
9	References	7
9.1	Normative References	7
9.2	Informative References	7
	Authors' Addresses	7

1 Introduction

This document describes how EVPN can be extended by access PE/NVE nodes and a controller in a data center to auto provision the L2 or L3 services needed to be enabled on the PE/NVE nodes.

Initially, all the PE/NVE nodes are configured with a default EVPN service that includes all Ethernet access ports. Based on the FSOL packets received on any of the Ethernet trunk ports, an EVPN MAC/IP Advertisement route is sent to the controller containing the MAC and IP information associated with this FSOL packet. The ESI field of the route encodes both the Ethernet port information as well as the Ethernet Tag associated with the FSOL packet.

Once the controller receives the MAC/IP Advertisement route from the PE/NVE node, it consults a pre-configured policy for any L2 or L3 services that need to be enabled on this PE/NVE node based on the information in the route. Any combination of fields encoded in the EVPN route may be used to that effect. If such service is required to be pushed to the PE/NVE node, the controller pushes the provisioning information to the access PE/NVE node and other PE/NVE nodes involved in this L2/L3 or subscriber service.

The alternative is to configure every EVPN instance on all PE/NVEs and that poses a scale concern on the PE/NVEs deployed in the DC.

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. Requirements

This section describes the requirements specific to this draft. These requirements are in addition to the ones described in [EVPN-REQ], [EVPN], and [EVPN-VPWS].

2.1 Auto-Provisioning

Auto provisioning of L2/L3 and subscriber services on PE/NVE nodes connected to a IP/MPLS fabric based on the FSOL packets received by the PE/NVE nodes.

2.2 Scalability

A single controller node can provision many access PE/NVE nodes.

A single controller node must be able to handle all EVPN routes received from all the access PE/NVE nodes that it is controlling.

2.3 Redundancy

TBD

2.4 Multi-homing

TBD

2.5 Fast Convergence

TBD

3. Benefits

This section describes some of the major benefits of EVPN Auto-provisioning.

Major benefits are:

- An easy and scalable mechanism for auto provisioning access PE/NVE nodes connected to a DC fabric based on FSOL using EVPN control plane.
- Auto-provision features such as QOS access lists (ACL), tunnel preference, bandwidth, L3VPN, EVPN, etc.. based on the policy plane previously available to the controller.

4. Solution Overview

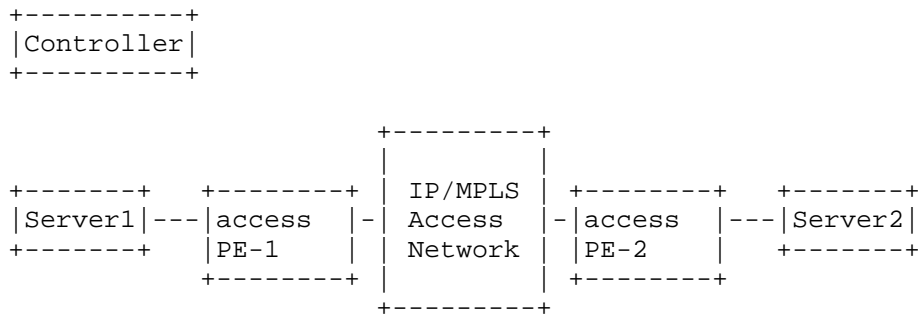


Figure 1:

EVPN-Auto provisioning Operation

Initially all the access PE/NVE nodes trunk ports will be associated with a default bridge and will be associated with a default EVPN instance that all PE/NVE node(s) and the controller are part of.

Based on FSOL packet received from Server1, an EVPN MAC/IP Advertisement route will be sent by PE-1 to the controller, the ESI value will be encoded to contain the access port number and the Ethernet Tag(s) associated with the FSOL packet, the IP and MAC fields will be set based on the source IP and MAC information on the FSOL packet.

Assuming for example, an operator previously provisioned a policy to associate a VLAN identifier on a given PE or set of PE(s) with a L2 or L3 service.

An operator may as well have previously provisioned an IPoE, MAC session or an unclassified VLAN or MAC service associated on with a given port on the access PE/NVE.

When the BGP EVPN advertisement is received by the controller, the controller checks the policy, and pushes down to the PE/NVE node or set of PE/NVE nodes(s) the L2/L3 or subscriber service to be provisioned on those access routers/switches.

A controller may as well based on the type of service, do authentication and authorization of service first before pushing the configuration associated with the service to the access PE/NVE.

When the service configured by the controller is an EVPN service, the provisioned access PE/NVE will advertise to other BGP Peers Inclusive Multicast route, the receiving PE/NVE(s) will check if an EVPN

service/EVI is configured with same RT or not. If the service is not configured with received RT the receiving PE may send the received Inclusive Mcast route to the controller. The Inclusive Mcast route may have the Ethernet Tag field set. Upon receiving the Inclusive Mcast route a controller may do authentication and authorization service and may push service configuration associated with the service to the PE/NVE.

Please note that controller's capability is outside of the scope of this draft.

5 Ethernet Segment identifier encoding

This document proposes a new ESI type to encode the Ethernet port on which the FSOL packet was received, and the Ethernet Tag(s) that are encoded on the FSOL packet.

```

+---+---+---+---+---+---+---+---+---+---+
| T |           ESI Value           |
+---+---+---+---+---+---+---+---+---+---+

```

The ESI 9 octets value will be as follow:

```

+---+---+---+---+---+---+---+---+---+---+
| T | Ethernet Port # | Vlan-1 | Vlan-2 | 0's |
+---+---+---+---+---+---+---+---+---+---+

```

Ethernet Port number encoded on the 1st 4 bytes, this Ethernet port number will be used on the controller to infer the actual physical port on the access node/router.

The Vlan-1 and Vlan-2 values are used to encode the Ethernet Tag identifiers found on the FSOL packet received on the Ethernet port.

6 Acknowledgements

The authors would like to thank Samer Salam for his valuable comments.

7 Security Considerations

This document does not introduce any additional security constraints.

8 IANA Considerations

New ESI type need to be allocated to specify the encoding in section 5.

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.

9.2 Informative References

[RFC7209] A. Sajassi, R. Aggarwal et. al., "Requirements for Ethernet VPN".

[EVPN] A. Sajassi, R. Aggarwal et. al., "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-11.txt.

[EVPN-VPWS] S. Boutros et. al., "EVPN-VPWS", draft-ietf-bess-evpn-
vpws-00.txt.

Authors' Addresses

Sami Boutros
VMware
Email: sboutros@vmware.com

Rex Fernando
Cisco
Email: rex@cisco.com

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Kitty Pang
Alibaba
Email: kittypang@alibaba-inc.com

Tapraj Singh
Juniper
Email: tsingh@juniper.net

INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros
VMware

Patrice Brissette
Ali Sajassi
Cisco Systems

Daniel Voyer
Bell Canada

John Drake
Juniper Networks

Expires: April 24, 2017

October 21, 2016

EVPN-VPWS Service Edge Gateway
draft-boutros-bess-evpn-vpws-service-edge-gateway-03

Abstract

This document describes how a service node can dynamically terminate EVPN virtual private wire transport service (VPWS) from access nodes and offer Layer 2, Layer 3 and Ethernet VPN overlay services to Customer edge devices connected to the access nodes. Service nodes using EVPN will advertise to access nodes the L2, L3 and Ethernet VPN overlay services it can offer for the terminated EVPN VPWS transport service. On an access node an operator can specify the L2 or L3 or Ethernet VPN overlay service needed by the customer edge device connected to the access node that will be transported over the EVPN-VPWS service between access node and service node.

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) 2016 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.	Requirements	4
2.1	Auto-Discovery	4
2.2	Scalability	4
2.3	Head-end	4
2.5	Multi-homing	5
2.5	Fast Convergence	5
3.	Benefits	5
4.	Solution Overview	5
4.1	Multi-homing	7
4.2	Applicability to IP-VPN	8
5	Failure Scenarios	8
6	Acknowledgements	8
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

This document describes how a service node can act as a gateway terminating dynamically EVPN virtual private wire service (VPWS) from access nodes and offering Layer 2, EVPN and Layer 3 VPN overlay services to Customer edge devices connected to the access nodes.

The service node would initially advertise using EVPN the different L2, L3 and Ethernet VPN overlay services that can be transported from access nodes over an EVPN-VPWS transport service.

The service node would advertise EVPN-VPWS per EVI Ethernet A-D routes with the Ethernet Segment Identifier field set to 0 and the Ethernet tag ID set to (0xFFFFFFFF wildcard), all those routes will be associated with the EVPN-VPWS service edge RT that will be imported by other service edge PEs, each route will have a unique RD and will be associated with another RT corresponding to the L2, L3 or Ethernet VPN overlay service that can be transported over the EVPN-VPWS transport service.

The access nodes will advertise EVPN-VPWS per EVI Ethernet A-D with the Ethernet Segment Identifier field set to 0 for single home customer edge CE device and set to the CE's ESI and the Ethernet Tag field is set to the VPWS service instance identifier. The route will have a unique RD and will be associated with an RT corresponding to the L2, L3 or Ethernet VPN overlay service that will be transported over the EVPN-VPWS transport service.

If more than one service node advertise the ability to terminate the EVPN-VPWS transport service and offer the L2, L3 or Ethernet VPN service required by CE device connected to a given access node, then all service node(s) will perform a DF election based on HWR algorithm using {Ethernet tag-id, Service node IP addresses} to determine which service node will be the primary service node to terminate the VPWS service and offer the L2, L3 or Ethernet overlay service for the customer edge, All active and single active redundancy can be offered.

The Service PE node that is a DF for a given VPWS service ID MUST respond to the Eth A-D route per EVI from the access node by sending its own Eth A-D per EVI route and by setting the same VPWS service instance ID and downstream assigned MPLS label to be used by Access node. When access node receives this Eth A-D route per EVI from the service node, it binds the two side of EVCs together.

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. Requirements

This section describes the requirements specific to this draft. These requirements are in addition to the ones described in [EVPN-REQ], [EVPN], and [EVPN-VPWS].

2.1 Auto-Discovery

A service node needs to support the following functionality of auto-discovery:

(R1a) A service node PE MUST be agnostic of all access nodes PEs connected on the same access network.

(R1b) A service node PE MUST advertise its associated overlay VRF(L2 and/or L3) to all service nodes PEs connected on the same network.

(R1c) A service node PE MUST resolve received overlay VRF(L2 and/or L3) from other service nodes with local configuration. The information is used to select proper service node PE for a given EVPN-VPWS connection from an access PE.

(R1d) A service node PE MUST accept EVPN-VPWS connection from any access node PE which require one of the service node PE available L2 or L3 overlay service.

2.2 Scalability

(R2a) A single service node PE can be associated with many access node PEs. The following requirements give a quantitative measure.

(R2b) A service node PE MUST support thousand(s) head-end connections for a a given access node PE connecting to different overlay VRF services on that service node.

(R2c) A service node PE MUST support thousand(s) head-end connections to many access node PEs.

2.3 Head-end

(R3a) A service node PE MUST support L2 and/or L3 head-end functionality.

(R3b) A service node PE SHALL support auto-configuration of L2 and/or

L3 head-end functionality.

2.5 Multi-homing

TBD

2.5 Fast Convergence

TBD

3. Benefits

This section describes some of the major benefits of EVPN-VPWS service edge gateway solution. This list is not considered as exhaustive.

Major benefits are:

- An easy and scalable mechanism for tunneling (head-end) customer traffic into a common IP/MPLS network infrastructure
- Auto-provision features such as QoS access lists (ACL), tunnel preference, bandwidth, L3VPN on a per head-end interface basis
- reduces CAPEX in the access or aggregation network and service PE
- Auto configuration of head-end functionality:

Configuring other Layer3 parameters, such as VRF and IP addresses, are optional for the head-end to be functional. However, they are required for Layer3 services to be operational (head-end L3 termination).

- Auto-discovery of access nodes by service nodes. Hence, there is no need to change any service node configuration when a new access node is being added to the access network.

4. Solution Overview

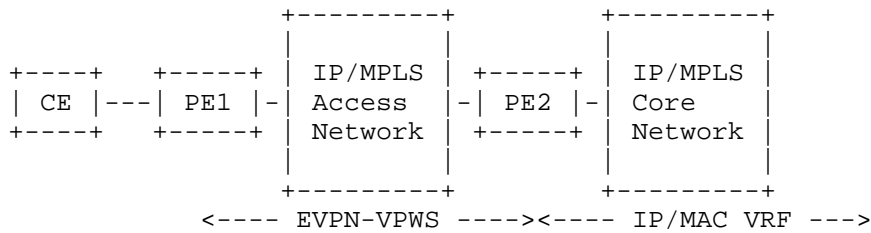


Figure 1: EVPN-VPWS Service Edge Gateway.

AN: Access node

SE: Service Edge node.

EVPN-VPWS Service Edge Gateway Operation

At the service edge node, the EVPN Per-EVI Ethernet A-D routes will be advertised with the ESI set to 0 and the Ethernet tag-id set to (wildcard 0xFFFFFFFF). The Ethernet A-D routes will have a unique RD and will be associated with 2 BGP RT(s), one RT corresponding to the underlay EVI i.e. the EVPN VPWS transport service that's configured only among the service edge nodes, and one corresponding to the L2, L3 or EVPN overlay service.

At the access nodes, the EVPN per-EVI Ethernet A-D routes will be advertised as described in [draft-ietf-bess-evpn-vpws] with the ESI field is set to 0 and for single homed CEs and to the CE's ESI for multi-homed CE's and the Ethernet Tag field will be set to the VPWS service instance identifier that identifies the EVPL or EPL service. The Ethernet-AD route will have a unique RD and will be associated with one BGP RT corresponding to the L2, L3 or EVPN overlay service that will be transported over this EVPN VPWS transport service.

Service edge nodes on the underlay EVI will determine the primary service node terminating the VPWS transport service and offering the L2, L3 or Ethernet VPN service by running the on HWR algorithm as described in [draft-mohanty-l2vpn-evpn-df-election] using weight [VPWS service identifier, Service Edge Node IP address]. This ensure that service node(s) will consistently pick the primary service node even after service node failure. Upon primary service node failure, all other remaining services nodes will choose another service node correctly and consistently.

Single-sided signaling mechanism is used. The Service PE node that is a DF for accepts to terminate the VPWS transport service from an access node, the primary service edge node shall:- Dynamically create an interface to terminate the service and shall attach this interface to the overlay VPN service required by the access node to service its

customer edge device.- Responds to the Eth A-D route per EVI from the access node by sending its own Eth A-D per EVI route by setting the same VPWS service instance ID and downstream assigned MPLS label to be used by the access node.

When access node receives this Eth A-D route per EVI from the service edge node, it binds the two side of EVCs together and it now knows what primary/backup service nodes to forward the traffic to.

The service edge node shall support per features such as QoS, ACL, etc. for the EVPN VPWS transport service it terminates.

4.1 Multi-homing

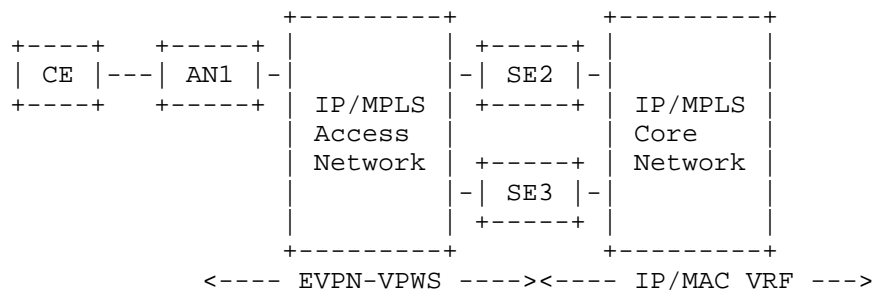


Figure 2: EVPN-VPWS SEG Multi-homing (same ASN)

AN: Access node
SE: Service Edge node.

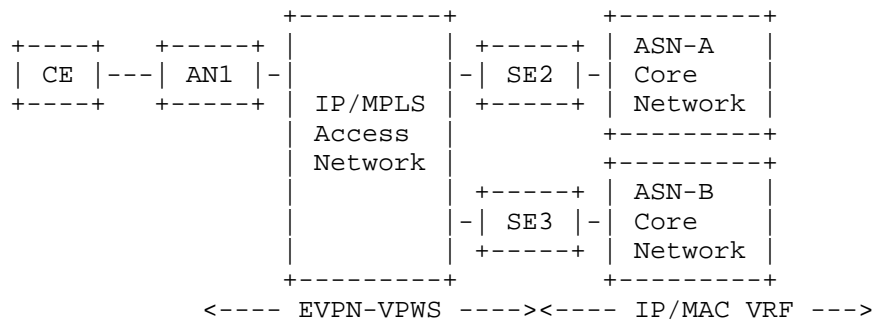


Figure 3: EVPN-VPWS SEG Multi-homing (different ASN)

AN: Access node
SE: Service Edge node.

Both All-active and single active redundancy can be supported.

A backup service node can be preprogrammed in data plane on an access node in order to switch traffic and based on how fast the data plane detect the failure of the primary service node traffic on an access node can switch to the backup node.

4.2 Applicability to IP-VPN TBD

5 Failure Scenarios TBD

6 Acknowledgements TBD.

7 Security Considerations

This document does not introduce any additional security constraints.

8 IANA Considerations

TBD.

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.

9.2 Informative References

[RFC7209] A. Sajassi, R. Aggarwal et. al., "Requirements for Ethernet VPN".

[EVPN] A. Sajassi, R. Aggarwal et. al., "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-11.txt.

[EVPN-VPWS] S. Boutros et. al., "EVPN-VPWS", draft-ietf-bess-evpn-
vpws-00.txt.

Authors' Addresses

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

Patrice Brissette
Cisco
Email: pbrisset@cisco.com

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Daniel Voyer
Bell Canada
Email: daniel.voyer@bell.ca

John Drake
Juniper Networks
Email: jdrake@juniper.net

INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros
VMware

Patrice Brissette
Ali Sajassi
Cisco Systems

Daniel Voyer
Bell Canada

John Drake
Juniper Networks

Expires: December 31, 2017

June 29, 2017

EVPN-VPWS Service Edge Gateway
draft-boutros-bess-evpn-vpws-service-edge-gateway-04

Abstract

This document describes how a service node can dynamically terminate EVPN virtual private wire transport service (VPWS) from access nodes and offer Layer 2, Layer 3 and Ethernet VPN overlay services to Customer edge devices connected to the access nodes. Service nodes using EVPN will advertise to access nodes the L2, L3 and Ethernet VPN overlay services it can offer for the terminated EVPN VPWS transport service. On an access node an operator can specify the L2 or L3 or Ethernet VPN overlay service needed by the customer edge device connected to the access node that will be transported over the EVPN-VPWS service between access node and service node.

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.	Requirements	4
2.1	Auto-Discovery	4
2.2	Scalability	4
2.3	Head-end	4
2.5	Multi-homing	5
2.5	Fast Convergence	5
3.	Benefits	5
4.	Solution Overview	5
4.1	Multi-homing	7
4.2	Applicability to IP-VPN	8
5	Failure Scenarios	8
6	Acknowledgements	8
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

This document describes how a service node can act as a gateway terminating dynamically EVPN virtual private wire service (VPWS) from access nodes and offering Layer 2, EVPN and Layer 3 VPN overlay services to Customer edge devices connected to the access nodes.

The service node would initially advertise using EVPN the different L2, L3 and Ethernet VPN overlay services that can be transported from access nodes over an EVPN-VPWS transport service.

The service node would advertise EVPN-VPWS per EVI Ethernet A-D routes with the Ethernet Segment Identifier field set to 0 and the Ethernet tag ID set to (0xFFFFFFFF wildcard), all those routes will be associated with the EVPN-VPWS service edge RT that will be imported by other service edge PEs, each route will have a unique RD and will be associated with another RT corresponding to the L2, L3 or Ethernet VPN overlay service that can be transported over the EVPN-VPWS transport service.

The access nodes will advertise EVPN-VPWS per EVI Ethernet A-D with the Ethernet Segment Identifier field set to 0 for single home customer edge CE device and set to the CE's ESI and the Ethernet Tag field is set to the VPWS service instance identifier. The route will have a unique RD and will be associated with an RT corresponding to the L2, L3 or Ethernet VPN overlay service that will be transported over the EVPN-VPWS transport service.

If more than one service node advertise the ability to terminate the EVPN-VPWS transport service and offer the L2, L3 or Ethernet VPN service required by CE device connected to a given access node, then all service node(s) will perform a DF election based on HWR algorithm using {Ethernet tag-id, Service node IP addresses} to determine which service node will be the primary service node to terminate the VPWS service and offer the L2, L3 or Ethernet overlay service for the customer edge, All active and single active redundancy can be offered.

The Service PE node that is a DF for a given VPWS service ID MUST respond to the Eth A-D route per EVI from the access node by sending its own Eth A-D per EVI route and by setting the same VPWS service instance ID and downstream assigned MPLS label to be used by Access node. When access node receives this Eth A-D route per EVI from the service node, it binds the two side of EVCs together.

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. Requirements

This section describes the requirements specific to this draft. These requirements are in addition to the ones described in [EVPN-REQ], [EVPN], and [EVPN-VPWS].

2.1 Auto-Discovery

A service node needs to support the following functionality of auto-discovery:

(R1a) A service node PE MUST be agnostic of all access nodes PEs connected on the same access network.

(R1b) A service node PE MUST advertise its associated overlay VRF(L2 and/or L3) to all service nodes PEs connected on the same network.

(R1c) A service node PE MUST resolve received overlay VRF(L2 and/or L3) from other service nodes with local configuration. The information is used to select proper service node PE for a given EVPN-VPWS connection from an access PE.

(R1d) A service node PE MUST accept EVPN-VPWS connection from any access node PE which require one of the service node PE available L2 or L3 overlay service.

2.2 Scalability

(R2a) A single service node PE can be associated with many access node PEs. The following requirements give a quantitative measure.

(R2b) A service node PE MUST support thousand(s) head-end connections for a a given access node PE connecting to different overlay VRF services on that service node.

(R2c) A service node PE MUST support thousand(s) head-end connections to many access node PEs.

2.3 Head-end

(R3a) A service node PE MUST support L2 and/or L3 head-end functionality.

(R3b) A service node PE SHALL support auto-configuration of L2 and/or

L3 head-end functionality.

2.5 Multi-homing

TBD

2.5 Fast Convergence

TBD

3. Benefits

This section describes some of the major benefits of EVPN-VPWS service edge gateway solution. This list is not considered as exhaustive.

Major benefits are:

- An easy and scalable mechanism for tunneling (head-end) customer traffic into a common IP/MPLS network infrastructure
- Auto-provision features such as QoS access lists (ACL), tunnel preference, bandwidth, L3VPN on a per head-end interface basis
- reduces CAPEX in the access or aggregation network and service PE
- Auto configuration of head-end functionality:

Configuring other Layer3 parameters, such as VRF and IP addresses, are optional for the head-end to be functional. However, they are required for Layer3 services to be operational (head-end L3 termination).

- Auto-discovery of access nodes by service nodes. Hence, there is no need to change any service node configuration when a new access node is being added to the access network.

4. Solution Overview

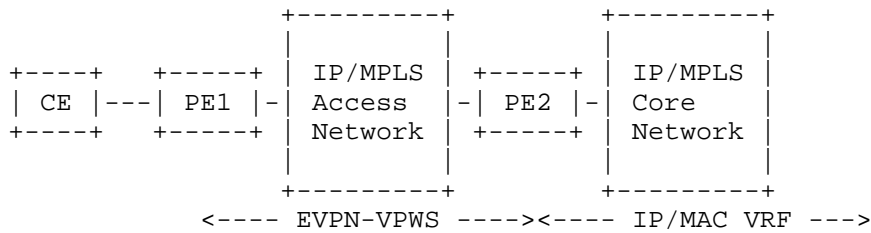


Figure 1: EVPN-VPWS Service Edge Gateway.

AN: Access node

SE: Service Edge node.

EVPN-VPWS Service Edge Gateway Operation

At the service edge node, the EVPN Per-EVI Ethernet A-D routes will be advertised with the ESI set to 0 and the Ethernet tag-id set to (wildcard 0xFFFFFFFF). The Ethernet A-D routes will have a unique RD and will be associated with 2 BGP RT(s), one RT corresponding to the underlay EVI i.e. the EVPN VPWS transport service that's configured only among the service edge nodes, and one corresponding to the L2, L3 or EVPN overlay service.

At the access nodes, the EVPN per-EVI Ethernet A-D routes will be advertised as described in [draft-ietf-bess-evpn-vpws] with the ESI field is set to 0 and for single homed CEs and to the CE's ESI for multi-homed CE's and the Ethernet Tag field will be set to the VPWS service instance identifier that identifies the EVPL or EPL service. The Ethernet-AD route will have a unique RD and will be associated with one BGP RT corresponding to the L2, L3 or EVPN overlay service that will be transported over this EVPN VPWS transport service.

Service edge nodes on the underlay EVI will determine the primary service node terminating the VPWS transport service and offering the L2, L3 or Ethernet VPN service by running the on HWR algorithm as described in [draft-mohanty-l2vpn-evpn-df-election] using weight [VPWS service identifier, Service Edge Node IP address]. This ensure that service node(s) will consistently pick the primary service node even after service node failure. Upon primary service node failure, all other remaining services nodes will choose another service node correctly and consistently.

Single-sided signaling mechanism is used. The Service PE node that is a DF for accepts to terminate the VPWS transport service from an access node, the primary service edge node shall:- Dynamically create an interface to terminate the service and shall attach this interface to the overlay VPN service required by the access node to service its

customer edge device.- Responds to the Eth A-D route per EVI from the access node by sending its own Eth A-D per EVI route by setting the same VPWS service instance ID and downstream assigned MPLS label to be used by the access node.

When access node receives this Eth A-D route per EVI from the service edge node, it binds the two side of EVCs together and it now knows what primary/backup service nodes to forward the traffic to.

The service edge node shall support per features such as QoS, ACL, etc. for the EVPN VPWS transport service it terminates.

4.1 Multi-homing

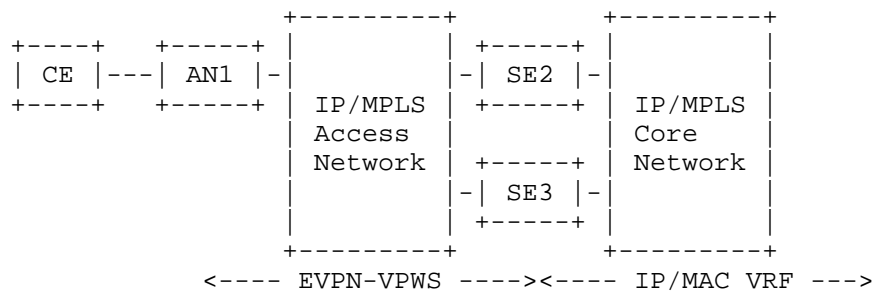


Figure 2: EVPN-VPWS SEG Multi-homing (same ASN)

AN: Access node
SE: Service Edge node.

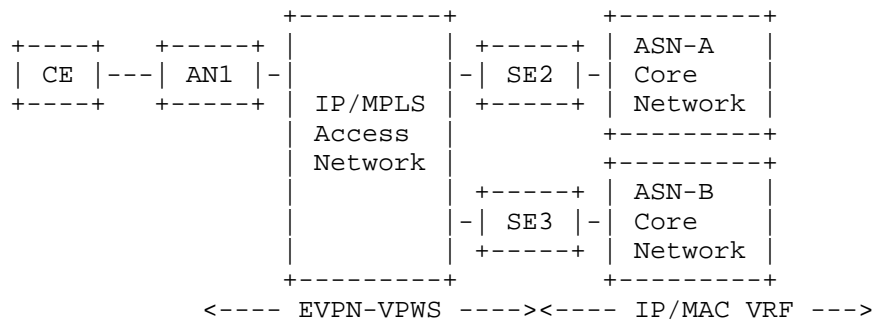


Figure 3: EVPN-VPWS SEG Multi-homing (different ASN)

AN: Access node
SE: Service Edge node.

Both All-active and single active redundancy can be supported.

A backup service node can be preprogrammed in data plane on an access node in order to switch traffic and based on how fast the data plane detect the failure of the primary service node traffic on an access node can switch to the backup node.

4.2 Applicability to IP-VPN TBD

5 Failure Scenarios TBD

6 Acknowledgements TBD.

7 Security Considerations

This document does not introduce any additional security constraints.

8 IANA Considerations

TBD.

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.

9.2 Informative References

[RFC7209] A. Sajassi, R. Aggarwal et. al., "Requirements for Ethernet VPN".

[EVPN] A. Sajassi, R. Aggarwal et. al., "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn-11.txt.

[EVPN-VPWS] S. Boutros et. al., "EVPN-VPWS", draft-ietf-bess-evpn-
vpws-00.txt.

Authors' Addresses

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

Patrice Brissette
Cisco
Email: pbrisset@cisco.com

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Daniel Voyer
Bell Canada
Email: daniel.voyer@bell.ca

John Drake
Juniper Networks
Email: jdrake@juniper.net

INTERNET-DRAFT
Intended Status: Informational

Sami Boutros
VMware

Ali Sajassi
Samer Salam
Dennis Cai
Samir Thoria
Cisco Systems

Tapraj Singh
John Drake
Juniper Networks

Jeff Tantsura
Ericsson

Expires: April 24, 2017

October 21, 2016

VXLAN DCI Using EVPN
draft-boutros-bess-vxlan-evpn-02.txt

Abstract

This document describes how Ethernet VPN (E-VPN) technology can be used to interconnect VXLAN or NVGRE networks over an MPLS/IP network. This is to provide intra-subnet connectivity at Layer 2 and control-plane separation among the interconnected VXLAN or NVGRE networks. The scope of the learning of host MAC addresses in VXLAN or NVGRE network is limited to data plane learning in this document.

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) 2016 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	4
2.	Requirements	4
2.1.	Control Plane Separation among VXLAN/NVGRE Networks	4
2.2	All-Active Multi-homing	5
2.3	Layer 2 Extension of VNIs/VSIDs over the MPLS/IP Network	5
2.4	Support for Integrated Routing and Bridging (IRB)	5
3.	Solution Overview	5
3.1.	Redundancy and All-Active Multi-homing	6
4.	EVPN Routes	7
4.1.	BGP MAC Advertisement Route	7
4.2.	Ethernet Auto-Discovery Route	8
4.3.	Per VPN Route Targets	8
4.4	Inclusive Multicast Route	8
4.5.	Unicast Forwarding	8
4.6.	Handling Multicast	9
4.6.2.	Multicast Stitching with Per-VNI Load Balancing	9
4.6.2.1	PIM SM operation	10
5.	NVGRE	11
6.	Use Cases Overview	11
6.1.	Homogeneous Network DCI interconnect Use cases	12
6.1.1.	VNI Base Mode EVPN Service Use Case	12
6.1.2.	VNI Bundle Service Use Case Scenario	13
6.1.3.	VNI Translation Use Case	13

- 6.2. Heterogeneous Network DCI Use Cases Scenarios 13
 - 6.2.1. VXLAN VLAN Interworking Over EVPN Use Case Scenario . . 13
- 7. Acknowledgements 14
- 8. Security Considerations 14
- 9. IANA Considerations 14
- 10. References 14
 - 10.1 Normative References 14
 - 10.2 Informative References 14
- Authors' Addresses 15

1 Introduction

[EVPN] introduces a solution for multipoint L2VPN services, with advanced multi-homing capabilities, using BGP control plane over the core MPLS/IP network. [VXLAN] defines a tunneling scheme to overlay Layer 2 networks on top of Layer 3 networks. [VXLAN] allows for optimal forwarding of Ethernet frames with support for multipathing of unicast and multicast traffic. VXLAN uses UDP/IP encapsulation for tunneling.

In this document, we discuss how Ethernet VPN (EVPN) technology can be used to interconnect VXLAN or NVGRE networks over an MPLS/IP network. This is achieved by terminating the VxLAN tunnel at the hand-off points, performing data plane MAC learning of customer traffic and providing intra-subnet connectivity for the customers at Layer 2 across the MPLS/IP core. The solution maintains control-plane separation among the interconnected VXLAN or NVGRE networks. The scope of the learning of host MAC addresses in VXLAN or NVGRE network is limited to data plane learning in this document. The distribution of MAC addresses in control plane using BGP in VXLAN or NVGRE network is outside of the scope of this document and it is covered in [EVPN-OVERLY].

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].

LDP: Label Distribution Protocol. MAC: Media Access Control MPLS: Multi Protocol Label Switching. OAM: Operations, Administration and Maintenance. PE: Provide Edge Node. PW: PseudoWire. TLV: Type, Length, and Value. VPLS: Virtual Private LAN Services. VXLAN: Virtual eXtensible Local Area Network. VTEP: VXLAN Tunnel End Point VNI: VXLAN Network Identifier (or VXLAN Segment ID) ToR: Top of Rack switch. LACP: Link Aggregation Control Protocol

2. Requirements

2.1. Control Plane Separation among VXLAN/NVGRE Networks

It is required to maintain control-plane separation for the underlay networks (e.g., among the various VXLAN/NVGRE networks) being interconnected over the MPLS/IP network. This ensures the following characteristics:

- scalability of the IGP control plane in large deployments and fault domain localization, where link or node failures in one site do not

trigger re-convergence in remote sites.

- scalability of multicast trees as the number of interconnected networks scales.

2.2 All-Active Multi-homing

It is important to allow for all-active multi-homing of the VXLAN/NVGRE network to MPLS/IP network where traffic from a VTEP can arrive at any of the PEs and can be forwarded accordingly over the MPLS/IP network. Furthermore, traffic destined to a VTEP can be received over the MPLS/IP network at any of the PEs connected to the VXLAN/NVGRE network and be forwarded accordingly. The solution MUST support all-active multi-homing to an VXLAN/NVGRE network.

2.3 Layer 2 Extension of VNIs/VSIDs over the MPLS/IP Network

It is required to extend the VXLAN VNIs or NVGRE VSIDs over the MPLS/IP network to provide intra-subnet connectivity between the hosts (e.g. VMs) at Layer 2.

2.4 Support for Integrated Routing and Bridging (IRB)

The data center WAN edge node is required to support integrated routing and bridging in order to accommodate both inter-subnet routing and intra-subnet bridging for a given VNI/VSID. For example, inter-subnet switching is required when a remote host connected to an enterprise IP-VPN site wants to access an application resided on a VM.

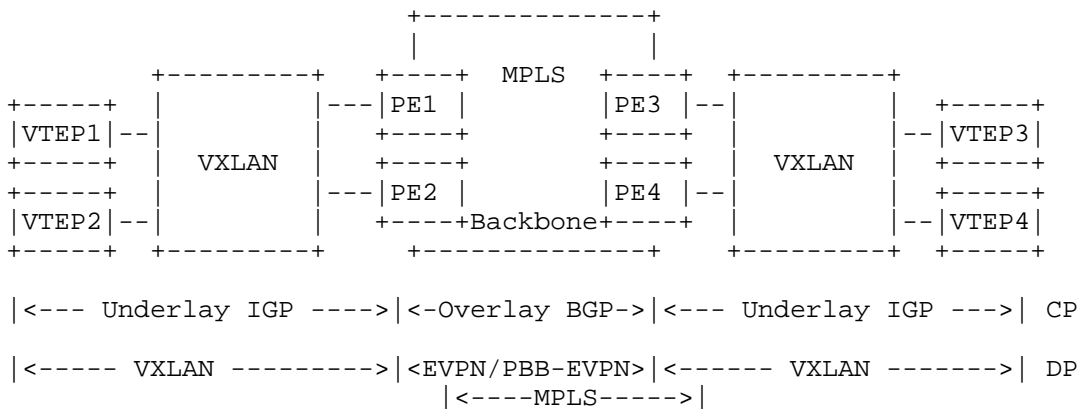
3. Solution Overview

Every VXLAN/NVGRE network, which is connected to the MPLS/IP core, runs an independent instance of the IGP control-plane. Each PE participates in the IGP control plane instance of its VXLAN/NVGRE network.

Each PE node terminates the VXLAN or NVGRE data-plane encapsulation where each VNI or VSID is mapped to a bridge-domain. The PE performs data plane MAC learning on the traffic received from the VXLAN/NVGRE network.

Each PE node implements EVPN or PBB-EVPN to distribute in BGP either the client MAC addresses learnt over the VXLAN tunnel in case of EVPN, or the PEs' B-MAC addresses in case of PBB-EVPN. In the PBB-EVPN case, client MAC addresses will continue to be learnt in data plane.

Each PE node would encapsulate the Ethernet frames with MPLS when sending the packets over the MPLS core and with the VXLAN or NVGRE tunnel header when sending the packets over the VXLAN or NVGRE Network.



Legend: CP = Control Plane View DP = Data Plane View

Figure 1: Interconnecting VXLAN Networks with VXLAN-EVPN

3.1. Redundancy and All-Active Multi-homing

When a VXLAN network is multi-homed to two or more PEs, and provided that these PEs have the same IGP distance to a given NVE, the solution MUST support load-balancing of traffic between the NVE and the MPLS network, among all the multi-homed PEs. This maximizes the use of the bisectional bandwidth of the VXLAN network. One of the main capabilities of EVPN/PBB-EVPN is the support for all-active multi-homing, where the known unicast traffic to/from a multi-homed site can be forwarded by any of the PEs attached to that site. This ensures optimal usage of multiple paths and load balancing. EVPN/PBB-EVPN, through its DF election and split-horizon filtering mechanisms, ensures that no packet duplication or forwarding loops result in such scenarios. In this solution, the VXLAN network is treated as a multi-homed site for the purpose of EVPN operation.

Since the context of this solution is VXLAN networks with data-plane learning paradigm, it is important for the multi-homing mechanism to ensure stability of the MAC forwarding tables at the NVEs, while supporting all-active forwarding at the PEs. For example, in Figure 1 above, if each PE uses a distinct IP address for its VTEP tunnel, then for a given VNI, when an NVE learns a host's MAC address against the originating VTEP source address, its MAC forwarding table will

keep flip-flopping among the VTEP addresses of the local PEs. This is because a flow associated with the same host MAC address can arrive at any of the PE devices. In order to ensure that there is no flip/flopping of MAC-to-VTEP address associations, an IP Anycast address MUST be used as the VTEP address on all PEs multi-homed to a given VXLAN network. The use of IP Anycast address has two advantages:

- a) It prevents any flip/flopping in the forwarding tables for the MAC-to-VTEP associations
- b) It enables load-balancing via ECMP for DCI traffic among the multi-homed PEs

In the baseline [EVPN] draft, the all-active multi-homing is described for a multi-homed device (MHD) using [LACP] and the single-active multi-homing is described for a multi-homed network (MHN) using [802.1Q]. In this draft, the all-active multi-homing is described for a VXLAN MHN. This implies some changes to the filtering which will be described in details in the multicast section (Section 4.6.2).

The filtering used for BUM traffic of all-active multi-homing in [EVPN] is asymmetric; where the BUM traffic from the MPLS/IP network towards the multi-homed site is filtered on non-DF PE(s) and it passes thorough the DF PE. There is no filtering of BUM traffic originating from the multi-homed site because of the use of Ethernet Link Aggregation: the MHD hashes the BUM traffic to only a single link. However, in this solution because BUM traffic can arrive at both PEs in both core-to-site and site-to-core directions, the filtering needs to be symmetric just like the filtering of BUM traffic for single-active multi-homing (on a per service instance/VLAN basis).

4. EVPN Routes

This solution leverages the same BGP Routes and Attributes defined in [EVPN], adapted as follows:

4.1. BGP MAC Advertisement Route

This route and its associated modes are used to distribute the customer MAC addresses learnt in data plane over the VXLAN tunnel in case of EVPN. Or can be used to distribute the provider Backbone MAC addresses in case of PBB-EVPN.

In case of EVPN, the Ethernet Tag ID of this route is set to zero for VNI-based mode, where there is one-to-one mapping between a VNI and

an EVI. In such case, there is no need to carry the VNI in the MAC advertisement route because BD ID can be derived from the RT associated with this route. However, for VNI-aware bundle mode, where there is multiple VNIs can be mapped to the same EVI, the Ethernet Tag ID MUST be set to the VNI. At the receiving PE, the BD ID is derived from the combination of RT + VNI - e.g., the RT identifies the associated EVI on that PE and the VNI identifies the corresponding BD ID within that EVI.

The Ethernet Tag field can be set to a normalized value that maps to the VNI, in VNI aware bundling services, this would make the VNI value of local significance in multiple Data centers. Data plane need to map to this normalized VNI value and have it on the IP VxLAN packets exchanged between the DCIs.

4.2. Ethernet Auto-Discovery Route

When EVPN is used, the application of this route is as specified in [EVPN]. However, when PBB-EVPN is used, there is no need for this route per [PBB-EVPN].

4.3. Per VPN Route Targets

VXLAN-EVPN uses the same set of route targets defined in [EVPN].

4.4 Inclusive Multicast Route

The EVPN Inclusive Multicast route is used for auto-discovery of PE devices participating in the same tenant virtual network identified by a VNI over the MPLS network. It also enables the stitching of the IP multicast trees, which are local to each VXLAN site, with the Label Switched Multicast (LSM) trees of the MPLS network.

The Inclusive Multicast Route is encoded as follow:

- Ethernet Tag ID is set to zero for VNI-based mode and to VNI for VNI-aware bundle mode.
- Originating Router's IP Address is set to one of the PE's IP addresses.

All other fields are set as defined in [EVPN].

Please see section 4.6 "Handling Multicast"

4.5. Unicast Forwarding

Host MAC addresses will be learnt in data plane from the VXLAN

network and associated with the corresponding VTEP identified by the source IP address. Host MAC addresses will be learnt in control plane if EVPN is implemented over the MPLS/IP core, or in the data-plane if PBB-EVPN is implemented over the MPLS core. When Host MAC addresses are learned in data plane over MPLS/IP core [in case of PBB-EVPN], they are associated with their corresponding BMAC addresses.

L2 Unicast traffic destined to the VXLAN network will be encapsulated with the IP/UDP header and the corresponding customer bridge VNI.

L2 Unicast traffic destined to the MPLS/IP network will be encapsulated with the MPLS label.

4.6. Handling Multicast

Each VXLAN network independently builds its P2MP or MP2MP shared multicast trees. A P2MP or MP2MP tree is built for one or more VNIs local to the VXLAN network.

In the MPLS/IP network, multiple options are available for the delivery of multicast traffic:

- Ingress replication
- LSM with Inclusive trees
- LSM with Aggregate Inclusive trees
- LSM with Selective trees
- LSM with Aggregate Selective trees

When LSM is used, the trees are P2MP.

The PE nodes are responsible for stitching the IP multicast trees, on the access side, to the ingress replication tunnels or LSM trees in the MPLS/IP core. The stitching must ensure that the following characteristics are maintained at all times:

1. Avoiding Packet Duplication: In the case where the VXLAN network is multi-homed to multiple PE nodes, if all of the PE nodes forward the same multicast frame, then packet duplication would arise. This applies to both multicast traffic from site to core as well as from core to site.

2. Avoiding Forwarding Loops: In the case of VXLAN network multi-homing, the solution must ensure that a multicast frame forwarded by a given PE to the MPLS core is not forwarded back by another PE (in the same VXLAN network) to the VXLAN network of origin. The same applies for traffic in the core to site direction.

The following approach of per-VNI load balancing can guarantee proper stitching that meets the above requirements.

4.6.2. Multicast Stitching with Per-VNI Load Balancing

To setup multicast trees in the VXLAN network for DC applications, PIM Bidir can be of special interest because it reduces the amount of multicast state in the network significantly. Furthermore, it alleviates any special processing for RPF check since PIM Bidir doesn't require any RPF check. The RP for PIM Bidir can be any of the spine nodes. Multiple trees can be built (e.g., one tree rooted per spine node) for efficient load-balancing within the network. All PEs participating in the multi-homing of the VXLAN network join all the trees. Therefore, for a given tree, all PEs receive BUM traffic. DF election procedures of [EVPN] are used to ensure that only traffic to/from a single PE is forwarded, thus avoiding packet duplications and forwarding loops. For load-balancing of BUM traffic, when a PE or an NVE wants to send BUM traffic over the VXLAN network, it selects one of the trees based on its VNI and forwards all the traffic for that VNI on that tree.

Multicast traffic from VXLAN/NVGRE is first subjected to filtering based on DF election procedures of [EVPN] using the VNI as the Ethernet Tag. This is similar to filtering in [EVPN] in principal; however, instead of VLAN ID, VNI is used for filtering, and instead of being 802.1Q frame, it is a VXLAN encapsulated packet. On the DF PE, where the multicast traffic is allowed to be forwarded, the VNI is used to select a bridge domain,. After the packet is de-encapsulated, an L2 lookup is performed based on host MAC DA. It should be noted that the MAC learning is performed in data-plane for the traffic received from the VXLAN/NVGRE network and the host MAC SA is learnt against the source VTEP address.

The PE nodes, connected to a multi-homed VXLAN network, perform BGP DF election to decide which PE node is responsible for forwarding multicast traffic associated with a given VNI. A PE would forward multicast traffic for a given VNI only when it is the DF for this VNI. This forwarding rule applies in both the site-to-core as well as core-to-site directions.

4.6.2.1 PIM SM operation

With PIM SM, multicast traffic from the core-to-site could be dropped since a transit router may decide that the RPF path towards the anycast address source is toward a PE node that is not the DF.

The PE nodes whether DF or not, has to forward forward multicast traffic from core-to-side.

