INTERNET-DRAFT
Intended Status: Standard Track

Sami Boutros
VMware
Rex Fernando
Ali Sajassi
Cisco Systems

Kitty Pang
Alibaba

Tapraj Singh
Juniper

Expires: September 17, 2016 March 16, 2016

EVPN auto provisioning using a controller
draft-boutros-bess-evpn-auto-provisoning-01

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/1id-abstracts.html

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

Table of Contents

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.

Majors 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

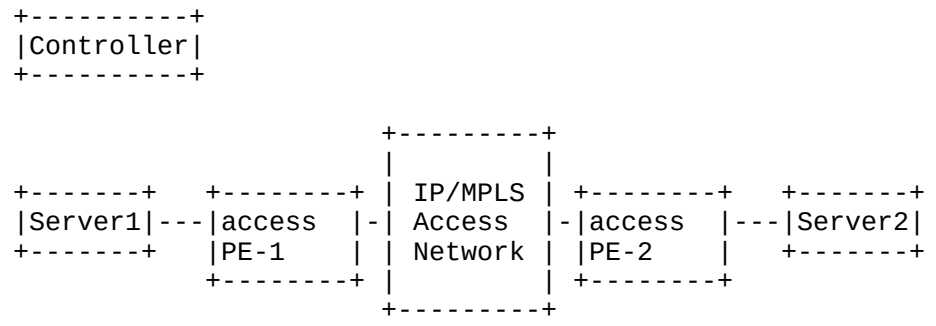


Figure 1:

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.

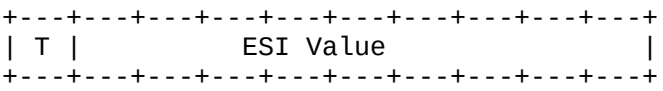
When the service configured by the controller is an EVPN service, the provisioned access PE/NVE will advertise to other BGP Peers Inclusive Multicast route, the receiving PE/NVE(s) will check if an EVPN

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.

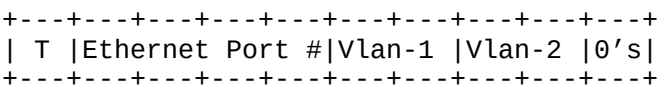
Please note that controller's capability is outside of the scope of this draft.

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.



The ESI 9 octets value will be as follow:



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
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/1id-abstracts.html

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

Table of Contents

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.

Majors 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

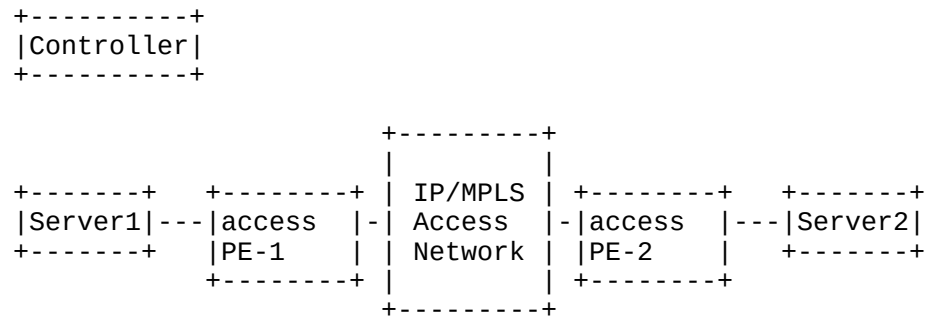


Figure 1:

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.

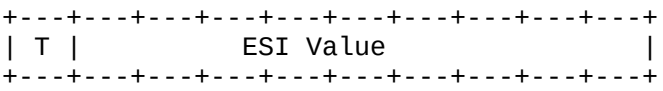
When the service configured by the controller is an EVPN service, the provisioned access PE/NVE will advertise to other BGP Peers Inclusive Multicast route, the receiving PE/NVE(s) will check if an EVPN

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.