The operation would work as follow:

Initially, the PE nodes connected to the multi-homed VXLAN network as well the VTEPs, join towards the RP for the multicast group for a

particular VXLAN.

When BUM traffic needs to be flooded from core to site, all the PE nodes connected to the multi-homed VXLAN network send PIM register messages to the RP. The multicast flow is identified as (anycast address, group) in the register message, and the source address for the PIM-SM register message should be a unique address on the PE node not the anycast address.

The RP will send a join for the (anycast address, group) upon receiving the register message, routed towards the closest PE which could be either the DF or the non-DF. This PE will switch to send traffic natively. Upon receiving the native traffic, the RP will send register-stop messages for other PEs that keep sending registering messages, given that only one PE will get the (anycast address, group) join.

When VTEPs receive traffic from the RP, VTEPs will send (anycast address, group) join, routed towards the closest PE to each VTEP. This starts native forwarding on multiple PE nodes connected to the VXLAN network, but each VTEP or transit router will only accept multicast traffic from one of the multi-homed PE nodes.

If PIM state times out when multicast traffic stops for a period of time, the next flooded packet will trigger the above process again.

It is to be noted that before the RP receives the first natively sent packet from one particular PE node connected to the multihomed VXLAN network, all packets encapsulated in the register messages from all PEs will be forwarded by the RP, causing duplications.

A possible optimization is for all PE nodes connected to the multihomed VXLAN network to send null-register periodically to maintain the PIM state at the RP, instead of encapsulating flooded packets in register messages.

The site-to-core operations for flooding BUM traffic would still be subject to DF election per VNI as described above.

5. NVGRE

Just like VXLAN, all the above specification would apply for NVGRE, replacing the VNI with Virtual Subnet Identifier (VSID) and the VTEP with NVGRE Endpoint.

6. Use Cases Overview

6.1. Homogeneous Network DCI interconnect Use cases

This covers DCI interconnect of two or more VXLAN based Data center over MPLS enabled EVPN core.

6.1.1. VNI Base Mode EVPN Service Use Case

This use case handles the EVPN service where there is one to one mapping between a VNI and an EVI. Ethernet TAG ID of EVPN BGP NLRI should be set to Zero. BD ID can be derived from the RT associated with the EVI/VNI.

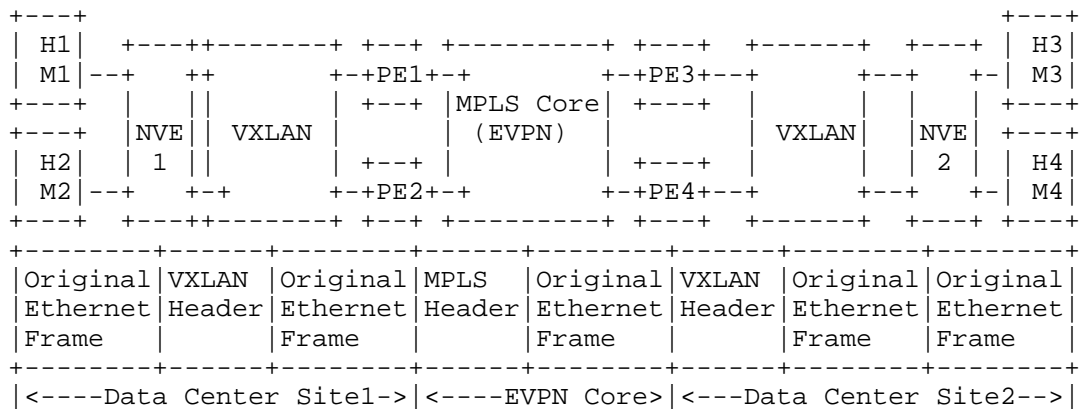


Figure 2 VNI Base Service Packet Flow.

VNI base Service(One VNI mapped to one EVI).

Hosts H1, H2, H3 and H4 are hosts and there associated MAC addresses are M1, M2, M3 and M4. PE1, PE2, PE3 and PE4 are the VXLAN-EVPN gateways. NVE1 and NVE2 are the originators of the VXLAN based network.

When host H1 in Data Center Site1 communicates with H3 in Data Center Site2, H1 forms a layer2 packet with source IP address as IP1 and Source MAC M1, Destination IP as IP3 and Destination MAC as M3(assuming that ARP resolution already happened). VNE1 learns Source MAC and lookup in bridge domain for the Destination MAC. Based on the MAC lookup, the frame needs to be sent to VXLAN network. VXLAN encapsulation is added to the original Ethernet frame and frame is sent over the VXLAN tunnel. Frames arrives at PE1. PE1(i.e. VXLAN gateway), identifies that frame is a VXLAN frame. The VXLAN header is de-capsulated and Destination MAC lookup is done in the bridge domain table of the EVI. Lookup of destination MAC results in the EVPN unicast NH. This NH will be used for identifying the labels (tunnel

label and service label) to be added over the EVPN core. Similar processing is done on the other side of DCI.

6.1.2. VNI Bundle Service Use Case Scenario

In the case of VNI-aware bundle service mode, there are multiple VNIs are mapped to one EVI. The Ethernet TAG ID must be set to the VNI ID in the EVPN BGP NLRIs. MPLS label allocation in this use case scenario can be done either per EVI or per EVI, VNI ID basis. If MPLS label allocation is done per EVI basis, then in data path there is a need to push a VLAN TAG for identifying bridge-domain at egress PE so that Destination MAC address lookup can be done on the bridge domain.

6.1.3. VNI Translation Use Case

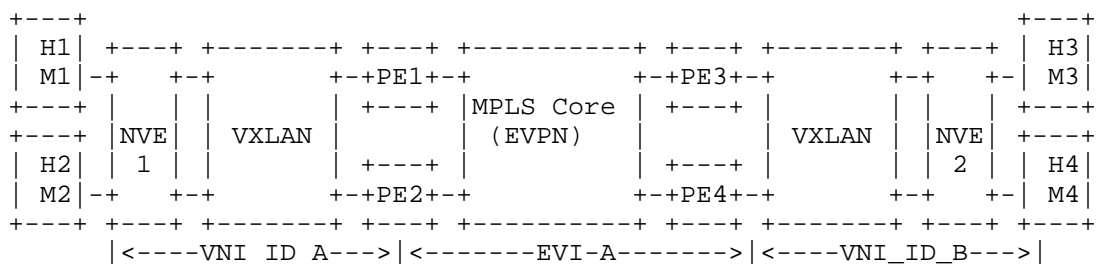


Figure 3 VNI Translation Use Case Scenarios.

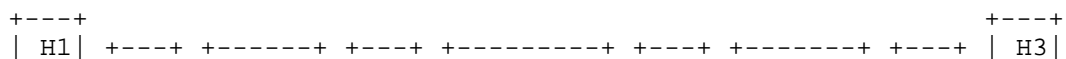
There are two or more Data Center sites. These Data Center sites might use different VNI ID for same service. For example, Service A usage "VNI_ID_A" at data center sitel and "VNI_ID_B" for same service in data center site 2. VNI ID A is terminated at ingress EVPN PE and VNI ID B is encapsulated at the egress EVPN PE.

6.2. Heterogeneous Network DCI Use Cases Scenarios

Data Center sites are upgraded slowly; so heterogeneous network DCI solution is required from the perspective of migration approach from traditional data center to VXLAN based data center. For Example Data Center Site1 is upgrade to VXLAN but Data Center Site 2 and 3 are still layer2/VLAN based data centers. For these use cases, it is required to provide VXLAN VLAN interworking over EVPN core.

6.2.1. VXLAN VLAN Interworking Over EVPN Use Case Scenario

The new data center site is VXLAN based data center site. But the older data center sites are still based on the VLAN.



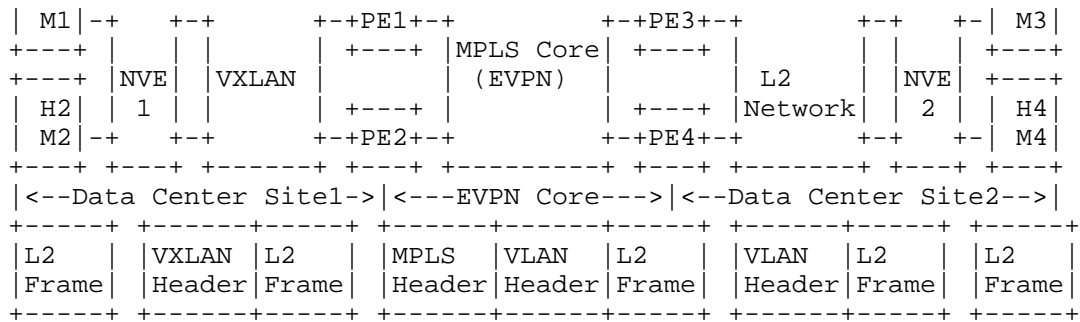


Figure 5 VXLAN VLAN interworking over EVPN Use Case.

If a service that are represented by VXLAN on one site of data center and via VLAN at different data center sites, then it is a recommended to model the service as a VNI base EVPN service. The BGP NLRIs will always advertise VLAN ID TAG as '0' in BGP routes. The advantage with this approach is that there is no requirement to do the VNI normalization at EVPN core. VNI ID A is terminated at ingress EVPN PE and "VLAN ID B" is encapsulated at the egress EVPN PE.

7. Acknowledgements

The authors would like to acknowledge Wen Lin contributions to this document.

8. Security Considerations

There are no additional security aspects that need to be discussed here.

9. IANA Considerations

10. References

10.1 Normative References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

10.2 Informative References

[EVPN] Sajassi et al., "BGP MPLS Based Ethernet VPN", RFC 7432, February, 2012.

[PBB-EVPN] Sajassi et al., "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, September, 2015.

[VXLAN] Mahalingam, Dutt et al., A Framework for Overlaying

Virtualized Layer 2 Networks over Layer 3 Networks, RFC 7348, August, 2012.

[NVGRE] Sridharan et al., Network Virtualization using Generic Routing Encapsulation, RFC 7637, July, 2012.

Authors' Addresses

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

Ali Sajassi
Cisco Systems
EMail: sajassi@cisco.com

Samer Salam
Cisco Systems
EMail: ssalam@cisco.com

Dennis Cai
Cisco Systems
EMail: dcai@cisco.com

Tapraj Singh
Juniper Networks
Email: tsingh@juniper.net

John Drake
Juniper Networks
Email: jdrake@juniper.net

Samir Thoria
Cisco
EMail: sthoria@cisco.com

Jeff Tantsura
Ericsson
Email: jeff.tantsura@ericsson.com

BESS Working Group
Internet-Draft
Intended status: Informational
Expires: April 19, 2017

J. Drake
A. Farrel
E. Rosen
Juniper Networks
K. Patel
Arccus, Inc.
L. Jalil
Verizon
October 16, 2016

Gateway Auto-Discovery and Route Advertisement for Segment Routing
Enabled Data Center Interconnection
draft-drake-bess-datacenter-gateway-02

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.

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 data center site to which it provides access, and also to advertise on behalf of each other gateway to the same data center site.

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 April 19, 2017.

Copyright Notice

Copyright (c) 2016 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. DC Gateway Auto-Discovery	5
3. Relationship to BGP Link State and Egress Peer Engineering	6
4. Advertising a DC Route Externally	6
5. Encapsulation	7
6. IANA Considerations	7
7. Security Considerations	7
8. Manageability Considerations	7
9. Acknowledgements	7
10. References	7
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.

Suppose that there are two gateways, GW1 and GW2 as shown in Figure 1, for a given egress DC and that they each advertise a route to prefix X which is located within the egress DC 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.

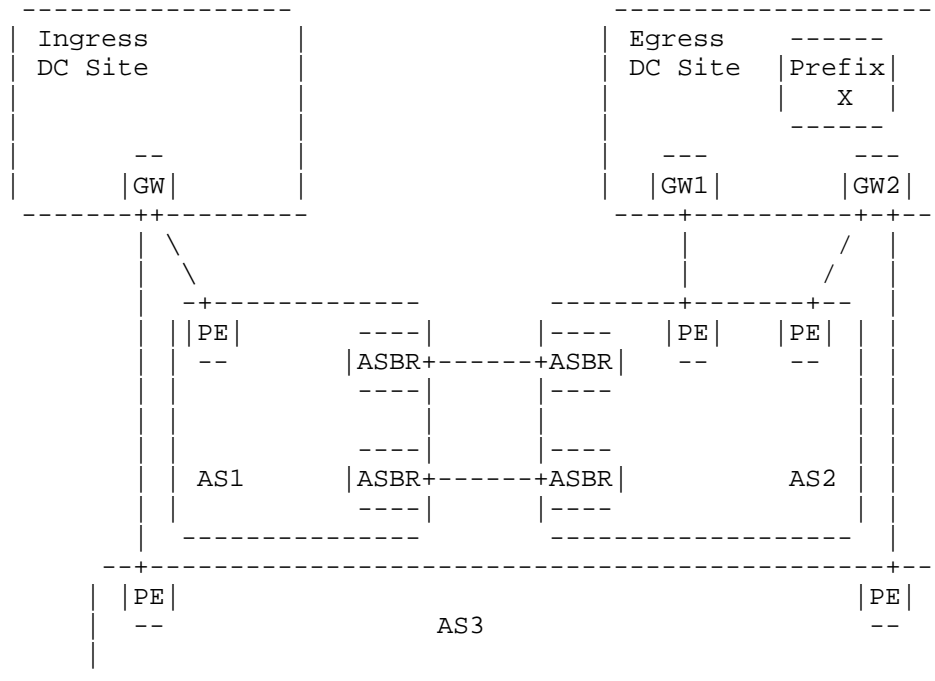


Figure 1: Example Data Center Interconnection

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 DC sites 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 DC advertise a route to a prefix X within the DC, 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 DC (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 data center site. 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 data center site can now see all possible paths to the egress data center site regardless of which route advertisement is propagated to it, and it can choose one or balance traffic flows as it sees fit.

2. DC Gateway Auto-Discovery

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

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 DC site (including itself), and will include these in every route advertised externally to the DC site 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 DC 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 DC 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 a DC Route Externally

When a packet destined for prefix X is sent on an SR TE path to a GW for the DC site 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 DC are configured to allow remote GWs to perform SR TE through that DC for a prefix X, then each GW computes an SR TE path through that DC 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 DC are configured to allow remote GWs to send them a packet in that DC'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., Ray, S., Patel, K., Dong, J.,
and M. Chen, "Segment Routing BGP Egress Peer Engineering
BGP-LS Extensions", draft-ietf-idr-bgpls-segment-routing-
epe-05 (work in progress), May 2016.
- [I-D.ietf-idr-tunnel-encaps]
Rosen, E., Patel, K., and G. Velde, "The BGP Tunnel
Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-02
(work in progress), May 2016.
- [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. jefftant@gmail.com, "BGP Link-State extensions
for Segment Routing", draft-gredler-idr-bgp-ls-segment-
routing-ext-03 (work in progress), July 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-09 (work in progress), July 2016.
- [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: adrian@olddog.co.uk

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
Arccus, 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.

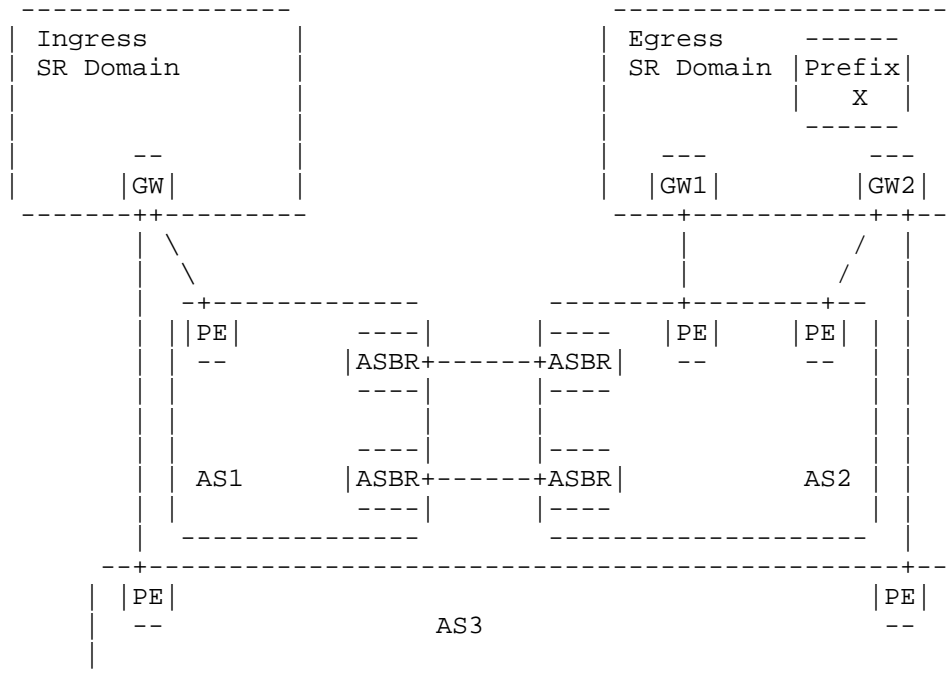


Figure 1: Example Segment Routing Domain Interconnection

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-bgppls-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 Workgroup
Internet Draft

Intended status: Standards Track

J. Rabadan, Ed.
W. Henderickx
S. Palislaovic
Nokia

A. Isaac
J. Drake
W. Lin
Juniper

A. Sajassi
Cisco

Expires: March 17, 2017

September 13, 2016

IP Prefix Advertisement in EVPN
draft-ietf-bess-evpn-prefix-advertisement-03

Abstract

EVPN provides a flexible control plane that allows intra-subnet connectivity in an IP/MPLS and/or an NVO-based network. In NVO networks, there is also a need for a dynamic and efficient inter-subnet connectivity across Tenant Systems and End Devices that can be physical or virtual and may not support their own routing protocols. This document defines a new EVPN route type for the advertisement of IP Prefixes and explains some use-case examples where this new route-type is used.

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 March 17, 2017.

Copyright Notice

Copyright (c) 2016 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. Terminology	3
2. Introduction and problem statement	4
2.1 Inter-subnet connectivity requirements in Data Centers	4
2.2 The requirement for a new EVPN route type	7
3. The BGP EVPN IP Prefix route	8
3.1 IP Prefix Route encoding	9
4. Benefits of using the EVPN IP Prefix route	11
5. IP Prefix overlay index use-cases	12
5.1 TS IP address overlay index use-case	12
5.2 Floating IP overlay index use-case	15
5.3 ESI overlay index ("Bump in the wire") use-case	16
5.4 IP-VRF-to-IP-VRF model	19
5.4.1 Interface-less IP-VRF-to-IP-VRF model	20
5.4.2 Interface-full IP-VRF-to-IP-VRF with core-facing IRB	23
5.4.3 Interface-full IP-VRF-to-IP-VRF with unnumbered core-facing IRB	25
6. Conclusions	28
7. Conventions used in this document	29
8. Security Considerations	29
9. IANA Considerations	29
10. References	30
10.1 Normative References	30
10.2 Informative References	30

11. Acknowledgments 30
 12. Contributors 30
 13. Authors' Addresses 30

1. Terminology

GW IP: Gateway IP Address

IPL: IP address length

IRB: Integrated Routing and Bridging interface

ML: MAC address length

NVE: Network Virtualization Edge

TS: Tenant System

VA: Virtual Appliance

RT-2: EVPN route type 2, i.e. MAC/IP advertisement route

RT-5: EVPN route type 5, i.e. IP Prefix route

AC: Attachment Circuit

Overlay index: object used in the IP Prefix route, as described in this document. It can be an IP address in the tenant space or an ESI, and identifies a pointer yielded by the IP route lookup at the routing context importing the route. An overlay index always needs a recursive route resolution on the NVE receiving the IP Prefix route, so that the NVE knows to which egress NVE it needs to forward the packets.

Underlay next-hop: IP address sent by BGP along with any EVPN route, i.e. BGP next-hop. It identifies the NVE sending the route and it is used at the receiving NVE as the VXLAN destination VTEP or NVGRE destination end-point.

Ethernet NVO tunnel: it refers to Network Virtualization Overlay tunnels with Ethernet payload. Examples of this type of tunnels are VXLAN or nvGRE.

IP NVO tunnel: it refers to Network Virtualization Overlay tunnels

with IP payload (no MAC header in the payload). Examples of IP NVO tunnels are VXLAN GPE or MPLSoGRE (both with IP payload).

2. Introduction and problem statement

Inter-subnet connectivity is required for certain tenants within the Data Center. [EVPN-INTERSUBNET] defines some fairly common inter-subnet forwarding scenarios where TSes can exchange packets with TSes located in remote subnets. In order to meet this requirement, [EVPN-INTERSUBNET] describes how MAC/IPs encoded in TS RT-2 routes are not only used to populate MAC-VRF and overlay ARP tables, but also IP-VRF tables with the encoded TS host routes (/32 or /128). In some cases, EVPN may advertise IP Prefixes and therefore provide aggregation in the IP-VRF tables, as opposed to program individual host routes. This document complements the scenarios described in [EVPN-INTERSUBNET] and defines how EVPN may be used to advertise IP Prefixes.

Section 2.1 describes the inter-subnet connectivity requirements in Data Centers. Section 2.2 explains why a new EVPN route type is required for IP Prefix advertisements. Once the need for a new EVPN route type is justified, sections 3, 4 and 5 will describe this route type and how it is used in some specific use cases.

2.1 Inter-subnet connectivity requirements in Data Centers

[RFC7432] is used as the control plane for a Network Virtualization Overlay (NVO3) solution in Data Centers (DC), where Network Virtualization Edge (NVE) devices can be located in Hypervisors or TORs, as described in [EVPN-OVERLAY].

If we use the term Tenant System (TS) to designate a physical or virtual system identified by MAC and IP addresses, and connected to an EVPN instance, the following considerations apply:

- o The Tenant Systems may be Virtual Machines (VMs) that generate traffic from their own MAC and IP.
- o The Tenant Systems may be Virtual Appliance entities (VAs) that forward traffic to/from IP addresses of different End Devices seating behind them.
 - o These VAs can be firewalls, load balancers, NAT devices, other appliances or virtual gateways with virtual routing instances.
 - o These VAs do not have their own routing protocols and hence rely on the EVPN NVEs to advertise the routes on their behalf.

- o In all these cases, the VA will forward traffic to the Data Center using its own source MAC but the source IP will be the one associated to the End Device seating behind or a translated IP address (part of a public NAT pool) if the VA is performing NAT.
- o Note that the same IP address could exist behind two of these TS. One example of this would be certain appliance resiliency mechanisms, where a virtual IP or floating IP can be owned by one of the two VAs running the resiliency protocol (the master VA). VRRP is one particular example of this. Another example is multi-homed subnets, i.e. the same subnet is connected to two VAs.
- o Although these VAs provide IP connectivity to VMs and subnets behind them, they do not always have their own IP interface connected to the EVPN NVE, e.g. layer-2 firewalls are examples of VAs not supporting IP interfaces.

The following figure illustrates some of the examples described above.

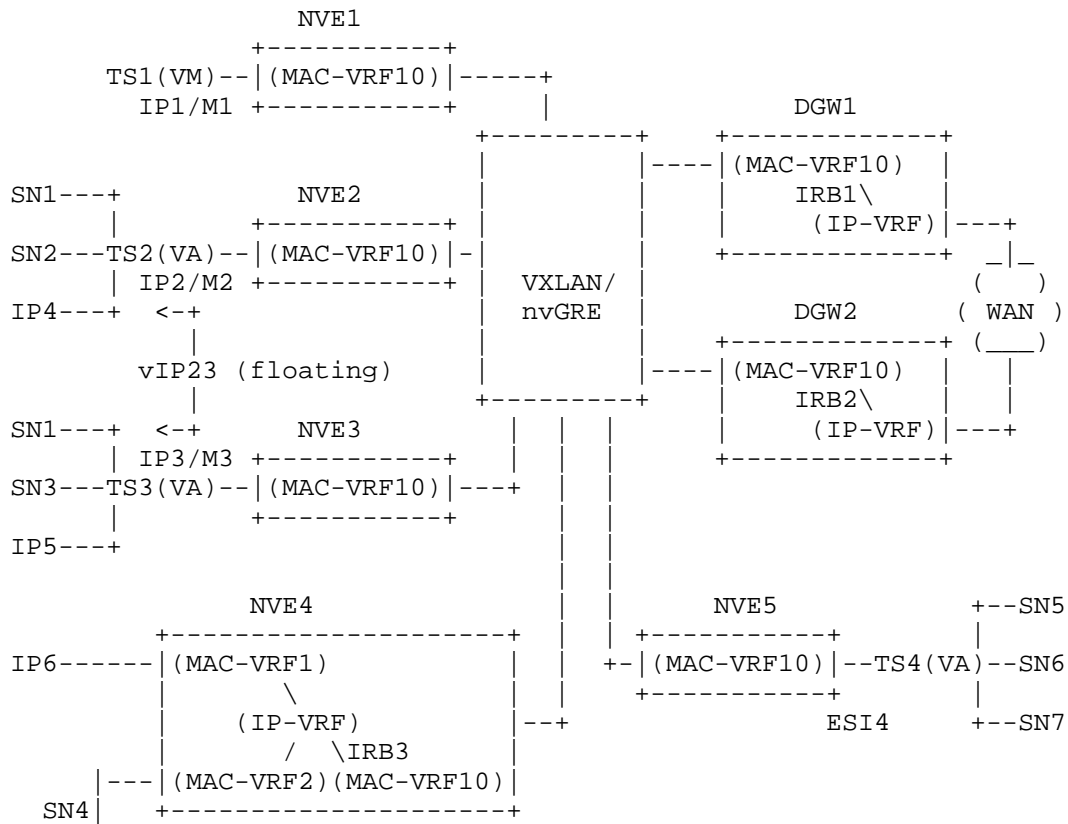


Figure 1 DC inter-subnet use-cases

Where:

NVE1, NVE2, NVE3, NVE4, NVE5, DGW1 and DGW2 share the same EVI for a particular tenant. EVI-10 is comprised of the collection of MAC-VRF10 instances defined in all the NVEs. All the hosts connected to EVI-10 belong to the same IP subnet. The hosts connected to EVI-10 are listed below:

- o TS1 is a VM that generates/receives traffic from/to IP1, where IP1 belongs to the EVI-10 subnet.
- o TS2 and TS3 are Virtual Appliances (VA) that generate/receive traffic from/to the subnets and hosts seating behind them (SN1, SN2, SN3, IP4 and IP5). Their IP addresses (IP2 and IP3) belong to the EVI-10 subnet and they can also generate/receive traffic. When these VAs receive packets destined to their own MAC addresses (M2 and M3) they will route the packets to the

proper subnet or host. These VAs do not support routing protocols to advertise the subnets connected to them and can move to a different server and NVE when the Cloud Management System decides to do so. These VAs may also support redundancy mechanisms for some subnets, similar to VRRP, where a floating IP is owned by the master VA and only the master VA forwards traffic to a given subnet. E.g.: vIP23 in figure 1 is a floating IP that can be owned by TS2 or TS3 depending on who the master is. Only the master will forward traffic to SN1.

- o Integrated Routing and Bridging interfaces IRB1, IRB2 and IRB3 have their own IP addresses that belong to the EVI-10 subnet too. These IRB interfaces connect the EVI-10 subnet to Virtual Routing and Forwarding (IP-VRF) instances that can route the traffic to other connected subnets for the same tenant (within the DC or at the other end of the WAN).
- o TS4 is a layer-2 VA that provides connectivity to subnets SN5, SN6 and SN7, but does not have an IP address itself in the EVI-10. TS4 is connected to a physical port on NVE5 assigned to Ethernet Segment Identifier 4.

All the above DC use cases require inter-subnet forwarding and therefore the individual host routes and subnets:

- a) MUST be advertised from the NVEs (since VAs and VMs do not run routing protocols) and
- b) MAY be associated to an overlay index that can be a VA IP address, a floating IP address or an ESI.

2.2 The requirement for a new EVPN route type

[RFC7432] defines a MAC/IP route (also referred as RT-2) where a MAC address can be advertised together with an IP address length (IPL) and IP address (IP). While a variable IPL might have been used to indicate the presence of an IP prefix in a route type 2, there are several specific use cases in which using this route type to deliver IP Prefixes is not suitable.

One example of such use cases is the "floating IP" example described in section 2.1. In this example we need to decouple the advertisement of the prefixes from the advertisement of the floating IP (vIP23 in figure 1) and MAC associated to it, otherwise the solution gets highly inefficient and does not scale.

E.g.: if we are advertising 1k prefixes from M2 (using RT-2) and the floating IP owner changes from M2 to M3, we would need to withdraw 1k

routes from M2 and re-advertise 1k routes from M3. However if we use a separate route type, we can advertise the 1k routes associated to the floating IP address (vIP23) and only one RT-2 for advertising the ownership of the floating IP, i.e. vIP23 and M2 in the route type 2. When the floating IP owner changes from M2 to M3, a single RT-2 withdraw/update is required to indicate the change. The remote DGW will not change any of the 1k prefixes associated to vIP23, but will only update the ARP resolution entry for vIP23 (now pointing at M3).

Other reasons to decouple the IP Prefix advertisement from the MAC/IP route are listed below:

- o Clean identification, operation of troubleshooting of IP Prefixes, not subject to interpretation and independent of the IPL and the IP value. E.g.: a default IP route 0.0.0.0/0 must always be easily and clearly distinguished from the absence of IP information.
- o MAC address information must not be compared by BGP when selecting two IP Prefix routes. If IP Prefixes were to be advertised using MAC/IP routes, the MAC information would always be present and part of the route key.
- o IP Prefix routes must not be subject to MAC/IP route procedures such as MAC mobility or aliasing. Prefixes advertised from two different ESIs do not mean mobility; MACs advertised from two different ESIs do mean mobility. Similarly load balancing for IP prefixes is achieved through IP mechanisms such as ECMP, and not through MAC route mechanisms such as aliasing.
- o NVEs that do not require processing IP Prefixes must have an easy way to identify an update with an IP Prefix and ignore it, rather than processing the MAC/IP route to find out only later that it carries a Prefix that must be ignored.

The following sections describe how EVPN is extended with a new route type for the advertisement of IP prefixes and how this route is used to address the current and future inter-subnet connectivity requirements existing in the Data Center.

3. The BGP EVPN IP Prefix route

The current BGP EVPN NLRI as defined in [RFC7432] is shown below:

```
+-----+
|   Route Type (1 octet)   |
+-----+
|   Length (1 octet)     |
+-----+
| Route Type specific (variable) |
+-----+
```

Where the route type field can contain one of the following specific values:

- + 1 - Ethernet Auto-Discovery (A-D) route
- + 2 - MAC/IP advertisement route
- + 3 - Inclusive Multicast Route
- + 4 - Ethernet Segment Route

This document defines an additional route type that will be used for the advertisement of IP Prefixes:

- + 5 - IP Prefix 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.

The detailed encoding of this route and associated procedures are described in the following sections.

3.1 IP Prefix Route encoding

An IP Prefix advertisement route NLRI consists of the following fields:

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

Where:

- o RD, Ethernet Tag ID and MPLS Label fields will be used as defined in [RFC7432] and [EVPN-OVERLAY].
- o The Ethernet Segment Identifier will be a non-zero 10-byte identifier if the ESI is used as an overlay index. It will be zero otherwise.
- o The IP Prefix Length can be set to a value between 0 and 32 (bits) for ipv4 and between 0 and 128 for ipv6.
- o The IP Prefix will be a 32 or 128-bit field (ipv4 or ipv6).
- o The GW IP (Gateway IP Address) will be a 32 or 128-bit field (ipv4 or ipv6), and will encode an overlay IP index for the IP Prefixes. The GW IP field SHOULD be zero if it is not used as an overlay index.
- o The MPLS Label field is encoded as 3 octets, where the high-order 20 bits contain the label value. The value SHOULD be null when the IP Prefix route is used for a recursive lookup resolution.
- o The total route length will indicate the type of prefix (ipv4 or ipv6) and the type of GW IP address (ipv4 or ipv6). Note that the IP Prefix + the GW IP should have a length of either 64 or 256 bits, but never 160 bits (ipv4 and ipv6 mixed values are not allowed).

The Eth-Tag ID, IP Prefix Length and IP Prefix will be part of the route key used by BGP to compare routes. The rest of the fields will

not be part of the route key.

The route will contain a single overlay index at most, i.e. if the ESI field is different from zero, the GW IP field will be zero, and vice versa. The following table shows the different inter-subnet use-cases described in this document and the corresponding coding of the overlay index in the route type 5 (RT-5). The IP-VRF-to-IP-VRF or IRB forwarding on NVEs case is a special use-case, where there may be no need for overlay index, since the actual next-hop is given by the BGP next-hop. When an overlay index is present in the RT-5, the receiving NVE will need to perform a recursive route resolution to find out to which egress NVE to forward the packets.

Use-case	Overlay Index in the RT-5 BGP update
TS IP address	Overlay GW IP Address
Floating IP address	Overlay GW IP Address
"Bump in the wire"	ESI
IP-VRF-to-IP-VRF	Overlay GW IP, MAC or N/A

4. Benefits of using the EVPN IP Prefix route

This section clarifies the different functions accomplished by the EVPN RT-2 and RT-5 routes, and provides a list of benefits derived from using a separate route type for the advertisement of IP Prefixes in EVPN.

[RFC7432] describes the content of the BGP EVPN RT-2 specific NLRI, i.e. MAC/IP Advertisement Route, where the IP address length (IPL) and IP address (IP) of a specific advertised MAC are encoded. The subject of the MAC advertisement route is the MAC address (M) and MAC address length (ML) encoded in the route. The MAC mobility and other procedures are defined around that MAC address. The IP address information carries the host IP address required for the ARP resolution of the MAC according to [RFC7432] and the host route to be programmed in the IP-VRF [EVPN-INTERSUBNET].

The BGP EVPN route type 5 defined in this document, i.e. IP Prefix Advertisement route, decouples the advertisement of IP prefixes from the advertisement of any MAC address related to it. This brings some major benefits to NVO-based networks where certain inter-subnet forwarding scenarios are required. Some of those benefits are:

a) Upon receiving a route type 2 or type 5, an egress NVE can easily

distinguish MACs and IPs from IP Prefixes. E.g. an IP prefix with IPL=32 being advertised from two different ingress NVEs (as RT-5) can be identified as such and be imported in the designated routing context as two ECMP routes, as opposed to two MACs competing for the same IP.

- b) Similarly, upon receiving a route, an ingress NVE not supporting processing of IP Prefixes can easily ignore the update, based on the route type.
- c) A MAC route includes the ML, M, IPL and IP in the route key that is used by BGP to compare routes, whereas for IP Prefix routes, only IPL and IP (as well as Ethernet Tag ID) are part of the route key. Advertised IP Prefixes are imported into the designated routing context, where there is no MAC information associated to IP routes. In the example illustrated in figure 1, subnet SN1 should be advertised by NVE2 and NVE3 and interpreted by DGW1 as the same route coming from two different next-hops, regardless of the MAC address associated to TS2 or TS3. This is easily accomplished in the RT-5 by including only the IP information in the route key.
- d) By decoupling the MAC from the IP Prefix advertisement procedures, we can leave the IP Prefix advertisements out of the MAC mobility procedures defined in [RFC7432] for MACs. In addition, this allows us to have an indirection mechanism for IP Prefixes advertised from a MAC/IP that can move between hypervisors. E.g. if there are 1,000 prefixes seating behind TS2 (figure 1), NVE2 will advertise all those prefixes in RT-5 routes associated to the overlay index IP2. Should TS2 move to a different NVE, a single MAC/IP advertisement route withdraw for the M2/IP2 route from NVE2 will invalidate the 1,000 prefixes, as opposed to have to wait for each individual prefix to be withdrawn. This may be easily accomplished by using IP Prefix routes that are not tied to a MAC address, and use a different MAC/IP route to advertise the location and resolution of the overlay index to a MAC address.

5. IP Prefix overlay index use-cases

The IP Prefix route can use a GW IP or an ESI as an overlay index as well as no overlay index whatsoever. This section describes some use-cases for these index types.

5.1 TS IP address overlay index use-case

The following figure illustrates an example of inter-subnet forwarding for subnets seating behind Virtual Appliances (on TS2 and TS3).

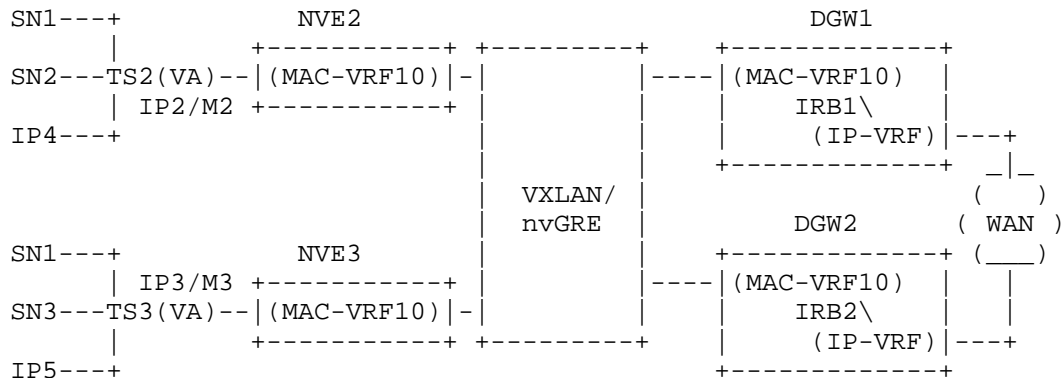


Figure 2 TS IP address use-case

An example of inter-subnet forwarding between subnet SN1/24 and a subnet seating in the WAN is described below. NVE2, NVE3, DGW1 and DGW2 are running BGP EVPN. TS2 and TS3 do not support routing protocols, only a static route to forward the traffic to the WAN.

- (1) NVE2 advertises the following BGP routes on behalf of TS2:
 - o Route type 2 (MAC/IP route) containing: ML=48, M=M2, IPL=32, IP=IP2 and [RFC5512] BGP Encapsulation Extended Community with the corresponding Tunnel-type.
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=0, GW IP address=IP2.
- (2) NVE3 advertises the following BGP routes on behalf of TS3:
 - o Route type 2 (MAC/IP route) containing: ML=48, M=M3, IPL=32, IP=IP3 (and BGP Encapsulation Extended Community).
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=0, GW IP address=IP3.
- (3) DGW1 and DGW2 import both received routes based on the route-targets:
 - o Based on the MAC-VRF10 route-target in DGW1 and DGW2, the MAC/IP route is imported and M2 is added to the MAC-VRF10 along with its corresponding tunnel information. For instance, if VXLAN is used, the VTEP will be derived from the MAC/IP route BGP next-hop (underlay next-hop) and VNI from the MPLS Label1 field. IP2 - M2 is added to the ARP table.

- o Based on the MAC-VRF10 route-target in DGW1 and DGW2, the IP Prefix route is also imported and SN1/24 is added to the IP-VRF with overlay index IP2 pointing at the local MAC-VRF10. Should ECMP be enabled in the IP-VRF, SN1/24 would also be added to the routing table with overlay index IP3.
- (4) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:
- o A destination IP lookup is performed on the DGW1 IP-VRF routing table and overlay index=IP2 is found. Since IP2 is an overlay index a recursive route resolution is required for IP2.
 - o IP2 is resolved to M2 in the ARP table, and M2 is resolved to the tunnel information given by the MAC-VRF FIB (e.g. remote VTEP and VNI for the VXLAN case).
 - o The IP packet destined to IPx is encapsulated with:
 - . Source inner MAC = IRB1 MAC.
 - . Destination inner MAC = M2.
 - . Tunnel information provided by the MAC-VRF (VNI, VTEP IPs and MACs for the VXLAN case).
- (5) When the packet arrives at NVE2:
- o Based on the tunnel information (VNI for the VXLAN case), the MAC-VRF10 context is identified for a MAC lookup.
 - o Encapsulation is stripped-off and based on a MAC lookup (assuming MAC forwarding on the egress NVE), the packet is forwarded to TS2, where it will be properly routed.
- (6) Should TS2 move from NVE2 to NVE3, MAC Mobility procedures will be applied to the MAC route IP2/M2, as defined in [RFC7432]. Route type 5 prefixes are not subject to MAC mobility procedures, hence no changes in the DGW IP-VRF routing table will occur for TS2 mobility, i.e. all the prefixes will still be pointing at IP2 as overlay index. There is an indirection for e.g. SN1/24, which still points at overlay index IP2 in the routing table, but IP2 will be simply resolved to a different tunnel, based on the outcome of the MAC mobility procedures for the MAC/IP route IP2/M2.

Note that in the opposite direction, TS2 will send traffic based on

its static-route next-hop information (IRB1 and/or IRB2), and regular EVPN procedures will be applied.

5.2 Floating IP overlay index use-case

Sometimes Tenant Systems (TS) work in active/standby mode where an upstream floating IP - owned by the active TS - is used as the overlay index to get to some subnets behind. This redundancy mode, already introduced in section 2.1 and 2.2, is illustrated in Figure 3.

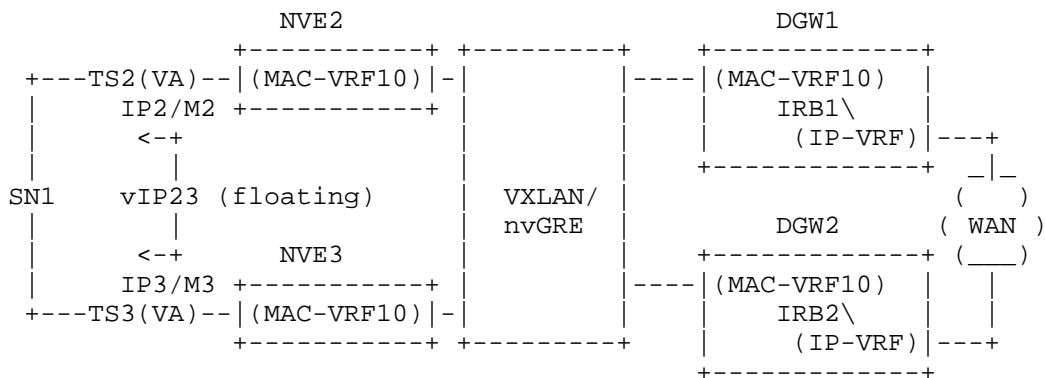


Figure 3 Floating IP overlay index for redundant TS

In this example, assuming TS2 is the active TS and owns IP23:

- (1) NVE2 advertises the following BGP routes for TS2:
 - o Route type 2 (MAC/IP route) containing: ML=48, M=M2, IPL=32, IP=IP23 (and BGP Encapsulation Extended Community).
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=0, GW IP address=IP23.
- (2) NVE3 advertises the following BGP routes for TS3:
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=0, GW IP address=IP23.
- (3) DGW1 and DGW2 import both received routes based on the route-target:
 - o M2 is added to the MAC-VRF10 FIB along with its corresponding tunnel information. For the VXLAN use case, the VTEP will be derived from the MAC/IP route BGP next-hop and VNI from the

VNI/VSID field. IP23 - M2 is added to the ARP table.

- o SN1/24 is added to the IP-VRF in DGW1 and DGW2 with overlay index IP23 pointing at the local MAC-VRF10.
- (4) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:
- o A destination IP lookup is performed on the DGW1 IP-VRF routing table and overlay index=IP23 is found. Since IP23 is an overlay index, a recursive route resolution for IP23 is required.
 - o IP23 is resolved to M2 in the ARP table, and M2 is resolved to the tunnel information given by the MAC-VRF (remote VTEP and VNI for the VXLAN case).
 - o The IP packet destined to IPx is encapsulated with:
 - . Source inner MAC = IRB1 MAC.
 - . Destination inner MAC = M2.
 - . Tunnel information provided by the MAC-VRF FIB (VNI, VTEP IPs and MACs for the VXLAN case).
- (5) When the packet arrives at NVE2:
- o Based on the tunnel information (VNI for the VXLAN case), the MAC-VRF10 context is identified for a MAC lookup.
 - o Encapsulation is stripped-off and based on a MAC lookup (assuming MAC forwarding on the egress NVE), the packet is forwarded to TS2, where it will be properly routed.
- (6) When the redundancy protocol running between TS2 and TS3 appoints TS3 as the new active TS for SN1, TS3 will now own the floating IP23 and will signal this new ownership (GARP message or similar). Upon receiving the new owner's notification, NVE3 will issue a route type 2 for M3-IP23. DGW1 and DGW2 will update their ARP tables with the new MAC resolving the floating IP. No changes are carried out in the IP-VRF routing table.

5.3 ESI overlay index ("Bump in the wire") use-case

Figure 5 illustrates an example of inter-subnet forwarding for an IP Prefix route that carries a subnet SN1 and uses an ESI as an overlay index (ESI23). In this use-case, TS2 and TS3 are layer-2 VA devices

without any IP address that can be included as an overlay index in the GW IP field of the IP Prefix route. Their MAC addresses are M2 and M3 respectively and are connected to EVI-10. Note that IRB1 and IRB2 (in DGW1 and DGW2 respectively) have IP addresses in a subnet different than SN1.

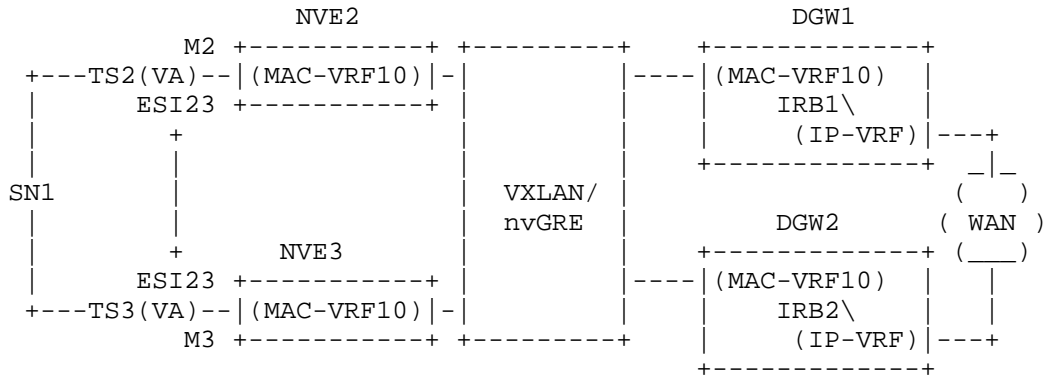


Figure 5 ESI overlay index use-case

Since neither TS2 nor TS3 can run any routing protocol and have no IP address assigned, an ESI, i.e. ESI23, will be provisioned on the attachment ports of NVE2 and NVE3. This model supports VA redundancy in a similar way as the one described in section 5.2 for the floating IP overlay index use-case, only using the EVPN Ethernet A-D route instead of the MAC advertisement route to advertise the location of the overlay index. The procedure is explained below:

- (1) NVE2 advertises the following BGP routes for TS2:
 - o Route type 1 (Ethernet A-D route for EVI-10) containing: ESI=ESI23 and the corresponding tunnel information (VNI/VSID field), as well as the BGP Encapsulation Extended Community as per [EVPN-OVERLAY].
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=ESI23, GW IP address=0. The Router's MAC Extended Community defined in [EVPN-INTERSUBNET] is added and carries the MAC address (M2) associated to the TS behind which SN1 seats.
- (2) NVE3 advertises the following BGP routes for TS3:
 - o Route type 1 (Ethernet A-D route for EVI-10) containing: ESI=ESI23 and the corresponding tunnel information (VNI/VSID

field), as well as the BGP Encapsulation Extended Community.

- o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=23, GW IP address=0. The Router's MAC Extended Community is added and carries the MAC address (M3) associated to the TS behind which SN1 seats.
- (3) DGW1 and DGW2 import the received routes based on the route-target:
- o The tunnel information to get to ESI23 is installed in DGW1 and DGW2. For the VXLAN use case, the VTEP will be derived from the Ethernet A-D route BGP next-hop and VNI from the VNI/VSID field (see [EVPN-OVERLAY]).
 - o SN1/24 is added to the IP-VRF in DGW1 and DGW2 with overlay index ESI23.
- (4) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:
- o A destination IP lookup is performed on the DGW1 IP-VRF routing table and overlay index=ESI23 is found. Since ESI23 is an overlay index, a recursive route resolution is required to find the egress NVE where ESI23 resides.
 - o The IP packet destined to IPx is encapsulated with:
 - . Source inner MAC = IRB1 MAC.
 - . Destination inner MAC = M2 (this MAC will be obtained from the Router's MAC Extended Community received along with the RT-5 for SN1).
 - . Tunnel information for the NVO tunnel is provided by the Ethernet A-D route per-EVI for ESI23 (VNI and VTEP IP for the VXLAN case).
- (5) When the packet arrives at NVE2:
- o Based on the tunnel demultiplexer information (VNI for the VXLAN case), the MAC-VRF10 context is identified for a MAC lookup (assuming MAC disposition model) or the VNI MAY directly identify the egress interface (for a label or VNI disposition model).
 - o Encapsulation is stripped-off and based on a MAC lookup (assuming MAC forwarding on the egress NVE) or a VNI lookup

(in case of VNI forwarding), the packet is forwarded to TS2, where it will be forwarded to SN1.

- (6) If the redundancy protocol running between TS2 and TS3 follows an active/standby model and there is a failure, appointing TS3 as the new active TS for SN1, TS3 will now own the connectivity to SN1 and will signal this new ownership. Upon receiving the new owner's notification, NVE3's AC will become active and issue a route type 1 for ESI23, whereas NVE2 will withdraw its Ethernet A-D route for ESI23. DGW1 and DGW2 will update their tunnel information to resolve ESI23. The destination inner MAC will be changed to M3.

5.4 IP-VRF-to-IP-VRF model

This use-case is similar to the scenario described in "IRB forwarding on NVEs for Tenant Systems" in [EVPN-INTERSUBNET], however the new requirement here is the advertisement of IP Prefixes as opposed to only host routes.

In the examples described in sections 5.1, 5.2 and 5.3, the MAC-VRF instance can connect IRB interfaces and any other Tenant Systems connected to it. EVPN provides connectivity for:

1. Traffic destined to the IRB IP interfaces as well as
2. Traffic destined to IP subnets seating behind the TS, e.g. SN1 or SN2.

In order to provide connectivity for (1), MAC/IP routes (RT-2) are needed so that IRB MACs and IPs can be distributed. Connectivity type (2) is accomplished by the exchange of IP Prefix routes (RT-5) for IPs and subnets seating behind certain overlay indexes, e.g. GW IP or ESI.

In some cases, IP Prefix routes may be advertised for subnets and IPs seating behind an IRB. We refer to this use-case as the "IP-VRF-to-IP-VRF" model.

[EVPN-INTERSUBNET] defines an asymmetric IRB model and a symmetric IRB model, based on the required lookups at the ingress and egress NVE: the asymmetric model requires an ip-lookup and a mac-lookup at the ingress NVE, whereas only a mac-lookup is needed at the egress NVE; the symmetric model requires ip and mac lookups at both, ingress and egress NVE. From that perspective, the IP-VRF-to-IP-VRF use-case described in this section is a symmetric IRB model. Note that in an IP-VRF-to-IP-VRF scenario, a PE may not be configured with any MAC-VRF for a given tenant, in which case it will only be doing IP

lookups and forwarding for that tenant.

Based on the way the IP-VRFs are interconnected, there are three different IP-VRF-to-IP-VRF scenarios identified and described in this document:

- 1) Interface-less model
- 2) Interface-full with core-facing IRB model
- 3) Interface-full with unnumbered core-facing IRB model

5.4.1 Interface-less IP-VRF-to-IP-VRF model

Figure 6 will be used for the description of this model.

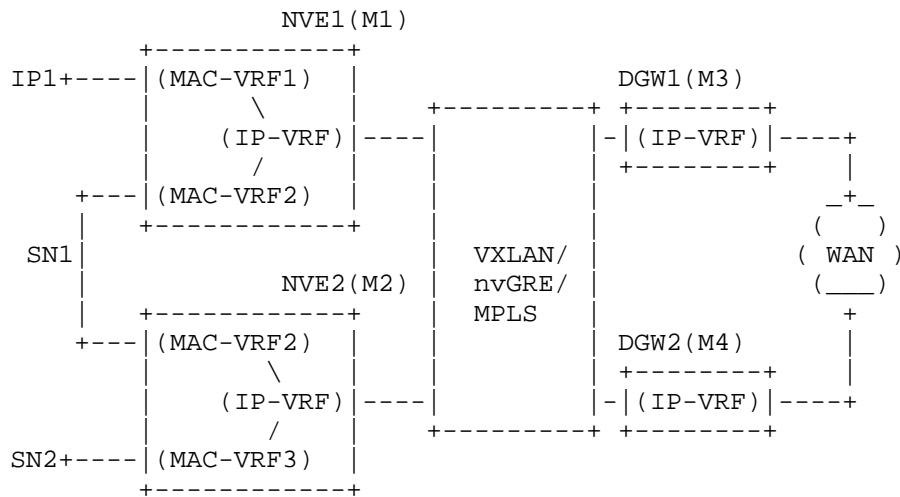


Figure 6 Interface-less IP-VRF-to-IP-VRF model

In this case, the requirements are the following:

- a) The NVEs and DGWs must provide connectivity between hosts in SN1, SN2, IP1 and hosts seating at the other end of the WAN.
- b) The IP-VRF instances in the NVE/DGWs are directly connected through NVO tunnels, and no IRBs and/or MAC-VRF instances are defined at the core.

- c) The solution must provide layer-3 connectivity among the IP-VRFs for Ethernet NVO tunnels, for instance, VXLAN or nvGRE.
- d) The solution may provide layer-3 connectivity among the IP-VRFs for IP NVO tunnels, for example, VXLAN GPE (with IP payload).

In order to meet the above requirements, the EVPN route type 5 will be used to advertise the IP Prefixes, along with the Router's MAC Extended Community as defined in [EVPN-INTERSUBNET] if the advertising NVE/DGW uses Ethernet NVO tunnels. Each NVE/DGW will advertise an RT-5 for each of its prefixes with the following fields:

- o RD as per [RFC7432].
- o Eth-Tag ID=0 assuming VLAN-based service.
- o IP address length and IP address, as explained in the previous sections.
- o GW IP address= SHOULD be set to 0.
- o ESI=0
- o MPLS label or VNI corresponding to the IP-VRF.

Each RT-5 will be sent with a route-target identifying the tenant (IP-VRF) and two BGP extended communities:

- o The first one is the BGP Encapsulation Extended Community, as per [RFC5512], identifying the tunnel type.
- o The second one is the Router's MAC Extended Community as per [EVPN-INTERSUBNET] containing the MAC address associated to the NVE advertising the route. This MAC address identifies the NVE/DGW and MAY be re-used for all the IP-VRFs in the NVE. The Router's MAC Extended Community MUST be sent if the route is associated to an Ethernet NVO tunnel, for instance, VXLAN. If the route is associated to an IP NVO tunnel, for instance VXLAN GPE with IP payload, the Router's MAC Extended Community SHOULD NOT be sent.

The following example illustrates the procedure to advertise and forward packets to SN1/24 (ipv4 prefix advertised from NVE1) for VXLAN tunnels:

(1) NVE1 advertises the following BGP route:

- o Route type 5 (IP Prefix route) containing:

- . IPL=24, IP=SN1, VNI=10.
- . GW IP= SHOULD be set to 0.
- . [RFC5512] BGP Encapsulation Extended Community with Tunnel-type=VXLAN.
- . Router's MAC Extended Community that contains M1.
- . Route-target identifying the tenant (IP-VRF).

(2) DGW1 imports the received routes from NVE1:

- o DGW1 installs SN1/24 in the IP-VRF identified by the RT-5 route-target.
- o Since GW IP=0 and the VNI is a valid value, DGW1 will use the VNI and next-hop of the RT-5, as well as the MAC address conveyed in the Router's MAC Extended Community (as inner destination MAC address) to encapsulate the routed IP packets.

(3) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:

- o A destination IP lookup is performed on the DGW1 IP-VRF routing table. The lookup yields SN1/24.
- o Since the RT-5 for SN1/24 had a GW IP=0 and a valid VNI and next-hop (used as destination VTEP), DGW1 will not need a recursive lookup to resolve the route.
- o The IP packet destined to IPx is encapsulated with: Source inner MAC = DGW1 MAC, Destination inner MAC = M1, Source outer IP (source VTEP) = DGW1 IP, Destination outer IP (destination VTEP) = NVE1 IP.

(4) When the packet arrives at NVE1:

- o NVE1 will identify the IP-VRF for an IP-lookup based on the VNI.
- o An IP lookup is performed in the routing context, where SN1 turns out to be a local subnet associated to MAC-VRF2. A subsequent lookup in the ARP table and the MAC-VRF FIB will provide the forwarding information for the packet in MAC-VRF2.

The implementation of this Interface-less model is REQUIRED.

5.4.2 Interface-full IP-VRF-to-IP-VRF with core-facing IRB

Figure 7 will be used for the description of this model.

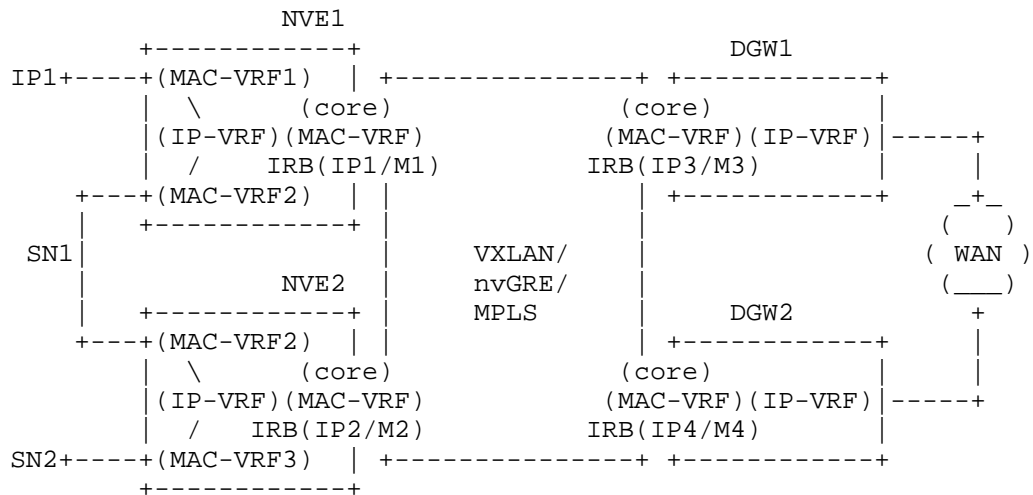


Figure 7 Interface-full with core-facing IRB model

In this model, the requirements are the following:

- a) As in section 5.4.1, the NVEs and DGWs must provide connectivity between hosts in SN1, SN2, IP1 and hosts seating at the other end of the WAN.
- b) However, the NVE/DGWs are now connected through Ethernet NVO tunnels terminated in core-MAC-VRF instances. The IP-VRFs use IRB interfaces for their connectivity to the core MAC-VRFs.
- c) Each core-facing IRB has an IP and a MAC address, where the IP address must be reachable from other NVEs or DGWs.
- d) The core EVI is composed of the NVE/DGW MAC-VRFs and may contain other MAC-VRFs without IRB interfaces. Those non-IRB MAC-VRFs will typically connect TSeS that need layer-3 connectivity to remote subnets.
- e) The solution must provide layer-3 connectivity for Ethernet NVO tunnels, for instance, VXLAN or nvGRE.

EVPN type 5 routes will be used to advertise the IP Prefixes, whereas

EVPN RT-2 routes will advertise the MAC/IP addresses of each core-facing IRB interface. Each NVE/DGW will advertise an RT-5 for each of its prefixes with the following fields:

- o RD as per [RFC7432].
- o Eth-Tag ID=0 assuming VLAN-based service.
- o IP address length and IP address, as explained in the previous sections.
- o GW IP address=IRB-IP (this is the overlay index that will be used for the recursive route resolution).
- o ESI=0
- o MPLS label or VNI corresponding to the IP-VRF. Note that the value SHOULD be zero since the RT-5 route requires a recursive lookup resolution to an RT-2 route. The MPLS label or VNI to be used when forwarding packets will be derived from the RT-2's MPLS Label1 field.

Each RT-5 will be sent with a route-target identifying the tenant (IP-VRF). The Router's MAC Extended Community SHOULD NOT be sent in this case.

The following example illustrates the procedure to advertise and forward packets to SN1/24 (ipv4 prefix advertised from NVE1) for VXLAN tunnels:

(1) NVE1 advertises the following BGP routes:

- o Route type 5 (IP Prefix route) containing:
 - . IPL=24, IP=SN1, VNI= SHOULD be set to 0.
 - . GW IP=IP1 (core-facing IRB's IP)
 - . Route-target identifying the tenant (IP-VRF).
- o Route type 2 (MAC/IP route for the core-facing IRB) containing:
 - . ML=48, M=M1, IPL=32, IP=IP1, VNI=10.
 - . A [RFC5512] BGP Encapsulation Extended Community with Tunnel-type= VXLAN.

. Route-target identifying the tenant. This route-target MAY be the same as the one used with the RT-5.

(2) DGW1 imports the received routes from NVE1:

- o DGW1 installs SN1/24 in the IP-VRF identified by the RT-5 route-target.

. Since GW IP is different from zero, the GW IP (IP1) will be used as the overlay index for the recursive route resolution to the RT-2 carrying IP1.

(3) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:

- o A destination IP lookup is performed on the DGW1 IP-VRF routing table. The lookup yields SN1/24, which is associated to the overlay index IP1. The forwarding information is derived from the RT-2 received for IP1.
- o The IP packet destined to IPx is encapsulated with: Source inner MAC = M3, Destination inner MAC = M1, Source outer IP (source VTEP) = DGW1 IP, Destination outer IP (destination VTEP) = NVE1 IP.

(4) When the packet arrives at NVE1:

- o NVE1 will identify the IP-VRF for an IP-lookup based on the VNI and the inner MAC DA.
- o An IP lookup is performed in the routing context, where SN1 turns out to be a local subnet associated to MAC-VRF2. A subsequent lookup in the ARP table and the MAC-VRF FIB will provide the forwarding information for the packet in MAC-VRF2.

The implementation of the Interface-full with core-facing IRB model is REQUIRED.

5.4.3 Interface-full IP-VRF-to-IP-VRF with unnumbered core-facing IRB

Figure 8 will be used for the description of this model. Note that this model is similar to the one described in section 5.4.2, only without IP addresses on the core-facing IRB interfaces.

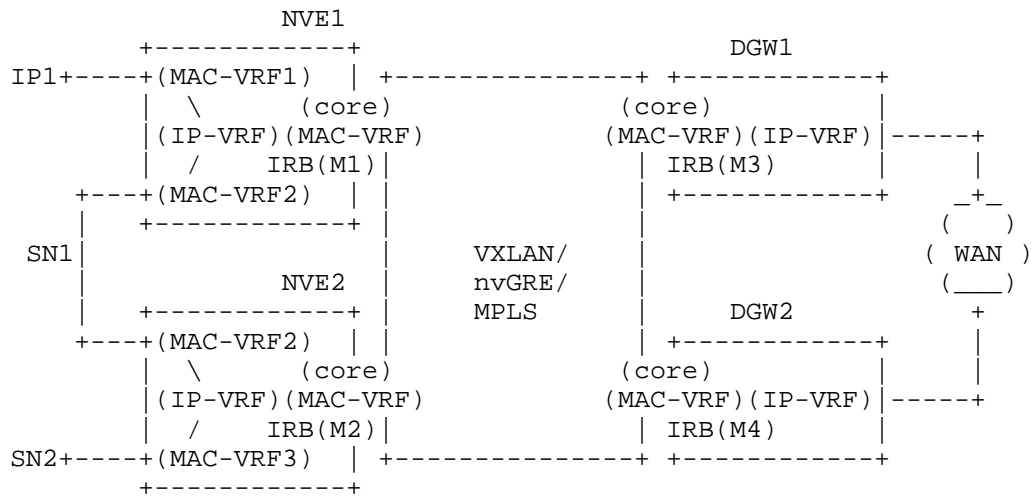


Figure 8 Interface-full with unnumbered core-facing IRB model

In this model, the requirements are the following:

- a) As in section 5.4.1 and 5.4.2, the NVEs and DGWs must provide connectivity between hosts in SN1, SN2, IP1 and hosts seating at the other end of the WAN.
- b) As in section 5.4.2, the NVE/DGWs are connected through Ethernet NVO tunnels terminated in core-MAC-VRF instances. The IP-VRFs use IRB interfaces for their connectivity to the core MAC-VRFs.
- c) However, each core-facing IRB has a MAC address only, and no IP address (that is why the model refers to an 'unnumbered' core-facing IRB). In this model, there is no need to have IP reachability to the core-facing IRB interfaces themselves and there is a requirement to save IP addresses on those interfaces.
- d) As in section 5.4.2, the core EVI is composed of the NVE/DGW MAC-VRFs and may contain other MAC-VRFs.
- e) As in section 5.4.2, the solution must provide layer-3 connectivity for Ethernet NVO tunnels, for instance, VXLAN or nvGRE.

This model will also make use of the RT-5 recursive resolution. EVPN type 5 routes will advertise the IP Prefixes along with the Router's MAC Extended Community used for the recursive lookup, whereas EVPN RT-2 routes will advertise the MAC addresses of each core-facing IRB

interface (this time without an IP). Each NVE/DGW will advertise an RT-5 for each of its prefixes with the following fields:

- o RD as per [RFC7432].
- o Eth-Tag ID=0 assuming VLAN-based service.
- o IP address length and IP address, as explained in the previous sections.
- o GW IP address= SHOULD be set to 0.
- o ESI=0
- o MPLS label or VNI corresponding to the IP-VRF. Note that the value SHOULD be zero since the RT-5 route requires a recursive lookup resolution to an RT-2 route. The MPLS label or VNI to be used when forwarding packets will be derived from the RT-2's MPLS Label1 field.

Each RT-5 will be sent with a route-target identifying the tenant (IP-VRF) and the Router's MAC Extended Community containing the MAC address associated to core-facing IRB interface. This MAC address MAY be re-used for all the IP-VRFs in the NVE.

The following example illustrates the procedure to advertise and forward packets to SN1/24 (ipv4 prefix advertised from NVE1) for VXLAN tunnels:

(1) NVE1 advertises the following BGP routes:

- o Route type 5 (IP Prefix route) containing:
 - . IPL=24, IP=SN1, VNI= SHOULD be set to 0.
 - . GW IP= SHOULD be set to 0.
 - . Router's MAC Extended Community containing M1 (this will be used for the recursive lookup to a RT-2).
 - . Route-target identifying the tenant (IP-VRF).
- o Route type 2 (MAC route for the core-facing IRB) containing:
 - . ML=48, M=M1, IPL=0, VNI=10.
 - . A [RFC5512] BGP Encapsulation Extended Community with Tunnel-type=VXLAN.

. Route-target identifying the tenant. This route-target MAY be the same as the one used with the RT-5.

(2) DGW1 imports the received routes from NVE1:

- o DGW1 installs SN1/24 in the IP-VRF identified by the RT-5 route-target.

- . The MAC contained in the Router's MAC Extended Community sent along with the RT-5 (M1) will be used as the overlay index for the recursive route resolution to the RT-2 carrying M1.

(3) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:

- o A destination IP lookup is performed on the DGW1 IP-VRF routing table. The lookup yields SN1/24, which is associated to the overlay index M1. The forwarding information is derived from the RT-2 received for M1.

- o The IP packet destined to IPx is encapsulated with: Source inner MAC = M3, Destination inner MAC = M1, Source outer IP (source VTEP) = DGW1 IP, Destination outer IP (destination VTEP) = NVE1 IP.

(4) When the packet arrives at NVE1:

- o NVE1 will identify the IP-VRF for an IP-lookup based on the VNI and the inner MAC DA.

- o An IP lookup is performed in the routing context, where SN1 turns out to be a local subnet associated to MAC-VRF2. A subsequent lookup in the ARP table and the MAC-VRF FIB will provide the forwarding information for the packet in MAC-VRF2.

The implementation of the Interface-full with unnumbered core-facing IRB model is OPTIONAL.

6. Conclusions

An EVPN route (type 5) for the advertisement of IP Prefixes is described in this document. This new route type has a differentiated role from the RT-2 route and addresses all the Data Center (or NVO-based networks in general) inter-subnet connectivity scenarios in which an IP Prefix advertisement is required. Using this new RT-5, an IP Prefix may be advertised along with an overlay index that can be a

GW IP address, a MAC or an ESI, or without an overlay index, in which case the BGP next-hop will point at the egress NVE and the MAC in the Router's MAC Extended Community will provide the inner MAC destination address to be used. As discussed throughout the document, the EVPN RT-2 does not meet the requirements for all the DC use cases, therefore this EVPN route type 5 is required.

The EVPN route type 5 decouples the IP Prefix advertisements from the MAC/IP route advertisements in EVPN, hence:

- a) Allows the clean and clear advertisements of ipv4 or ipv6 prefixes in an NLRI with no MAC addresses in the route key, so that only IP information is used in BGP route comparisons.
- b) Since the route type is different from the MAC/IP Advertisement route, the advertisement of prefixes will be excluded from all the procedures defined for the advertisement of VM MACs, e.g. MAC Mobility or aliasing. As a result of that, the current [RFC7432] procedures do not need to be modified.
- c) Allows a flexible implementation where the prefix can be linked to different types of overlay indexes: overlay IP address, overlay MAC addresses, overlay ESI, underlay IP next-hops, etc.
- d) An EVPN implementation not requiring IP Prefixes can simply discard them by looking at the route type value. An unknown route type MUST be ignored by the receiving NVE/PE.

7. 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].

8. Security Considerations

The security considerations discussed in [RFC7432] apply to this document.

9. IANA Considerations

This document requests the allocation of value 5 in the "EVPN Route Types" registry defined by [RFC7432] and modification of the registry as follows:

Value	Description	Reference
5	IP Prefix route	[this document]

6-255 Unassigned

10. References

10.1 Normative References

[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>>.

[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>>.

10.2 Informative References

[EVPN-OVERLAY] Sajassi-Drake et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-04.txt, work in progress, June, 2016

[EVPN-INTERSUBNET] Sajassi et al., "IP Inter-Subnet Forwarding in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-01.txt, work in progress, October, 2015

11. Acknowledgments

The authors would like to thank Mukul Katiyar for their valuable feedback and contributions. The following people also helped improving this document with their feedback: Tony Przygienda and Thomas Morin.

12. Contributors

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

Senthil Sathappan
Florin Balus

13. Authors' Addresses

Jorge Rabadan (Editor)
Nokia
777 E. Middlefield Road

Mountain View, CA 94043 USA
Email: jorge.rabadan@nokia.com

Wim Henderickx
Nokia
Email: wim.henderickx@nokia.com

Aldrin Isaac
Juniper
Email: aisaac@juniper.net

Senad Palislamovic
Nokia
Email: senad.palislamovic@nokia.com

John E. Drake
Juniper
Email: jdrake@juniper.net

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Wen Lin
Juniper
Email: wlin@juniper.net

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
W. Henderickx
Nokia

J. Drake
W. Lin
Juniper

A. Sajassi
Cisco

Expires: November 19, 2018

May 18, 2018

IP Prefix Advertisement in EVPN
draft-ietf-bess-evpn-prefix-advertisement-11

Abstract

The BGP MPLS-based Ethernet VPN (EVPN) [RFC7432] mechanism provides a flexible control plane that allows intra-subnet connectivity in an MPLS and/or NVO (Network Virtualization Overlay) [RFC7365] network. In some networks, there is also a need for a dynamic and efficient inter-subnet connectivity across Tenant Systems and End Devices that can be physical or virtual and do not necessarily participate in dynamic routing protocols. This document defines a new EVPN route type for the advertisement of IP Prefixes and explains some use-case examples where this new route-type is used.

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 November 19, 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 (<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. Problem Statement	5
2.1 Inter-Subnet Connectivity Requirements in Data Centers	5
2.2 The Need for the EVPN IP Prefix Route	8
3. The BGP EVPN IP Prefix Route	10
3.1 IP Prefix Route Encoding	11
3.2 Overlay Indexes and Recursive Lookup Resolution	13
4. Overlay Index Use-Cases	15
4.1 TS IP Address Overlay Index Use-Case	16
4.2 Floating IP Overlay Index Use-Case	18
4.3 Bump-in-the-Wire Use-Case	20
4.4 IP-VRF-to-IP-VRF Model	23
4.4.1 Interface-less IP-VRF-to-IP-VRF Model	24
4.4.2 Interface-ful IP-VRF-to-IP-VRF with SBD IRB	27
4.4.3 Interface-ful IP-VRF-to-IP-VRF with Unnumbered SBD IRB	30
5. Security Considerations	33
6. IANA Considerations	33
7. References	34
7.1 Normative References	34
7.2 Informative References	34
8. Acknowledgments	35
9. Contributors	35
10. Authors' Addresses	36

1. Introduction

[RFC7365] provides a framework for Data Center (DC) Network Virtualization over Layer 3 and specifies that the Network Virtualization Edge devices (NVEs) must provide layer 2 and layer 3 virtualized network services in multi-tenant DCs. [RFC8365] discusses the use of EVPN as the technology of choice to provide layer 2 or intra-subnet services in these DCs. This document, along with [EVPN-INTERSUBNET], specifies the use of EVPN for layer 3 or inter-subnet connectivity services.

[EVPN-INTERSUBNET] defines some fairly common inter-subnet forwarding scenarios where TSes can exchange packets with TSes located in remote subnets. In order to achieve this, [EVPN-INTERSUBNET] describes how MAC/IPs encoded in TS RT-2 routes are not only used to populate MAC-VRF and overlay ARP tables, but also IP-VRF tables with the encoded TS host routes (/32 or /128). In some cases, EVPN may advertise IP Prefixes and therefore provide aggregation in the IP-VRF tables, as opposed to propagate individual host routes. This document complements the scenarios described in [EVPN-INTERSUBNET] and defines how EVPN may be used to advertise IP Prefixes. Interoperability between EVPN and L3VPN [RFC4364] IP Prefix routes is out of the scope of this document.

Section 2.1 describes the inter-subnet connectivity requirements in Data Centers. Section 2.2 explains why a new EVPN route type is required for IP Prefix advertisements. Sections 3, 4 and 5 will describe this route type and how it is used in some specific use cases.

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.

AC: Attachment Circuit.

ARP: Address Resolution Protocol.

BD: Broadcast Domain. As per [RFC7432], an EVI consists of a single or multiple BDs. In case of VLAN-bundle and VLAN-based service models (see [RFC7432]), a BD is equivalent to an EVI. In case of VLAN-aware bundle service model, an EVI contains multiple BDs. Also, in this document, BD and subnet are equivalent terms.

BD Route Target: refers to the Broadcast Domain assigned Route Target [RFC4364]. In case of VLAN-aware bundle service model, all the BD instances in the MAC-VRF share the same Route Target.

BT: Bridge Table. The instantiation of a BD in a MAC-VRF, as per [RFC7432].

DGW: Data Center Gateway.

Ethernet A-D route: Ethernet Auto-Discovery (A-D) route, as per [RFC7432].

Ethernet NVO tunnel: refers to Network Virtualization Overlay tunnels with Ethernet payload. Examples of this type of tunnels are VXLAN or GENEVE.

EVI: EVPN Instance spanning the NVE/PE devices that are participating on that EVPN, as per [RFC7432].

EVPN: Ethernet Virtual Private Networks, as per [RFC7432].

GRE: Generic Routing Encapsulation.

GW IP: Gateway IP Address.

IPL: IP Prefix Length.

IP NVO tunnel: it refers to Network Virtualization Overlay tunnels with IP payload (no MAC header in the payload).

IP-VRF: A VPN Routing and Forwarding table for IP routes on an NVE/PE. The IP routes could be populated by EVPN and IP-VPN address families. An IP-VRF is also an instantiation of a layer 3 VPN in an NVE/PE.

IRB: Integrated Routing and Bridging interface. It connects an IP-VRF to a BD (or subnet).

MAC-VRF: A Virtual Routing and Forwarding table for Media Access Control (MAC) addresses on an NVE/PE, as per [RFC7432]. A MAC-VRF is also an instantiation of an EVI in an NVE/PE.

ML: MAC address length.

ND: Neighbor Discovery Protocol.

NVE: Network Virtualization Edge.

GENEVE: Generic Network Virtualization Encapsulation, [GENEVE].

NVO: Network Virtualization Overlays.

RT-2: EVPN route type 2, i.e., MAC/IP advertisement route, as defined in [RFC7432].

RT-5: EVPN route type 5, i.e., IP Prefix route. As defined in Section 3.

SBD: Supplementary Broadcast Domain. A BD that does not have any ACs, only IRB interfaces, and it is used to provide connectivity among all the IP-VRFs of the tenant. The SBD is only required in IP-VRF-to-IP-VRF use-cases (see Section 4.4.).

SN: Subnet.

TS: Tenant System.

VA: Virtual Appliance.

VNI: Virtual Network Identifier. As in [RFC8365], the term is used as a representation of a 24-bit NVO instance identifier, with the understanding that VNI will refer to a VXLAN Network Identifier in VXLAN, or Virtual Network Identifier in GENEVE, etc. unless it is stated otherwise.

VTEP: VXLAN Termination End Point, as in [RFC7348].

VXLAN: Virtual Extensible LAN, as in [RFC7348].

This document also assumes familiarity with the terminology of [RFC7432], [RFC8365] and [RFC7365].

2. Problem Statement

This Section describes the inter-subnet connectivity requirements in Data Centers and why a specific route type to advertise IP Prefixes is needed.

2.1 Inter-Subnet Connectivity Requirements in Data Centers

[RFC7432] is used as the control plane for a Network Virtualization Overlay (NVO) solution in Data Centers (DC), where Network Virtualization Edge (NVE) devices can be located in Hypervisors or

Top of Rack switches (ToRs), as described in [RFC8365].

The following considerations apply to Tenant Systems (TS) that are physical or virtual systems identified by MAC and maybe IP addresses and connected to BDs by Attachment Circuits:

- o The Tenant Systems may be Virtual Machines (VMs) that generate traffic from their own MAC and IP.
- o The Tenant Systems may be Virtual Appliance entities (VAs) that forward traffic to/from IP addresses of different End Devices sitting behind them.
 - o These VAs can be firewalls, load balancers, NAT devices, other appliances or virtual gateways with virtual routing instances.
 - o These VAs do not necessarily participate in dynamic routing protocols and hence rely on the EVPN NVEs to advertise the routes on their behalf.
 - o In all these cases, the VA will forward traffic to other TSes using its own source MAC but the source IP will be the one associated to the End Device sitting behind or a translated IP address (part of a public NAT pool) if the VA is performing NAT.
 - o Note that the same IP address and endpoint could exist behind two of these TSes. One example of this would be certain appliance resiliency mechanisms, where a virtual IP or floating IP can be owned by one of the two VAs running the resiliency protocol (the master VA). Virtual Router Redundancy Protocol (VRRP), RFC5798, is one particular example of this. Another example is multi-homed subnets, i.e., the same subnet is connected to two VAs.
 - o Although these VAs provide IP connectivity to VMs and subnets behind them, they do not always have their own IP interface connected to the EVPN NVE, e.g., layer 2 firewalls are examples of VAs not supporting IP interfaces.

Figure 1 illustrates some of the examples described above.

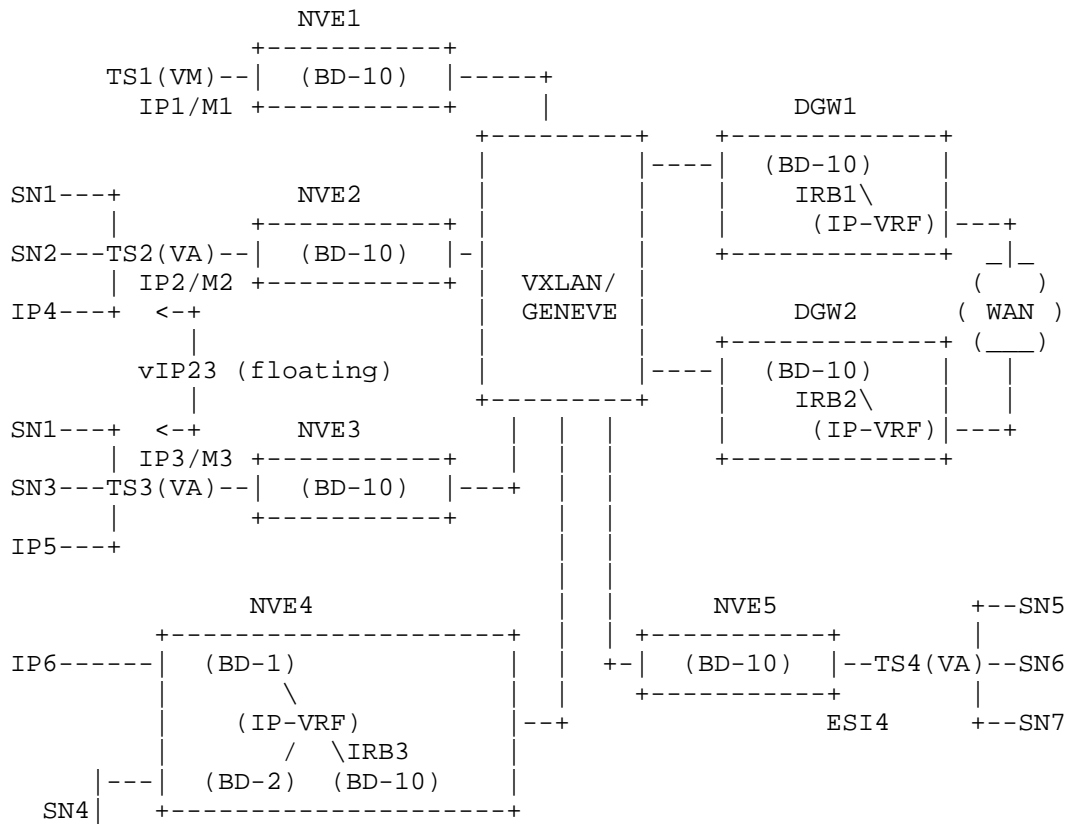


Figure 1 DC inter-subnet use-cases

Where:

NVE1, NVE2, NVE3, NVE4, NVE5, DGW1 and DGW2 share the same BD for a particular tenant. BD-10 is comprised of the collection of BD instances defined in all the NVEs. All the hosts connected to BD-10 belong to the same IP subnet. The hosts connected to BD-10 are listed below:

- o TS1 is a VM that generates/receives traffic from/to IP1, where IP1 belongs to the BD-10 subnet.
- o TS2 and TS3 are Virtual Appliances (VA) that send/receive traffic from/to the subnets and hosts sitting behind them (SN1, SN2, SN3, IP4 and IP5). Their IP addresses (IP2 and IP3) belong to the BD-10 subnet and they can also generate/receive traffic. When these VAs receive packets destined to their own MAC addresses (M2 and M3) they will route the packets to the proper subnet or host. These VAs

do not support routing protocols to advertise the subnets connected to them and can move to a different server and NVE when the Cloud Management System decides to do so. These VAs may also support redundancy mechanisms for some subnets, similar to VRRP, where a floating IP is owned by the master VA and only the master VA forwards traffic to a given subnet. E.g.,: vIP23 in Figure 1 is a floating IP that can be owned by TS2 or TS3 depending on which system is the master. Only the master will forward traffic to SN1.

- o Integrated Routing and Bridging interfaces IRB1, IRB2 and IRB3 have their own IP addresses that belong to the BD-10 subnet too. These IRB interfaces connect the BD-10 subnet to Virtual Routing and Forwarding (IP-VRF) instances that can route the traffic to other subnets for the same tenant (within the DC or at the other end of the WAN).
- o TS4 is a layer 2 VA that provides connectivity to subnets SN5, SN6 and SN7, but does not have an IP address itself in the BD-10. TS4 is connected to a port on NVE5 assigned to Ethernet Segment Identifier 4.

For a BD that an ingress NVE is attached to, "Overlay Index" is defined as an identifier that the ingress EVPN NVE requires in order to forward packets to a subnet or host in a remote subnet. As an example, vIP23 (Figure 1) is an Overlay Index that any NVE attached to BD-10 needs to know in order to forward packets to SN1. IRB3 IP address is an Overlay Index required to get to SN4, and ESI4 (Ethernet Segment Identifier 4) is an Overlay Index needed to forward traffic to SN5. In other words, the Overlay Index is a next-hop in the overlay address space that can be an IP address, a MAC address or an ESI. When advertised along with an IP Prefix, the Overlay Index requires a recursive resolution to find out to what egress NVE the EVPN packets need to be sent.

All the DC use cases in Figure 1 require inter-subnet forwarding and therefore, the individual host routes and subnets:

- a) must be advertised from the NVEs (since VAs and VMs do not participate in dynamic routing protocols) and
- b) may be associated to an Overlay Index that can be a VA IP address, a floating IP address, a MAC address or an ESI. The Overlay Index is further discussed in Section 3.2.

2.2 The Need for the EVPN IP Prefix Route

[RFC7432] defines a MAC/IP route (also referred as RT-2) where a MAC

address can be advertised together with an IP address length and IP address (IP). While a variable IP address length might have been used to indicate the presence of an IP prefix in a route type 2, there are several specific use cases in which using this route type to deliver IP Prefixes is not suitable.

One example of such use cases is the "floating IP" example described in Section 2.1. In this example it is needed to decouple the advertisement of the prefixes from the advertisement of MAC address of either M2 or M3, otherwise the solution gets highly inefficient and does not scale.

For example, if 1,000 prefixes are advertised from M2 (using RT-2) and the floating IP owner changes from M2 to M3, 1,000 routes would be withdrawn from M2 and readvertise 1k routes from M3. However if a separate route type is used, 1,000 routes can be advertised as associated to the floating IP address (vIP23) and only one RT-2 for advertising the ownership of the floating IP, i.e., vIP23 and M2 in the route type 2. When the floating IP owner changes from M2 to M3, a single RT-2 withdraw/update is required to indicate the change. The remote DGW will not change any of the 1,000 prefixes associated to vIP23, but will only update the ARP resolution entry for vIP23 (now pointing at M3).

An EVPN route (type 5) for the advertisement of IP Prefixes is described in this document. This new route type has a differentiated role from the RT-2 route and addresses the Data Center (or NVO-based networks in general) inter-subnet connectivity scenarios described in this document. Using this new RT-5, an IP Prefix may be advertised along with an Overlay Index that can be a GW IP address, a MAC or an ESI, or without an Overlay Index, in which case the BGP next-hop will point at the egress NVE/ASBR/ABR and the MAC in the Router's MAC Extended Community will provide the inner MAC destination address to be used. As discussed throughout the document, the EVPN RT-2 does not meet the requirements for all the DC use cases, therefore this EVPN route type 5 is required.

The EVPN route type 5 decouples the IP Prefix advertisements from the MAC/IP route advertisements in EVPN, hence:

- a) Allows the clean and clear advertisements of IPv4 or IPv6 prefixes in an NLRI (Network Layer Reachability Information message) with no MAC addresses.
- b) Since the route type is different from the MAC/IP Advertisement route, the current [RFC7432] procedures do not need to be modified.

- c) Allows a flexible implementation where the prefix can be linked to different types of Overlay/Underlay Indexes: overlay IP address, overlay MAC addresses, overlay ESI, underlay BGP next-hops, etc.
- d) An EVPN implementation not requiring IP Prefixes can simply discard them by looking at the route type value.

The following Sections describe how EVPN is extended with a route type for the advertisement of IP prefixes and how this route is used to address the inter-subnet connectivity requirements existing in the Data Center.

3. The BGP EVPN IP Prefix Route

The BGP EVPN NLRI as defined in [RFC7432] is shown below:

```

+-----+
|   Route Type (1 octet)   |
+-----+
|   Length (1 octet)     |
+-----+
| Route Type specific (variable) |
+-----+

```

Figure 2 BGP EVPN NLRI

This document defines an additional route type (RT-5) in the IANA EVPN Route Types registry [EVPNRouteTypes], to be used for the advertisement of EVPN routes using IP Prefixes:

Value: 5
Description: IP Prefix Route

According to Section 5.4 in [RFC7606], a node that doesn't recognize the Route Type 5 (RT-5) will ignore it. Therefore an NVE following this document can still be attached to a BD where an NVE ignoring RT-5s is attached to. Regular [RFC7432] procedures would apply in that case for both NVEs. In case two or more NVEs are attached to different BDs of the same tenant, they MUST support RT-5 for the proper Inter-Subnet Forwarding operation of the tenant.

The detailed encoding of this route and associated procedures are described in the following Sections.

3.1 IP Prefix Route Encoding

An IP Prefix Route Type for IPv4 has the Length field set to 34 and consists of the following fields:

```

+-----+
|      RD      (8 octets)      |
+-----+
|Ethernet Segment Identifier (10 octets)|
+-----+
| Ethernet Tag ID (4 octets)   |
+-----+
| IP Prefix Length (1 octet, 0 to 32) |
+-----+
| IP Prefix (4 octets)        |
+-----+
| GW IP Address (4 octets)    |
+-----+
| MPLS Label (3 octets)      |
+-----+

```

Figure 3 EVPN IP Prefix route NLRI for IPv4

An IP Prefix Route Type for IPv6 has the Length field set to 58 and consists of the following fields:

```

+-----+
|      RD      (8 octets)      |
+-----+
|Ethernet Segment Identifier (10 octets)|
+-----+
| Ethernet Tag ID (4 octets)   |
+-----+
| IP Prefix Length (1 octet, 0 to 128) |
+-----+
| IP Prefix (16 octets)       |
+-----+
| GW IP Address (16 octets)    |
+-----+
| MPLS Label (3 octets)      |
+-----+

```

Figure 4 EVPN IP Prefix route NLRI for IPv6

Where:

- o The Length field of the BGP EVPN NLRI for an EVPN IP Prefix route MUST be either 34 (if IPv4 addresses are carried) or 58 (if IPv6 addresses are carried). The IP Prefix and Gateway IP Address MUST be from the same IP address family.
- o Route Distinguisher (RD) and Ethernet Tag ID MUST be used as defined in [RFC7432] and [RFC8365]. In particular, the RD is unique per MAC-VRF (or IP-VRF). The MPLS Label field is set to either an MPLS label or a VNI, as described in [RFC8365] for other EVPN route types.
- o The Ethernet Segment Identifier MUST be a non-zero 10-octet identifier if the ESI is used as an Overlay Index (see the definition of Overlay Index in Section 3.2). It MUST be all bytes zero otherwise. The ESI format is described in [RFC7432].
- o The IP Prefix Length can be set to a value between 0 and 32 (bits) for IPv4 and between 0 and 128 for IPv6, and specifies the number of bits in the Prefix. The value MUST NOT be greater than 128.
- o The IP Prefix is a 4 or 16-octet field (IPv4 or IPv6).
- o The GW (Gateway) IP Address field is a 4 or 16-octet field (IPv4 or IPv6), and will encode a valid IP address as an Overlay Index for the IP Prefixes. The GW IP field MUST be all bytes zero if it is not used as an Overlay Index. Refer to Section 3.2 for the definition and use of the Overlay Index.
- o The MPLS Label field is encoded as 3 octets, where the high-order 20 bits contain the label value, as per [RFC7432]. When sending, the label value SHOULD be zero if recursive resolution based on overlay index is used. If the received MPLS Label value is zero, the route MUST contain an Overlay Index and the ingress NVE/PE MUST do recursive resolution to find the egress NVE/PE. If the received Label is zero and the route does not contain an Overlay Index, it MUST be treat-as-withdraw [RFC7606].