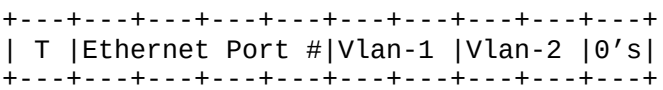
Please note that controller's capability is outside of the scope of this draft.

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.



The ESI 9 octets value will be as follow:



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: September 17, 2016

March 16, 2016

EVPN-VPWS Service Edge Gateway
draft-boutros-bess-evpn-vpws-service-edge-gateway-02

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/1id-abstracts.html

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

Table of Contents

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 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.

Majors benefits are:

- An easy and scalable mechanism for tunneling (head-end) customer traffic into a common IP/MPLS network infra structure
- 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

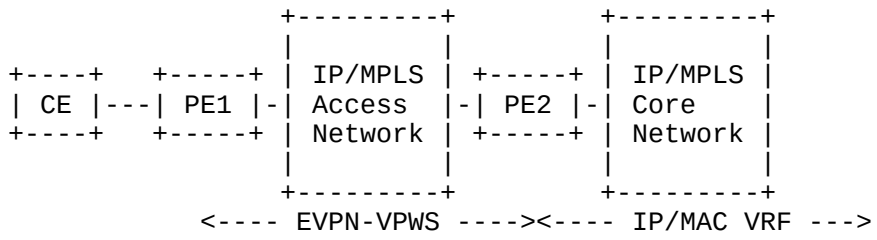


Figure 1: EVPN-VPWS Service Edge Gateway.
AN: Access node
SE: Service Edge node.

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 0xFFFFFFF). 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

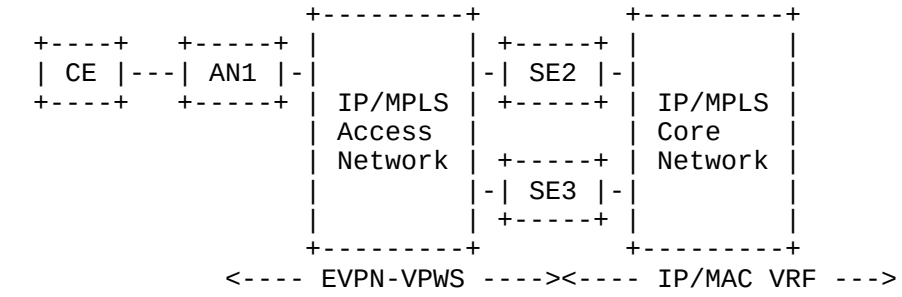


Figure 2: EVPN-VPWS SEG Multi-homing (same ASN)
AN: Access node
SE: Service Edge node.

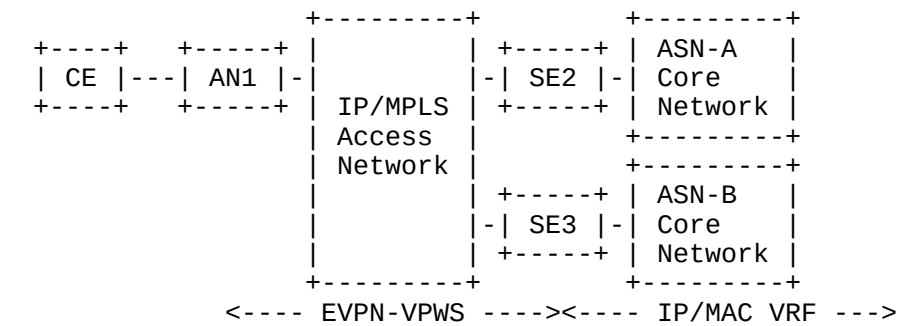


Figure 3: EVPN-VPWS SEG Multi-homing (different ASN)
AN: Access node
SE: Service Edge node.

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."

Copyright and License Notice

Table of Contents

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 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.