The RD, Ethernet Tag ID, IP Prefix Length and IP Prefix are part of the route key used by BGP to compare routes. The rest of the fields are not part of the route key.

An IP Prefix Route MAY be sent along with a Router's MAC Extended Community (defined in [EVPN-INTERSUBNET]) to carry the MAC address that is used as the overlay index. Note that the MAC address may be that of an TS.

As described in Section 3.2, certain data combinations in a received routes would imply a "treat-as-withdraw" handling of the route

[RFC7606].

3.2 Overlay Indexes and Recursive Lookup Resolution

RT-5 routes support recursive lookup resolution through the use of Overlay Indexes as follows:

- o An Overlay Index can be an ESI, IP address in the address space of the tenant or MAC address and it is used by an NVE as the next-hop for a given IP Prefix. An Overlay Index always needs a recursive route resolution on the NVE/PE that installs the RT-5 into one of its IP-VRFs, so that the NVE knows to which egress NVE/PE it needs to forward the packets. It is important to note that recursive resolution of the Overlay Index applies upon installation into an IP-VRF, and not upon BGP propagation (for instance, on an ASBR). Also, as a result of the recursive resolution, the egress NVE/PE is not necessarily the same NVE that originated the RT-5.
- o The Overlay Index is indicated along with the RT-5 in the ESI field, GW IP field or Router's MAC Extended Community, depending on whether the IP Prefix next-hop is an ESI, IP address or MAC address in the tenant space. The Overlay Index for a given IP Prefix is set by local policy at the NVE that originates an RT-5 for that IP Prefix (typically managed by the Cloud Management System).
- o In order to enable the recursive lookup resolution at the ingress NVE, an NVE that is a possible egress NVE for a given Overlay Index must originate a route advertising itself as the BGP next hop on the path to the system denoted by the Overlay Index. For instance:
 - . If an NVE receives an RT-5 that specifies an Overlay Index, the NVE cannot use the RT-5 in its IP-VRF unless (or until) it can recursively resolve the Overlay Index.
 - . If the RT-5 specifies an ESI as the Overlay Index, recursive resolution can only be done if the NVE has received and installed an RT-1 (Auto-Discovery per-EVI) route specifying that ESI.
 - . If the RT-5 specifies a GW IP address as the Overlay Index, recursive resolution can only be done if the NVE has received and installed an RT-2 (MAC/IP route) specifying that IP address in the IP address field of its NLRI.
 - . If the RT-5 specifies a MAC address as the Overlay Index, recursive resolution can only be done if the NVE has received and installed an RT-2 (MAC/IP route) specifying that MAC address in the MAC address field of its NLRI.

Note that the RT-1 or RT-2 routes needed for the recursive resolution may arrive before or after the given RT-5 route.

- o Irrespective of the recursive resolution, if there is no IGP or BGP route to the BGP next-hop of an RT-5, BGP MUST NOT install the RT-5 even if the Overlay Index can be resolved.
- o The ESI and GW IP fields may both be zero at the same time. However, they MUST NOT both be non-zero at the same time. A route containing a non-zero GW IP and a non-zero ESI (at the same time) SHOULD be treat-as-withdraw [RFC7606].
- o If either the ESI or GW IP are non-zero, then the non-zero one is the Overlay Index, regardless of whether the Router's MAC Extended Community is present or the value of the Label. In case the GW IP is the Overlay Index (hence ESI is zero), the Router's MAC Extended Community is ignored if present.
- o A route where ESI, GW IP, MAC and Label are all zero at the same time SHOULD be treat-as-withdraw.

The indirection provided by the Overlay Index and its recursive lookup resolution is required to achieve fast convergence in case of a failure of the object represented by the Overlay Index (see the example described in Section 2.2).

Table 1 shows the different RT-5 field combinations allowed by this specification and what Overlay Index must be used by the receiving NVE/PE in each case. Those cases where there is no Overlay Index, are indicated as "None" in Table 1. If there is no Overlay Index the receiving NVE/PE will not perform any recursive resolution, and the actual next-hop is given by the RT-5's BGP next-hop.

ESI	GW IP	MAC*	Label	Overlay Index
Non-Zero	Zero	Zero	Don't Care	ESI
Non-Zero	Zero	Non-Zero	Don't Care	ESI
Zero	Non-Zero	Zero	Don't Care	GW IP
Zero	Zero	Non-Zero	Zero	MAC
Zero	Zero	Non-Zero	Non-Zero	MAC or None**
Zero	Zero	Zero	Non-Zero	None***

Table 1 - RT-5 fields and Indicated Overlay Index

Table NOTES:

* MAC with Zero value means no Router's MAC extended community is present along with the RT-5. Non-Zero indicates that the extended community is present and carries a valid MAC address. The

encoding of a MAC address MUST be the 6-octet MAC address specified by [802.1Q] and [802.1D-REV]. Examples of invalid MAC addresses are broadcast or multicast MAC addresses. The route MUST be treat-as-withdraw in case of an invalid MAC address. The presence of the Router's MAC extended community alone is not enough to indicate the use of the MAC address as the Overlay Index, since the extended community can be used for other purposes.

- ** In this case, the Overlay Index may be the RT-5's MAC address or None, depending on the local policy of the receiving NVE/PE. Note that the advertising NVE/PE that sets the Overlay Index SHOULD advertise an RT-2 for the MAC Overlay Index if there are receiving NVE/PEs configured to use the MAC as the Overlay Index. This case in Table 1 is used in the IP-VRF-to-IP-VRF implementations described in 4.4.1 and 4.4.3. The support of a MAC Overlay Index in this model is OPTIONAL.
- *** The Overlay Index is None. This is a special case used for IP-VRF-to-IP-VRF where the NVE/PEs are connected by IP NVO tunnels as opposed to Ethernet NVO tunnels.

If the combination of ESI, GW IP, MAC and Label in the receiving RT-5 is different than the combinations shown in Table 1, the router will process the route as per the rules described at the beginning of this Section (3.2).

Table 2 shows the different inter-subnet use-cases described in this document and the corresponding coding of the Overlay Index in the route type 5 (RT-5).

Section	Use-case	Overlay Index in the RT-5
4.1	TS IP address	GW IP
4.2	Floating IP address	GW IP
4.3	"Bump in the wire"	ESI or MAC
4.4	IP-VRF-to-IP-VRF	GW IP, MAC or None

Table 2 - Use-cases and Overlay Indexes for Recursive Resolution

The above use-cases are representative of the different Overlay Indexes supported by RT-5 (GW IP, ESI, MAC or None).

4. Overlay Index Use-Cases

This Section describes some use-cases for the Overlay Index types used with the IP Prefix route. Although the examples use IPv4 Prefixes and subnets, the descriptions of the RT-5 are valid for the same cases with IPv6, only replacing the IP Prefixes, IPL and GW IP by the corresponding IPv6 values.

4.1 TS IP Address Overlay Index Use-Case

Figure 5 illustrates an example of inter-subnet forwarding for subnets sitting behind Virtual Appliances (on TS2 and TS3).

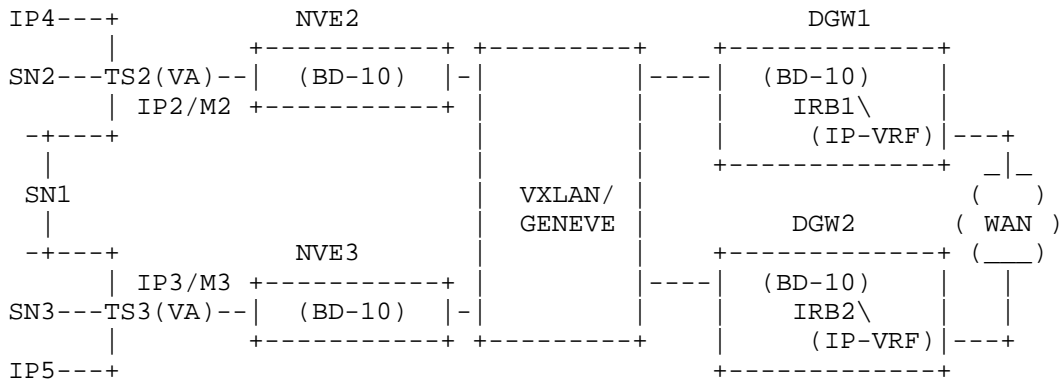


Figure 5 TS IP address use-case

An example of inter-subnet forwarding between subnet SN1, which uses a 24 bit IP prefix (written as SN1/24 in future), and a subnet sitting in the WAN is described below. NVE2, NVE3, DGW1 and DGW2 are running BGP EVPN. TS2 and TS3 do not participate in dynamic routing protocols, and they only have a static route to forward the traffic to the WAN. SN1/24 is dual-homed to NVE2 and NVE3.

In this case, a GW IP is used as an Overlay Index. Although a different Overlay Index type could have been used, this use-case assumes that the operator knows the VA's IP addresses beforehand, whereas the VA's MAC address is unknown and the VA's ESI is zero. Because of this, the GW IP is the suitable Overlay Index to be used with the RT-5s. The NVEs know the GW IP to be used for a given Prefix by policy.

- (1) NVE2 advertises the following BGP routes on behalf of TS2:
 - o Route type 2 (MAC/IP route) containing: ML=48 (MAC Address Length), M=M2 (MAC Address), IPL=32 (IP Prefix Length), IP=IP2 and [RFC5512] BGP Encapsulation Extended Community with the corresponding Tunnel type. The MAC and IP addresses may be

learned via ARP snooping.

- o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=0, GW IP address=IP2. The prefix and GW IP are learned by policy.
- (2) Similarly, NVE3 advertises the following BGP routes on behalf of TS3:
- o Route type 2 (MAC/IP route) containing: ML=48, M=M3, IPL=32, IP=IP3 (and BGP Encapsulation Extended Community).
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=0, GW IP address=IP3.
- (3) DGW1 and DGW2 import both received routes based on the Route Targets:
- o Based on the BD-10 Route Target in DGW1 and DGW2, the MAC/IP route is imported and M2 is added to the BD-10 along with its corresponding tunnel information. For instance, if VXLAN is used, the VTEP will be derived from the MAC/IP route BGP next-hop and VNI from the MPLS Label1 field. IP2 - M2 is added to the ARP table. Similarly, M3 is added to BD-10 and IP3 - M3 to the ARP table.
 - o Based on the BD-10 Route Target in DGW1 and DGW2, the IP Prefix route is also imported and SN1/24 is added to the IP-VRF with Overlay Index IP2 pointing at the local BD-10. In this example, it is assumed that the RT-5 from NVE2 is preferred over the RT-5 from NVE3. If both routes were equally preferable and ECMP enabled, SN1/24 would also be added to the routing table with Overlay Index IP3.
- (4) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:
- o A destination IP lookup is performed on the DGW1 IP-VRF routing table and Overlay Index=IP2 is found. Since IP2 is an Overlay Index a recursive route resolution is required for IP2.
 - o IP2 is resolved to M2 in the ARP table, and M2 is resolved to the tunnel information given by the BD FIB (e.g., remote VTEP and VNI for the VXLAN case).
 - o The IP packet destined to IPx is encapsulated with:

- . Source inner MAC = IRB1 MAC.
- . Destination inner MAC = M2.
- . Tunnel information provided by the BD (VNI, VTEP IPs and MACs for the VXLAN case).

(5) When the packet arrives at NVE2:

- o Based on the tunnel information (VNI for the VXLAN case), the BD-10 context is identified for a MAC lookup.
- o Encapsulation is stripped off and based on a MAC lookup (assuming MAC forwarding on the egress NVE), the packet is forwarded to TS2, where it will be properly routed.

(6) Should TS2 move from NVE2 to NVE3, MAC Mobility procedures will be applied to the MAC route IP2/M2, as defined in [RFC7432]. Route type 5 prefixes are not subject to MAC mobility procedures, hence no changes in the DGW IP-VRF routing table will occur for TS2 mobility, i.e., all the prefixes will still be pointing at IP2 as Overlay Index. There is an indirection for e.g., SN1/24, which still points at Overlay Index IP2 in the routing table, but IP2 will be simply resolved to a different tunnel, based on the outcome of the MAC mobility procedures for the MAC/IP route IP2/M2.

Note that in the opposite direction, TS2 will send traffic based on its static-route next-hop information (IRB1 and/or IRB2), and regular EVPN procedures will be applied.

4.2 Floating IP Overlay Index Use-Case

Sometimes Tenant Systems (TS) work in active/standby mode where an upstream floating IP - owned by the active TS - is used as the Overlay Index to get to some subnets behind. This redundancy mode, already introduced in Section 2.1 and 2.2, is illustrated in Figure 6.

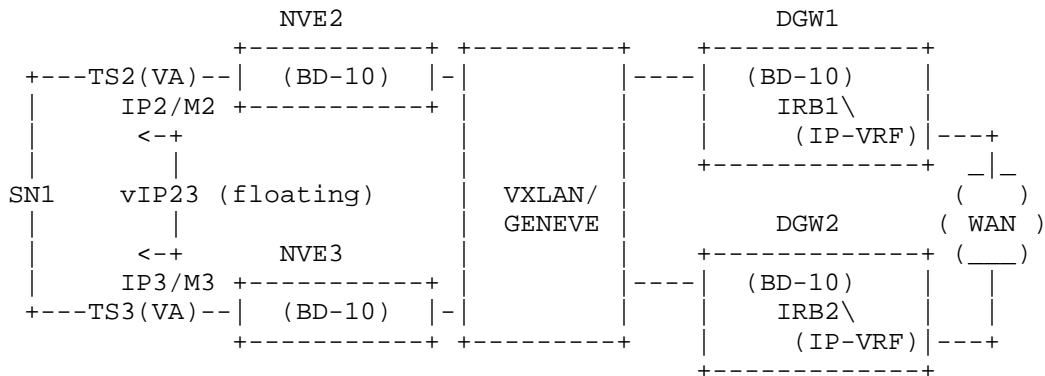


Figure 6 Floating IP Overlay Index for redundant TS

In this use-case, a GW IP is used as an Overlay Index for the same reasons as in 4.1. However, this GW IP is a floating IP that belongs to the active TS. Assuming TS2 is the active TS and owns vIP23:

- (1) NVE2 advertises the following BGP routes for TS2:
 - o Route type 2 (MAC/IP route) containing: ML=48, M=M2, IPL=32, IP=vIP23 (and BGP Encapsulation Extended Community). The MAC and IP addresses may be learned via ARP snooping.
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=0, GW IP address=vIP23. The prefix and GW IP are learned by policy.
- (2) NVE3 advertises the following BGP route for TS3 (it does not advertise an RT-2 for vIP23/M3):
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=0, GW IP address=vIP23. The prefix and GW IP are learned by policy.
- (3) DGW1 and DGW2 import both received routes based on the Route Target:
 - o M2 is added to the BD-10 FIB along with its corresponding tunnel information. For the VXLAN use case, the VTEP will be derived from the MAC/IP route BGP next-hop and VNI from the VNI field. vIP23 - M2 is added to the ARP table.
 - o SN1/24 is added to the IP-VRF in DGW1 and DGW2 with Overlay index vIP23 pointing at M2 in the local BD-10.

- (4) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:
- o A destination IP lookup is performed on the DGW1 IP-VRF routing table and Overlay Index=vIP23 is found. Since vIP23 is an Overlay Index, a recursive route resolution for vIP23 is required.
 - o vIP23 is resolved to M2 in the ARP table, and M2 is resolved to the tunnel information given by the BD (remote VTEP and VNI for the VXLAN case).
 - o The IP packet destined to IPx is encapsulated with:
 - . Source inner MAC = IRB1 MAC.
 - . Destination inner MAC = M2.
 - . Tunnel information provided by the BD FIB (VNI, VTEP IPs and MACs for the VXLAN case).
- (5) When the packet arrives at NVE2:
- o Based on the tunnel information (VNI for the VXLAN case), the BD-10 context is identified for a MAC lookup.
 - o Encapsulation is stripped off and based on a MAC lookup (assuming MAC forwarding on the egress NVE), the packet is forwarded to TS2, where it will be properly routed.
- (6) When the redundancy protocol running between TS2 and TS3 appoints TS3 as the new active TS for SN1, TS3 will now own the floating vIP23 and will signal this new ownership, using a gratuitous ARP REPLY message (explained in [RFC5227]) or similar. Upon receiving the new owner's notification, NVE3 will issue a route type 2 for M3-vIP23 and NVE2 will withdraw the RT-2 for M2-vIP23. DGW1 and DGW2 will update their ARP tables with the new MAC resolving the floating IP. No changes are made in the IP-VRF routing table.

4.3 Bump-in-the-Wire Use-Case

Figure 7 illustrates an example of inter-subnet forwarding for an IP Prefix route that carries a subnet SN1. In this use-case, TS2 and TS3 are layer 2 VA devices without any IP address that can be included as an Overlay Index in the GW IP field of the IP Prefix route. Their MAC addresses are M2 and M3 respectively and are connected to BD-10. Note that IRB1 and IRB2 (in DGW1 and DGW2 respectively) have IP addresses

in a subnet different than SN1.

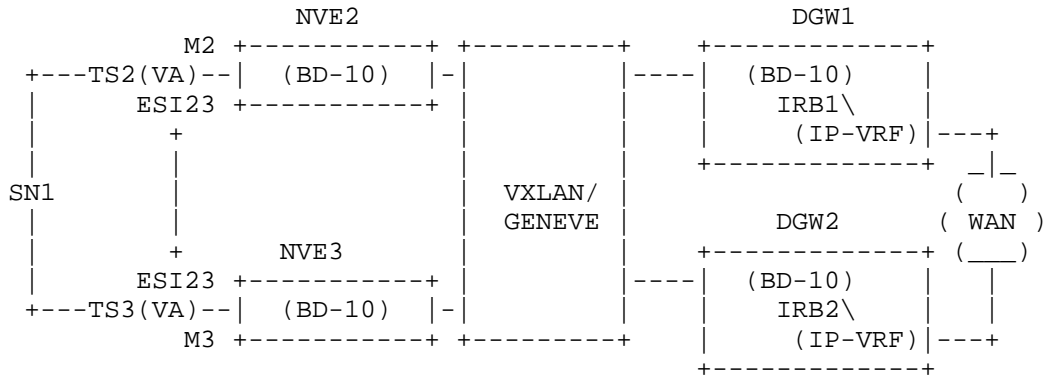


Figure 7 Bump-in-the-wire use-case

Since neither TS2 nor TS3 can participate in any dynamic routing protocol and have no IP address assigned, there are two potential Overlay Index types that can be used when advertising SN1:

- a) an ESI, i.e., ESI23, that can be provisioned on the attachment ports of NVE2 and NVE3, as shown in Figure 7.
- b) or the VA's MAC address, that can be added to NVE2 and NVE3 by policy.

The advantage of using an ESI as Overlay Index as opposed to the VA's MAC address, is that the forwarding to the egress NVE can be done purely based on the state of the AC in the ES (notified by the Ethernet A-D per-EVI route) and all the EVPN multi-homing redundancy mechanisms can be reused. For instance, the [RFC7432] mass-withdrawal mechanism for fast failure detection and propagation can be used. This Section assumes that an ESI Overlay Index is used in this use-case but it does not prevent the use of the VA's MAC address as an Overlay Index. If a MAC is used as Overlay Index, the control plane must follow the procedures described in Section 4.4.3.

The model supports VA redundancy in a similar way to the one described in Section 4.2 for the floating IP Overlay Index use-case, except that it uses the EVPN Ethernet A-D per-EVI route instead of the MAC advertisement route to advertise the location of the Overlay Index. The procedure is explained below:

- (1) Assuming TS2 is the active TS in ESI23, NVE2 advertises the following BGP routes:

- o Route type 1 (Ethernet A-D route for BD-10) containing: ESI=ESI23 and the corresponding tunnel information (VNI field), as well as the BGP Encapsulation Extended Community as per [RFC8365].
 - o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=ESI23, GW IP address=0. The Router's MAC Extended Community defined in [EVPN-INTERSUBNET] is added and carries the MAC address (M2) associated to the TS behind which SN1 sits. M2 may be learned by policy, however the MAC in the Extended Community is preferred if sent with the route.
- (2) NVE3 advertises the following BGP route for TS3 (no AD per-EVI route is advertised):
- o Route type 5 (IP Prefix route) containing: IPL=24, IP=SN1, ESI=23, GW IP address=0. The Router's MAC Extended Community is added and carries the MAC address (M3) associated to the TS behind which SN1 sits. M3 may be learned by policy, however the MAC in the Extended Community is preferred if sent with the route.
- (3) DGW1 and DGW2 import the received routes based on the Route Target:
- o The tunnel information to get to ESI23 is installed in DGW1 and DGW2. For the VXLAN use case, the VTEP will be derived from the Ethernet A-D route BGP next-hop and VNI from the VNI/VSID field (see [RFC8365]).
 - o The RT-5 coming from the NVE that advertised the RT-1 is selected and SN1/24 is added to the IP-VRF in DGW1 and DGW2 with Overlay Index ESI23 and MAC = M2.
- (4) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:
- o A destination IP lookup is performed on the DGW1 IP-VRF routing table and Overlay Index=ESI23 is found. Since ESI23 is an Overlay Index, a recursive route resolution is required to find the egress NVE where ESI23 resides.
 - o The IP packet destined to IPx is encapsulated with:
 - . Source inner MAC = IRB1 MAC.
 - . Destination inner MAC = M2 (this MAC will be obtained from the Router's MAC Extended Community received along

with the RT-5 for SN1). Note that the Router's MAC Extended Community is used in this case to carry the TS' MAC address, as opposed to the NVE/PE's MAC address.

- . Tunnel information for the NVO tunnel is provided by the Ethernet A-D route per-EVI for ESI23 (VNI and VTEP IP for the VXLAN case).

(5) When the packet arrives at NVE2:

- o Based on the tunnel demultiplexer information (VNI for the VXLAN case), the BD-10 context is identified for a MAC lookup (assuming MAC-based disposition model [RFC7432]) or the VNI may directly identify the egress interface (for a MPLS-based disposition model, which in this context is a VNI-based disposition model).
- o Encapsulation is stripped off and based on a MAC lookup (assuming MAC forwarding on the egress NVE) or a VNI lookup (in case of VNI forwarding), the packet is forwarded to TS2, where it will be forwarded to SN1.

(6) If the redundancy protocol running between TS2 and TS3 follows an active/standby model and there is a failure, appointing TS3 as the new active TS for SN1, TS3 will now own the connectivity to SN1 and will signal this new ownership. Upon receiving the new owner's notification, NVE3's AC will become active and issue a route type 1 for ESI23, whereas NVE2 will withdraw its Ethernet A-D route for ESI23. DGW1 and DGW2 will update their tunnel information to resolve ESI23. The destination inner MAC will be changed to M3.

4.4 IP-VRF-to-IP-VRF Model

This use-case is similar to the scenario described in "IRB forwarding on NVEs for Tenant Systems" in [EVPN-INTERSUBNET], however the new requirement here is the advertisement of IP Prefixes as opposed to only host routes.

In the examples described in Sections 4.1, 4.2 and 4.3, the BD instance can connect IRB interfaces and any other Tenant Systems connected to it. EVPN provides connectivity for:

1. Traffic destined to the IRB or TS IP interfaces as well as
2. Traffic destined to IP subnets sitting behind the TS, e.g., SN1 or SN2.

In order to provide connectivity for (1), MAC/IP routes (RT-2) are needed so that IRB or TS MACs and IPs can be distributed. Connectivity type (2) is accomplished by the exchange of IP Prefix routes (RT-5) for IPs and subnets sitting behind certain Overlay Indexes, e.g., GW IP or ESI or TS MAC.

In some cases, IP Prefix routes may be advertised for subnets and IPs sitting behind an IRB. This use-case is referred to as the "IP-VRF-to-IP-VRF" model.

[EVPN-INTERSUBNET] defines an asymmetric IRB model and a symmetric IRB model, based on the required lookups at the ingress and egress NVE: the asymmetric model requires an IP lookup and a MAC lookup at the ingress NVE, whereas only a MAC lookup is needed at the egress NVE; the symmetric model requires IP and MAC lookups at both, ingress and egress NVE. From that perspective, the IP-VRF-to-IP-VRF use-case described in this Section is a symmetric IRB model.

Note that, in an IP-VRF-to-IP-VRF scenario, out of the many subnets that a tenant may have, it may be the case that only a few are attached to a given NVE/PE's IP-VRF. In order to provide inter-subnet connectivity among the set of NVE/PEs where the tenant is connected, a new SBD is created on all of them if recursive resolution is needed. This SBD is instantiated as a regular BD (with no ACs) in each NVE/PE and has an IRB interface that connects the SBD to the IP-VRF. The IRB interface's IP or MAC address is used as the overlay index for recursive resolution.

Depending on the existence and characteristics of the SBD and IRB interfaces for the IP-VRFs, there are three different IP-VRF-to-IP-VRF scenarios identified and described in this document:

- 1) Interface-less model: no SBD and no overlay indexes required.
- 2) Interface-ful with SBD IRB model: it requires SBD, as well as GW IP addresses as overlay indexes.
- 3) Interface-ful with unnumbered SBD IRB model: it requires SBD, as well as MAC addresses as overlay indexes.

Inter-subnet IP multicast is outside the scope of this document.

4.4.1 Interface-less IP-VRF-to-IP-VRF Model

Figure 8 will be used for the description of this model.

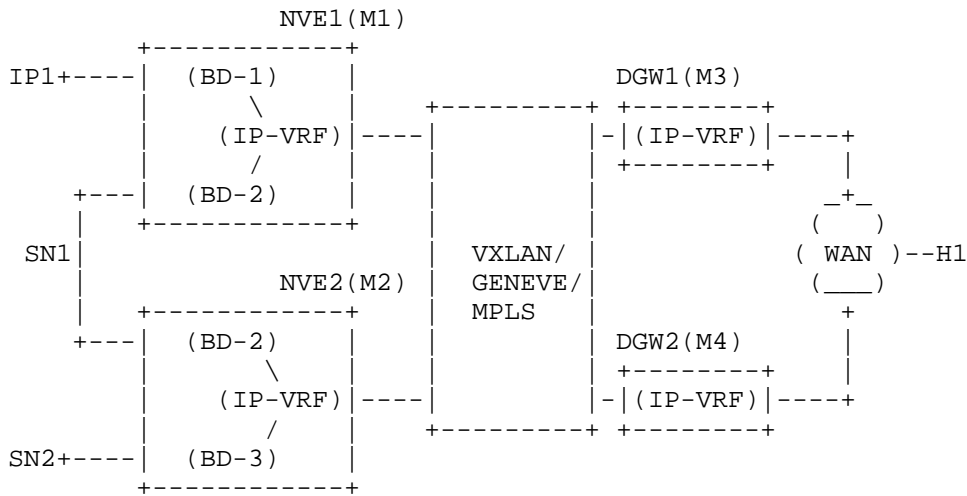


Figure 8 Interface-less IP-VRF-to-IP-VRF model

In this case:

- a) The NVEs and DGWs must provide connectivity between hosts in SN1, SN2, IP1 and hosts sitting at the other end of the WAN, for example, H1. It is assumed that the DGWs import/export IP and/or VPN-IP routes from/to the WAN.
- b) The IP-VRF instances in the NVE/DGWs are directly connected through NVO tunnels, and no IRBs and/or BD instances are instantiated to connect the IP-VRFs.
- c) The solution must provide layer 3 connectivity among the IP-VRFs for Ethernet NVO tunnels, for instance, VXLAN or GENEVE.
- d) The solution may provide layer 3 connectivity among the IP-VRFs for IP NVO tunnels, for example, GENEVE (with IP payload).

In order to meet the above requirements, the EVPN route type 5 will be used to advertise the IP Prefixes, along with the Router's MAC Extended Community as defined in [EVPN-INTERSUBNET] if the advertising NVE/DGW uses Ethernet NVO tunnels. Each NVE/DGW will advertise an RT-5 for each of its prefixes with the following fields:

- o RD as per [RFC7432].

- o Ethernet Tag ID=0.
- o IP Prefix Length and IP address, as explained in the previous Sections.
- o GW IP address=0.
- o ESI=0
- o MPLS label or VNI corresponding to the IP-VRF.

Each RT-5 will be sent with a Route Target identifying the tenant (IP-VRF) and may be sent with two BGP extended communities:

- o The first one is the BGP Encapsulation Extended Community, as per [RFC5512], identifying the tunnel type.
- o The second one is the Router's MAC Extended Community as per [EVPN-INTERSUBNET] containing the MAC address associated to the NVE advertising the route. This MAC address identifies the NVE/DGW and MAY be reused for all the IP-VRFs in the NVE. The Router's MAC Extended Community must be sent if the route is associated to an Ethernet NVO tunnel, for instance, VXLAN. If the route is associated to an IP NVO tunnel, for instance GENEVE with IP payload, the Router's MAC Extended Community should not be sent.

The following example illustrates the procedure to advertise and forward packets to SN1/24 (IPv4 prefix advertised from NVE1):

(1) NVE1 advertises the following BGP route:

- o Route type 5 (IP Prefix route) containing:
 - . IPL=24, IP=SN1, Label=10.
 - . GW IP= set to 0.
 - . [RFC5512] BGP Encapsulation Extended Community.
 - . Router's MAC Extended Community that contains M1.
 - . Route Target identifying the tenant (IP-VRF).

(2) DGW1 imports the received routes from NVE1:

- o DGW1 installs SN1/24 in the IP-VRF identified by the RT-5 Route Target.

- o Since GW IP=ESI=0, the Label is a non-zero value and the local policy indicates this interface-less model, DGW1 will use the Label and next-hop of the RT-5, as well as the MAC address conveyed in the Router's MAC Extended Community (as inner destination MAC address) to set up the forwarding state and later encapsulate the routed IP packets.
- (3) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:
- o A destination IP lookup is performed on the DGW1 IP-VRF routing table. The lookup yields SN1/24.
 - o Since the RT-5 for SN1/24 had a GW IP=ESI=0, a non-zero Label and next-hop and the model is interface-less, DGW1 will not need a recursive lookup to resolve the route.
 - o The IP packet destined to IPx is encapsulated with: Source inner MAC = DGW1 MAC, Destination inner MAC = M1, Source outer IP (tunnel source IP) = DGW1 IP, Destination outer IP (tunnel destination IP) = NVE1 IP. The Source and Destination inner MAC addresses are not needed if IP NVO tunnels are used.
- (4) When the packet arrives at NVE1:
- o NVE1 will identify the IP-VRF for an IP lookup based on the Label (the Destination inner MAC is not needed to identify the IP-VRF).
 - o An IP lookup is performed in the routing context, where SN1 turns out to be a local subnet associated to BD-2. A subsequent lookup in the ARP table and the BD FIB will provide the forwarding information for the packet in BD-2.

The model described above is called Interface-less model since the IP-VRFs are connected directly through tunnels and they don't require those tunnels to be terminated in SBDs instead, as in Sections 4.4.2 or 4.4.3.

4.4.2 Interface-ful IP-VRF-to-IP-VRF with SBD IRB

Figure 9 will be used for the description of this model.

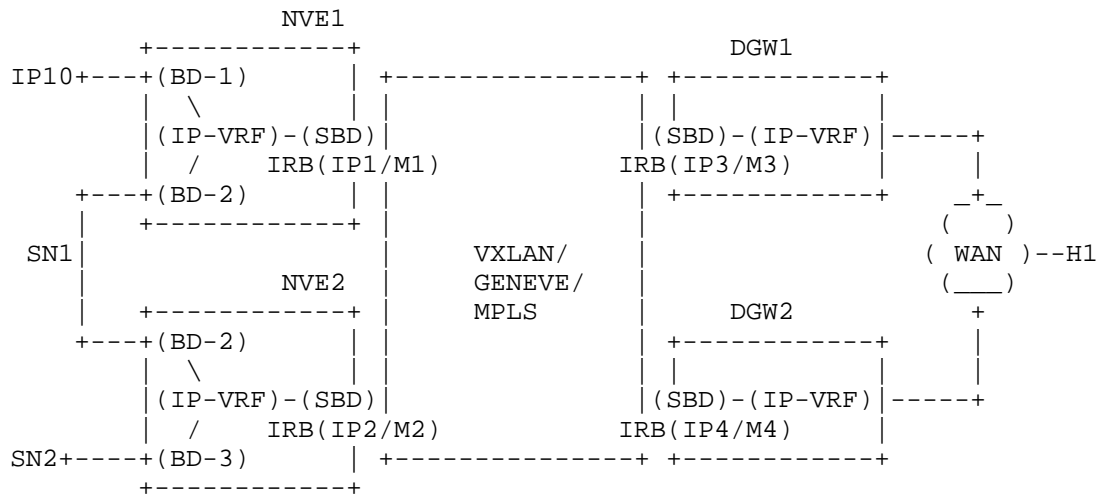


Figure 9 Interface-ful with SBD IRB model

In this model:

- a) As in Section 4.4.1, the NVEs and DGWs must provide connectivity between hosts in SN1, SN2, IP10 and hosts sitting at the other end of the WAN.
- b) However, the NVE/DGWs are now connected through Ethernet NVO tunnels terminated in the SBD instance. The IP-VRFs use IRB interfaces for their connectivity to the SBD.
- c) Each SBD IRB has an IP and a MAC address, where the IP address must be reachable from other NVEs or DGWs.
- d) The SBD is attached to all the NVE/DGWs in the tenant domain BDs.
- e) The solution must provide layer 3 connectivity for Ethernet NVO tunnels, for instance, VXLAN or GENEVE (with Ethernet payload).

EVPN type 5 routes will be used to advertise the IP Prefixes, whereas EVPN RT-2 routes will advertise the MAC/IP addresses of each SBD IRB interface. Each NVE/DGW will advertise an RT-5 for each of its prefixes with the following fields:

- o RD as per [RFC7432].
- o Ethernet Tag ID=0.

- o IP Prefix Length and IP address, as explained in the previous Sections.
- o GW IP address=IRB-IP of the SBD (this is the Overlay Index that will be used for the recursive route resolution).
- o ESI=0
- o Label value should be zero since the RT-5 route requires a recursive lookup resolution to an RT-2 route. It is ignored on reception, and, when forwarding packets, the MPLS label or VNI from the RT-2's MPLS Label field is used.

Each RT-5 will be sent with a Route Target identifying the tenant (IP-VRF). The Router's MAC Extended Community should not be sent in this case.

The following example illustrates the procedure to advertise and forward packets to SN1/24 (IPv4 prefix advertised from NVE1):

(1) NVE1 advertises the following BGP routes:

- o Route type 5 (IP Prefix route) containing:
 - . IPL=24, IP=SN1, Label= SHOULD be set to 0.
 - . GW IP=IP1 (SBD IRB's IP)
 - . Route Target identifying the tenant (IP-VRF).
- o Route type 2 (MAC/IP route for the SBD IRB) containing:
 - . ML=48, M=M1, IPL=32, IP=IP1, Label=10.
 - . A [RFC5512] BGP Encapsulation Extended Community.
 - . Route Target identifying the SBD. This Route Target may be the same as the one used with the RT-5.

(2) DGW1 imports the received routes from NVE1:

- o DGW1 installs SN1/24 in the IP-VRF identified by the RT-5 Route Target.
 - . Since GW IP is different from zero, the GW IP (IP1) will be used as the Overlay Index for the recursive route resolution to the RT-2 carrying IP1.

- (3) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:
- o A destination IP lookup is performed on the DGW1 IP-VRF routing table. The lookup yields SN1/24, which is associated to the Overlay Index IP1. The forwarding information is derived from the RT-2 received for IP1.
 - o The IP packet destined to IPx is encapsulated with: Source inner MAC = M3, Destination inner MAC = M1, Source outer IP (source VTEP) = DGW1 IP, Destination outer IP (destination VTEP) = IP1.
- (4) When the packet arrives at NVE1:
- o NVE1 will identify the IP-VRF for an IP lookup based on the Label and the inner MAC DA.
 - o An IP lookup is performed in the routing context, where SN1 turns out to be a local subnet associated to BD-2. A subsequent lookup in the ARP table and the BD FIB will provide the forwarding information for the packet in BD-2.

The model described above is called 'Interface-ful with SBD IRB model' because the tunnels connecting the DGWs and NVEs need to be terminated into the SBD. The SBD is connected to the IP-VRFs via SBD IRB interfaces, and that allows the recursive resolution of RT-5s to GW IP addresses.

4.4.3 Interface-ful IP-VRF-to-IP-VRF with Unnumbered SBD IRB

Figure 10 will be used for the description of this model. Note that this model is similar to the one described in Section 4.4.2, only without IP addresses on the SBD IRB interfaces.

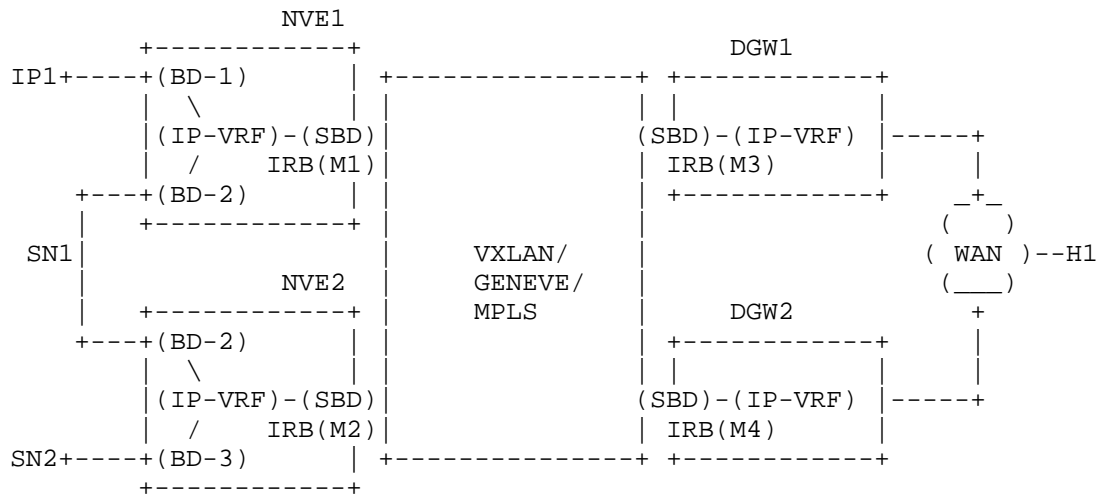


Figure 10 Interface-ful with unnumbered SBD IRB model

In this model:

- a) As in Section 4.4.1 and 4.4.2, the NVEs and DGWs must provide connectivity between hosts in SN1, SN2, IP1 and hosts sitting at the other end of the WAN.
- b) As in Section 4.4.2, the NVE/DGWs are connected through Ethernet NVO tunnels terminated in the SBD instance. The IP-VRFs use IRB interfaces for their connectivity to the SBD.
- c) However, each SBD IRB has a MAC address only, and no IP address (that is why the model refers to an 'unnumbered' SBD IRB). In this model, there is no need to have IP reachability to the SBD IRB interfaces themselves and there is a requirement to limit the number of IP addresses used.
- d) As in Section 4.4.2, the SBD is composed of all the NVE/DGW BDs of the tenant that need inter-subnet-forwarding.
- e) As in Section 4.4.2, the solution must provide layer 3 connectivity for Ethernet NVO tunnels, for instance, VXLAN or GENEVE (with Ethernet payload).

This model will also make use of the RT-5 recursive resolution. EVPN type 5 routes will advertise the IP Prefixes along with the Router's MAC Extended Community used for the recursive lookup, whereas EVPN RT-2 routes will advertise the MAC addresses of each SBD IRB

interface (this time without an IP).

Each NVE/DGW will advertise an RT-5 for each of its prefixes with the same fields as described in 4.4.2 except for:

- o GW IP address= set to 0.

Each RT-5 will be sent with a Route Target identifying the tenant (IP-VRF) and the Router's MAC Extended Community containing the MAC address associated to SBD IRB interface. This MAC address may be reused for all the IP-VRFs in the NVE.

The example is similar to the one in Section 4.4.2:

(1) NVE1 advertises the following BGP routes:

- o Route type 5 (IP Prefix route) containing the same values as in the example in Section 4.4.2, except for:
 - . GW IP= SHOULD be set to 0.
 - . Router's MAC Extended Community containing M1 (this will be used for the recursive lookup to a RT-2).
- o Route type 2 (MAC route for the SBD IRB) with the same values as in Section 4.4.2 except for:
 - . ML=48, M=M1, IPL=0, Label=10.

(2) DGW1 imports the received routes from NVE1:

- o DGW1 installs SN1/24 in the IP-VRF identified by the RT-5 Route Target.
 - . The MAC contained in the Router's MAC Extended Community sent along with the RT-5 (M1) will be used as the Overlay Index for the recursive route resolution to the RT-2 carrying M1.

(3) When DGW1 receives a packet from the WAN with destination IPx, where IPx belongs to SN1/24:

- o A destination IP lookup is performed on the DGW1 IP-VRF routing table. The lookup yields SN1/24, which is associated to the Overlay Index M1. The forwarding information is derived from the RT-2 received for M1.
- o The IP packet destined to IPx is encapsulated with: Source

inner MAC = M3, Destination inner MAC = M1, Source outer IP (source VTEP) = DGW1 IP, Destination outer IP (destination VTEP) = NVE1 IP.

(4) When the packet arrives at NVE1:

- o NVE1 will identify the IP-VRF for an IP lookup based on the Label and the inner MAC DA.
- o An IP lookup is performed in the routing context, where SN1 turns out to be a local subnet associated to BD-2. A subsequent lookup in the ARP table and the BD FIB will provide the forwarding information for the packet in BD-2.

The model described above is called Interface-ful with unnumbered SBD IRB model (as in Section 4.4.2), only this time the SBD IRB does not have an IP address.

5. Security Considerations

This document provides a set of procedures to achieve Inter-Subnet Forwarding across NVEs or PEs attached to a group of BDs that belong to the same tenant (or VPN). The security considerations discussed in [RFC7432] apply to the Intra-Subnet Forwarding or communication within each of those BDs. In addition, the security considerations in [RFC4364] should also be understood, since this document and [RFC4364] may be used in similar applications.

Contrary to [RFC4364], this document does not describe PE/CE route distribution techniques, but rather considers the CEs as TSeS or VAS that do not run dynamic routing protocols. This can be considered a security advantage, since dynamic routing protocols can be blocked on the NVE/PE ACs, not allowing the tenant to interact with the infrastructure's dynamic routing protocols.

In this document, the RT-5 may use a regular BGP Next Hop for its resolution or an Overlay Index that requires a recursive resolution to a different EVPN route (an RT-2 or an RT-1). In the latter case, it is worth noting that any action that ends up filtering or modifying the RT-2/RT-1 routes used to convey the Overlay Indexes, will modify the resolution of the RT-5 and therefore the forwarding of packets to the remote subnet.

6. IANA Considerations

This document requests value 5 in the [EVPNRouteTypes] registry

defined by [RFC7432]:

Value	Description	Reference
5	IP Prefix route	[this document]

7. References

7.1 Normative References

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

[RFC5512] Mohapatra, P. and E. Rosen, "The BGP Encapsulation Subsequent Address Family Identifier (SAFI) and the BGP Tunnel Encapsulation Attribute", RFC 5512, DOI 10.17487/RFC5512, April 2009, <<http://www.rfc-editor.org/info/rfc5512>>.

[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>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<http://www.rfc-editor.org/info/rfc8174>>.

[RFC8365] Sajassi-Drake et al., "A Network Virtualization Overlay Solution using EVPN", RFC 8365, DOI 10.17487/RFC8365, March, 2018.

[EVPN-INTERSUBNET] Sajassi et al., "IP Inter-Subnet Forwarding in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-03.txt, work in progress, February, 2017

[EVPNRouteTypes] IANA EVPN Route Type registry, <https://www.iana.org/assignments/evpn>

7.2 Informative References

[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>>.

[RFC7606] Chen, E., Scudder, J., Mohapatra, P., and K. Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, August

2015, <<http://www.rfc-editor.org/info/rfc7606>>.

[802.1D-REV] "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges", IEEE Std. 802.1D, June 2004.

[802.1Q] "IEEE Standard for Local and metropolitan area networks - Media Access Control (MAC) Bridges and Virtual Bridged Local Area Networks", IEEE Std 802.1Q(tm), 2014 Edition, November 2014.

[RFC7365] Lasserre, M., Balus, F., Morin, T., Bitar, N., and Y. Rekhter, "Framework for Data Center (DC) Network Virtualization", RFC 7365, DOI 10.17487/RFC7365, October 2014, <<https://www.rfc-editor.org/info/rfc7365>>.

[RFC5227] Cheshire, S., "IPv4 Address Conflict Detection", RFC 5227, DOI 10.17487/RFC5227, July 2008, <<https://www.rfc-editor.org/info/rfc5227>>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.

[GENEVE] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", Work in Progress, draft-ietf-nvo3-geneve-06, March 2018.

8. Acknowledgments

The authors would like to thank Mukul Katiyar and Jeffrey Zhang for their valuable feedback and contributions. The following people also helped improving this document with their feedback: Tony Przygienda and Thomas Morin. Special THANK YOU to Eric Rosen for his detailed review, it really helped improve the readability and clarify the concepts. Thank you to Alvaro Retana for his thorough review.

9. Contributors

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

Senthil Sathappan
Florin Balus
Aldrin Isaac
Senad Palislamovic

Samir Thoria

10. Authors' Addresses

Jorge Rabadan (Editor)
Nokia
777 E. Middlefield Road
Mountain View, CA 94043 USA
Email: jorge.rabadan@nokia.com

Wim Henderickx
Nokia
Email: wim.henderickx@nokia.com

John E. Drake
Juniper
Email: jdrake@juniper.net

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Wen Lin
Juniper
Email: wlin@juniper.net

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
pe
+--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
        +--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

```

```

|
|         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 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
dentifier-type
|
|         +---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
```



```

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

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

```

4. YANG Module

The EVPN configuration container is logically divided into

following high level configuration areas:

4.1 Ethernet Segment Yang Module

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

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

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

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

  import iETF-interfaces {
    prefix "if";
  }

  import iETF-pseudowires {
    prefix "pw";
  }

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

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

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

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

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

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

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

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

/* Features */
```

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

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

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

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

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

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

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

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

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

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

4.2 EVPN Yang Module

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

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

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

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

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

  import ietf-l2vpn {
    prefix "l2vpn";
  }

  import ietf-pseudowires {
    prefix "pw";
  }

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

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



```
description    "evpn";

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

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

revision "2017-10-21" {
  description " - Modified the operationalstate 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-constraints {
        must "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:pw/l2vpn:name]" +
            "    /evpn-pw/local-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:primary-pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:primary-pw/l2vpn:name]" +
            "    /evpn-pw/local-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:backup-pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:backup-pw/l2vpn:name]" +
            "    /evpn-pw/local-id))" {
            description "A VPLS pseudowire must not be EVPN PW";
        }
        description "VPLS constraints";
    }
}

/* Notifications */

notification evpn-state-change-notification {
    description "EVPN state change notification";
}

```

```
    leaf evpn-instance {
      type evpn-instance-ref;
      description "Related EVPN instance";
    }
    leaf state {
      type identityref {
        base evpn-notification-state;
      }
      description "State change notification";
    }
  }
}
<CODE ENDS>
```

5. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

The security concerns listed above are, however, no different than faced by other routing protocols. Hence, this draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg]

6. IANA Considerations

None.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <<https://www.rfc-editor.org/info/rfc6991>>.
- [RFC8294] Liu, X., Qu, Y., Lindem, A., Hopps, C., and L. Berger, "Common YANG Data Types for the Routing Area", RFC 8294,

DOI 10.17487/RFC8294, December 2017, <<https://www.rfc-editor.org/info/rfc8294>>.

7.2. Informative References

- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC7209] Sajassi, A., Aggarwal, R., Uttaro, J., Bitar, N., Henderickx, W., and A. Isaac, "Requirements for Ethernet VPN (EVPN)", RFC 7209, DOI 10.17487/RFC7209, May 2014, <<https://www.rfc-editor.org/info/rfc7209>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<https://www.rfc-editor.org/info/rfc7432>>.
- [RFC7623] Sajassi, A., Ed., Salam, S., Bitar, N., Isaac, A., and W. Henderickx, "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, DOI 10.17487/RFC7623, September 2015, <<https://www.rfc-editor.org/info/rfc7623>>.
- [RFC8214] Boutros, S., Sajassi, A., Salam, S., Drake, J., and J. Rabadan, "Virtual Private Wire Service Support in Ethernet VPN", RFC 8214, DOI 10.17487/RFC8214, August 2017, <<https://www.rfc-editor.org/info/rfc8214>>.

Authors' Addresses

Patrice Brissette
Cisco Systems, Inc.
EMail: pbrisset@cisco.com

Himanshu Shah
Ciena Corporation
EMail: hshah@ciena.com

Helen Chen
Jabil
EMail: Ing-Wher_Chen@jabil.com

Iftekar Hussain
Infinera Corporation
EMail: ihussain@infinera.com

Kishore Tiruveedhula
Juniper Networks
EMail: kishoret@juniper.net

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

Ali Sajassi
Cisco Systems, Inc.
EMail: sajassi@cisco.com

Zhenbin Li
Huawei Technologies
EMail: lizhenbin@huawei.com

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2020

H. Shah, Ed.
Ciena Corporation
P. Brissette, Ed.
Cisco Systems, Inc.
I. Chen, Ed.
The MITRE Corporation
I. Hussain, Ed.
Infinera Corporation
B. Wen, Ed.
Comcast
K. Tiruveedhula, Ed.
Juniper Networks
July 02, 2019

YANG Data Model for MPLS-based L2VPN
draft-ietf-bess-l2vpn-yang-10.txt

Abstract

This document describes a YANG data model for Layer 2 VPN (L2VPN) services over MPLS networks. These services include point-to-point Virtual Private Wire Service (VPWS) and multipoint Virtual Private LAN service (VPLS) that uses LDP and BGP signaled Pseudowires. It is expected that this model will be used by the management tools run by the network operators in order to manage and monitor the network resources that they use to deliver L2VPN services.

This document also describes the YANG data model for the Pseudowires. The independent definition of the Pseudowires facilitates its use in Ethernet Segment and EVPN data models defined in separate document.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 3, 2020.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Specification of Requirements	4
3. L2VPN YANG Model	4
3.1. Overview	4
3.2. Latest addition	7
3.3. Open issues and next steps	8
3.4. Pseudowire Common	8
3.4.1. Pseudowire	8
3.4.2. pw-templates	8
3.5. L2VPN Common	8
3.5.1. redundancy-group-templates	8
3.6. L2VPN instance	9
3.6.1. common attributes	9
3.6.2. PW list	9
3.6.3. List of endpoints	9
3.6.4. point-to-point or multipoint service	10
3.6.5. multi-segment pseudowire	11
3.7. Operational State	11
3.8. Yang tree	11
4. YANG Module	14
5. Security Considerations	43
6. IANA Considerations	43
7. Acknowledgments	43
8. References	44
8.1. Normative References	44
8.2. Informative References	44
Appendix A. Example Configuration	47
Appendix B. Contributors	47
Authors' Addresses	48

1. Introduction

The Network Configuration Protocol (NETCONF) [RFC6241] is a network management protocol that defines mechanisms to manage network devices. YANG [RFC7950] is a modular language that represents data structures in an XML or JSON tree format, and is used as a data modeling language for the NETCONF.

This document defines a YANG data model for MPLS based Layer 2 VPN services (L2VPN) [RFC4664] and includes switching between the local attachment circuits. The L2VPN model covers point-to-point VPWS and Multipoint VPLS services. These services use signaling of Pseudowires across MPLS networks using LDP [RFC8077][RFC4762] or BGP[RFC4761].

The data model covers Ethernet based Layer 2 services. The Ethernet Attachment Circuits are not defined. Instead, they are leveraged from other standards organizations such as IEEE802.1 and Metro Ethernet Forum (MEF).

Other Layer 2 services, such as ATM, Frame Relay, TDM, etc are included in the scope but will be covered as the future work items.

The objective of the model is to define building blocks that can easily be assembled in different order to realize different services.

The data model uses following constructs for configuration and management:

- o Configuration
- o Operational State
- o Executables (Actions)
- o Notifications

This document focuses on definition of configuration, state and notification objects.

The L2VPN data object model uses the instance centric approach. The L2VPN instance is recognized by network instance model. The network-instance container is defined in network instance model [I-D.ietf-netmod-ni-model].

Within this network instance, L2VPN container contains definitions of a set of common parameters, a list of PWs and a list of endpoints. A

special constraint is added for the VPWS configuration such that only two endpoints are allowed in the list of endpoints.

The Pseudowire data object model is defined independent of the L2VPN data object model to allow its inclusion in the Ethernet Segment and EVPN data objects.

The L2VPN data object model augments Pseudowire 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 number 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
rrrent () / ../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
    |         +---rw vccv-ability?    boolean
    |         +---rw request-vlanid?  uint16
    |         +---rw vlan-tpid?       string
    |         +---rw ttl?              uint8
    |         +---rw split-horizon-group? string
    +---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
    +---rw vccv-ability?    boolean
    +---rw request-vlanid?  uint16
    +---rw vlan-tpid?       string
    +---rw ttl?              uint8
    +---rw split-horizon-group? string
  
```

```

+--:(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 "Ninth revision " +
      " - Used bgp parameters hierarchy common to L2VPN and EVPN " +
      "";
    reference "";
  }

  revision "2018-02-06" {
    description "Eighth revision " +
      " - Incorporated ietf-network-instance model " +
      " - change the type of attachment circuit to " +
      " if:interface-ref " +
      "";
    reference "";
  }
}
```

```
}  
revision "2017-09-21" {  
  description "Seventh revision " +  
    " - Fixed yangdump errors " +  
    "";  
  reference "";  
}  
revision "2017-06-26" {  
  description "Sixth revision " +  
    " - Removed unused module mpls " +  
    " - Renamed l2vpn-instances-state to l2vpn-instances " +  
    " - Added pseudowire status as defined in RFC4446 and " +  
    " IANA Pseudowire Status Codes Register " +  
    " - Added notifications " +  
    " - Moved PW definition out of L2VPN " +  
    " - Moved model to NMDA style specified in " +  
    " draft-dsdt-nmda-guidelines-01.txt " +  
    " - Renamed l2vpn-instances and l2vpn-instance to " +  
    " instances and instance to shorten xpaths " +  
    "";  
  reference "";  
}  
revision "2017-03-06" {  
  description "Sixth revision " +  
    " - Removed the 'common' container and move pw-templates " +  
    " and redundancy-group-templates up a level " +  
    " - Consolidated the endpoint configuration such that " +  
    " all L2VPN instances has a list of endpoint. For " +  
    " certain types of L2VPN instances such as VPWS where " +  
    " each L2VPN instance is limited to at most two " +  
    " endpoint, additional augment statements were included " +  
    " to add necessary constraints " +  
    " - Removed discovery-type and signaling-type operational " +  
    " state from VPLS pseudowires, as these two parameters " +  
    " are configured as L2VPN parameters rather than " +  
    " pseudowire paramteres " +  
    " - Renamed l2vpn-instances to l2vpn-instances-state " +  
    " in the operational state branch " +  
    " - Removed BGP parameter groupings and reused " +  
    " ietf-routing-types.yang module instead " +  
    "";  
  reference "";  
}  
revision "2016-10-24" {  
  description "Fifth revision " +
```

```
    " - Edits based on Giles's comments " +
    " 5) Remove relative leafrefs in groupings, " +
    " and the resulting new groupings are: " +
    " (a) bgp-auto-discovery-parameters-grp " +
    " (b) bgp-signaling-parameters-grp " +
    " (c) endpoint-grp " +
    " 11) Merge VPLS and VPWS into one single list " +
    " and use augment statements to handle " +
    " differences between VPLS and VPWS " +
    " - Add a new grouping l2vpn-common-parameters-grp " +
    " to make VPLS and VPWS more consistent";
  reference "";
}

revision "2016-05-31" {
  description "Fourth revision " +
    " - Edits based on Giles's comments " +
    " 1) Change enumeration to identityref type for: " +
    " (a) l2vpn-service-type " +
    " (b) l2vpn-discovery-type " +
    " (c) l2vpn-signaling-type " +
    " bgp-rt-type, cw-negotiation, and " +
    " pbb-component remain enumerations " +
    " 2) Define i-sid-type for leaf 'i-sid' " +
    " (which is renamed from 'i-tag') " +
    " 3) Rename 'vpn-targets' to 'vpn-target' " +
    " 4) Import ietf-mpls.yang and reuse the " +
    " 'mpls-label' type defined in ietf-mpls.yang " +
    " transmit-label and receive-label " +
    " 8) Change endpoint list's key to name " +
    " 9) Changed MTU to type uint16 " +
    "";
  reference "";
}

revision "2016-03-07" {
  description "Third revision " +
    " - Changed the module name to ietf-l2vpn " +
    " - Merged EVPN into L2VPN " +
    " - Eliminated the definitions of attachment " +
    " circuit with the intention to reuse other " +
    " layer-2 definitions " +
    " - Added state branch";
  reference "";
}

revision "2015-10-08" {
  description "Second revision " +
```

```
        " - Added container vpls-instances " +
        " - Rearranged groupings and typedefs to be " +
        "   reused across vpls-instance and vpws-instances";
    reference "";
}

revision "2015-06-30" {
    description "Initial revision";
    reference  "";
}

/* identities */

identity l2vpn-instance-type {
    description "Base identity from which identities of " +
               "l2vpn service instance types are derived";
}

identity vpws-instance-type {
    base l2vpn-instance-type;
    description "This identity represents VPWS instance type";
}

identity vpls-instance-type {
    base l2vpn-instance-type;
    description "This identity represents VPLS instance type";
}

identity link-discovery-protocol {
    description "Base identity from which identities describing " +
               "link discovery protocols are derived";
}

identity lacp {
    base "link-discovery-protocol";
    description "This identity represents LACP";
}

identity lldp {
    base "link-discovery-protocol";
    description "This identity represents LLDP";
}

identity bpdu {
    base "link-discovery-protocol";
    description "This identity represents BPDU";
}
```

```
identity cpd {
  base "link-discovery-protocol";
  description "This identity represents CPD";
}

identity udld {
  base "link-discovery-protocol";
  description "This identity represens UDLD";
}

identity l2vpn-service {
  description "Base identity from which identities describing " +
    "L2VPN services are derived";
}

identity Ethernet {
  base "l2vpn-service";
  description "This identity represents Ethernet service";
}

identity ATM {
  base "l2vpn-service";
  description "This identity represents Asynchronous Transfer " +
    "Mode service";
}

identity FR {
  base "l2vpn-service";
  description "This identity represent Frame-Relay service";
}

identity TDM {
  base "l2vpn-service";
  description "This identity represent Time Devision " +
    "Multiplexing service";
}

identity l2vpn-discovery {
  description "Base identity from which identities describing " +
    "L2VPN discovery protocols are derived";
}

identity manual-discovery {
  base "l2vpn-discovery";
  description "Manual configuration of l2vpn service";
}

identity bgp-auto-discovery {
  base "l2vpn-discovery";
```

```
    description "Border Gateway Protocol (BGP) auto-discovery of " +
                "l2vpn service";
}

identity ldp-discovery {
    base "l2vpn-discovery";
    description "Label Distribution Protocol (LDP) discovery of " +
                "l2vpn service";
}

identity mixed-discovery {
    base "l2vpn-discovery";
    description "Mixed discovery methods of l2vpn service";
}

identity l2vpn-signaling {
    description "Base identity from which identities describing " +
                "L2VPN signaling protocols are derived";
}

identity static-configuration {
    base "l2vpn-signaling";
    description "Static configuration of labels (no signaling)";
}

identity ldp-signaling {
    base "l2vpn-signaling";
    description "Label Distribution Protocol (LDP) signaling";
}

identity bgp-signaling {
    base "l2vpn-signaling";
    description "Border Gateway Protocol (BGP) signaling";
}

identity mixed-signaling {
    base "l2vpn-signaling";
    description "Mixed signaling methods";
}

identity l2vpn-notification-state {
    description "The base identity on which notification states " +
                "are based";
}

identity MAC-limit-reached {
    base "l2vpn-notification-state";
    description "MAC limit is reached";
}
```

```
}
identity MAC-limit-cleared {
  base "l2vpn-notification-state";
  description "MAC limit is cleared";
}

identity MTU-mismatched {
  base "l2vpn-notification-state";
  description "MAC is mismatched";
}

identity MTU-mismatched-cleared {
  base "l2vpn-notification-state";
  description "MAC is mismatch is cleared";
}

identity state-changed-to-up {
  base "l2vpn-notification-state";
  description "State is changed to UP";
}

identity state-changed-to-down {
  base "l2vpn-notification-state";
  description "State is changed to down";
}

identity MAC-move-limit-exceeded {
  base "l2vpn-notification-state";
  description "MAC move limit is exceeded";
}

identity MAC-move-limit-exceeded-cleared {
  base "l2vpn-notification-state";
  description "MAC move limit exceeded is cleared";
}

identity MAC-flap-detected {
  base "l2vpn-notification-state";
  description "MAC flap detected";
}

identity port-disabled-due-to-MAC-flap {
  base "l2vpn-notification-state";
  description "Port disabled due to MAC flap";
}

/* typedefs */
```



```
typedef l2vpn-service-type {
  type identityref {
    base "l2vpn-service";
  }
  description "L2VPN service type";
}

typedef l2vpn-discovery-type {
  type identityref {
    base "l2vpn-discovery";
  }
  description "L2VPN discovery type";
}

typedef l2vpn-signaling-type {
  type identityref {
    base "l2vpn-signaling";
  }
  description "L2VPN signaling type";
}

typedef link-discovery-protocol-type {
  type identityref {
    base "link-discovery-protocol";
  }
  description "This type is used to identify " +
    "link discovery protocol";
}

typedef pbb-component-type {
  type enumeration {
    enum "b-component" {
      description "Identifies as a b-component";
    }
    enum "i-component" {
      description "Identifies as an i-component";
    }
  }
  description "This type is used to identify " +
    "the type of PBB component";
}

typedef redundancy-group-template-ref {
  type leafref {
    path "/l2vpn:l2vpn/l2vpn:redundancy-group-templates" +
      "/l2vpn:redundancy-group-template/l2vpn:name";
  }
  description "redundancy-group-template-ref";
}
```

```
}
typedef l2vpn-instance-name-ref {
  type leafref {
    path "/ni:network-instances/ni:network-instance" +
        "/ni:name";
  }
  description "l2vpn-instance-name-ref";
}

typedef l2vpn-instance-type-ref {
  type leafref {
    path "/ni:network-instances/ni:network-instance" +
        "/l2vpn:type";
  }
  description "l2vpn-instance-type-ref";
}

typedef operational-state-type {
  type enumeration {
    enum 'up' {
      description "Operational state is up";
    }
    enum 'down' {
      description "Operational state is down";
    }
  }
  description "operational-state-type";
}

typedef i-sid-type {
  type uint32 {
    range "0..16777216";
  }
  description "I-SID type that is 24-bits. " +
    "This should be moved to ieee-types.yang at " +
    "http://www.ieee802.org/1/files/public/docs2015" +
    "/new-mholness-ieee-types-yang-v01.yang";
}

/* groupings */

grouping pbb-parameters-grp {
  description "PBB parameters grouping";
  container pbb-parameters {
    description "pbb-parameters";
    choice component-type {
      description "PBB component type";
      case i-component {
```

```

    leaf i-sid {
        type i-sid-type;
        description "I-SID";
    }
    leaf backbone-src-mac {
        type yang:mac-address;
        description "backbone-src-mac";
    }
}
case b-component {
    leaf bind-b-component-name {
        type l2vpn-instance-name-ref;
        must "/ni:network-instances" +
            "/ni:network-instance[ni:name=current()]" +
            "/l2vpn:type = 'l2vpn:vpls-instance-type'" {
            description "A b-component must be an L2VPN instance " +
                "of type vpls-instance-type";
        }
        description "Reference to the associated b-component";
    }
    leaf bind-b-component-type {
        type identityref {
            base l2vpn-instance-type;
        }
        must ". = 'l2vpn:vpls-instance-type'" {
            description "The associated b-component must have " +
                "type vpls-instance-type";
        }
        config false;
        description "Type of the associated b-component";
    }
}
}
}
}

grouping pbb-parameters-state-grp {
    description "PBB parameters grouping";
    container pbb-parameters {
        description "pbb-parameters";
        choice component-type {
            description "PBB component type";
            case i-component {
                leaf i-sid {
                    type i-sid-type;
                    description "I-SID";
                }
                leaf backbone-src-mac {

```

```
        type yang:mac-address;
        description "backbone-src-mac";
    }
}
case b-component {
    leaf bind-b-component-name {
        type string;
        description "Name of the associated b-component";
    }
    leaf bind-b-component-type {
        type identityref {
            base l2vpn-instance-type;
        }
        must ". = 'l2vpn:vpls-instance-type'" {
            description "The associated b-component must have " +
                "type vpls-instance-type";
        }
        description "Type of the associated b-component";
    }
}
}
}
}

grouping l2vpn-common-parameters-grp {
    description "L2VPN common parameters";
    leaf type {
        type identityref {
            base l2vpn-instance-type;
        }
        description "Type of L2VPN service instance";
    }
    leaf mtu {
        type uint16;
        description "MTU of L2VPN service";
    }
    leaf mac-aging-timer {
        type uint32;
        description "mac-aging-timer, the duration after which" +
            "a MAC entry is considered aged out";
    }
    leaf service-type {
        type l2vpn-service-type;
        default Ethernet;
        description "L2VPN service type";
    }
    leaf discovery-type {
        type l2vpn-discovery-type;
    }
}
```

```
        default manual-discovery;
        description "L2VPN service discovery type";
    }
    leaf signaling-type {
        type l2vpn-signaling-type;
        mandatory true;
        description "L2VPN signaling type";
    }
}
grouping bgp-signaling-parameters-grp {
    description "BGP parameters for signaling";
    leaf site-id {
        type uint16;
        description "Site ID";
    }
    leaf site-range {
        type uint16;
        description "Site Range";
    }
}

grouping redundancy-group-properties-grp {
    description "redundancy-group-properties-grp";
    leaf protection-mode {
        type enumeration {
            enum "frr" {
                value 0;
                description "fast reroute";
            }
            enum "master-slave" {
                value 1;
                description "master-slave";
            }
            enum "independent" {
                value 2;
                description "independent";
            }
        }
        description "protection-mode";
    }
    leaf reroute-mode {
        type enumeration {
            enum "immediate" {
                value 0;
                description "immediate reroute";
            }
            enum "delayed" {
                value 1;
            }
        }
    }
}
```

```
        description "delayed reroute";
    }
    enum "never" {
        value 2;
        description "never reroute";
    }
}
description "reroute-mode";
}
leaf dual-receive {
    type boolean;
    description
        "allow extra traffic to be carried by backup";
}
leaf revert {
    type boolean;
    description "allow forwarding to revert to primary " +
        "after restoring primary";
}
leaf reroute-delay {
    when "../reroute-mode = 'delayed'" {
        description "Specify amount of time to " +
            "delay reroute only when " +
            "delayed route is configured";
    }
    type uint16;
    description "amount of time to delay reroute";
}
leaf revert-delay {
    when "../revert = 'true'" {
        description "Specify the amount of time to " +
            "wait to revert to primary " +
            "only if reversion is configured";
    }
    type uint16;
    description "amount of time to wait to revert to primary";
}
}

grouping endpoint-grp {
    description "A grouping that defines the structure of " +
        "an endpoint";
    choice ac-or-pw-or-redundancy-grp {
        description "A choice of attachment circuit or " +
            "pseudowire or redundancy group";
        case ac {
            description "Attachment circuit(s) as an endpoint";
        }
    }
}
```

```
    case pw {
      description "Pseudowire(s) as an endpoint";
    }
    case redundancy-grp {
      description "Redundancy group as an endpoint";
      choice primary {
        mandatory true;
        description "primary options";
        case primary-ac {
          description "primary-ac";
        }
        case primary-pw {
          description "primary-pw";
        }
      }
      choice backup {
        description "backup options";
        case backup-ac {
          description "backup-ac";
        }
        case backup-pw {
          description "backup-pw";
        }
      }
    }
  }
}

/* L2VPN YANG Model */

container l2vpn {
  description "L2VPN specific data";

  container redundancy-group-templates {
    description "redundancy group templates";
    list redundancy-group-template {
      key "name";
      description "redundancy-group-template";
      leaf name {
        type string;
        description "name";
      }
      uses redundancy-group-properties-grp;
    }
  }
}

/* augments */
```

```
augment "/ni:network-instances/ni:network-instance/ni:ni-type" {
  description
    "Augmentation for L2VPN instance";
  case l2vpn {
    description "An L2VPN service instance";
    uses l2vpn-common-parameters-grp;
    container bgp-parameters {
      when "../discovery-type = 'l2vpn:bgp-auto-discovery'" {
        description "Parameters used when discovery type is " +
          "bgp-auto-discovery";
      }
      description "BGP auto-discovery parameters";
      leaf vpn-id {
        type string;
        description "VPN ID";
      }
      container rd-rt {
        leaf route-distinguisher {
          type rt-types:route-distinguisher;
          description "BGP route distinguisher";
        }
        uses rt-types:vpn-route-targets;
        description "Route distinguisher and " +
          "corresponding VPN route targets";
      }
    }
  }
  container bgp-signaling {
    when "../signaling-type = 'l2vpn:bgp-signaling'" {
      description "Check signaling type: " +
        "Can only configure BGP signaling if " +
        "signaling type is BGP";
    }
    description "BGP signaling parameters";
    uses bgp-signaling-parameters-grp;
  }
  list endpoint {
    key "name";
    description "An endpoint";
    leaf name {
      type string;
      description "endpoint name";
    }
    uses endpoint-grp {
      augment "ac-or-pw-or-redundancy-grp/ac" {
        description "Augment for attachment circuit(s) " +
          "as an endpoint";
      }
      list ac {
        key "name";
      }
    }
  }
}
```



```

    leaf name {
        type if:interface-ref;
        description "Name of attachment circuit";
    }
    leaf state {
        type operational-state-type;
        config false;
        description "attachment circuit up/down state";
    }
    description "An L2VPN instance's " +
        "attachment circuit list";
}
}
augment "ac-or-pw-or-redundancy-grp/pw" {
    description "Augment for pseudowire(s) as an endpoint";
    list pw {
        key "name";
        leaf name {
            type pw:pseudowire-ref;
            must "(../../../../type = " +
                "'l2vpn:vpws-instance-type') or " +
                "(not(boolean(/pw:pseudowires" +
                    "/pw:pseudowire[pw:name = current()]" +
                    "/vccv-ability)) and " +
                "not(boolean(/pw:pseudowires" +
                    "/pw:pseudowire[pw:name = current()]" +
                    "/request-vlanid)) and " +
                "not(boolean(/pw:pseudowires" +
                    "/pw:pseudowire[pw:name = current()]" +
                    "/vlan-tpid)) and " +
                "not(boolean(/pw:pseudowires" +
                    "/pw:pseudowire[pw:name = current()]" +
                    "/ttl)))" {
                description "Only a VPWS PW has parameters " +
                    "vccv-ability, request-vlanid, " +
                    "vlan-tpid, and ttl";
            }
        }
        description "Pseudowire name";
    }
    leaf state {
        type leafref {
            path "/pw:pseudowires" +
                "/pw:pseudowire[pw:name=current()]/../name]" +
                "/pw:state";
        }
        config false;
        description "Pseudowire state";
    }
}

```

```

        description "An L2VPN instance's pseudowire list";
    }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "primary/primary-ac" {
    description "Augment for primary-ac";
    container primary-ac {
        description "Primary AC";
        leaf name {
            type if:interface-ref;
            description "Name of attachment circuit";
        }
        leaf state {
            type operational-state-type;
            config false;
            description "attachment circuit up/down state";
        }
    }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "primary/primary-pw" {
    description "Augment for primary-pw";
    list primary-pw {
        key "name";
        leaf name {
            type pw:pseudowire-ref;
            must "(../../../../../type = " +
                "'l2vpn:vpws-instance-type') or " +
                "(not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /vccv-ability)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /request-vlanid)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /vlan-tpid)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /ttl)))" {
                description "Only a VPWS PW has parameters " +
                    "vccv-ability, request-vlanid, " +
                    "vlan-tpid, and ttl";
            }
        }
        description "Pseudowire name";
    }
    leaf state {
        type leafref {

```

```

        path "/pw:pseudowires" +
            "/pw:pseudowire[pw:name=current()/../name]" +
            "/pw:state";
    }
    config false;
    description "Pseudowire state";
}
description "An L2VPN instance's pseudowire list";
}
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "backup/backup-ac" {
    description "Augment for backup-ac";
    container backup-ac {
        description "Backup AC";
        leaf name {
            type if:interface-ref;
            description "Name of attachment circuit";
        }
        leaf state {
            type operational-state-type;
            config false;
            description "attachment circuit up/down state";
        }
    }
}
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "backup/backup-pw" {
    description "Augment for backup-pw";
    list backup-pw {
        key "name";
        leaf name {
            type pw:pseudowire-ref;
            must "(../../../../../type = " +
                "'l2vpn:vpws-instance-type') or " +
                "(not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /vccv-ability)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /request-vlanid)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /vlan-tpid)) and " +
                "not(boolean(/pw:pseudowires" +
                "    /pw:pseudowire[pw:name = current()]" +
                "    /ttl))" {
            description "Only a VPWS PW has parameters " +

```

```

        "vccv-ability, request-vlanid, " +
        "vlan-tpid, and ttl";
    }
    description "Pseudowire name";
}
leaf state {
    type leafref {
        path "/pw:pseudowires" +
            "/pw:pseudowire[pw:name=current()/../name]" +
            "/pw:state";
    }
    config false;
    description "Pseudowire state";
}
description "A list of backup pseudowires";
}
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp" {
    description "Augment for redundancy group properties";
    leaf template {
        type redundancy-group-template-ref;
        description "Reference a redundancy group " +
            "properties template";
    }
    uses redundancy-group-properties-grp;
}
}
}
}
}

augment "/pw:pseudowires/pw:pseudowire" {
    description "Augment for pseudowire parameters for " +
        "VPWS pseudowires";
    leaf vccv-ability {
        type boolean;
        description "vccvability";
    }
    leaf request-vlanid {
        type uint16;
        description "request vlanid";
    }
    leaf vlan-tpid {
        type string;
        description "vlan tpid";
    }
    leaf ttl {
        type uint8;
    }
}

```

```

        description "time-to-live";
    }
}

augment "/pw:pseudowires/pw:pseudowire/pw:pw-type" {
    description "Additional pseudowire types";
    case bgp-pw {
        container bgp-pw {
            description "BGP pseudowire";
            leaf remote-pe-id {
                type inet:ip-address;
                description "remote pe id";
            }
        }
    }
    case bgp-ad-pw {
        container bgp-ad-pw {
            description "BGP auto-discovery pseudowire";
            leaf remote-ve-id {
                type uint16;
                description "remote ve id";
            }
        }
    }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
    "/l2vpn:l2vpn" {
    when "l2vpn:type = 'l2vpn:vpws-instance-type'" {
        description "Constraints only for VPWS pseudowires";
    }
    description "Augment for VPWS instance";
    container vpws-constraints {
        must "(count(..endpoint) <= 2) and " +
            "(count(..endpoint/pw) <= 1) and " +
            "(count(..endpoint/ac) <= 1) and " +
            "(count(..endpoint/primary-pw) <= 1) and " +
            "(count(..endpoint/backup-pw) <= 1) " {
            description "A VPWS L2VPN instance has at most 2 endpoints " +
                "and each endpoint has at most 1 pseudowire or " +
                "1 attachment circuit";
        }
        description "VPWS constraints";
    }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
    "/l2vpn:l2vpn" {

```

```
    when "l2vpn:type = 'l2vpn:vpls-instance-type'" {
      description "Parameters specifically for a VPLS instance";
    }
    description "Augment for parameters for a VPLS instance";
    uses pbb-parameters-grp;
  }

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
  "/l2vpn:l2vpn/l2vpn:endpoint" {
  when "../l2vpn:type = 'l2vpn:vpls-instance-type'" {
    description "Endpoint parameter specifically for " +
      "a VPLS instance";
  }
  description "Augment for endpoint parameters for a VPLS instance";
  leaf split-horizon-group {
    type string;
    description "Identify a split horizon group";
  }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
  "/l2vpn:l2vpn/l2vpn:endpoint" +
  "/l2vpn:ac-or-pw-or-redundancy-grp" +
  "/l2vpn:redundancy-grp/l2vpn:backup" +
  "/l2vpn:backup-pw/l2vpn:backup-pw" {
  when "../..//l2vpn:type = 'l2vpn:vpls-instance-type'" {
    description "Backup pseudowire parameter specifically for " +
      "a VPLS instance";
  }
  description "Augment for backup pseudowire paramters for " +
    "a VPLS instance";
  leaf precedence {
    type uint32;
    description "precedence of the pseudowire";
  }
}

/* Notifications */

notification l2vpn-state-change-notification {
  description "L2VPN and constituents state change notification";
  leaf l2vpn-instance-name {
    type l2vpn-instance-name-ref;
    description "The L2VPN instance name";
  }
  leaf l2vpn-instance-type {
    type leafref {
      path "/ni:network-instances" +

```

```

        "/ni:network-instance" +
          "[ni:name=current()/../l2vpn-instance-name]" +
          "/l2vpn:type";
    }
    description "The L2VPN instance type";
}
leaf endpoint {
    type leafref {
        path "/ni:network-instances" +
            "/ni:network-instance" +
            "[ni:name=current()/../l2vpn-instance-name]" +
            "/l2vpn:endpoint/l2vpn:name";
    }
    description "The endpoint";
}
uses endpoint-grp {
    augment "ac-or-pw-or-redundancy-grp/ac" {
        description "Augment for attachment circuit(s) " +
            "as an endpoint";
        leaf ac {
            type leafref {
                path "/ni:network-instances" +
                    "/ni:network-instance" +
                    "[ni:name=current()/../l2vpn-instance-name]" +
                    "/l2vpn:endpoint" +
                    "[l2vpn:name=current()/../endpoint]" +
                    "/l2vpn:ac/l2vpn:name";
            }
            description "Related attachment circuit";
        }
    }
}
augment "ac-or-pw-or-redundancy-grp/pw" {
    description "Augment for pseudowire(s) as an endpoint";
    leaf pw {
        type leafref {
            path "/ni:network-instances" +
                "/ni:network-instance" +
                "[ni:name=current()/../l2vpn-instance-name]" +
                "/l2vpn:endpoint[l2vpn:name=current()/../endpoint]" +
                "/l2vpn:pw/l2vpn:name";
        }
        description "Related pseudowire";
    }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
    "primary/primary-ac" {
    description "Augment for primary-ac";
    leaf primary-ac {

```

```

    type leafref {
      path "/ni:network-instances" +
        "/ni:network-instance" +
          "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:endpoint" +
          "[l2vpn:name=current()/../endpoint]" +
        "/l2vpn:primary-ac/l2vpn:name";
    }
    description "Related primary attachment circuit";
  }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
  "primary/primary-pw" {
  description "Augment for primary-pw";
  leaf primary-pw {
    type leafref {
      path "/ni:network-instances" +
        "/ni:network-instance" +
          "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:endpoint" +
          "[l2vpn:name=current()/../endpoint]" +
        "/l2vpn:primary-pw/l2vpn:name";
    }
    description "Related primary pseudowire";
  }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
  "backup/backup-ac" {
  description "Augment for backup-ac";
  leaf backup-ac {
    type leafref {
      path "/ni:network-instances" +
        "/ni:network-instance" +
          "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:endpoint" +
          "[l2vpn:name=current()/../endpoint]" +
        "/l2vpn:backup-ac/l2vpn:name";
    }
    description "Related backup attachment circuit";
  }
}
augment "ac-or-pw-or-redundancy-grp/redundancy-grp/" +
  "backup/backup-pw" {
  description "Augment for backup-pw";
  leaf backup-pw {
    type leafref {
      path "/ni:network-instances" +
        "/ni:network-instance" +

```