Majors benefits are:

- An easy and scalable mechanism for tunneling (head-end) customer traffic into a common IP/MPLS network infra structure
- 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

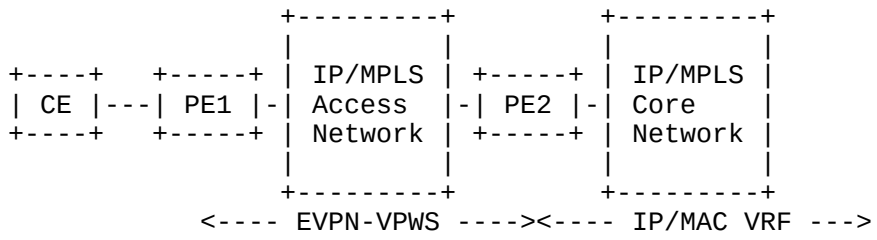


Figure 1: EVPN-VPWS Service Edge Gateway.
AN: Access node
SE: Service Edge node.

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 0xFFFFFFF). 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

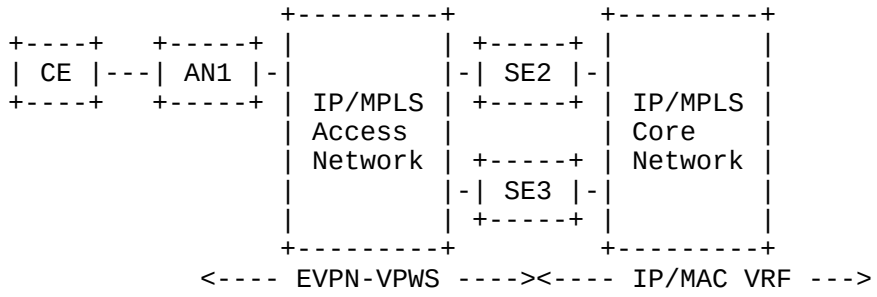


Figure 2: EVPN-VPWS SEG Multi-homing (same ASN)
AN: Access node
SE: Service Edge node.

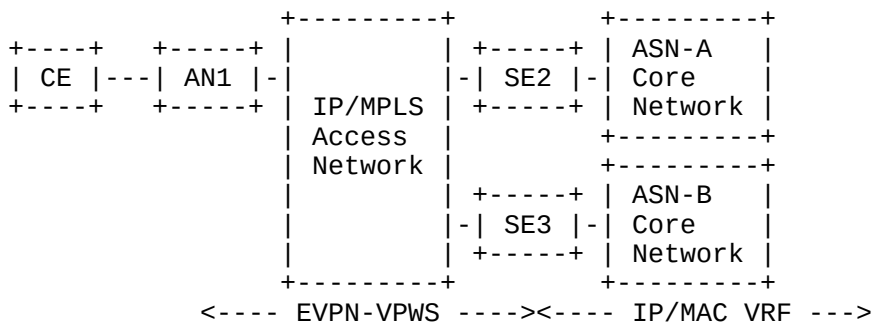


Figure 3: EVPN-VPWS SEG Multi-homing (different ASN)
AN: Access node
SE: Service Edge node.

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

BESS Working Group
Internet Draft
Intended Status: Proposed Standard
Expires: June 4, 2016

P. Brissette
Cisco System
H. Shah
Ciena Corporation
Z. Li
Huawei Technologies
A. liu
Ericsson
K. Tiruveedhula
T. Singh
Juniper Networks
I. Hussain
Infinera Corporation
J. Rabadan
December 2, 2015

Yang Data Model for EVPN
draft-brissette-bess-evpn-yang-01

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. The merging of this model with L2 services model is for future investigation. Any "add-on" features such as EVPN IRB, EVPN overlay, etc. are for future investigation. This document mainly focuses on EVPN 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."

Copyright and License Notice

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

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 will leverage 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 services. The definition work is undertaken initially by a smaller working group with members representing various vendors and service providers. The EVPN basic framework definition is covered first. Merging with L2 services model is left for future study. The EVPN basic framework consist of two modules: evpn and ethernet-segment. These models 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:

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

The document is organized to first define the data model for the configuration, operational state, actions 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 specific an interface. The interface can be a physical interface,

a bundle interface or virtual interface. The latter includes pseudowires. 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 service. For example, an access node can be dual-homed to two service nodes servicing a VPLS core. The access connectivity can be represented by an ethernet-segment where EVPN BGP DF election is performed over both service nodes. The core remains VPLS. Therefore, there is no EVPN instance being used here.