```
        "[ni:name=current()/../l2vpn-instance-name]" +
        "/l2vpn:endpoint" +
        "[l2vpn:name=current()/../endpoint]" +
        "/l2vpn:backup-pw/l2vpn:name";
    }
    description "Related backup pseudowire";
}
}
}
leaf state {
    type identityref {
        base l2vpn-notification-state;
    }
    description "State change notification";
}
}
}
}
<CODE ENDS>
```

Figure 3

5. Security Considerations

The configuration, state, action and notification data defined in this document are designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the mandatory-to-implement secure transport is SSH [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and content.

The security concerns listed above are, however, no different than faced by other routing protocols. Hence, this draft does not change any underlying security issues inherent in [I-D.ietf-netmod-routing-cfg]

6. IANA Considerations

None.

7. Acknowledgments

The authors would like to acknowledge Giles Heron and others for their useful comments.

MITRE has approved this document for Public Release, Distribution Unlimited, with Public Release Case Number 19-0683.

8. References

8.1. Normative References

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

8.2. Informative References

- [RFC3916] Xiao, X., Ed., McPherson, D., Ed., and P. Pate, Ed., "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, DOI 10.17487/RFC3916, September 2004, <<https://www.rfc-editor.org/info/rfc3916>>.
- [RFC3985] Bryant, S., Ed. and P. Pate, Ed., "Pseudo Wire Emulation Edge-to-Edge (PWE3) Architecture", RFC 3985, DOI 10.17487/RFC3985, March 2005, <<https://www.rfc-editor.org/info/rfc3985>>.
- [RFC4385] Bryant, S., Swallow, G., Martini, L., and D. McPherson, "Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN", RFC 4385, DOI 10.17487/RFC4385, February 2006, <<https://www.rfc-editor.org/info/rfc4385>>.
- [RFC4446] Martini, L., "IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3)", BCP 116, RFC 4446, DOI 10.17487/RFC4446, April 2006, <<https://www.rfc-editor.org/info/rfc4446>>.
- [RFC4448] Martini, L., Ed., Rosen, E., El-Aawar, N., and G. Heron, "Encapsulation Methods for Transport of Ethernet over MPLS Networks", RFC 4448, DOI 10.17487/RFC4448, April 2006, <<https://www.rfc-editor.org/info/rfc4448>>.
- [RFC4664] Andersson, L., Ed. and E. Rosen, Ed., "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, DOI 10.17487/RFC4664, September 2006, <<https://www.rfc-editor.org/info/rfc4664>>.
- [RFC4665] Augustyn, W., Ed. and Y. Serbest, Ed., "Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks", RFC 4665, DOI 10.17487/RFC4665, September 2006, <<https://www.rfc-editor.org/info/rfc4665>>.

- [RFC4761] Kompella, K., Ed. and Y. Rekhter, Ed., "Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, DOI 10.17487/RFC4761, January 2007, <<https://www.rfc-editor.org/info/rfc4761>>.
- [RFC4762] Lasserre, M., Ed. and V. Kompella, Ed., "Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling", RFC 4762, DOI 10.17487/RFC4762, January 2007, <<https://www.rfc-editor.org/info/rfc4762>>.
- [RFC5003] Metz, C., Martini, L., Balus, F., and J. Sugimoto, "Attachment Individual Identifier (AII) Types for Aggregation", RFC 5003, DOI 10.17487/RFC5003, September 2007, <<https://www.rfc-editor.org/info/rfc5003>>.
- [RFC5254] Bitar, N., Ed., Bocci, M., Ed., and L. Martini, Ed., "Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3)", RFC 5254, DOI 10.17487/RFC5254, October 2008, <<https://www.rfc-editor.org/info/rfc5254>>.
- [RFC5659] Bocci, M. and S. Bryant, "An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge", RFC 5659, DOI 10.17487/RFC5659, October 2009, <<https://www.rfc-editor.org/info/rfc5659>>.
- [RFC6073] Martini, L., Metz, C., Nadeau, T., Bocci, M., and M. Aissaoui, "Segmented Pseudowire", RFC 6073, DOI 10.17487/RFC6073, January 2011, <<https://www.rfc-editor.org/info/rfc6073>>.
- [RFC6074] Rosen, E., Davie, B., Radoaca, V., and W. Luo, "Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs)", RFC 6074, DOI 10.17487/RFC6074, January 2011, <<https://www.rfc-editor.org/info/rfc6074>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.

- [RFC6391] Bryant, S., Ed., Filsfils, C., Drafz, U., Kompella, V., Regan, J., and S. Amante, "Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network", RFC 6391, DOI 10.17487/RFC6391, November 2011, <<https://www.rfc-editor.org/info/rfc6391>>.
- [RFC6423] Li, H., Martini, L., He, J., and F. Huang, "Using the Generic Associated Channel Label for Pseudowire in the MPLS Transport Profile (MPLS-TP)", RFC 6423, DOI 10.17487/RFC6423, November 2011, <<https://www.rfc-editor.org/info/rfc6423>>.
- [RFC6478] Martini, L., Swallow, G., Heron, G., and M. Bocci, "Pseudowire Status for Static Pseudowires", RFC 6478, DOI 10.17487/RFC6478, May 2012, <<https://www.rfc-editor.org/info/rfc6478>>.
- [RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <<https://www.rfc-editor.org/info/rfc6536>>.
- [RFC6624] Kompella, K., Kothari, B., and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, DOI 10.17487/RFC6624, May 2012, <<https://www.rfc-editor.org/info/rfc6624>>.
- [RFC7041] Balus, F., Ed., Sajassi, A., Ed., and N. Bitar, Ed., "Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging", RFC 7041, DOI 10.17487/RFC7041, November 2013, <<https://www.rfc-editor.org/info/rfc7041>>.
- [RFC7361] Dutta, P., Balus, F., Stokes, O., Calvignac, G., and D. Fedyk, "LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS)", RFC 7361, DOI 10.17487/RFC7361, September 2014, <<https://www.rfc-editor.org/info/rfc7361>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8077] Martini, L., Ed. and G. Heron, Ed., "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", STD 84, RFC 8077, DOI 10.17487/RFC8077, February 2017, <<https://www.rfc-editor.org/info/rfc8077>>.

Appendix A. Example Configuration

This section shows an example configuration using the YANG data model defined in the document.

Appendix B. Contributors

The editors gratefully acknowledge the following people for their contributions to this document.

Reshad Rahman
Cisco Systems, Inc.
Email: rrahman@cisco.com

Kamran Raza
Cisco Systems, Inc.
Email: skraza@cisco.com

Giles Heron
Cisco Systems, Inc.
Email: giheron@cisco.com

Tapraj Singh
Cisco Systems, Inc.
Email: tsingh@cisco.com

Zhenbin Li
Huawei Technologies
Email: lizhenbin@huawei.com

Zhuang Shunwan
Huawei Technologies
Email: Zhuangshunwan@huawei.com

Wang Haibo
Huawei Technologies
Email: rainsword.wang@huawei.com

Sajjad Ahmed
Ericsson
Email: sajjad.ahmed@ericsson.com

Matthew Bocci
Nokia
Email: matthew.bocci@nokia.com

Jorge Rabadan
Nokia

Email: jorge.rabadan@nokia.com

Jonathan Hardwick
Metaswitch
Email: jonathan.hardwick@metaswitch.com

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

Nick Delregno
Verizon
Email: nick.deregn@verizon.com

Luay Jalil
Verizon
Email: luay.jalil@verizon.com

Maria Joecylyn
Verizon
Email: joecylyn.malit@verizon.com

Figure 4

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Ing-When Chen
The MITRE Corporation

Email: ingwherchen@mitre.org

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Kishore Tiruveedhula
Juniper Networks

Email: kishoret@juniper.net

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 22, 2019

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
Oct 19, 2018

Yang Data Model for BGP/MPLS L3 VPNs
draft-ietf-bess-l3vpn-yang-04.txt

Abstract

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

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 April 22, 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.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	3
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	5
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
8.1. Normative References	20
8.2. Informative References	21
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.

1.1. 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].

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 to 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-defini
tions/policy-definition/name
          +--rw import-from-global
            +--rw enable?            boolean
            +--rw advertise-as-vpn?  boolean
            +--rw route-policy?     -> /rt-pol:routing-policy/policy-de
finitions/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 t
o 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

8.1. Normative References

[I-D.ietf-idr-bgp-model]

Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Zhdankin, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", draft-ietf-idr-bgp-model-02 (work in progress), July 2016.

[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., Liu, X., and A. Shaikh, "A YANG Data Model for Routing Policy Management", draft-ietf-rtgwg-policy-model-02 (work in progress), March 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>>.

[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>>.

8.2. Informative References

- [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>>.
- [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>>.
- [RFC7951] Lhotka, L., "JSON Encoding of Data Modeled with YANG", RFC 7951, DOI 10.17487/RFC7951, August 2016, <<https://www.rfc-editor.org/info/rfc7951>>.
- [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>>.

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
Huawei Bld., No.156 Beiqing Rd.
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

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 3, 2017

A. Farrel
J. Drake
E. Rosen
Juniper Networks
J. Uttaro
AT&T
L. Jalil
Verizon
October 30, 2016

BGP Control Plane for NSH SFC
draft-mackie-bess-nsh-bgp-control-plane-01

Abstract

This document describes the use of BGP as a control plane for networks that support Service Function Chaining (SFC). The document introduces a new BGP address family called the SFC AFI/SAFI with two route types. One route type is originated by a node to advertise that it hosts a particular instance of a specified service function. This route type also provides "instructions" on how to send a packet to the hosting node in a way that indicates that the service function has to be applied to the packet. The other route type is used by a Controller to advertise the paths of "chains" of service functions, and to give a unique designator to each such path so that they can be used in conjunction with the Network Service Header.

This document adopts the SFC architecture described in RFC 7665.

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 May 3, 2017.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction	3
1.1.	Terminology	4
2.	Overview	5
2.1.	Functional Overview	5
2.2.	Control Plane Overview	7
3.	BGP SFC Routes	9
3.1.	Service Function Instance Route (SFIR)	10
3.2.	Service Function Path Route (SFPR)	10
3.2.1.	The SFP Attribute	11
3.2.2.	General Rules For The SFP Attribute	15
4.	Mode of Operation	16
4.1.	Route Targets	16
4.2.	Service Function Instance Routes	17
4.3.	Service Function Path Routes	17
4.4.	Classifier Operation	19
4.5.	Service Function Forwarder Operation	19
5.	Selection in Service Function Paths	20
6.	Looping, Jumping, and Branching	21
6.1.	Protocol Control of Looping, Jumping, and Branching	22
6.2.	Implications for Forwarding State	23
7.	Advanced Topics	23
7.1.	Preserving Entropy	23
7.2.	Correlating Service Function Path Instances	24
7.3.	VPN Considerations and Private Service Functions	24
8.	Examples	25
8.1.	Example Explicit SFP With No Choices	26

8.2.	Example SFP With Choice of SFIs	27
8.3.	Example SFP With Open Choice of SFIs	28
8.4.	Example SFP With Choice of SFTs	28
8.5.	Example Correlated Bidirectional SFPs	29
8.6.	Example Correlated Asymmetrical Bidirectional SFPs	29
8.7.	Example Looping in an SFP	30
8.8.	Example Branching in an SFP	31
9.	Security Considerations	31
10.	IANA Considerations	32
10.1.	New BGP AF/SAFI	32
10.2.	New BGP Path Attribute	32
10.3.	New SFP Attribute TLVs Type Registry	32
10.4.	New SFP Association Type Registry	33
10.5.	New Service Function Type Registry	34
11.	Contributors	34
12.	Acknowledgements	35
13.	References	35
13.1.	Normative References	35
13.2.	Informative References	36
	Authors' Addresses	36

1. Introduction

As described in [RFC7498], the delivery of end-to-end services can require a packet to pass through a series of Service Functions (SFs) (e.g., classifiers, firewalls, TCP accelerators, and server load balancers) in a specified order: this is termed "Service Function Chaining" (SFC). There are a number of issues associated with deploying and maintaining service function chaining in production networks, which are described below.

Conventionally, if a packet needs to travel through a particular service chain, the nodes hosting the service functions of that chain are placed in the network topology in such a way that the packet cannot reach its ultimate destination without first passing through all the service functions in the proper order. This need to place the service functions at particular topological locations limits the ability to adapt a service function chain to changes in network topology (e.g., link or node failures), network utilization, or offered service load. These topological restrictions on where the service functions can be placed raise the following issues:

1. The process of configuring or modifying a service function chain is operationally complex and may require changes to the network topology.
2. Alternate or redundant service functions may need to be co-located with the primary service functions.

3. When there is more than one path between source and destination, forwarding may be asymmetric and it may be difficult to support bidirectional service function chains using simple routing methodologies and protocols without adding mechanisms for traffic steering or traffic engineering.

In order to address these issues, the SFC architecture includes Service Function Chains that are built in their own overlay network (the service function overlay network), coexisting with other overlay networks, over a common underlay network [RFC7665]. A Service Function Chain is a sequence of Service Functions through which packet flows satisfying specified criteria will pass.

This document describes the use of BGP as a control plane for networks that support Service Function Chaining (SFC). The document introduces a new BGP address family called the SFC AFI/SAFI with two route types. One route type is originated by a node to advertise that it hosts a particular instance of a specified service function. This route type also provides "instructions" on how to send a packet to the hosting node in a way that indicates that the service function has to be applied to the packet. The other route type is used by a Controller to advertise the paths of "chains" of service functions, and to give a unique designator to each such path so that they can be used in conjunction with the Network Service Header.

This document adopts the SFC architecture described in [RFC7665].

1.1. Terminology

This document uses the following terms from [RFC7665]:

- o Bidirectional Service Function Chain
- o Classifier
- o Service Function (SF)
- o Service Function Chain (SFC)
- o Service Function Forwarder (SFF)
- o Service Function Instance (SFI)
- o Service Function Path (SFP)
- o SFC branching

Additionally, this document uses the following terms from [I-D.ietf-sfc-nsh]:

- o Network Service Header (NSH)
- o Service Index (SI)
- o Service Path Identifier (SPI)

This document introduces the following terms:

- o Service Function Instance Route (SFIR)
- o Service Function Overlay Network
- o Service Function Path Route (SFPR)
- o Service Function Type (SFT)

2. Overview

2.1. Functional Overview

In [I-D.ietf-sfc-nsh] a Service Function Chain (SFC) is an ordered list of Service Functions (SFs). A Service Function Path (SFP) is an indication of which instances of SFs are acceptable to be traversed in an instantiation of an SFC in a service function overlay network. The Service Path Identifier (SPI) is a 24-bit number that identifies a specific SFP, and a Service Index (SI) is an 8-bit number that identifies a specific point in that path. In the context of a particular SFP (identified by an SPI), an SI represents a particular Service Function, and indicates the order of that SF in the SFP.

In fact, each SI is mapped to one or more SFs that are implemented by one or more Service Function Instances (SFIs) that support those specified SFs. Thus an SI may represent a choice of SFIs of one or more Service Function Types. By deploying multiple SFIs for a single SF, one can provide load balancing and redundancy.

A special Service Function, called a Classifier, is located at each ingress point to a service function overlay network. It assigns the packets of a given packet flow to a specific Service Function Path. This may be done by comparing specific fields in a packet's header with local policy, which may be customer/network/service specific. The classifier picks an SFP and sets the SPI accordingly it then sets the SI to the value of the SI for the first hop in the SFP and then prepending a Network Services Header (NSH) [I-D.ietf-sfc-nsh], to that packet containing the assigned SPI/SI. Note that the Classifier

and the node that hosts the first Service Function in a Service Function Path need not be located at the same point in the service function overlay network.

Note that the presence of the NSH can make it difficult for nodes in the underlay network to locate the fields in the original packet that would normally be used to constrain equal cost multipath (ECMP) forwarding. Therefore, it is recommended, as described in Section 7.1, that the node prepending the NSH also provide some form of entropy indicator that can be used in the underlay network.

The Service Function Forwarder (SFF) receives a packet from the previous node in a Service Function Path, removes the packet's link layer or tunnel encapsulation and hands the packet and the NSH to the Service Function Instance for processing.

When the SFF receives the packet and the NSH back from the SFI it must select the next SFI along the path using the SPI and SI in the NSH and potentially choosing between multiple SFIs (possibly of different Service Function Types) as described in Section 5. In the normal case the SPI remains unchanged and the SI will have been decremented to indicate the next SF along the path. But other possibilities exist if the SF makes other changes to the NSH through a process of re-classification:

- o The SI in the NSH may indicate:
 - * A previous SF in the path: known as "looping" (see Section 6).
 - * An SF further down the path: known as "jumping" (see also Section 6).
- o The SPI and the SI may point to an SF on a different SFP: known as "branching" (see also Section 6).

Such modifications are limited to within the same service function overlay network. That is, an SPI is known within the scope of service function overlay network. Furthermore, the new SI value is interpreted in the context of the SFP identified by the SPI, and SI values that do not form part of the definition of the path are invalid.

An unknown or invalid SPI/SI combination SHALL be treated as an error and the SFF MUST drop the packet. Such errors SHOULD be logged, and such logs MUST be subject to rate limits. See [I-D.ietf-sfc-nsh] for more details of handling this situation in received NSH packets.

The SFF then selects an SFI that provides the SF denoted by the SPI/SI, and forwards the packet to the SFF that supports that SFI.

2.2. Control Plane Overview

To accomplish the function described in Section 2.1, this document introduces a new BGP AFI/SAFI [values to be assigned by IANA] for "SFC Routes". Two SFC Route Types are defined by this document: the Service Function Instance Route (SFIR), and the Service Function Path Route (SFPR). As detailed in Section 3, the route type is indicated by a sub-field in the NLRI.

- o The SFIR is advertised by the node hosting the service function instance. The SFIR describes a particular instance of a particular Service Function and the way to forward a packet to it through the underlay network, i.e., IP address and encapsulation information.
- o The SFPRs are originated by Controllers. One SFPR is originated for each Service Function Path. The SFPR specifies:
 - A. the SPI of the path
 - B. the sequence of SFTs and/or SFIs of which the path consists
 - C. for each such SFT or SFI, the SI that represents it in the identified path.

This approach assumes that there is an underlay network that provides connectivity between SFFs and Controllers, and that the SFFs are grouped to form one or more service function overlay networks through which SFPs are built. We assume BGP connectivity between the Controllers and all SFFs within each service function overlay network.

In addition, we also introduce the Service Function Type (SFT) that is the category of SF that is supported by an SFF (such as "firewall"). An IANA registry of Service Function Types is introduced in Section 10. An SFF may support SFs of multiple different SFTs, and may support multiple SFIs of each SF.

When choosing the next SFI in a path, the SFF uses the SPI and SI as well as the SFT to choose among the SFIs, applying, for example, a load balancing algorithm or direct knowledge of the underlay network topology as described in Section 4.

The SFF then encapsulates the packet using the encapsulation specified by the SFIR of the selected SFI and forwards the packet. See Figure 1.

Thus the SFF can be seen as a portal in the underlay network through which a particular SFI is reached.

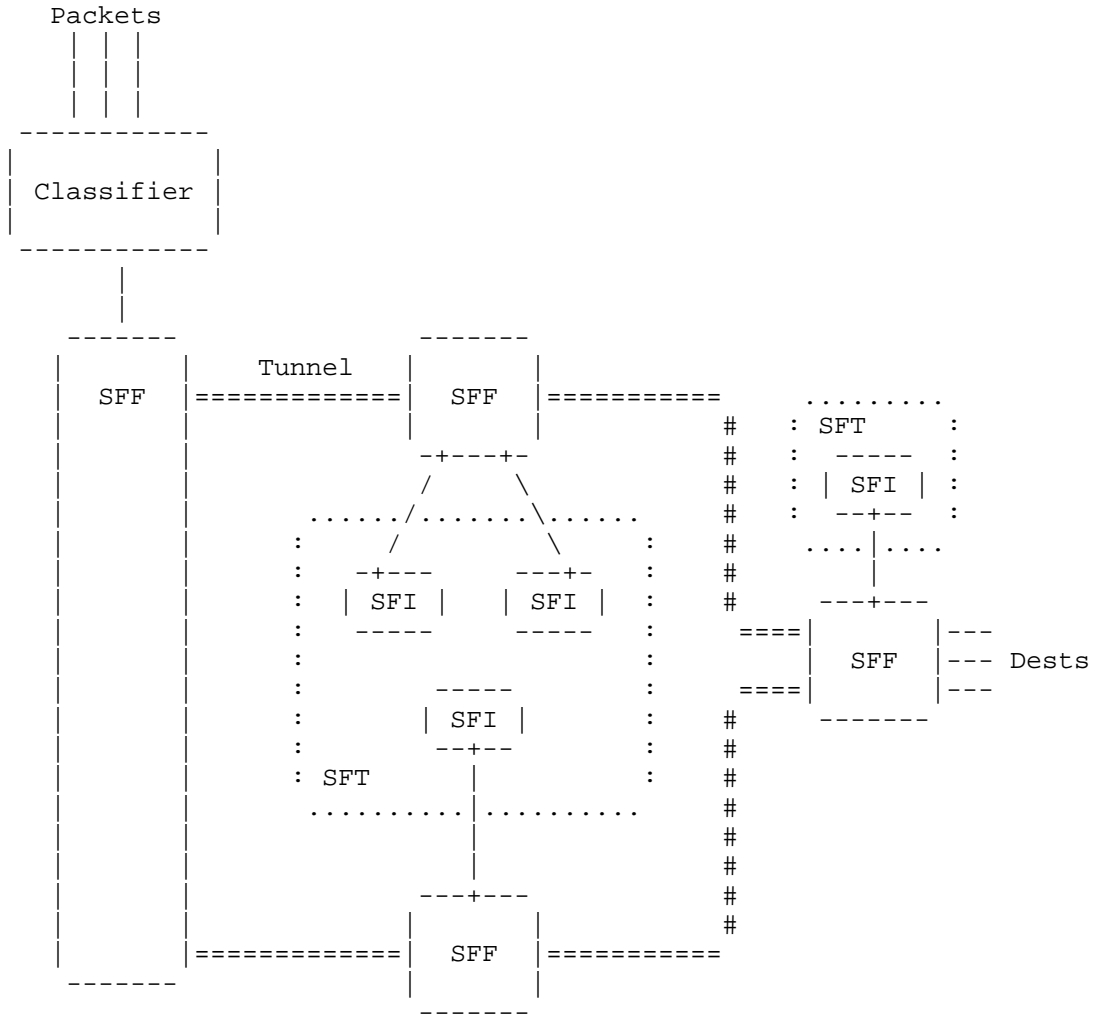


Figure 1: The SFC Architecture Reference Model

3. BGP SFC Routes

This document defines a new AFI/SAFI for BGP, known as "SFC", with an NLRI that is described in this section.

The format of the SFC NLRI is shown in Figure 2.

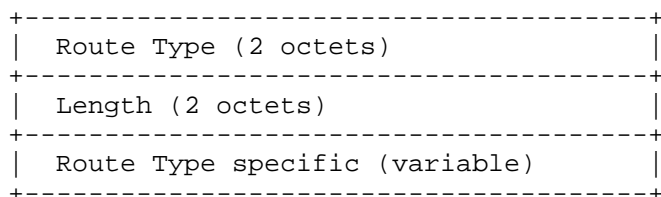


Figure 2: The Format of the SFC NLRI

The Route Type field determines the encoding of the rest of the route type specific SFC NLRI.

The Length field indicates the length in octets of the route type specific field of the SFC NLRI.

This document defines the following Route Types:

1. Service Function Instance Route (SFIR)
2. Service Function Path Route (SFPR)

A Service Function Instance Route (SFIR) is used to identify an SFI. A Service Function Path Route (SFPR) defines a sequence of Service Functions (each of which has at least one instance advertised in an SFIR) that form an SFP.

The detailed encoding and procedures for these Route Types are described in subsequent sections.

The SFC NLRI is carried in BGP [RFC4271] using BGP Multiprotocol Extensions [RFC4760] with an Address Family Identifier (AFI) of TBD1 and a Subsequent Address Family Identifier (SAFI) of TBD2. The NLRI field in the MP_REACH_NLRI/MP_UNREACH_NLRI attribute contains the SFC NLRI, encoded as specified above.

In order for two BGP speakers to exchange SFC NLRIs, they must use BGP Capabilities Advertisements to ensure that they both are capable of properly processing such NLRIs. This is done as specified in

[RFC4760], by using capability code 1 (Multiprotocol BGP) with an AFI of TBD1 and a SAFI of TBD2.

3.1. Service Function Instance Route (SFIR)

Figure 3 shows the Route Type specific NLRI of the SFIR.

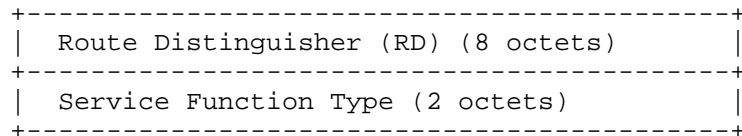


Figure 3: SFIR Route Type specific NLRI

Per [RFC4364] the RD field comprises a two byte Type field and a six byte Value field. Two SFIs of the same SFT must be associated with different RDs, where the association of an SFI with an RD is determined by provisioning. If two SFIRs are originated from different administrative domains, they must have different RDs. In particular, SFIRs from different VPNs (for different service function overlay networks) must have different RDs, and those RDs must be different from any non-VPN SFIRs.

The Service Function Type identifies a service function, e.g., classifier, firewall, load balancer, etc. There may be several SFIs that can perform a given Service Function. Each node hosting an SFI must originate an SFIR for each SFI that it hosts. The SFIR representing a given SFI will contain an NLRI with RD field set to an RD as specified above, and with SFT field set to identify that SFI's Service Function Type. The values for the SFT field are taken from a registry administered by IANA (see Section 10). A BGP Update containing one or more SFIRs will also include a Tunnel Encapsulation attribute [I-D.ietf-idr-tunnel-encaps]. If a data packet needs to be sent to an SFI identified in one of the SFIRs, it will be encapsulated as specified by the Tunnel Encapsulation attribute, and then transmitted through the underlay network.

3.2. Service Function Path Route (SFPR)

Figure 4 shows the Route Type specific NLRI of the SFPR.

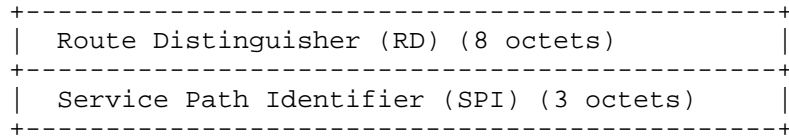


Figure 4: SFPR Route Type Specific NLRI

Per [RFC4364] the RD field comprises a two byte Type field and a six byte Value field. All SFPs must be associated with different RDs. The association of an SFP with an RD is determined by provisioning. If two SFPRs are originated from different Controllers they must have different RDs. Additionally, SFPRs from different VPNs (i.e., in different service function overlay networks) must have different RDs, and those RDs must be different from any non-VPN SFPRs.

The Service Path Identifier is defined in [I-D.ietf-sfc-nsh] and is the value to be placed in the Service Path Identifier field of the NSH header of any packet sent on this Service Function Path. It is expected that one or more Controllers will originate these routes in order to configure a service function overlay network.

The SFP is described in a new BGP Path attribute, the SFP attribute. Section 3.2.1 shows the format of that attribute.

3.2.1. The SFP Attribute

[RFC4271] defines the BGP Path attribute. This document introduces a new Path attribute called the SFP attribute with value TBD3 to be assigned by IANA. The first SFP attribute MUST be processed and subsequent instances MUST be ignored.

The common fields of the SFP attribute are set as follows:

- o Optional bit is set to 1 to indicate that this is an optional attribute.
- o The Transitive bit is set to 1 to indicate that this is a transitive attribute.
- o The Extended Length bit is set according to the length of the SFP attribute as defined in [RFC4271].
- o The Attribute Type Code is set to TBD3.

The content of the SFP attribute is a series of Type-Length-Variable (TLV) constructs. Each TLV may include sub-TLVs. All TLVs and sub-TLVs have a common format that is:

- o Type: A single octet indicating the type of the SFP attribute TLV. Values are taken from the registry described in Section 10.3.
- o Length: A two octet field indicating the length of the data following the Length field counted in octets.
- o Value: The contents of the TLV.

The formats of the TLVs defined in this document are shown in the following sections. The presence rules and meanings are as follows.

- o The SFP attribute contains a sequence of zero or more Association TLVs. That is, the Association TLV is optional. Each Association TLV provides an association between this SFPR and another SFPR. Each associated SFPR is indicated using the RD with which it is advertised (we say the SFPR-RD to avoid ambiguity).
- o The SFP attribute contains a sequence of one or more Hop TLVs. Each Hop TLV contains all of the information about a single hop in the SFP.
- o Each Hop TLV contains an SI value and a sequence of one or more SFT TLVs. Each SFT TLV contains an SFI reference for each instance of an SF that is allowed at this hop of the SFP for the specific SFT. Each SFI is indicated using the RD with which it is advertised (we say the SFIR-RD to avoid ambiguity).

3.2.1.1. The Association TLV

The Association TLV is an optional TLV in the SFP attribute. It may be present multiple times. Each occurrence provides an association with another SFP as advertised in another SFPR. The format of the Association TLV is shown in Figure 5

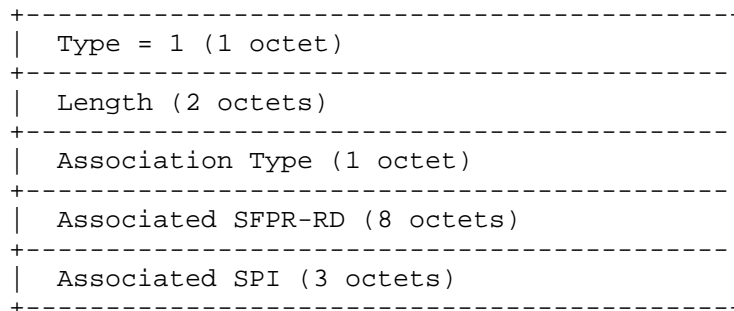


Figure 5: The Format of the Association TLV

The fields are as follows:

Type is set to 1 to indicate an Association TLV.

Length indicates the length in octets of the Association Type and Associated SFPR-RD fields. The value of the Length field is 12.

The Association Type field indicate the type of association. The values are tracked in an IANA registry (see Section 10.4). Only one value is defined in this document: type 1 indicates association of two unidirectional SFPs to form a bidirectional SFP. An SFP attribute SHOULD NOT contain more than one Association TLV with Association Type 1: if more than one is present, the first one MUST be processed and subsequent instances MUST be ignored. Note that documents that define new Association Types must also define the presence rules for Association TLVs of the new type.

The Associated SFPR-RD contains the RD of some other SFPR advertisement that contains the SFP with which this SFP is associated.

The Associated SPI contains the SPI of the associated SFP as advertised in the SFPR indicated by the Associated SFPR-RD field.

Association TLVs with unknown Association Type values SHOULD be ignored. Association TLVs that contain an Associated SFPR-RD value equal to the RD of the SFPR in which they are contained SHOULD be ignored. If the Associated SPI is not equal to the SPI advertised in the SFPR indicated by the Associated SFPR-RD then the Association TLV SHOULD be ignored.

Note that when two SFPRs reference each other using the Association TLV, one SFPR advertisement will be received before the other. Therefore, processing of an association MUST NOT be rejected simply because the Associated SFPR-RD is unknown.

Further discussion of correlation of SFPRs is provided in Section 7.2.

3.2.1.2. The Hop TLV

There is one Hop TLV in the SFP attribute for each hop in the SFP. The format of the Hop TLV is shown in Figure 6. At least one Hop TLV must be present in an SFP attribute.

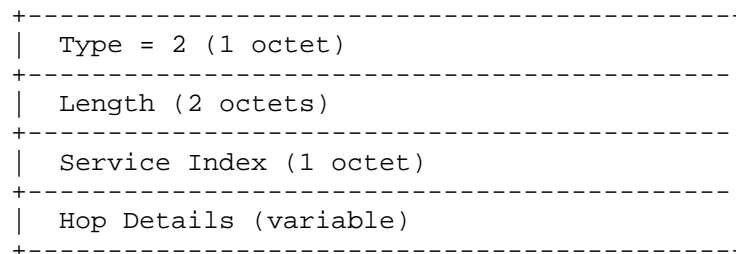


Figure 6: The Format of the Hop TLV

The fields are as follows:

Type is set to 2 to indicate a Hop TLV.

Length indicates the length in octets of the Service Index and Hop Details fields.

The Service Index is defined in [I-D.ietf-sfc-nsh] and is the value found in the Service Index field of the NSH header that an SFP will use to lookup to which next SFI a packet should be sent.

The Hop Details consist of a sequence of one or more SFT TLVs.

3.2.1.3. The SFT TLV

There is one or more SFT TLV in each Hop TLV. There is one SFT TLV for each SFT supported in the specific hop of the SFP. The format of the SFT TLV is shown in Figure 7.

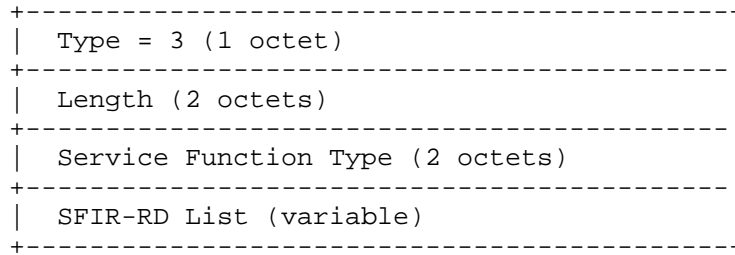


Figure 7: The Format of the SFT TLV

The fields are as follows:

Type is set to 3 to indicate an SFT TLV.

Length indicates the length in octets of the Service Function Type and SFIR-RD List fields.

The Service Function Type is used to identify a Service Function Instance Route in the service function overlay network which, in turn, will allow lookup of routes to SFIs implementing the SF. SFT values in the range 1-31 are Special Purpose SFT values and have meanings defined by the documents that describe them - the value 'Change Sequence' is defined in Section 6.1 of this document.

The SFIR-RD List is made up of one or more SFIR-RD values from the advertisements of SFIs in SFIRs. An SFIR-RD of value zero has special meaning as described in Section 5. Each entry in the list is 8 octets long, and the number of entries in the list can be deduced from the value of the Length field.

3.2.2. General Rules For The SFP Attribute

It is possible for the same SFI, as described by an SFIR, to be used in multiple SFPRs.

When two SFPRs have the same SPI but different SFPR-RDs there can be three cases:

- o Two or more Controllers are originating SFPRs for the same SFP. In this case the content of the SFPRs is identical and the duplication is to ensure receipt and to provide Controller redundancy.

- o There is a transition in content of the advertised SFP and the advertisements may originate from one or more Controllers. In this case the content of the SFPRs will be different.
- o The reuse of an SPI may result from a configuration error.

In all cases, there is no way for the receiving SFF to know which SFPR to process, and the SFPRs could be received in any order. At any point in time, when multiple SFPRs have the same SPI but different SFPR-RDs, the SFF MUST use the SFPR with the numerically lowest SFPR-RD. The SFF SHOULD log this occurrence to assist with debugging.

Furthermore, a Controller that wants to change the content of an SFP is RECOMMENDED to use a new SPI and so create a new SFP onto which the Classifiers can transition packet flows before the SFPR for the old SFP is withdrawn. This avoids any race conditions with SFPR advertisements.

Additionally, a Controller SHOULD NOT re-use an SPI after it has withdrawn the SFPR that used it until at least a configurable amount of time has passed. This timer SHOULD have a default of one hour.

4. Mode of Operation

This document describes the use of BGP as a control plane to create and manage a service function overlay network.

4.1. Route Targets

The main feature introduced by this document is the ability to create multiple service function overlay networks through the use of Route Targets (RTs) [RFC4364].

Every BGP UPDATE containing an SFIR or SFPR carries one or more RTs. The RT carried by a particular SFIR or SFPR is determined by the provisioning of the route's originator.

Every node in a service function overlay network is configured with one or more import RTs. Thus, each SFF will import only the SFPRs with matching RTs allowing the construction of multiple service function overlay networks or the instantiation of Service Function Chains within an L3VPN or EVPN instance (see Section 7.3). An SFF that has a presence in multiple service function overlay networks (i.e., imports more than one RT) may find it helpful to maintain separate forwarding state for each overlay network.

4.2. Service Function Instance Routes

The SFIR (see Section 3.1) is used to advertise the existence and location of a specific Service Function Instance and consists of:

- o The RT as just described.
- o A Service Function Type (SFT) that is the category of Service Function that is provided (such as "firewall").
- o A Route Distinguisher (RD) that is unique to a specific instance of a service function.

4.3. Service Function Path Routes

The SFPR (see Section 3.2) describes a specific path of a Service Function Chain. The SFPR contains the Service Path Identifier (SPI) used to identify the SFP in the NSH in the data plane. It also contains a sequence of Service Indexes (SIs). Each SI identifies a hop in the SFP, and each hop is a choice between one of more SFIs.

As described in this document, each Service Function Path Route is identified in the service function overlay network by an RD and an SPI. The SPI is unique across all service function overlay networks supported by the underlay network.

The SFPR advertisement comprises:

- o An RT as described in Section 4.1.
- o A tuple that identifies the SFPR
 - * An RD that identifies an advertisement of an SFPR.
 - * The SPI that uniquely identifies this path within all service function overlay networks supported by the underlay network. This SPI also appears in the NSH.
- o A series of Service Indexes. Each SI is used in the context of a particular SPI and identifies one or more SFs (distinguished by their SFTs) and for each SF a set of SFIs that instantiate the SF. The values of the SI indicate the order in which the SFs are to be executed in the SFP that is represented by the SPI.
- o The SI is used in the NSH to identify the entries in the SFP. Note that the SI values have meaning only relative to a specific path. They have no semantic other than to indicate the order of Service Functions within the path and are assumed to be

monotonically decreasing from the start to the end of the path [I-D.ietf-sfc-nsh].

- o Each Service Index is associated with a set of one or more Service Function Instances that can be used to provide the indexed Service Function within the path. Each member of the set comprises:
 - * The RD used in an SFIR advertisement of the SFI.
 - * The SFT that indicates the type of function as used in the same SFIR advertisement of the SFI.

This may be summarized as follows where the notations "SFPR-RD" and "SFIR-RD" are used to distinguish the two different RDs:

$$RT, \{SFPR-RD, SPI\}, m * \{SI, \{n * \{SFT, p * SFIR-RD\} \} \}$$

Where:

RT: Route Target

SFPR-RD: The Route Descriptor of the Service Function Path Route advertisement

SPI: Service Path Identifier used in the NSH

m: The number of hops in the Service Function Path

n: The number of choices of Service Function Type for a specific hop

p: The number of choices of Service Function Instance for given Service Function Type in a specific hop

SI: Service Index used in the NSH to indicate a specific hop

SFT: The Service Function Type used in the same advertisement of the Service Function Instance Route

SFIR-RD: The Route Descriptor used in an advertisement of the Service Function Instance Route

Note that the values of SI are from the set {255, ..., 1} and are monotonically decreasing within the SFP. SIs MUST appear in order within the SFPR (i.e., monotonically decreasing) and MUST NOT appear more than once. Malformed SFPRs MUST be discarded and MUST cause any previous instance of the SFPR (same SFPR-RD and SPI) to be discarded.

The choice of SFI is explained further in Section 5. Note that an SFIR-RD value of zero has special meaning as described in that Section.

4.4. Classifier Operation

As shown in Figure 1, the Classifier is a special Service Function that is used to assign packets to an SFP.

The Classifier is responsible for determining to which packet flow a packet belongs (usually by inspecting the packet header), imposing an NSH, and initializing the NSH to include the SPI of the selected SFPR and to include the SI from first hop of the selected SFP.

The Classifier may also provide an entropy indicator as described in Section 7.1.

4.5. Service Function Forwarder Operation

Each packet sent to an SFF is transmitted encapsulated in an NSH. The NSH includes an SPI and SI: the SPI indicates the SFPR advertisement that announced the Service Function Path; the tuple SPI/SI indicates a specific hop in a specific path and maps to the RD/SFT of a particular SFIR advertisement.

When an SFF gets an SFPR advertisement it will first determine whether to import the route by examining the RT. If the SFPR is imported the SFF then determines whether it is on the SFP by looking for its own SFIR-RDs in the SFPR. For each occurrence in the SFP, the SFF creates forwarding state for incoming packets and forwarding state for outgoing packets that have been processed by the specified SFI.

The SFF creates local forwarding state for packets that it receives from other SFFs. This state makes the association between the SPI/SI in the NSH of the received packet and one or more specific local SFIs as identified by the SFIR-RD/SFT. If there are multiple local SFIs that match this is because a single advertisement was made for a set of equivalent SFIs and the SFF may use local policy (such as load balancing) to determine to which SFI to forward a received packet.

The SFF also creates next hop forwarding state for packets received back from the local SFI that need to be forwarded to the next hop in the SFP. There may be a choice of next hops as described in Section 4.3. The SFF could install forwarding state for all potential next hops, or it could choose to only install forwarding state to a subset of the potential next hops. If a choice is made then it will be as described in Section 5.

The installed forwarding state may change over time reacting to changes in the underlay network and the availability of particular SFIs.

Note that SFFs only create and store forwarding state for the SFPs on which they are included. They do not retain state for all SFPs advertised.

An SFF may also install forwarding state to support looping, jumping, and branching. The protocol mechanism for explicit control of looping, jumping, and branching is described in Section 6.1 using a special value of the SFT within an entry in an SFPR.

5. Selection in Service Function Paths

As described in Section 2 the SPI/SI in the NSH passed back from an SFI to the SFF may leave the SFF with a choice of next hop SFTs, and a choice of SFIs for each SFT. That is, the SPI indicates an SFPR, and the SI indicates an entry in that SFPR. Each entry in an SFPR is a set of one or more SFT/SFIR-RD pairs. The SFF must choose one of these, identify the SFF that supports the chosen SFI, and send the packet to that next hop SFF.

In the typical case, the SFF chooses a next hop SFF by looking at the set of all SFFs that support the SFs identified by the SI (that set having been advertised in individual SFIR advertisements), finding the one or more that are "nearest" in the underlay network, and choosing between next hop SFFs using its own load-balancing algorithm.

An SFI may influence this choice process by passing additional information back along with the packet and NSH. This information may influence local policy at the SFF to cause it to favor a next hop SFF (perhaps selecting one that is not nearest in the underlay), or to influence the load-balancing algorithm.

This selection applies to the normal case, but also applies in the case of looping, jumping, and branching (see Section 6).

Suppose an SFF in a particular service overlay network (identified by a particular import RT, RT-z) needs to forward an NSH-encapsulated packet whose SPI is SPI-x and whose SI is SI-y. It does the following:

1. It looks for an installed SFPR that carries RT-z and that has SPI-x in its NLRI. If there is none, then such packets cannot be forwarded.

2. From the SFP attribute of that SFPR, it finds the Hop TLV with SI value set to SI-y. If there is no such Hop TLV, then such packets cannot be forwarded.
3. It then finds the "relevant" set of SFIRs by going through the list of of SFT TLVs contained in the Hop TLV as follows:
 - A. An SFIR is relevant if it carries RT-z, the SFT in its NLRI matches the SFT value in one of the SFT TLVs, and the RD value in its NLRI matches an entry in the list of SFIR-RDs in that SFT TLV.
 - B. If an entry in the SFIR-RD list of an SFT TLV contains the value zero, then an SFIR is relevant if it carries RT-z and the SFT in its NLRI matches the SFT value in that SFT TLV. I.e., any SFIR in the service function overlay network defined by RT-z and with the correct SFT is relevant.

Each of the relevant SFIRs identifies a single SFI, and contains a Tunnel Encapsulation attribute that specifies how to send a packet to that SFI. For a particular packet, the SFF chooses a particular SFI from the set of relevant SFIRs. This choice is made according to local policy.

A typical policy might be to figure out the set of SFIs that are closest, and to load balance among them. But this is not the only possible policy.

6. Looping, Jumping, and Branching

As described in Section 2 an SFI or an SFF may cause a packets to "loop back" to a previous SF on a path in order that a sequence of functions may be re-executed. This is simply achieved by replacing the SI in the NSH with a higher value instead of decreasing it as would normally be the case to determine the next hop in the path.

Section 2 also describes how an SFI or an SFF may cause a packets to "jump forward" to an SF on a path that is not the immediate next SF in the SFP. This is simply achieved by replacing the SI in the NSH with a lower value than would be achieved by decreasing it by the normal amount.

A more complex option to move packets from one SFP to another is described in [I-D.ietf-sfc-nsh] and Section 2 where it is termed "branching". This mechanism allows an SFI or SFF to make a choice of downstream treatments for packets based on local policy and output of the local SF. Branching is achieved by changing the SPI in the NSH

to indicate the new path and setting the SI to indicate the point in the path at which the packets should enter.

Note that the NSH does not include a marker to indicate whether a specific packet has been around a loop before. Therefore, the use of NSH metadata may be required in order to prevent infinite loops.

6.1. Protocol Control of Looping, Jumping, and Branching

If the SFT value in an SFT TLV in an SFPR has the Special Purpose SFT value "Change Sequence" (see Section 10) then this is an indication that the SFF may make a loop, jump, or branch according to local policy and information returned by the local SFI.

In this case, the SPI and SI of the next hop is encoded in the eight bytes of an entry in the SFIR-RD list as follows:

3 bytes SPI

2 bytes SI

3 bytes Reserved (SHOULD be set to zero and ignored)

If the SI in this encoding is not part of the SFPR indicated by the SPI in this encoding, then this is an explicit error that SHOULD be detected by the SFF when it parses the SFPR. The SFPR SHOULD NOT cause any forwarding state to be installed in the SFF and packets received with the SPI that indicates this SFPR SHOULD be silently discarded.

If the SPI in this encoding is unknown, the SFF SHOULD NOT install any forwarding state for this SFPR, but MAY hold the SFPR pending receipt of another SFPR that does use the encoded SPI.

If the SPI matches the current SPI for the path, this is a loop or jump. In this case, if the SI is greater than to the current SI it is a loop. If the SPI matches and the SI is less than the next SI, it is a jump.

If the SPI indicates another path, this is a branch and the SI indicates the point at which to enter that path.

The Change Sequence SFT is just another SFT that may appear in a set of SFI/SFT tuples within an SI and is selected as described in Section 5.

Note that Special Purpose SFTs MUST NOT be advertised in SFIRs.

6.2. Implications for Forwarding State

Support for looping and jumping requires that the SFF has forwarding state established to an SFF that provides access to an instance of the appropriate SF. This means that the SFF must have seen the relevant SFIR advertisements and known that it needed to create the forwarding state. This is a matter of local configuration and implementation: for example, an implementation could be configured to install forwarding state for specific looping/jumping.

Support for branching requires that the SFF has forwarding state established to an SFF that provides access to an instance of the appropriate entry SF on the other SFP. This means that the SFF must have seen the relevant SFIR and SFPR advertisements and known that it needed to create the forwarding state. This is a matter of local configuration and implementation: for example, an implementation could be configured to install forwarding state for specific branching (identified by SPI and SI).

7. Advanced Topics

This section highlights several advanced topics introduced elsewhere in this document.

7.1. Preserving Entropy

Forwarding decisions in the underlay network in the presence of equal cost multipath (ECMP) are usually made by inspecting key invariant fields in a packet header so that all packets from the same packet flow receive the same forwarding treatment. However, when an NSH is included in a packet, those key fields may be inaccessible. For example, the fields may be too far inside the packet for a forwarding engine to quickly find them and extract their values, or the node performing the examination may be unaware of the format and meaning of the NSH and so unable to parse far enough into the packet.

Various mechanisms exist within forwarding technologies to include an "entropy indicator" within a forwarded packet. For example, in MPLS there is the entropy label [RFC6790], while for encapsulations in UDP the source port field is often used to carry an entropy indicator (such as for MPLS in UDP [RFC7510]).

Implementations of this specification are RECOMMENDED to include an entropy indicator within the packet's underlay network header, and SHOULD preserve any entropy indicator from a received packet for use on the same packet when it is forwarded along the path but MAY choose to generate a new entropy indicator so long as the method used is constant for all packets. Note that preserving per packet entropy

may require that the entropy indicator is passed to and returned by the SFI to prevent the SFF from having to maintain per-packet state.

7.2. Correlating Service Function Path Instances

It is often useful to create bidirectional SFPs to enable packet flows to traverse the same set of SFs, but in the reverse order. However, packets on SFPs in the data plane (per [I-D.ietf-sfc-nsh]) do not contain a direction indicator, so each direction must use a different SPI.

As described in Section 3.2.1.1 an SFPR can contain one or more correlators encoded in Association TLVs. If the Association Type indicates "Bidirectional SFP" then the SFP advertised in the SFPR is one direction of a bidirectional pair of SFPs where the other in the pair is advertised in the SFPR with RD as carried in the Associated SFPR-RD field of the Association TLV. The SPI carried in the Associated SPI field of the Association TLV provides a cross-check and should match the SPI advertised in the SFPR with RD as carried in the Associated SFPR-RD field of the Association TLV.

As noted in Section 3.2.1.1 SFPRs reference each other one SFPR advertisement will be received before the other. Therefore processing of an association will require that the first SFPR is not rejected simply because the Associated SFPR-RD it carries is unknown. However, the SFP defined by the first SFPR is valid and SHOULD be available for use as a unidirectional SFP even in the absence of an advertisement of its partner.

Furthermore, in error cases where SFPR-a associates with SFPR-b, but SFPR-b associates with SFPR-c such that a bidirectional pair of SFPs cannot be formed, the individual SFPs are still valid and SHOULD be available for use as unidirectional SFPs. An implementation SHOULD log this situation because it represents a Controller error.

Usage of a bidirectional SFP may be programmed into the Classifiers by the Controller. Alternatively, a Classifier may look at incoming packets on a bidirectional packet flow, extract the SPI from the received NSH, and look up the SFPR to find the reverse direction SFP to use when it sends packets.

See Section 8 for an example of how this works.

7.3. VPN Considerations and Private Service Functions

Likely deployments include reserving specific instances of Service Functions for specific customers or allowing customers to deploy their own Service Functions within the network. Building Service

Functions in such environments requires that suitable identifiers are used to ensure that SFFs distinguish which SFIs can be used and which cannot.

This problem is similar to how VPNs are supported and is solved in a similar way. The RT field is used to indicate a set of Service Functions from which all choices must be made.

8. Examples

Assume we have a service function overlay network with four SFFs (SFF1, SFF2, SFF3, and SFF4). The SFFs have addresses in the underlay network as follows:

```
SFF1 192.0.2.1
SFF2 192.0.2.2
SFF3 192.0.2.3
SFF4 192.0.2.4
```

Each SFF provides access to some SFIs from the four Service Function Types SFT=41, SFT=42, SFT=43, and SFT=44 as follows:

```
SFF1 SFT=41 and SFT=42
SFF2 SFT=41 and SFT=43
SFF3 SFT=42 and SFT=44
SFF4 SFT=43 and SFT=44
```

The service function network also contains a Controller with address 198.51.100.1.

This example service function overlay network is shown in Figure 8.

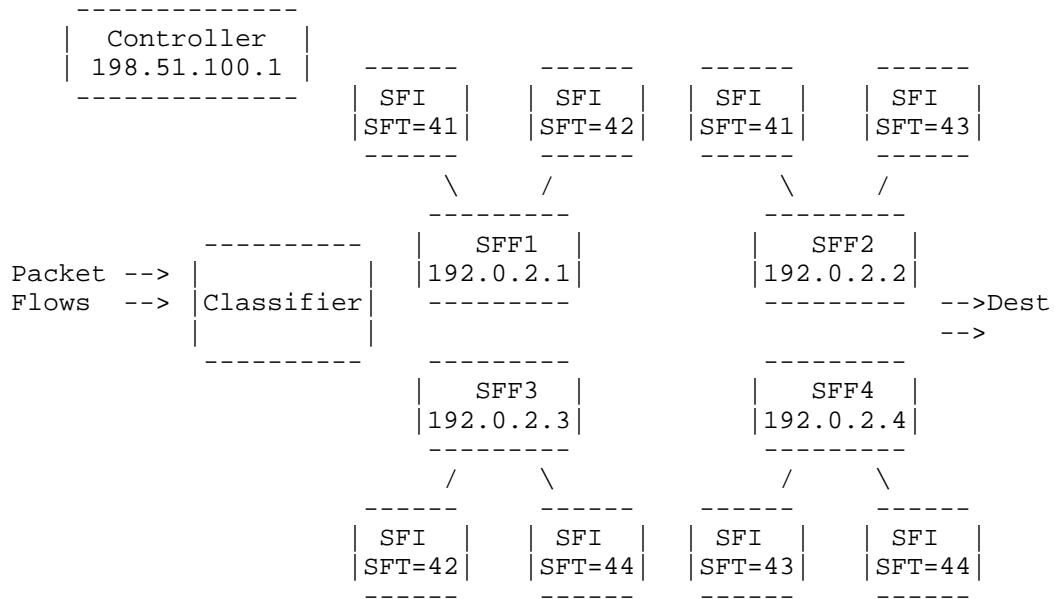


Figure 8: Example Service Function Overlay Network

The SFFs advertise routes to the SFIs they support. So we see the following SFIRs:

- RD = 192.0.2.1,1, SFT = 41
- RD = 192.0.2.1,2, SFT = 42
- RD = 192.0.2.2,1, SFT = 41
- RD = 192.0.2.2,2, SFT = 43
- RD = 192.0.2.3,7, SFT = 42
- RD = 192.0.2.3,8, SFT = 44
- RD = 192.0.2.4,5, SFT = 43
- RD = 192.0.2.4,6, SFT = 44

Note that the addressing used for communicating between SFFs is taken from the Tunnel Encapsulation attribute of the SFIR and not from the SFIR-RD.

8.1. Example Explicit SFP With No Choices

Consider the following SFPR.


```
SFP1:  RD = 198.51.100.1,101, SPI = 15,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
       [SI = 250, SFT = 43, RD = 192.0.2.2,2]
```

The Service Function Path consists of an SF of type 41 located at SFF1 followed by an SF of type 43 located at SFF2. This path is fully explicit and each SFF is offered no choice in forwarding packet along the path.

SFF1 will receive packets on the path from the Classifier and will identify the path from the SPI (15). The initial SI will be 255 and so SFF1 will deliver the packets to the SFI for SFT 41.

When the packets are returned to SFF1 by the SFI the SI will be decreased to 250 for the next hop. SFF1 has no flexibility in the choice of SFF to support the next hop SFI and will forward the packet to SFF2 which will send the packets to the SFI that supports SFT 43 before forwarding the packets to their destinations.

8.2. Example SFP With Choice of SFIs

```
SFP2:  RD = 198.51.100.1,102, SPI = 16,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,],  
       [SI = 250, SFT = 43, {RD = 192.0.2.2,2,  
                           RD = 192.0.2.4,5 } ]
```

In this example the path also consists of an SF of type 41 located at SFF1 and this is followed by an SF of type 43, but in this case the SI = 250 contains a choice between the SFI located at SFF2 and the SFI located at SFF4.

SFF1 will receive packets on the path from the Classifier and will identify the path from the SPI (16). The initial SI will be 255 and so SFF1 will deliver the packets to the SFI for SFT 41.

When the packets are returned to SFF1 by the SFI the SI will be decreased to 250 for the next hop. SFF1 now has a choice of next hop SFF to execute the next hop in the path. It can either forward packets to SFF2 or SFF4 to execute a function of type 43. It uses its local load balancing algorithm to make this choice. The chosen SFF will send the packets to the SFI that supports SFT 43 before forwarding the packets to their destinations.

8.3. Example SFP With Open Choice of SFIs

```
SFP3:  RD = 198.51.100.1,103, SPI = 17,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
       [SI = 250, SFT = 44, RD = 0]
```

In this example the path also consists of an SF of type 41 located at SFF1 and this is followed by an SI with an RD of zero and SF of type 44. This means that a choice can be made between any SFF that supports an SFI of type 44.

SFF1 will receive packets on the path from the Classifier and will identify the path from the SPI (17). The initial SI will be 255 and so SFF1 will deliver the packets to the SFI for SFT 41.

When the packets are returned to SFF1 by the SFI the SI will be decreased to 250 for the next hop. SFF1 now has a free choice of next hop SFF to execute the next hop in the path selecting between all SFFs that support SFs of type 44. Looking at the SFIRs it has received, SFF1 knows that SF type 44 is supported by SFF3 and SFF4. SFF1 uses its local load balancing algorithm to make this choice. The chosen SFF will send the packets to the SFI that supports SFT 44 before forwarding the packets to their destinations.

8.4. Example SFP With Choice of SFTs

```
SFP4:  RD = 198.51.100.1,104, SPI = 18,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
       [SI = 250, {SFT = 43, RD = 192.0.2.2,2,  
                 SFT = 44, RD = 192.0.2.3,8 } ]
```

This example provides a choice of SF type in the second hop in the path. The SI of 250 indicates a choice between SF type 43 located through SF2 and SF type 44 located at SF3.

SFF1 will receive packets on the path from the Classifier and will identify the path from the SPI (18). The initial SI will be 255 and so SFF1 will deliver the packets to the SFI for SFT 41.

When the packets are returned to SFF1 by the SFI the SI will be decreased to 250 for the next hop. SFF1 now has a free choice of next hop SFF to execute the next hop in the path selecting between all SFF2 that support an SF of type 43 and SFF3 that supports an SF of type 44. These may be completely different functions that are to

be executed dependent on specific conditions, or may be similar functions identified with different type identifiers (such as firewalls from different vendors). SFF1 uses its local policy and load balancing algorithm to make this choice, and may use additional information passed back from the local SFI to help inform its selection. The chosen SFF will send the packets to the SFI that supports the chose SFT before forwarding the packets to their destinations.

8.5. Example Correlated Bidirectional SFPs

```
SFP5: RD = 198.51.100.1,105, SPI = 19,  
      Assoc-Type = 1, Assoc-RD = 198.51.100.1,106, Assoc-SPI = 20,  
      [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
      [SI = 250, SFT = 43, RD = 192.0.2.2,2]
```

```
SFP6: RD = 198.51.100.1,106, SPI = 20,  
      Assoc-Type = 1, Assoc-RD = 198.51.100.1,105, Assoc-SPI = 19,  
      [SI = 254, SFT = 43, RD = 192.0.2.2,2],  
      [SI = 249, SFT = 41, RD = 192.0.2.1,1]
```

This example demonstrates correlation of two SFPs to form a bidirectional SFP as described in Section 7.2.

Two SFPRs are advertised by the Controller. They have different SPIs (19 and 20) so they are known to be separate SFPs, but they both have Association TLVs with Association Type set to 1 indicating bidirectional SFPs. Each has an Associated SFPR-RD fields containing the value of the other SFPR-RD to correlated the two SFPs as a bidirectional pair.

As can be seen from the SFPRs in this example, the paths are symmetric: the hops in SFP5 appear in the reverse order in SFP6.

8.6. Example Correlated Asymmetrical Bidirectional SFPs

```
SFP7:  RD = 198.51.100.1,107, SPI = 21,  
       Assoc-Type = 1, Assoc-RD = 198.51.100.1,108, Assoc-SPI = 22,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
       [SI = 250, SFT = 43, RD = 192.0.2.2,2]  
  
SFP8:  RD = 198.51.100.1,108, SPI = 22,  
       Assoc-Type = 1, Assoc-RD = 198.51.100.1,107, Assoc-SPI = 21,  
       [SI = 254, SFT = 44, RD = 192.0.2.4,6],  
       [SI = 249, SFT = 41, RD = 192.0.2.1,1]
```

Asymmetric bidirectional SFPs can also be created. This example shows a pair of SFPs with distinct SPIs (21 and 22) that are correlated in the same way as in the example in Section 8.5.

However, unlike in that example, the SFPs are different in each direction. Both paths include a hop of SF type 41, but SFP7 includes a hop of SF type 43 supported at SFF2 while SFP8 includes a hop of SF type 44 supported at SFF4.

8.7. Example Looping in an SFP

```
SFP9:  RD = 198.51.100.1,109, SPI = 23,  
       [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
       [SI = 250, SFT = 44, RD = 192.0.2.4,5],  
       [SI = 245, SFT = 1, RD = {SPI=23, SI=255, Rsv=0}],  
       [SI = 245, SFT = 42, RD = 192.0.2.3,7]
```

Looping and jumping are described in Section 6. This example shows an SFP that contains an explicit loop-back instruction that is presented as a choice within an SFP hop.

The first two hops in the path (SI = 255 and SI = 250) are normal. That is, the packets will be delivered to SFF1 and SFF4 in turn for execution of SFs of type 41 and 44 respectively.

The third hop (SI = 245) presents SFF4 with a choice of next hop. It can either forward the packets to SFF3 for an SF of type 42 (the second choice), or it can loop back.

The loop-back entry in the SFP for SI = 245 is indicated by the special purpose SFT value 1 ("Change Sequence"). Within this hop, the RD is interpreted as encoding the SPI and SI of the next hop (see Section 6.1. In this case the SPI is 23 which indicates that this is loop or branch: i.e., the next hop is on the same SFP. The SI is set

to 255: this is a higher number than the current SI (245) indicating a loop.

SFF4 must make a choice between these two next hops. Either the packets will be forwarded to SFF3 with the NSH SI decreased to 245 or looped back to SFF1 with the NSH SI reset to 255. This choice will be made according to local policy, information passed back by the local SFI, and details in the packets' metadata that are used to prevent infinite looping.

8.8. Example Branching in an SFP

```
SFP10:  RD = 198.51.100.1,110, SPI = 24,  
        [SI = 254, SFT = 42, RD = 192.0.2.3,7],  
        [SI = 249, SFT = 43, RD = 192.0.2.2,2]  
  
SFP11:  RD = 198.51.100.1,111, SPI = 25,  
        [SI = 255, SFT = 41, RD = 192.0.2.1,1],  
        [SI = 250, SFT = 1, RD = {SPI=24, SI=254, Rsv=0}]
```

Branching follows a similar procedure to that for looping (and jumping) as shown in Section 8.7 however there are two SFPs involved.

SFP10 shows a normal path with packets forwarded to SFF3 and SFF2 for execution of service functions of type 42 and 43 respectively.

SFP11 starts as normal (SFF1 for an SF of type 41), but then SFF1 processes the next hop in the path and finds a "Change Sequence" Special Purpose SFT. The SFIR-RD field includes an SPI of 24 which indicates SFP10, not the current SFP. The SI in the SFIR-RD is 254, so SFF1 knows that it must set the SPI/SI in the NSH to 24/254 and send the packets to the appropriate SFF as advertised in the SFPR for SFP10 (that is, SFF3).

9. Security Considerations

This document inherits all the security considerations discussed in the documents that specify BGP, the documents that specify BGP Multiprotocol Extensions, and the documents that define the attributes that are carried by BGP UPDATEs of the SFC AFI/SAFI. For more information look in [RFC4271], [RFC4760], and [I-D.ietf-idr-tunnel-encaps].

Service Function Chaining provides a significant attack opportunity: packets can be diverted from their normal paths through the network, can be made to execute unexpected functions, and the functions that

are instantiated in software can be subverted. However, this specification does not change the existence of Service Function Chaining and security issues specific to Service Function Chaining are covered in [RFC7665] and [I-D.ietf-sfc-nsh].

This document defines a control plane for Service Function Chaining. Clearly, this provides an attack vector for a Service Function Chaining system as an attack on this control plane could be used to make the system misbehave. Thus, the security of the BGP system is critically important to the security of the whole Service Function Chaining system.

10. IANA Considerations

10.1. New BGP AF/SAFI

IANA maintains a registry of "Address Family Numbers". IANA is requested to assign a new Address Family Number from the "Standards Action" range called "BGP SFC" (TBD1 in this document) with this document as a reference.

IANA maintains a registry of "Subsequent Address Family Identifiers (SAFI) Parameters". IANA is requested to assign a new SAFI value from the "Standards Action" range called "BGP SFC" (TBD2 in this document) with this document as a reference.

10.2. New BGP Path Attribute

IANA maintains a registry of "Border Gateway Protocol (BGP) Parameters" with a subregistry of "BGP Path Attributes". IANA is requested to assign a new Path attribute called "SFP attribute" (TBD3 in this document) with this document as a reference.

10.3. New SFP Attribute TLVs Type Registry

IANA maintains a registry of "Border Gateway Protocol (BGP) Parameters". IANA is request to create a new subregistry called the "SFP Attribute TLVs" registry.

Valid values are in the range 0 to 65535.

- o Values 0 and 65535 are to be marked "Reserved, not to be allocated".
- o Values 1 through 65524 are to be assigned according to the "First Come First Served" policy [RFC5226].

This document should be given as a reference for this registry.

The new registry should track:

- o Type
- o Name
- o Reference Document or Contact
- o Registration Date

The registry should initially be populated as follows:

Type	Name	Reference	Date
1	Association TLV	[This.I-D]	Date-to-be-set
2	Hop TLV	[This.I-D]	Date-to-be-set
3	SFT TLV	[This.I-D]	Date-to-be-set

10.4. New SFP Association Type Registry

IANA maintains a registry of "Border Gateway Protocol (BGP) Parameters". IANA is request to create a new subregistry called the "SFP Association Type" registry.

Valid values are in the range 0 to 65535.

- o Values 0 and 65535 are to be marked "Reserved, not to be allocated".
- o Values 1 through 65524 are to be assigned according to the "First Come First Served" policy [RFC5226].

This document should be given as a reference for this registry.

The new registry should track:

- o Association Type
- o Name
- o Reference Document or Contact
- o Registration Date

The registry should initially be populated as follows:

Association Type	Name	Reference	Date
1	Bidirectional SFP	[This.I-D]	Date-to-be-set

10.5. New Service Function Type Registry

IANA is request to create a new top-level registry called "Service Function Chaining Service Function Types".

Valid values are in the range 0 to 65535.

- o Values 0 and 65535 are to be marked "Reserved, not to be allocated".
- o Values 1 through 31 are to be assigned by "Standards Action" [RFC5226] and are referred to as the Special Purpose SFT values.
- o Other values (32 through 65534) are to be assigned according to the "First Come First Served" policy [RFC5226].

This document should be given as a reference for this registry.

The new registry should track:

- o Value
- o Name
- o Reference Document or Contact
- o Registration Date

The registry should initially be populated as follows:

Value	Name	Reference	Date
1	Change Sequence	[This.I-D]	Date-to-be-set

11. Contributors

Stuart Mackie
Juniper Networks

Email: wsmackie@juinper.net

Keyur Patel
Arrcus, Inc.

Email: keyur@arrcus.com

12. Acknowledgements

Thanks to Tony Przygienda for helpful comments, and to Joel Halpern for discussions that improved this document.

13. References

13.1. Normative References

- [I-D.ietf-idr-tunnel-encaps]
Rosen, E., Patel, K., and G. Velde, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-02 (work in progress), May 2016.
- [I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.
- [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>>.
- [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>>.
- [RFC4760] Bates, T., Chandra, R., Katz, D., and Y. Rekhter, "Multiprotocol Extensions for BGP-4", RFC 4760, DOI 10.17487/RFC4760, January 2007, <<http://www.rfc-editor.org/info/rfc4760>>.

- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.

13.2. Informative References

- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<http://www.rfc-editor.org/info/rfc6790>>.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <<http://www.rfc-editor.org/info/rfc7498>>.
- [RFC7510] Xu, X., Sheth, N., Yong, L., Callon, R., and D. Black, "Encapsulating MPLS in UDP", RFC 7510, DOI 10.17487/RFC7510, April 2015, <<http://www.rfc-editor.org/info/rfc7510>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

Authors' Addresses

Adrian Farrel
Juniper Networks

Email: adrian@olddog.co.uk

John Drake
Juniper Networks

Email: jdrake@juniper.net

Eric Rosen
Juniper Networks

Email: erosen@juniper.net

Jim Uttaro
AT&T

Email: jul738@att.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

BESS Working Group
Internet-Draft
Intended Status: Standards Track

Ali Sajassi
Samir Thoria
Cisco
Keyur Patel
Derek Yeung
Arrcus
John Drake
Wen Lin
Juniper

Expires: April 28, 2017

October 28, 2016

IGMP and MLD Proxy for EVPN
draft-sajassi-bess-evpn-igmp-mld-proxy-01

Abstract

Ethernet Virtual Private Network (EVPN) solution [RFC 7432] is becoming pervasive in data center (DC) applications for Network Virtualization Overlay (NVO) and DC interconnect (DCI) services, and in service provider (SP) applications for next generation virtual private LAN services.

This draft describes how to support efficiently endpoints running IGMP for the above services over an EVPN network by incorporating IGMP proxy procedures on 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) 2013 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	IGMP Proxy	5
2.1	Proxy Reporting	5
2.1.1	IGMP Membership Report Advertisement in BGP	5
2.1.1	IGMP Leave Group Advertisement in BGP	7
2.2	Proxy Querier	8
3	Operation	8
3.1	PE with only attached hosts/VMs for a given subnet	9
3.2	PE with mixed of attached hosts/VMs and multicast source	10
3.3	PE with mixed of attached hosts/VMs, multicast source and router	10
4	All-Active Multi-Homing	10
4.1	Local IGMP Join Synchronization	11
4.2	Local IGMP Leave Group Synchronization	11
4.2.1	Remote Leave Group Synchronization	12
4.2.2	Common Leave Group Synchronization	13
5	Single-Active Multi-Homing	13
6	Discovery of Selective P-Tunnel Types	13
7	BGP Encoding	15
7.1	Selective Multicast Ethernet Tag Route	15
7.1.1	Constructing the Selective Multicast route	16
7.2	IGMP Join Synch Route	17
7.2.1	Constructing the IGMP Join Synch Route	19
7.3	IGMP Leave Synch Route	20

7.3.1 Constructing the IGMP Leave Synch Route 21

7.4 Multicast Flags Extended Community 22

7.5 EVI-RT Extended Community 23

8 Acknowledgement 24

9 Security Considerations 24

10 IANA Considerations 24

11 References 24

11.1 Normative References 24

11.2 Informative References 24

Authors' Addresses 25

1 Introduction

Ethernet Virtual Private Network (EVPN) solution [RFC 7432] is becoming pervasive in data center (DC) applications for Network Virtualization Overlay (NVO) and DC interconnect (DCI) services, and in service provider (SP) applications for next generation virtual private LAN services.

In DC applications, a POD can consist of a collection of servers supported by several TOR and Spine switches. This collection of servers and switches are self contained and may have their own control protocol for intra-POD communication and orchestration. However, EVPN is used as way of standard inter-POD communication for both intra-DC and inter-DC. A subnet can span across multiple PODs and DCs. EVPN provides robust multi-tenant solution with extensive multi-homing capabilities to stretch a subnet (e.g., VLAN) across multiple PODs and DCs. There can be many hosts/VMs (e.g., several hundreds) attached to a subnet that is stretched across several PODs and DCs.

These hosts/VMs express their interests in multicast groups on a given subnet/VLAN by sending IGMP membership reports (Joins) for their interested multicast group(s). Furthermore, an IGMP router (e.g., IGMPv1) periodically sends membership queries to find out if there are hosts on that subnet still interested in receiving multicast traffic for that group. The IGMP/MLD Proxy solution described in this draft has three objectives to accomplish:

- 1) Just like ARP/ND suppression mechanism in EVPN to reduce the flooding of ARP messages over EVPN, it is also desired to have a mechanism to reduce the flood of IGMP messages (both Queries and Reports) in EVPN.
- 2) If there is no physical/virtual multicast router attached to the EVPN network for a given (*,G) or (S,G), it is desired for the EVPN network to act as a distributed anycast multicast router for all the hosts attached to that subnet.
- 3) To forward multicast traffic efficiently over EVPN network such that it only gets forwarded to the PEs that have interest in the multicast group(s) - i.e., multicast traffic will not be forwarded to the PEs that have no receivers attached to them for that multicast group. This draft shows how the above objectives are achieved.

The first two objectives are achieved by using IGMP/MLD proxy on the PE and the third objective is achieved by setting up a multicast tunnel (ingress replication or P2MP) only among the PEs that have interest in that multicast group(s) based on the trigger from

IGMP/MLD proxy processes. The proposed solutions for each of these objectives are discussed in the following sections.

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 [KEYWORDS].

2 IGMP Proxy

IGMP Proxy mechanism is used to reduce the flooding of IGMP messages over EVPN network similar to ARP proxy used in reducing the flooding of ARP messages over EVPN. It also provides triggering mechanism for the PEs to setup their underlay multicast tunnels. IGMP Proxy mechanism consist of two components: a) Proxy for IGMP Reports and b) Proxy for IGMP Queries.

2.1 Proxy Reporting

When IGMP protocol is used between host/VMs and its first hop EVPN router (EVPN PE), Proxy-reporting is used by the EVPN PE to summarize (when possible) reports received from downstream hosts and propagate it in BGP to other PEs that are interested in the info. This is done by terminating IGMP Reports in the first hop PE, translating and exchanging the relevant information among EVPN BGP speakers. The information is again translated back to IGMP message at the recipient EVPN speaker. Thus it helps create an IGMP overlay subnet using BGP. In order to facilitate such an overlay, this document also defines a new EVPN route type NLRI (EVPN Selective Multicast Ethernet Tag route) along with its procedures to help exchange and register IGMP multicast groups [section 5].

2.1.1 IGMP Membership Report Advertisement in BGP

When a PE wants to advertise an IGMP membership report (Join) using the BGP EVPN route, it follows the following rules:

1) When the first hop PE receives several IGMP membership reports (Joins) , belonging to the same IGMP version, from different attached hosts/VMs for the same (*,G) or (S,G), it only sends a single BGP message corresponding to the very first IGMP Join. This is because BGP is a statefull protocol and no further transmission of the same report is needed. If the IGMP Join is for (*,G), then multicast group address along with the corresponding version flag (v1, v2, or v3) are set. In case of IGMPv3, exclude flag also needs to be set to indicate

that no source IP address to be excluded (e.g., include all sources "*"). If the IGMP Join is for (S,G), then besides setting multicast group address along with the version flag v3, the source IP address and the include/exclude flag must be set. It should be noted that when advertising the EVPN route for (S,G), the only valid version flag is v3 (i.e., v1 and v2 flags must be set to zero).

2) When the first hop PE receives an IGMPv3 Join for (S,G), then the PE checks to see if the source (S) is attached to self. If so, it does not send the corresponding BGP EVPN route advertisement.

3) When the first hop PE receives an IGMP version-X Join first for (*,G) and then later it receives an IGMP version-Y Join for the same (*,G), then it will readvertise the same EVPN Selective Multicast route with flag for version-Y set in addition to any previously-set version flag(s). In other words, the first hop PE does not withdraw the EVPN route before sending the new route because the flag field is not part of BGP route key processing.

4) When the first hop PE receives an IGMP version-X Join first for (*,G) and then later it receives an IGMPv3 Join for the same multicast group address but for a specific source address S, then the PE will advertise a new EVPN Selective Multicast route with v3 flag set (and v1 and v2 reset). Include/exclude flag also need to be set accordingly. Since source IP address is used as part of BGP route key processing, it is considered as a new BGP route advertisement.

5) When a PE receives an EVPN Selective Multicast route with more than one version flag set, it will generate the corresponding IGMP report for (*,G) for each version specified in the flag field. With multiple version flags set, there should be no source IP address in the receive EVPN route. If there is, then an error should be logged. If v3 flag is set (in addition to v1 or v2), then the include/exclude flag needs to indicate "exclude". If not, then an error should be logged. The PE MUST generate an IGMP membership report (Join) for that (*,G) and each IGMP version in the version flag.

6) When a PE receives a list of EVPN Selective Multicast NLRI's in its BGP update message, each with a different source IP address and the multicast group address, and the version flag is set to v3, then the PE generates an IGMPv3 membership report with a record corresponding to the list of source IP addresses and the group address along with the proper indication of inclusion/exclusion.

7) Upon receiving EVPN Selective Multicast route(s) and before

generating the corresponding IGMP Join(s), the PE checks to see whether it has any multicast router's AC(s) (Attachment Circuits connected to multicast routers). If it has router's ACs, then the generated IGMP Join(s) are sent to those ACs. If it doesn't have any router's AC, then no IGMP Join(s) needs to be generated because sending IGMP Joins to other hosts can result in unintentionally preventing a host from joining a specific multicast group for IGMPv1 and IGMPv2 - i.e., if the PE does not receive a join from the host it will not forward multicast data to it. Per [RFC4541], when an IGMPv1 or IGMPv2 host receives a membership report for a group address that it intends to join, the host will suppress its own membership report for the same group. This message suppression is a requirement for IGMPv1 and IGMPv2 hosts. This is not a problem for hosts running IGMPv3 because there is no suppression of IGMP Membership reports.

2.1.1 IGMP Leave Group Advertisement in BGP

When a PE wants to withdraw an EVPN Selective Multicast route corresponding to an IGMPv2 Leave Group (Leave) or IGMPv3 "Leave" equivalent message, it follows the following rules:

- 1) For IGMPv1, there is no explicit membership leave; therefore, the PE needs to periodically send out an IGMP membership query to determine whether there is any host left who is interested in receiving traffic directed to this multicast group (this proxy query function will be described in more details in section 2.2). If there is no host left, then the PE re-advertises EVPN Selective Multicast route with the v1 version flag reset. If this is the last version flag to be reset, then instead of re-advertising the EVPN route with all version flags reset, the PE withdraws the EVPN route for that (*,G).
- 2) When a PE receives an IGMPv2 Leave Group or its "Leave" equivalent message for IGMPv3 from its attached host, it checks to see if this host is the last host who is interested in this multicast group by sending a query for the multicast group. If the host was indeed the last one, then the PE re-advertises EVPN Selective Multicast route with the corresponding version flag reset. If this is the last version flag to be reset, then instead of re-advertising the EVPN route with all version flags reset, the PE withdraws the EVPN route for that (*,G).
- 3) When a PE receives an EVPN Selective Multicast route for a given (*,G), it compares the received version flags from the route with its per-PE stored version flags. If the PE finds that a version flag associated with the (*,G) for the remote PE is reset, then the PE generates IGMP Leave for that (*,G) toward its local interface (if

any) attached to the multicast router for that multicast group. It should be noted that the received EVPN route should at least have one version flag set. If all version flags are reset, it is an error because the PE should have received an EVPN route withdraw for the last version flag. If the PE receives an EVPN Selective Multicast route withdraw, then it must remove the remote PE from the OIF list associated with that multicast group.

4) When a PE receives an EVPN Selective Multicast route withdraw, it removes the remote PE from its OIF list for that multicast group and if there are no more OIF entries for that multicast group (either locally or remotely), then the PE MUST stop responding to queries from the locally attached router (if any). If there is a source for that multicast group, the PE stops sending multicast traffic for that source.

2.2 Proxy Querier

As mentioned in the previous sections, each PE need to have proxy querier functionality for the following reasons:

1) To enable the collection of EVPN PEs providing L2VPN service to act as distributed multicast router with Anycast IP address for all attached hosts/VMs in that subnet.

2) To enable suppression of IGMP membership reports and queries over MPLS/IP core.

3) To enable generation of query messages locally to their attached host. In case of IGMPv1, the PE needs to send out an IGMP membership query to verify that at least one host on the subnet is still interested in receiving traffic directed to that group. When there is no reply to three consecutive IGMP membership queries, the PE times out the group, stops forwarding multicast traffic to the attached hosts for that (*,G), and sends a EVPN Selective Multicast route associated with that (*,G) with the version-1 flag reset or withdraws that route.

3 Operation

Consider the EVPN network of figure-1, where there is an EVPN instance configured across the PEs shown in this figure (namely PE1, PE2, and PE3). Lets consider that this EVPN instance consist of a single bridge domain (single subnet) with all the hosts, sources and the multicast router shown in this figure connected to this subnet. PE1 only has hosts connected to it. PE2 has a mix of hosts and

send it to any of its port for that subnet - namely ports associated with H6 and H7.

When PE1 receives the second IGMPv1 Join from H2 for the same multicast group (*,G1), it only adds that port to its OIF list but it doesn't send any EVPN BGP route because there is no change in information. However, when it receives the IGMPv2 Join from H3 for the same (*,G1), besides adding the corresponding port to its OIF list, it re-advertises the previously sent EVPN Selective Multicast route with the version-2 flag set.

Finally when PE1 receives the IMGMPv3 Join from H4 for (S2,G2), it advertises a new EVPN Selective Multicast route corresponding to it.

3.2 PE with mixed of attached hosts/VMs and multicast source

The main difference in here is that when PE2 receives IGMPv3 Join from H7 for (S2,G2), it does not advertises it in BGP because PE2 knows that S2 is attached to its local AC. PE2 adds the port associated with H7 to its OIF list for (S2,G2). The processing for IGMPv2 received from H6 is the same as the v2 Join described in previous section.

3.3 PE with mixed of attached hosts/VMs, multicast source and router

The main difference in here relative to the previous two sections is that Join messages received locally needs to be sent to the port associated with router R1. Furthermore, the Joins received via BGP need to be passed to the R1 port but filtered for all other ports.

4 All-Active Multi-Homing

Because a CE's LAG flow hashing algorithm is unknown, in an All-Active redundancy mode it must be assumed that the CE can send a given IGMP message to any one of the multi-homed PEs, either DF or non-DF - i.e., different IGMP Join messages can arrive at different PEs in the redundancy group and furthermore their corresponding Leave messages can arrive at PEs that are different from the ones received the Join messages. Therefore, all PEs attached to a given ES must coordinate IGMP Join and Leave Group (x, G) state, where x may be either '*' or a particular source S, for each [EVI, broadcast domain (BD)] on that ES. This allows the DF for that [ES, EVI, BD] to correctly advertise or withdraw a Selective Multicast Ethernet Tag (SMET) route for that (x, G) group in that [EVI, BD] when needed.

All-Active multihoming PEs for a given ES MUST support IGMP synch procedures described in this section if they want to perform IGMP proxy for hosts connects to that ES.

4.1 Local IGMP Join Synchronization

When a PE, either DF or non-DF, receives, on a given multihomed ES operating in All-Active redundancy mode, an IGMP Membership Report for (x, G), it determines the [EVI, BD] to which the IGMP Membership Report belongs. If the PE doesn't already have local IGMP Join (x, G) state for that [EVI, BD] on that ES, it instantiates local IGMP Join (x, G) state and advertises a BGP IGMP Join Synch route for that [ES, EVI, BD]. Local IGMP Join (x, G) state refers to IGMP Join (x, G) state that is created as the result of processing an IGMP Membership Report for (x, G).

The IGMP Join Synch route carries the ES-Import RT for the ES on which the IGMP Membership Report was received. Thus it may only go to the PEs attached to that ES (and not any other PEs).

When a PE, either DF or non-DF, receives an IGMP Join Synch route it installs that route and if it doesn't already have IGMP Join (x, G) state for that [ES, EVI, BD], it instantiates that IGMP Join (x,G) state - i.e., IGMP Join (x, G) state is the union of local IGMP Join (x, G) state and installed IGMP Join Synch route. If the DF is not currently advertising (originating) a SMET route for that (x, G) group in that [EVI, BD], it does so now.

When a PE, either DF or non-DF, deletes its local IGMP Join (x, G) state for that [ES, EVI, BD], it withdraws its BGP IGMP Join Synch route for that [ES, EVI, BD].

When a PE, either DF or non-DF, receives the withdrawal of an IGMP Join Synch route from another PE it removes that route. When a PE has no local IGMP Join (x, G) state and it has no installed IGMP Join Synch routes, it removes IGMP Join (x, G) state for that [ES, EVI, BD]. If the DF no longer has IGMP Join (x, G) state for that [EVI, BD] on any ES for which it is DF, it withdraws its SMET route for that (x, G) group in that [EVI, BD].

I.e., A PE advertises an SMET route for that (x, G) group in that [EVI, BD] when it has IGMP Join (x, G) state in that [EVI, BD] on at least one ES for which it is DF and it withdraws that SMET route when it does not have IGMP Join (x, G) state in that [EVI, BD] on any ES for which it is DF.

4.2 Local IGMP Leave Group Synchronization

When a PE, either DF or non-DF, receives, on a given multihomed ES operating in All-Active redundancy mode, an IGMP Leave Group message for (x, G) from the attached CE, it determines the [EVI, BD] to which the IGMPv2 Leave Group belongs. Regardless of whether it has IGMP Join (x, G) state for that [ES, EVI, BD], it initiates the (x, G) leave group synchronization procedure, which consists of the following steps:

- 1) It computes the Maximum Response Time, which is the duration of (x, G) leave group synchronization procedure. This is the product of two locally configured values, Last Member Query Count and Last Member Query Interval (described in Section 3 of [RFC2236]), plus delta, the time it takes for a BGP advertisement to propagate between the PEs attached to the multihomed ES (delta is a consistently configured value on all PEs attached to the multihomed ES).
- 2) It starts the Maximum Response Time timer. Note that the receipt of subsequent IGMP Leave Group messages or BGP Leave Synch routes for (x, G) do not change the value of a currently running Maximum Response Time timer and are ignored by the PE.
- 3) It initiates the Last Member Query procedure described in Section 3 of [RFC2236]; viz, it sends a number of Group-Specific Query (x, G) messages (Last Member Query Count) at a fixed interval (Last Member Query Interval) to the attached CE.
- 4) It advertises an IGMP Leave Synch route for that that [ES, EVI, BD]. This route notifies the other multihomed PEs attached to the given multihomed ES that it has initiated an (x, G) leave group synchronization procedure; i.e., it carries the ES-Import RT for the ES on which the IGMP Leave Group was received. It also contains the Maximum Response Time and the Leave Group Synchronization Procedure Sequence number. The latter identifies the specific (x, G) leave group synchronization procedure initiated by the advertising PE, which increments the value whenever it initiates a procedure.
- 5) When the Maximum Response Timer expires, the PE that has advertised the IGMP Leave Synch route withdraws it.

4.2.1 Remote Leave Group Synchronization

When a PE, either DF or non-DF, receives an IGMP Leave Synch route it installs that route and it starts a timer for (x, G) on the specified [ES, EVI, BD] whose value is set to the Maximum Response Time in the received IGMP Leave Synch route. Note that the receipt of subsequent IGMPv2 Leave Group messages or BGP Leave Synch routes for (x, G) do not change the value of a currently running Maximum Response Time

timer and are ignored by the PE.

4.2.2 Common Leave Group Synchronization

If a PE attached to the multihomed ES receives an IGMP Membership Report for (x, G) before the Maximum Response Time timer expires, it advertises a BGP IGMP Join Synch route for that [ES, EVI, BD]. If it doesn't already have local IGMP Join (x, G) state for that [ES, EVI, BD], it instantiates local IGMP Join (x, G) state. If the DF is not currently advertising (originating) a SMET route for that (x, G) group in that [EVI, BD], it does so now.

If a PE attached to the multihomed ES receives an IGMP Join Synch route for (x, G) before the Maximum Response Time timer expires, it installs that route and if it doesn't already have IGMP Join (x, G) state for that [EVI, BD] on that ES, it instantiates that IGMP Join (x, G) state. If the DF is not currently advertising (originating) a SMET route for that (x, G) group in that [EVI, BD], it does so now.

When the Maximum Response Timer expires a PE that has advertised an IGMP Leave Synch route, withdraws it. Any PE attached to the multihomed ES, that started the Maximum Response Time and has no local IGMP Join (x, G) state and no installed IGMP Join Synch routes, it removes IGMP Join (x, G) state for that [ES, EVI, BD]. If the DF no longer has IGMP Join (x, G) state for that [EVI, BD] on any ES for which it is DF, it withdraws its SMET route for that (x, G) group in that [EVI, BD].

5 Single-Active Multi-Homing

Note that to facilitate state synchronization after failover, the PEs attached to a multihomed ES operating in Single-Active redundancy mode should also coordinate IGMP Join (x, G) state. In this case all IGMP Join messages are received by the DF and distributed to the non-DF PEs using the procedures described above.

6 Discovery of Selective P-Tunnel Types

To allow an ingress PE that supports IGMP proxy procedures and SMET route to properly assign a selective P-tunnel supported by the receiving PEs, the ingress PE needs to discover the types of selective P-tunnels supported by the receiving PEs and select the preferred tunnel type among the ones that it has in common with the receiving PEs.

In order to support such discovery mechanism, the Multicast Flags extended community defined in section 7.2 is used. Each PE that

supports different types of P-tunnels, marks the corresponding bits and advertise this extended community along with its IMET route. Therefore, the ingress PE can discover types of P-tunnels supported by the receiving PEs. If the ingress PE does not receive this extended community along with an IMET route for a given EVI, it assumes the only P-tunnel type supported by the egress PE, is ingress replication.

If besides ingress-replication P-tunnel type, there is no other P-tunnel types in common among the participant PEs for an EVI, then the ingress PE MUST use ingress-replication P-tunnel type.

If besides ingress-replication P-tunnel type, there is one or more P-tunnel types in common among the participant PEs for an EVI, then the ingress PE can choose the P-tunnel type that it prefers.

If besides ingress-replication P-tunnel type, there is no other P-tunnel types in common among the participant PEs for an EVI, then the ingress PE MAY choose several different P-tunnel types where the union of them covers the tunnel types supported by the participant PEs for that EVI. This implies that the ingress PE replicates the multicast traffic into different P-tunnels - i.e., to replicate the multicast traffic onto P2MP mLDP P-tunnel and ingress-replication P-tunnel.

If an ingress PE uses ingress replication, then for a given (x, G) group in a given [EVI, BD]:

- 1) It sends (x, G) traffic to the set of PEs not supporting IGMP Proxy. This set consists of any PE that has advertised an Inclusive Multicast Tag route for the [EVI, BD] without the "IGMP Proxy Support" flag.
- 2) It sends (x, G) traffic to the set of PEs supporting IGMP Proxy and having listeners for that (x, G) group in that [EVI, BD]. This set consists of any PE that has advertised an Inclusive Multicast Tag route for the [EVI, BD] with the "IGMP Proxy Support" flag and that has advertised an SMET route for that (x, G) group in that [EVI, BD].

If an ingress PE's Selective P-Tunnel for a given [EVI, BD] uses P2MP and all of the PEs in the [EVI, BD] support that tunnel type and IGMP, then for a given (x, G) group in a given [EVI, BD] it sends (x, G) traffic using the Selective P-Tunnel for that (x, G) group in that [EVI, BD]. This tunnel will include those PEs that have advertised an SMET route for that (x, G) group on that [EVI, BD] (for Selective P-tunnel) but it may include other PEs as well (for Aggregate Selective P-tunnel).

7 BGP Encoding

This document defines three new BGP EVPN routes to carry IGMP membership reports. This route type is known as:

- + 6 - Selective Multicast Ethernet Tag Route
- + 7 - IGMP Join Synch Route
- + 8 - IGMP Leave Synch Route

The detailed encoding and procedures for this route type is described in subsequent section.

7.1 Selective Multicast Ethernet Tag Route

An Selective Multicast Ethernet Tag route type specific EVPN NLRI consists of the following:

```

+-----+
|  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)                |
+-----+

```

For the purpose of BGP route key processing, all the fields are considered to be part of the prefix in the NLRI except for the one-octet optional flag field (if included). The Flags fields are defined as follows:

```

      0  1  2  3  4  5  6  7
+-----+-----+-----+-----+
| reserved | IE|v3|v2|v1|
+-----+-----+-----+-----+

```

The least significant bit, bit 7 indicates support for IGMP version 1.

The second least significant bit, bit 6 indicates support for IGMP version 2.

The third least significant bit, bit 5 indicates support for IGMP version 3.

The fourth least significant bit, bit 4 indicates whether the (S, G) information carried within the route-type is of Include Group type (bit value 0) or an Exclude Group type (bit value 1). The Exclude Group type bit MUST be ignored if bit 5 is not set.

This EVPN route type is used to carry tenant IGMP multicast group information. The flag field assists in distributing IGMP membership interest of a given host/VM for a given multicast route. The version bits help associate IGMP version of receivers participating within the EVPN domain.

The include/exclude bit helps in creating filters for a given multicast route.

7.1.1 Constructing the Selective Multicast route

This section describes the procedures used to construct the Selective Multicast route. Support for this route type is optional.

The Route Distinguisher (RD) SHOULD be a Type 1 RD [RFC4364]. The value field comprises an IP address of the PE (typically, the loopback address) followed by a number unique to the PE.

The Ethernet Tag ID MUST be set as follows:

```

EVI is VLAN-Based or VLAN Bundle service - set to 0
EVI is VLAN-Aware Bundle service without translation - set to
the customer VID for the [EVI, BD]
EVI is VLAN-Aware Bundle service with translation - set to the
normalized Ethernet Tag ID for the [EVI, BD]

```

The Multicast Source length MUST be set to length of multicast source address in bits. In case of a (*, G) Join, the Multicast Source Length is set to 0.

The Multicast Source is the Source IP address of the IGMP membership report. In case of a (*, G) Join, this field does not exist.

The Multicast Group length MUST be set to length of multicast group address in bits.

The Multicast Group is the Group address of the IGMP membership report.

The Originator Router Length is the length of the Originator Router address in bits.

The Originator Router Address is the IP address of Router Originating the prefix. It should be noted that using the "Originating Router's IP address" field to get the PE IP address, needed for building multicast underlay tunnels, allows for inter-AS operations where BGP next hop can get over written.

The Flags field indicates the version of IGMP protocol from which the membership report was received. It also indicates whether the multicast group had INCLUDE or EXCLUDE bit set.

IGMP protocol is used to receive group membership information from hosts/VMs by TORs. Upon receiving the hosts/VMs expression of interest of a particular group membership, this information is then forwarded using Ethernet Multicast Source Group Route NLRI. The NLRI also keeps track of receiver's IGMP protocol version and any "source filtering" for a given group membership. All EVPN Selective Multicast Group routes are announced with per-EVI Route Target extended communities.

7.2 IGMP Join Synch Route

This EVPN route type is used to coordinate IGMP Join (x,G) state for a given [EVI, BD] between the PEs attached to a given ES operating in All-Active (or Single-Active) redundancy mode and it consists of following:

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)

For the purpose of BGP route key processing, all the fields are considered to be part of the prefix in the NLRI except for the one-octet Flags field, whose fields are defined as follows:

0	1	2	3	4	5	6	7
reserved				IE	v3	v2	v1

The least significant bit, bit 7 indicates support for IGMP version 1. The second least significant bit, bit 6 indicates support for IGMP version 2. The third least significant bit, bit 5 indicates support for IGMP version 3. The fourth least significant bit, bit 4 indicates whether the (S, G) information carried within the route-type is of Include Group type (bit value 0) or an Exclude Group type (bit value 1). The Exclude Group type bit MUST be ignored if bit 5 is not set.

The Flags field assists in distributing IGMP membership interest of a given host/VM for a given multicast route. The version bits help associate IGMP version of receivers participating within the EVPN domain. The include/exclude bit helps in creating filters for a

given multicast route.

7.2.1 Constructing the IGMP Join Synch Route

This section describes the procedures used to construct the IGMP Join Synch route. Support for this route type is optional. If a PE does not support this route, then it MUST not indicate that it supports 'IGMP proxy' in Multicast Flag extended community for the EVIs corresponding to its multi-homed Ethernet Segments. An IGMP Join Synch route is advertised with an ES-Import Route Target extended community whose value is set to the ESI for the ES on which the IGMP Join was received.

The Route Distinguisher (RD) SHOULD be a Type 1 RD [RFC4364]. The value field comprises an IP address of the PE (typically, the loopback address) followed by a number unique to the PE.

The Ethernet Segment Identifier (ESI) MUST be set to the 10-octet value defined for the ES.

The Ethernet Tag ID MUST be set as follows:

EVI is VLAN-Based or VLAN Bundle service - set to 0
EVI is VLAN-Aware Bundle service without translation - set to the customer VID for the [EVI, BD]
EVI is VLAN-Aware Bundle service with translation - set to the normalized Ethernet Tag ID for the [EVI, BD]

The Multicast Source length MUST be set to length of multicast source address in bits. In case of a (*, G) Join, the Multicast Source Length is set to 0.

The Multicast Source is the Source IP address of the IGMP membership report. In case of a (*, G) Join, this field does not exist.

The Multicast Group length MUST be set to length of multicast group address in bits.

The Multicast Group is the Group address of the IGMP membership report.

The Originator Router Length is the length of the Originator Router address in bits.

The Originator Router Address is the IP address of Router Originating the prefix.

The Flags field indicates the version of IGMP protocol from which the membership report was received. It also indicates whether the multicast group had INCLUDE or EXCLUDE bit set.

7.3 IGMP Leave Synch Route This EVPN route type is used to coordinate IGMP Leave Group (x,G) state for a given [EVI, BD] between the PEs attached to a given ES operating in All-Active (or Single-Active) redundancy mode and it consists of following:

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)
Leave Group Synchronization # (4 octets)
Maximum Response Time (1 octet)
Flags (1 octet)

For the purpose of BGP route key processing, all the fields are considered to be part of the prefix in the NLRI except for the Maximum Response Time and the one-octet Flags field, whose fields are defined as follows:


```

      0  1  2  3  4  5  6  7
+-----+-----+-----+-----+
| reserved | IE|v3|v2|v1|
+-----+-----+-----+-----+

```

The least significant bit, bit 7 indicates support for IGMP version 1. The second least significant bit, bit 6 indicates support for IGMP version 2. The third least significant bit, bit 5 indicates support for IGMP version 3. The fourth least significant bit, bit 4 indicates whether the (S, G) information carried within the route-type is of Include Group type (bit value 0) or an Exclude Group type (bit value 1). The Exclude Group type bit MUST be ignored if bit 5 is not set.

The Flags field assists in distributing IGMP membership interest of a given host/VM for a given multicast route. The version bits help associate IGMP version of receivers participating within the EVPN domain. The include/exclude bit helps in creating filters for a given multicast route.

7.3.1 Constructing the IGMP Leave Synch Route

This section describes the procedures used to construct the IGMP Join Synch route. Support for this route type is optional. If a PE does not support this route, then it MUST not indicate that it supports 'IGMP proxy' in Multicast Flag extended community for the EVIs corresponding to its multi-homed Ethernet Segments. An IGMP Join Synch route is advertised with an ES-Import Route Target extended community whose value is set to the ESI for the ES on which the IGMP Join was received.

The Route Distinguisher (RD) SHOULD be a Type 1 RD [RFC4364]. The value field comprises an IP address of the PE (typically, the loopback address) followed by a number unique to the PE.

The Ethernet Segment Identifier (ESI) MUST be set to the 10-octet value defined for the ES.

The Ethernet Tag ID MUST be set as follows:

```

EVI is VLAN-Based or VLAN Bundle service - set to 0
EVI is VLAN-Aware Bundle service without translation - set to
the customer VID for the [EVI, BD]
EVI is VLAN-Aware Bundle service with translation - set to the
normalized Ethernet Tag ID for the [EVI, BD]

```

The Multicast Source length MUST be set to length of multicast source address in bits. In case of a (*, G) Join, the Multicast Source Length is set to 0.

The Multicast Source is the Source IP address of the IGMP membership report. In case of a (*, G) Join, this field does not exist.

The Multicast Group length MUST be set to length of multicast group address in bits.

The Multicast Group is the Group address of the IGMP membership report.

The Originator Router Length is the length of the Originator Router address in bits.

The Originator Router Address is the IP address of Router Originating the prefix.

The Flags field indicates the version of IGMP protocol from which the membership report was received. It also indicates whether the multicast group had INCLUDE or EXCLUDE bit set.

7.4 Multicast Flags Extended Community

The 'Multicast Flags' extended community is a new EVPN extended community. EVPN extended communities are transitive extended communities with a Type field value of 6. IANA will assign a Sub-Type from the 'EVPN Extended Community Sub-Types' registry.

A PE that supports IGMP proxy on a given [EVI, BD] MUST attach this extended community to the Inclusive Multicast Ethernet Tag (IMET) route it advertises for that [EVI, BD] and it Must set the IGMP Proxy Support flag to 1. Note that an [RFC7432] compliant PE will not advertise this extended community so its absence indicates that the advertising PE does not support IGMP Proxy.

The advertisement of this extended community enables more efficient multicast tunnel setup from the source PE specially for ingress replication - i.e., if an egress PE supports IGMP proxy but doesn't have any interest in a given (x, G), it advertises its IGMP proxy capability using this extended community but it does not advertise any SMET route for that (x, G). When the source PE (ingress PE) receives such advertisements from the egress PE, it doesn't not replicate the multicast traffic to that egress PE; however, it does replicate the multicast traffic to the egress PEs that don't

advertise such capability even if they don't have any interests in that (x, G).

A Multicast Flags extended community is encoded as an 8-octet value, as follows:

```

          1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type=0x06   | Sub-Type=TBD   |           Flags (2 Octets)           |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           Reserved=0           |           Tunnel Type           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The low-order bit of the Flags is defined as the "IGMP Proxy Support" bit. A value of 1 means that the PE supports IGMP Proxy as defined in this document, and a value of 0 means that the PE does not support IGMP proxy. The absence of this extended community also means that the PE doesn't support IGMP proxy.

Tunnel type field is a 2-octet field with the bits set according to the following:

```

LSB = 1, indicates the support for RSVP-TE P2MP LSP
2nd LSB = 1, indicates the support for P2MP LSP
3rd LSB = 1, indicates the support for PIM-SSM
4th LSB = 1, indicates the support for PIM-SM
5th LSB = 1, indicates the support for BIDIR-PIM
6th LSB = 1, indicates the support for mLDP MP2MP LSP

```

7.5 EVI-RT Extended Community

The 'EVI-RT' extended community is a new EVPN extended community. EVPN extended communities are transitive extended communities with a Type field value of 6. IANA will assign a Sub-Type from the 'EVPN Extended Community Sub-Types' registry.

A PE that supports IGMP synch procedures for All-Active (or Single-Active) multi-homed ES, MUST attach this extended community to either IGMP Join Synch route (sec 7.2) or IGMP Leave Synch route (sec 7.3). This extended community carries the RT associated with the EVI so that the receiving PE can identify the EVI properly. The reason standard format RT is not used, is to avoid distribution of these routes beyond the group of multihoming PEs for that ES.

1										2										3											
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Type=0x06										Sub-Type=TBD										RT associated with EVI											
										RT associated with the EVI (cont.)																					

8 Acknowledgement

9 Security Considerations

Same security considerations as [RFC7432].

10 IANA Considerations

Allocation of Extended Community Type and Sub-Type for EVPN.

11 References

11.1 Normative References

[KEYWORDS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4360] S. Sangli et al, "'BGP Extended Communities Attribute", February, 2006.

[RFC7432] Sajassi et al., "BGP MPLS Based Ethernet VPN", February, 2015.

11.2 Informative References

[ETREE-FMWK] Key et al., "A Framework for E-Tree Service over MPLS Network", draft-ietf-l2vpn-etree-frwk-03, work in progress, September 2013.

[PBB-EVPN] Sajassi et al., "PBB-EVPN", draft-ietf-l2vpn-pbb-evpn-05.txt, work in progress, October, 2013.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for IGMP and MLD snooping PEs", RFC 4541, 2006.

Authors' Addresses

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Samir Thoria
Cisco
Email: sthoria@cisco.com

Keyur Patel
Cisco
Email: keyur@arccus.com

Derek Yeung
Cisco
Email: Yeung@arccus.com

John Drake
Juniper
Email: jdrake@juniper.net

Wen Lin
Juniper
Email: wlin@juniper.net

BESS Workgroup
INTERNET-DRAFT
Intended Status: Standards Track

A. Sajassi
P. Brissette
Cisco
J. Uttaro
ATT
J. Drake
W. Lin
Juniper
S. Boutros
VMWare
J. Rabadan
Nokia

Expires: May 1, 2017

November 1, 2016

EVPN VPWS Flexible Cross-Connect Service
draft-sajassi-bess-evpn-vpws-fxc-01.txt

Abstract

This document describes a new EVPN VPWS VLAN-aware bundle service type referred to as flexible cross-connect service. It also describes the rationale for this new service as well as a solution to deliver such service.

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) 2012 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	Requirements	4
4	Solution	6
4.1	VLAN-Unaware Flexible Xconnect - Single-Homing	7
4.2	VLAN-Aware Flexible Xconnect	8
4.2.1	Local Switching	9
4.3	VLAN-Unaware Flexible Xconnect - Multi-Homing	9
5	BGP Extensions	9
6	Failure Scenarios	11
6.2	EVPN VPWS service Failure	11
6.2	Attachment Circuit Failure	11
6.3	PE Port Failure	11
6.4	PE Node Failure	11
7	Security Considerations	11
8	IANA Considerations	11
9	References	11
9.1	Normative References	11
9.2	Informative References	12
	Authors' Addresses	12

1 Introduction

[EVPN-VPWS] describes a solution to deliver P2P services using BGP constructs defined in [RFC7432]. It delivers this P2P service between a pair of Attachment Circuits (ACs), where an AC can designate on a PE, a port, a VLAN on a port, or a group of VLANs on a port. It also leverages multi-homing and fast convergence capabilities of [RFC7432] in delivering these VPWS services. Multi-homing capabilities include the support of single-active and all-active redundancy mode and fast convergence is provided using "mass withdraw" message in control-plane and fast protection switching using prefix independent convergence in data-plane upon node or link failure. Furthermore, the use of EVPN BGP constructs eliminates the need for multi-segment PW auto-discovery and signaling if the VPWS service need to span across multiple ASes.

Some service providers have very large number of ACs (in millions) that require tag manipulation (e.g., VLAN translation) to be back hauled across their MPLS/IP network. These service providers want to multiplex a large number of ACs across several physical interfaces (e.g., several Ethernet Segments) onto a single VPWS service tunnel in order to a) reduce number of EVPN service labels associated with VPWS service tunnels and thus the associated OAM monitoring, and b) reduce EVPN BGP signaling (e.g., not to signal each AC as it is the case in [EVPN-VPWS]).

These service provider want the above functionality without scarifying any of the capabilities of [EVPN-VPWS] including single-active and all-active multi-homing, and fast convergence.

This document presents a solution based on extensions to [EVPN-VPWS] to meet the above requirements.

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].

MAC: Media Access Control

MPLS: Multi Protocol Label Switching

OAM: Operations, Administration and Maintenance

PE: Provide Edge Node

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

EVPL: Ethernet Virtual Private Line

EPL: Ethernet Private Line

ES: Ethernet Segment

VPWS: Virtual private wire service

EVI: EVPN Instance

VPWS Service Tunnel: It is represented by a pair of EVPN service labels associated with a pair of endpoints. Each label is downstream assigned and advertised by the disposition PE through an Ethernet A-D per-EVI route. The downstream label identifies the endpoint on the disposition PE. A VPWS service tunnel can be associated with many VPWS service identifiers for VLAN-aware VPWS service where each identifier is a normalized VID.

Single-Active Mode: When a device or a network is multi-homed to two or more PEs and when only a single PE in such redundancy group can forward traffic to/from the multi-homed device or network for a given VLAN, then such multi-homing or redundancy is referred to as "Single-Active".

All-Active: When a device is multi-homed to two or more PEs and when all PEs in such redundancy group can forward traffic to/from the multi-homed device for a given VLAN, then such multi-homing or redundancy is referred to as "All-Active".

2 Requirements

Two of the main motivations for service providers seeking a new solution are: 1) to reduce number of VPWS service tunnels by multiplexing large number of ACs across different physical interfaces instead of having one VPWS service tunnel per AC, and 2) to reduce the signaling of ACs as much as possible. Besides these two requirements, they also want multi-homing and fast convergence capabilities of [EVPN-VPWS].

In [EVPN-VPWS], a PE signals an AC indirectly by first associating that AC to a VPWS service tunnel (e.g., a VPWS service instance) and then signaling the VPWS service tunnel via a per-EVI Ethernet AD route with Ethernet Tag field set to a 24-bit VPWS service instance identifier (which is unique within the EVI) and ESI field set to a 10-octet identifier of the Ethernet Segment corresponding to that AC.

Therefore, a PE device that receives such EVPN routes, can associate the VPWS service tunnel to the remote Ethernet Segment, and when the remote ES fails and the PE receives the "mass withdraw" message associated with the failed ES per [RFC7432], it can update its BGP path list for that VPWS service tunnel quickly and achieve fast convergence for multi-homing scenarios. Even if fast convergence were not needed, there would still be a need for signaling each AC failure (via its corresponding VPWS service tunnel) associated with the failed ES, so that the BGP path list for each of them gets updated accordingly and the packets are sent to backup PE (in case of single-active multi-homing) or to other PEs in the redundancy group (in case of all-active multi-homing). In absence of updating the BGP path list, the traffic for that VPWS service tunnel will be black-holed.

When a single VPWS service tunnel multiplexes many ACs across number of Ethernet Segments (number of physical interfaces) and the ACs are not signaled via EVPN BGP to remote PE devices, then the remote PE devices neither know the association of the received Ethernet Segment to these ACs (and in turn to their local ACs) nor they know the association of the VPWS service tunnel (e.g., EVPN service label) to the far-end ACs - i.e, the remote PEs only know the association of their local ACs to the VPWS service tunnel but not the far-end ACs. Thus upon a connectivity failure to the ES, they don't know how to redirect traffic via another multi-homing PE to that ES. In other words, even if an ES failure is signaled via EVPN to the remote PE devices, they don't know what to do with such message because they don't know the association among the ES, their ACs, and the VPWS service tunnel.

In order to address this issue when multiplexing large number of ACs onto a single VPWS service tunnel, two mechanisms are devised: one to support VPWS services between two single-homed endpoints and another one to support VPWS services where one of the endpoints is multi-homed. An endpoint can be an AC, MAC-VRF, IP-VRF, global table, or etc.

For single-homed endpoints, it is OK not to signal each AC in BGP because upon connection failure to the ES, there is no alternative path to that endpoint. However, the ramification for not signaling an AC failure is that the traffic destined to the failed AC, is sent over MPLS/IP core and then gets discarded at the destination PE - i.e., it can waste network resources. However, when there is a connection failure, the application layer will eventually stop sending traffic and thus this wastage of network resources should be transient. Section 4.1 describes a solution for such single-homing VPWS service which is called VLAN-Unaware flexible cross-connect service.

For VPWS services where one of the endpoints is multi-homed, there are two options:

1) to signal each AC via BGP so that the path list can be updated upon a failure that impacts those ACs. This solution is described in section 4.2 and it is called VLAN-Aware flexible cross-connect service.

2) to bundle several ACs on an ES together per destination end-point (e.g., ES, MAC-VRF, etc.) and associated such bundle to a single VPWS service tunnel. This is similar to VLAN-bundle service interface described in [EVPN-VPWS]. This solution is described in section 4.3.

4 Solution

This section describes a solution for providing a new VPWS service between two PE devices where a large number of ACs (e.g., VLANs) that span across many Ethernet Segments (i.e., physical interfaces) on each PE are multiplex onto a single P2P EVPN LSP tunnel. Since multiplexing is done across several physical interfaces, there can be overlapping VLAN IDs across these interfaces; therefore, in such scenarios, the VLAN IDs (VIDs) MUST be translated into unique VIDs to avoid collision. Furthermore, if the number of VLANs that are getting multiplex onto a single VPWS service tunnel, exceed 4K, then a single tag to double tag translation MUST be performed. This translation of VIDs into unique VIDs (either single or double) is referred to as "VID normalization". When single normalized VID is used, the lower 12-bit of Ethernet tag field in EVPN routes is set to that VID and when double normalized VID is used, the lower 12-bit of Ethernet tag field is set to inner VID and the higher 12-bit is set to the outer VID.

Since there is only a single EVPN VPWS service tunnel associated with many normalized VIDs (either single or double), MPLS lookup at the disposition PE is no longer sufficient to forward the packet to the right egress endpoint/interface. Therefore, in addition to an EVPN label lookup corresponding to the VPWS service tunnel, a VID lookup (either single or double) is also required. On the disposition PE, one can think of the lookup of EVPN label results in identification of a VID-VRF, and the lookup of normalized VID(s) in that table, results in identification of egress endpoint/interface. The tag manipulation (translation from normalized VID(s) to local VID) can be performed either as part of the VID table lookup or at the egress interface itself.

Since VID lookup (single or double) needs to be performed at the

disposition PE, then VID normalization MUST be performed prior to the MPLS encapsulation on the ingress PE. This requires that both imposition and disposition PE devices be capable of VLAN tag manipulation, such as re-write (single or double), addition, deletion (single or double) at their endpoints (e.g., their ES's, MAC-VRFs, etc.).

4.1 VLAN-Unaware Flexible Xconnect - Single-Homing

In this mode of operation, many ACs across several Ethernet Segments are multiplex into a single EVPN VPWS service tunnel represented by a single VPWS service ID. VLAN-Unaware mode for this solution means that VLANs (normalized VIDs) are not signaled via EVPN BGP among the PEs. In this solution, there is only a single P2P EVPN LSP tunnel between a pair of PEs for all their ACs that are single-homed.

As discussed previously, since the VPWS service tunnel is used to multiplex ACs across different ES's (e.g., physical interfaces), the EVPN label alone is not sufficient for proper forwarding of the received packets (over MPLS/IP network) to egress interfaces. Therefore, normalized VID lookup is required in the disposition direction to forward packets to their proper egress end-points - i.e., the EVPN label lookup identifies a VID-VRF and subsequently, the normalized VID lookup in that table, identifies the egress interface.

In this solution, on each PE, the single-homing ACs represented by their normalized VIDs are associated with a single VPWS service tunnel (in a given EVI). The EVPN route that gets generated is an EVPN Ethernet AD per EVI route with ESI=0, Ethernet Tag field set to VPWS service instance ID, MPLS label field set to dynamically generated EVPN service label representing the EVPN VPWS service tunnel. This route is sent with an RT representing the EVI. This RT can be auto-generated from the EVI per section 5.1.2.1 of [EVPN-Overlay]. Furthermore, this route is sent with the EVPN Layer-2 Extended Community defined in section 3.1 of [EVPN-VPWS] with two new flags (defined in section 5) that indicate: 1) this VPWS service tunnel is for VLAN-unaware Flexible Cross-Connect, and 2) normalized VID type (single versus double). The receiving PE uses these new flags for consistency check and MAY generate an alarm if it detects inconsistency but doesn't bring down the VPWS service because such inconsistency may be intentional - i.e., one side is configured for VLAN-aware VPWS service and another side is configured for VLAN-unaware VPWS service.

It should be noted that in this mode of operation, a single Ethernet AD per EVI route is sent upon configuration of the first AC (ie,

normalized VID). Later, when additional ACs are configured and associated with this EVPN VPWS service tunnel, the PE does not advertise any additional EVPN BGP routes. The PE only associates locally these ACs with the already created VPWS service tunnel.

4.2 VLAN-Aware Flexible Xconnect

In this mode of operation, just as the VLAN-unaware mode, many normalized VIDs (ACs) across several different ES's/interfaces are multiplexed into a single EVPN VPWS service tunnel; however, this single tunnel is represented by many VPWS service IDs (one per normalized VID) and these normalized VIDs are signaled using EVPN BGP.

In this solution, on each PE, the multi-homing ACs represented by their normalized VIDs are configured with a single EVI. There is no need to configure VPWS service instance ID in here as it is the same as the normalized VID. For each normalized VID on each ES, the PE generates an EVPN Ethernet AD per EVI route where ESI field represents the ES ID, the Ethernet Tag field is set to the normalized VID, MPLS label field is set to dynamically generated EVPN label representing the P2P EVPN LSP tunnel. This route is sent with an RT representing the EVI. As before, this RT can be auto-generated from the EVI per section 5.1.2.1 of [EVPN-Overlay]. Furthermore, this route is sent with the EVPN Layer-2 Extended Community defined in section 3.1 of [EVPN-VPWS] with two new flags (defined in section 5) that indicate: 1) this VPWS service tunnel is for VLAN-aware Flexible Cross-Connect, and 2) normalized VID type (single versus double). The receiving PE uses these new flags for consistency check and MAY generate an alarm if it detects inconsistency but doesn't bring down the VPWS service because such inconsistency may be intentional - i.e., one side is configured for VLAN-aware VPWS service and another side is configured for VLAN-unaware VPWS service.

It should be noted that in this mode of operation, the PE sends a single Ethernet AD route for each AC that is configured - i.e., each normalized VID that is configured per ES results in generation of an EVPN Ethernet AD per EVI.

This mode of operation provides automatic cross checking of normalized VIDs used for EVPL services because these VIDs are signaled in EVPN BGP. For example, if the same normalized VID is configured on three PE devices (instead of two) for the same EVI, then when a PE receives the second EVPN EAD per-EVI route, it generates an error message unless the two EVPN EAD per-EVI routes include the same ESI. Such cross-checking is not feasible in VLAN-unaware FXC because the normalized VIDs are not signaled.

4.2.1 Local Switching

When cross-connection is between two ACs belonging to two multi-homed Ethernet Segments on the same set of multi-homing PEs, then forwarding between the two ACs MUST be performed locally during normal operation (e.g., in absence of a local link failure) - i.e., the traffic between the two ACs MUST be locally switched within the PE.

In terms of control plane processing, this means that when the receiving PE receives an Ethernet A-D per-EVI route whose ESI is a local ESI, the PE does not alter its forwarding state based on the received route. This ensures that the local switching takes precedence over forwarding via MPLS/IP network. This scheme of locally switched preference is consistent with baseline EVPN [RFC 7432] where it describes the locally switched preference for MAC/IP routes.

In such scenarios, the Ethernet A-D per EVI route should be advertised with the MPLS label either associated with the destination Attachment Circuit or with the destination Ethernet Segment in order to avoid any ambiguity in forwarding. In other words, the MPLS label cannot represent the same VID-VRF used in section 4.2 because the same normalized VID can be reachable via two Ethernet Segments. In case of using MPLS label per destination AC, then this same solution can be used for VLAN-based VPWS or VLAN-bundle VPWS services per [EVPN-VPWS].

4.3 VLAN-Unaware Flexible Xconnect - Multi-Homing

In this mode of operation, a group of normalized VIDs (ACs) on a single ES that are destined to a single endpoint are multiplexed into a single EVPN VPWS service tunnel represented by a single VPWS service ID. This mode of operation is the same as VLAN-bundle service interface of [EVPN-VPWS] except for the fact that VIDs on Ethernet frames are normalized before getting sent over the LSP tunnel.

In the previous two modes of operation, only a single EVPN VPWS service tunnel is needed per pair of PEs. However, in this mode of operation, there can be lot more service tunnels per pair of PEs - i.e, there is one tunnel per group of VIDs per pair of PEs and there can be many groups between a pair of PEs, thus resulting in many EVPN service tunnels.

5. BGP Extensions

This draft uses the EVPN Layer-2 attribute extended community defined in [EVPN-VPWS] with two additional flags added to this EC as described below. This EC is to be advertised with Ethernet A-D per EVI route per section 4.

```

+-----+
| Type(0x06)/Sub-type(TBD)(2 octet) |
+-----+
| Control Flags (2 octets)          |
+-----+
| L2 MTU (2 octets)                |
+-----+
| Reserved (2 octets)              |
+-----+

```

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+
| MBZ          | V | M | C | P | B | (MBZ = MUST Be Zero)
+---+---+---+---+---+---+---+---+

```

The following bits in the Control Flags are defined; the remaining bits MUST be set to zero when sending and MUST be ignored when receiving this community.

Name	Meaning
B,P,C	per definition in [EVPN-VPWS]
M	00 mode of operation as defined in [EVPN-VPWS] 01 VLAN-aware FXC 10 VLAN-unaware FXC
V	00 operating per [EVPN-VPWS] 01 single-VID normalization 10 double-VID normalization

The M and V fields are OPTIONAL on transmission and ignored at reception for forwarding purposes. They are used for error notifications.

6 Failure Scenarios

6.2 EVPN VPWS service Failure

The failure detection of an EVPN VPWS service can be performed via OAM mechanisms such as VCCV-BFD and upon such failure detection, the switch over procedure to the backup S-PE is the same as the one described above.

6.2 Attachment Circuit Failure

6.3 PE Port Failure

6.4 PE Node Failure

In the case of PE node failure, the operation is similar to the steps described above, albeit that EVPN route withdrawals are performed by the Route Reflector instead of the PE.

7 Security Considerations

TBD.

8 IANA Considerations

TBD

9 References

9.1 Normative References

[RFC7432] Sajassi et al., "Ethernet VPN", RFC 7432, February 2015.

[EVPN-IRB] Sajassi et al., "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-00, work in progress, November 2014.

[EVPN-PREFIX] Rabadan et al., "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-02, work in progress, September 2015.

[RFC6718] Muley P., et al., "Pseudowire Redundancy", RFC 6718, August 2012.

[RFC6870] Muley P., et al., "Pseudowire Preferential Forwarding Status Bit", RFC 6870, February 2013.

9.2 Informative References

[BGP-PIC] Bashandy A. et al., "BGP Prefix Independent Convergence", draft-rtgwg-bgp-pic-02.txt, work in progress, October 2013.

Authors' Addresses

A. Sajassi
Cisco
EMail: sajassi@cisco.com

P. Brissette
Cisco
EMail: pbrisset@cisco.com

J. Uttaro
ATT
EMail: jul738@att.com

J. Drake
Juniper
EMail: jdrake@juniper.net

S. Boutros
ATT
EMail: boutros.sami@gmail.com

W. Lin
Juniper
EMail: wlin@juniper.net

J. Rabadan
jorge.rabadan@nokia.com

BESS Workgroup
INTERNET-DRAFT
Intended Status: Standards Track

A. Sajassi
P. Brissette
Cisco
J. Uttaro
ATT
J. Drake
W. Lin
Juniper
S. Boutros
VMWare
J. Rabadan
Nokia

Expires: August 26, 2018

February 26, 2018

EVPN VPWS Flexible Cross-Connect Service
draft-sajassi-bess-evpn-vpws-fxc-03.txt

Abstract

This document describes a new EVPN VPWS service type specifically for multiplexing multiple attachment circuits across different Ethernet Segments and physical interfaces into a single EVPN VPWS service tunnel and still providing Single-Active and All-Active multi-homing. This new service is referred to as flexible cross-connect service. It also describes the rationale for this new service type as well as a solution to deliver such service.

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) 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	3
1.1	Terminology	3
2	Requirements	4
4	Solution	6
4.1	Flexible Xconnect	7
4.2	VLAN-Signaled Flexible Xconnect	8
4.2.1	Local Switching	9
5	BGP Extensions	9
6	Failure Scenarios	11
6.1	EVPN VPWS service Failure	13
6.2	Attachment Circuit Failure	13
6.3	PE Port Failure	14
6.4	PE Node Failure	14
7	Security Considerations	14
8	IANA Considerations	14
9	References	14
9.1	Normative References	14
9.2	Informative References	15
	Authors' Addresses	15

1 Introduction

[RFC8214] describes a solution to deliver P2P services using BGP constructs defined in [RFC7432]. It delivers this P2P service between a pair of Attachment Circuits (ACs), where an AC can designate on a PE, a port, a VLAN on a port, or a group of VLANs on a port. It also leverages multi-homing and fast convergence capabilities of [RFC7432] in delivering these VPWS services. Multi-homing capabilities include the support of single-active and all-active redundancy mode and fast convergence is provided using "mass withdraw" message in control-plane and fast protection switching using prefix independent convergence in data-plane upon node or link failure [BGP-PIC]. Furthermore, the use of EVPN BGP constructs eliminates the need for multi-segment PW auto-discovery and signaling if the VPWS service need to span across multiple ASes.

Some service providers have very large number of ACs (in millions) that need to be back hauled across their MPLS/IP network. These ACs may or may not require tag manipulation (e.g., VLAN translation). These service providers want to multiplex a large number of ACs across several physical interfaces spread across one or more PEs (e.g., several Ethernet Segments) onto a single VPWS service tunnel in order to a) reduce number of EVPN service labels associated with EVPN-VPWS service tunnels and thus the associated OAM monitoring, and b) reduce EVPN BGP signaling (e.g., not to signal each AC as it is the case in [RFC8214]).

These service provider want the above functionality without scarifying any of the capabilities of [RFC8214] including single-active and all-active multi-homing, and fast convergence.

This document presents a solution based on extensions to [RFC8214] to meet the above requirements.

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].

MAC: Media Access Control

MPLS: Multi Protocol Label Switching

OAM: Operations, Administration and Maintenance

PE: Provide Edge Node

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

EVPL: Ethernet Virtual Private Line

EPL: Ethernet Private Line

ES: Ethernet Segment

VPWS: Virtual private wire service

EVI: EVPN Instance

VPWS Service Tunnel: It is represented by a pair of EVPN service labels associated with a pair of endpoints. Each label is downstream assigned and advertised by the disposition PE through an Ethernet A-D per-EVI route. The downstream label identifies the endpoint on the disposition PE. A VPWS service tunnel can be associated with many VPWS service identifiers for VLAN-signaled VPWS service where each identifier is a normalized VID.

Single-Active Mode: When a device or a network is multi-homed to two or more PEs and when only a single PE in such redundancy group can forward traffic to/from the multi-homed device or network for a given VLAN, then such multi-homing or redundancy is referred to as "Single-Active".

All-Active: When a device is multi-homed to two or more PEs and when all PEs in such redundancy group can forward traffic to/from the multi-homed device for a given VLAN, then such multi-homing or redundancy is referred to as "All-Active".

2 Requirements

Two of the main motivations for service providers seeking a new solution are: 1) to reduce number of VPWS service tunnels by multiplexing large number of ACs across different physical interfaces instead of having one VPWS service tunnel per AC, and 2) to reduce the signaling of ACs as much as possible. Besides these two requirements, they also want multi-homing and fast convergence capabilities of [RFC8214].

In [RFC8214], a PE signals an AC indirectly by first associating that AC to a VPWS service tunnel (e.g., a VPWS service instance) and then signaling the VPWS service tunnel via a per-EVI Ethernet AD route with Ethernet Tag field set to a 24-bit VPWS service instance identifier (which is unique within the EVI) and ESI field set to a 10-octet identifier of the Ethernet Segment corresponding to that AC.

Therefore, a PE device that receives such EVPN routes, can associate the VPWS service tunnel to the remote Ethernet Segment, and when the remote ES fails and the PE receives the "mass withdraw" message associated with the failed ES per [RFC7432], it can update its BGP path list for that VPWS service tunnel quickly and achieve fast convergence for multi-homing scenarios. Even if fast convergence were not needed, there would still be a need for signaling each AC failure (via its corresponding VPWS service tunnel) associated with the failed ES, so that the BGP path list for each of them gets updated accordingly and the packets are sent to backup PE (in case of single-active multi-homing) or to other PEs in the redundancy group (in case of all-active multi-homing). In absence of updating the BGP path list, the traffic for that VPWS service tunnel will be black-holed.

When a single VPWS service tunnel multiplexes many ACs across number of Ethernet Segments (number of physical interfaces) and the ACs are not signaled via EVPN BGP to remote PE devices, then the remote PE devices neither know the association of the received Ethernet Segment to these ACs (and in turn to their local ACs) nor they know the association of the VPWS service tunnel (e.g., EVPN service label) to the far-end ACs - i.e, the remote PEs only know the association of their local ACs to the VPWS service tunnel but not the far-end ACs. Thus upon a connectivity failure to the ES, they don't know how to redirect traffic via another multi-homing PE to that ES. In other words, even if an ES failure is signaled via EVPN to the remote PE devices, they don't know what to do with such message because they don't know the association among the remote ES, the remote ACs, and the VPWS service tunnel.

In order to address this issue when multiplexing large number of ACs onto a single VPWS service tunnel, two mechanisms are devised: one to support VPWS services between two single-homed endpoints and another one to support VPWS services where one of the endpoints is multi-homed. An endpoint can be an AC, MAC-VRF, IP-VRF, global table, or etc.

For single-homed endpoints, it is OK not to signal each AC in BGP because upon connection failure to the ES, there is no alternative path to that endpoint. However, the ramification for not signaling an AC failure is that the traffic destined to the failed AC, is sent over MPLS/IP core and then gets discarded at the destination PE - i.e., it can waste network resources. However, when there is a connection failure, the application layer will eventually stop sending traffic and thus this wastage of network resources should be transient. Section 4.1 describes a solution for such single-homing VPWS service.

For VPWS services where one of the endpoints is multi-homed, there

are two options:

1) to signal each AC via BGP so that the path list can be updated upon a failure that impacts those ACs. This solution is described in section 4.2 and it is called VLAN-signaled flexible cross-connect service.

2) to bundle several ACs on an ES together per destination end-point (e.g., ES, MAC-VRF, etc.) and associated such bundle to a single VPWS service tunnel. This is similar to VLAN-bundle service interface described in [RFC8214]. This solution is described in section 4.3.

4 Solution

This section describes a solution for providing a new VPWS service between two PE devices where a large number of ACs (e.g., VLANs) that span across many Ethernet Segments (i.e., physical interfaces) on each PE are multiplex onto a single P2P EVPN service tunnel. Since multiplexing is done across several physical interfaces, there can be overlapping VLAN IDs across these interfaces; therefore, in such scenarios, the VLAN IDs (VIDs) MUST be translated into unique VIDs to avoid collision. Furthermore, if the number of VLANs that are getting multiplex onto a single VPWS service tunnel, exceed 4K, then a single tag to double tag translation MUST be performed. This translation of VIDs into unique VIDs (either single or double) is referred to as "VID normalization". When single normalized VID is used, the lower 12-bit of Ethernet tag field in EVPN routes is set to that VID and when double normalized VID is used, the lower 12-bit of Ethernet tag field is set to inner VID and the higher 12-bit is set to the outer VID.

Since there is only a single EVPN VPWS service tunnel associated with many normalized VIDs (either single or double) across multiple physical interfaces, MPLS lookup at the disposition PE is no longer sufficient to forward the packet to the right egress endpoint/interface. Therefore, in addition to an EVPN label lookup corresponding to the VPWS service tunnel, a VID lookup (either single or double) is also required. On the disposition PE, one can think of the lookup of EVPN label results in identification of a VID-VRF, and the lookup of normalized VID(s) in that table, results in identification of egress endpoint/interface. The tag manipulation (translation from normalized VID(s) to local VID) can be performed either as part of the VID table lookup or at the egress interface itself.

Since VID lookup (single or double) needs to be performed at the

disposition PE, then VID normalization MUST be performed prior to the MPLS encapsulation on the ingress PE. This requires that both imposition and disposition PE devices be capable of VLAN tag manipulation, such as re-write (single or double), addition, deletion (single or double) at their endpoints (e.g., their ES's, MAC-VRFs, IP-VRFs, etc.).

4.1 Flexible Xconnect

In this mode of operation, many ACs across several Ethernet Segments are multiplex into a single EVPN VPWS service tunnel represented by a single VPWS service ID. This is the default mode of operation for FXC and the participating PEs do not need to signal the VLANs (normalized VIDs) in EVPN BGP.

With respect to the data-plane aspects of the solution, both imposition and disposition PEs are aware of the VLANs as the imposition PE performs VID normalization and the disposition PE does VID lookup and translation. In this solution, there is only a single P2P EVPN VPWS service tunnel between a pair of PEs for a set of ACs.

As discussed previously, since the EVPN VPWS service tunnel is used to multiplex ACs across different ES's (e.g., physical interfaces), the EVPN label alone is not sufficient for proper forwarding of the received packets (over MPLS/IP network) to egress interfaces. Therefore, normalized VID lookup is required in the disposition direction to forward packets to their proper egress end-points - i.e., the EVPN label lookup identifies a VID-VRF and subsequently, the normalized VID lookup in that table, identifies the egress interface.

This mode of operation is only suitable for single-homing because in multi-homing the association between EVPN VPWS service tunnel and remote AC changes during the failure and therefore the VLANs (normalized VIDs) need to be signaled.

In this solution, on each PE, the single-homing ACs represented by their normalized VIDs are associated with a single EVPN VPWS service tunnel (in a given EVI). The EVPN route that gets generated is an EVPN Ethernet AD per EVI route with ESI=0, Ethernet Tag field set to VPWS service instance ID, MPLS label field set to dynamically generated EVPN service label representing the EVPN VPWS service tunnel. This route is sent with an RT representing the EVI. This RT can be auto-generated from the EVI per section 5.1.2.1 of [EVPN-Overlay]. Furthermore, this route is sent with the EVPN Layer-2 Extended Community defined in section 3.1 of [RFC8214] with two new flags (defined in section 5) that indicate: 1) this VPWS service

tunnel is for default Flexible Cross-Connect, and 2) normalized VID type (single versus double). The receiving PE uses these new flags for consistency check and MAY generate an alarm if it detects inconsistency but doesn't bring down the VPWS service.

It should be noted that in this mode of operation, a single Ethernet AD per EVI route is sent upon configuration of the first AC (ie, normalized VID). Later, when additional ACs are configured and associated with this EVPN VPWS service tunnel, the PE does not advertise any additional EVPN BGP routes. The PE only associates locally these ACs with the already created VPWS service tunnel.

The default FXC mode can be used for multi-homing. In this mode, a group of normalized VIDs (ACs) on a single Ethernet segment that are destined to a single endpoint are multiplexed into a single EVPN VPWS service tunnel represented by a single VPWS service ID. When the default FXC mode is used for multi-homing, instead of a single EVPN VPWS service tunnel, there can be many service tunnels per pair of PEs - i.e, there is one tunnel per group of VIDs per pair of PEs and there can be many groups between a pair of PEs, thus resulting in many EVPN service tunnels.

4.2 VLAN-Signaled Flexible Xconnect

In this mode of operation, just as the default FXC mode in section 4.1, many normalized VIDs (ACs) across several different ES's/interfaces are multiplexed into a single EVPN VPWS service tunnel; however, this single tunnel is represented by many VPWS service IDs (one per normalized VID) and these normalized VIDs are signaled using EVPN BGP.

In this solution, on each PE, the multi-homing ACs represented by their normalized VIDs are configured with a single EVI. There is no need to configure VPWS service instance ID in here as it is the same as the normalized VID. For each normalized VID on each ES, the PE generates an EVPN Ethernet AD per EVI route where ESI field represents the ES ID, the Ethernet Tag field is set to the normalized VID, MPLS label field is set to dynamically generated EVPN label representing the P2P EVPN service tunnel and it is the same label for all the ACs that are multiplexed into a single EVPN VPWS service tunnel. This route is sent with an RT representing the EVI. As before, this RT can be auto-generated from the EVI per section 5.1.2.1 of [EVPN-Overlay]. Furthermore, this route is sent with the EVPN Layer-2 Extended Community defined in section 3.1 of [RFC8214] with two new flags (defined in section 5) that indicate: 1) this VPWS service tunnel is for VLAN-signaled Flexible Cross-Connect, and 2) normalized VID type (single versus double). The receiving PE uses

these new flags for consistency check and MAY generate an alarm if it detects inconsistency but doesn't bring down the VPWS service.

It should be noted that in this mode of operation, the PE sends a single Ethernet AD route for each AC that is configured - i.e., each normalized VID that is configured per ES results in generation of an EVPN Ethernet AD per EVI.

This mode of operation provides automatic cross checking of normalized VID's used for EVPL services because these VID's are signaled in EVPN BGP. For example, if the same normalized VID is configured on three PE devices (instead of two) for the same EVI, then when a PE receives the second EVPN EAD per-EVI route, it generates an error message unless the two EVPN EAD per-EVI routes include the same ESI. Such cross-checking is not feasible in default FXC mode because the normalized VID's are not signaled.

4.2.1 Local Switching

When cross-connection is between two ACs belonging to two multi-homed Ethernet Segments on the same set of multi-homing PEs, then forwarding between the two ACs MUST be performed locally during normal operation (e.g., in absence of a local link failure) - i.e., the traffic between the two ACs MUST be locally switched within the PE.

In terms of control plane processing, this means that when the receiving PE receives an Ethernet A-D per-EVI route whose ESI is a local ESI, the PE does not alter its forwarding state based on the received route. This ensures that the local switching takes precedence over forwarding via MPLS/IP network. This scheme of locally switched preference is consistent with baseline EVPN [RFC 7432] where it describes the locally switched preference for MAC/IP routes.

In such scenarios, the Ethernet A-D per EVI route should be advertised with the MPLS label either associated with the destination Attachment Circuit or with the destination Ethernet Segment in order to avoid any ambiguity in forwarding. In other words, the MPLS label cannot represent the same VID-VRF used in section 4.2 because the same normalized VID can be reachable via two Ethernet Segments. In case of using MPLS label per destination AC, then this same solution can be used for VLAN-based VPWS or VLAN-bundle VPWS services per [RFC8214].

5. BGP Extensions

This draft uses the EVPN Layer-2 attribute extended community defined in [RFC8214] with two additional flags added to this EC as described below. This EC is to be advertised with Ethernet A-D per EVI route per section 4.

```

+-----+
| Type(0x06)/Sub-type(TBD)(2 octet) |
+-----+
| Control Flags (2 octets)           |
+-----+
| L2 MTU (2 octets)                 |
+-----+
| Reserved (2 octets)                |
+-----+

```

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| MBZ           | V | M | C | P | B | (MBZ = MUST Be Zero)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The following bits in the Control Flags are defined; the remaining bits MUST be set to zero when sending and MUST be ignored when receiving this community.

Name	Meaning
B,P,C	per definition in [RFC8214]
M	00 mode of operation as defined in [RFC8214] 01 VLAN-Signaled FXC 10 Default FXC
V	00 operating per [RFC8214] 01 single-VID normalization 10 double-VID normalization

The M and V fields are OPTIONAL on transmission and ignored at reception for forwarding purposes. They are used for error notifications.

6 Failure Scenarios

Two examples will be used as an example to analyze the failure scenarios.

The first scenario is depicted in Figure 1 and shows the VLAN-signaled FXC mode with Multi-Homing. In this example:

- CE1 is connected to PE1 and PE2 via (port,vid)=(p1,1) and (p3,3) respectively. CE1's VIDs are normalized to value 1 on both PEs, and CE1 is Xconnected to CE3's VID 1 at the remote end.
- CE2 is connected to PE1 and PE2 via ports p2 and p4 respectively:
 - o (p2,1) and (p4,3) identify the ACs that are used to Xconnect CE2 to CE4's VID 2, and are normalized to value 2.
 - o (p2,2) and (p4,4) identify the ACs that are used to Xconnect CE2 to CE5's VID 3, and are normalized to value 3.

In this scenario, PE1 and PE2 advertise an AD per-EVI route per normalized VID (values 1, 2 and 3), however only two VPWS Service Tunnels are needed: VPWS Service Tunnel 1 (sv.T1) between PE1's FXC service and PE3's FXC, and VPWS Service Tunnel 2 (sv.T2) between PE2's FXC and PE3's FXC.

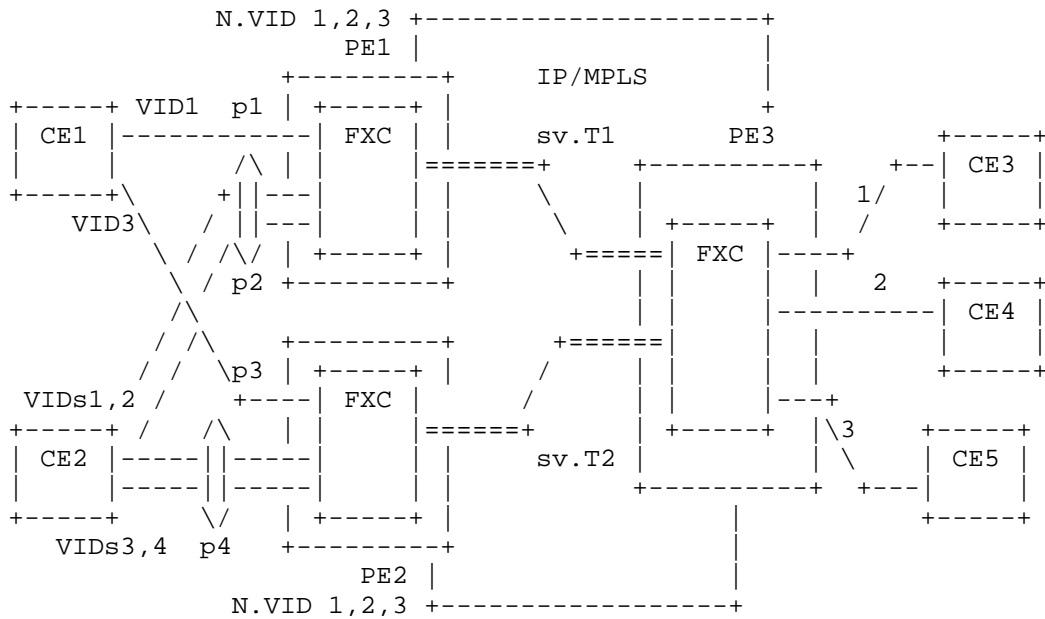


Figure 1 VLAN-Signaled Flexible Xconnect

The second scenario is a default Flexible Xconnect with Multi- Homing solution and it is depicted in Figure 2. In this case, the same VID Normalization as in the previous example is performed, however there is not an individual AD per-EVI route per normalized VID, but per bundle of ACs on an ES. That is, PE1 will advertise two AD per-EVI routes: the first one will identify the ACs on p1's ES and the second one will identify the AC2 in p2's ES. Similarly, PE2 will advertise two AD per-EVI routes.

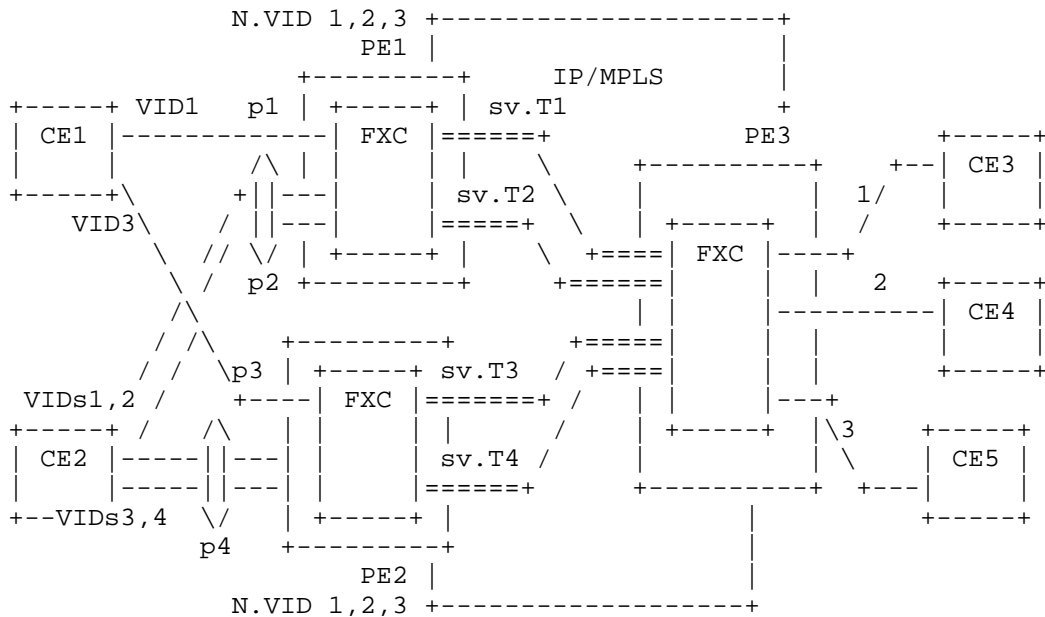


Figure 2 Default Flexible Xconnect

6.1 EVPN VPWS service Failure

The failure detection of an EVPN VPWS service can be performed via OAM mechanisms such as VCCV-BFD and upon such failure detection, the switch over procedure to the backup S-PE is the same as the one described above.

6.2 Attachment Circuit Failure

In case of AC Failure, the VLAN-Signaled and default FXC modes behave in a different way:

- o VLAN-signaled FXC (Figure 1): a VLAN or AC failure, e.g. VID1 on CE2, triggers the withdrawal of the AD per-EVI route for the corresponding Normalized VID, that is, Ethernet-Tag 2. When PE3 receives the route withdrawal, it will remove PE1 from its path-list for traffic coming from CE4.

- o Default FXC (Figure 2): a VLAN or AC failure is not signaled in the default mode, therefore in case of an AC failure, e.g. VID1 on CE2, nothing prevents PE3 from sending CE4's traffic to PE1, creating a

black-hole. Application layer OAM may be used if per-VLAN fault propagation is required in this case.

6.3 PE Port Failure

In case of PE port Failure, the failure will be signaled and the other PE will take over in both cases:

- o VLAN-signaled FXC (Figure 1): a port failure, e.g. p2, triggers the withdrawal of the AD per-EVI routes for Normalized VIDs 2 and 3, as well as the withdrawal of the AD per-ES route for p2's ES. Upon receiving the fault notification, PE3 will withdraw PE1 from its path-list for the traffic coming from CE4 and CE5.

- o Default FXC (Figure 2): a port failure, e.g. p2, is signaled by route for sv.T2 will also be withdrawn. Upon receiving the fault notification, PE3 will remove PE1 from its path-list for traffic coming from CE4 and CE5.

6.4 PE Node Failure

In the case of PE node failure, the operation is similar to the steps described above, albeit that EVPN route withdrawals are performed by the Route Reflector instead of the PE.

7 Security Considerations

There are no additional security considerations beyond what is already specified in [RFC8214].

8 IANA Considerations

TBD.

9 References

9.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC7432] Sajassi et al., "Ethernet VPN", RFC 7432, February 2015.

[RFC8214] Boutros et al., "Virtual Private Wire Service Support in Ethernet VPN", RFC 8214, August 2015.

9.2 Informative References

[BGP-PIC] Bashandy A. et al., "BGP Prefix Independent Convergence", draft-rtgwg-bgp-pic-02.txt, work in progress, October 2013.

[EVPN-Overlay] Sajassi et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-12, work in progress, February 2018.

Authors' Addresses

A. Sajassi
Cisco
EMail: sajassi@cisco.com

P. Brissette
Cisco
EMail: pbrisset@cisco.com

J. Uttaro
ATT
EMail: jul738@att.com

J. Drake
Juniper
EMail: jdrake@juniper.net

S. Boutros
ATT
EMail: boutros.sami@gmail.com

W. Lin
Juniper
EMail: wlin@juniper.net

J. Rabadan
jorge.rabadan@nokia.com