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 2 main containers: common and instance. The first one has common attributes to all VPNs where as the latter has attributes specific to an EVI. 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:

- Requirements for EVPN: RFC 7209
- EVPN: RFC 7432
- PBB-EVPN: RFC 7623

The integration with L2VPN instance Yang model is left for future study. Following documents will be covered at that time:

- VPWS support in EVPN: draft-ietf-bess-evpn-vpws-00
- E-TREE Support in EVPN & PBB-EVPN: draft-ietf-bess-evpn-etree-02
- (PBB-)EVPN Seamless Integration with (PBB-)VPLS: draft-ietf-bess-evpn-vpls-seamless-integ-00
- EVPN Virtual Ethernet Segment: draft-sajassi-bess-evpn-virtual-eth-segment-00

The VxLAN aspect and the work related to Layer 3 is also for future definition. Following documents will be covered at that time:

o IP Prefix Advertisement in EVPN:
draft-ietf-bess-evpn-prefix-advertisement-02
o VXLAN DCI Using EVPN:
draft-boutros-l2vpn-vxlan-evpn
o A Network Virtualization Overlay Solution using EVPN:
draft-ietf-bess-evpn-overlay-00
o Interconnect Solution for EVPN Overlay networks:
draft-ietf-bess-dci-evpn-overlay-00
o Integrated Routing and Bridging in EVPN:
draft-ietf-bess-evpn-inter-subnet-forwarding-00

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
         +--rw esi?                    empty
         +--rw (active-mode)
         |  +--:(single-active)
         |  |  +--rw single-active-mode?   empty
         |  +--:(all-active)
         |     +--rw all-active-mode?      empty
         +--rw bgp-parameters
         |  +--rw common
         |     +--rw route-distinguisher?   string
         |     +--rw vpn-targets* [rt-value]
         |        +--rw rt-value    string
         |        +--rw rt-type     bgp-rt-type
         +--rw df-election
            +--rw (df-election-method)?
            |  +--:(highest-random-weight)
            |     +--rw enable-hrw?           empty
            +--rw election-wait-time?   uint32
```

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 source-bmac?      yang:hex-string
               +--rw evpn-arp-proxy?   boolean
               +--rw nd-arp-proxy?     boolean
               +--rw bgp-parameters
                  +--rw common
                     +--rw route-distinguisher?   string
                     +--rw vpn-targets* [rt-value]
                        +--rw rt-value    string
                        +--rw rt-type     bgp-rt-type
```

4. YANG Module

The EVPN configuration container is logically divided into following high level config areas:

4.1 Ethernet Segment Yang Module

```
<CODE BEGINS> file "ietf-ethernet-segment@2015-12-02.yang"
module ietf-ethernet-segment {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ethernet-segment";
  prefix "es";

  import ietf-evpn {
    prefix "evpn";
  }

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

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

  /* 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 esi {
           type empty;
           description "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";
             }
           }
         }
         uses evpn:bgp-parameters-grp;
         container df-election {
           description "df-election";
           choice df-election-method {
             description "Choice of df election method";
             case highest-random-weight {
               leaf enable-hrw {
                 type empty;
                 description "enable-hrw";
               }
             }
           }
           leaf election-wait-time {
             type uint32;
             description "election-wait-time";
           }
         }
         description "An ethernet segment";
       }
     }
   }
```

<CODE ENDS>

4.2 EVPN Yang Module

```
<CODE BEGINS> file "ietf-evpn@2015-12-02.yang"
module ietf-evpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-evpn";
  prefix "evpn";

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

  organization  "ietf";
  contact       "ietf";
  description   "evpn";

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

  /* Typedefs */

  typedef bgp-rt-type {
    type enumeration {
      enum import {
        description "For import";
      }
      enum export {
        description "For export";
      }
      enum both {
        description "For both import and export";
      }
    }
    description "BGP route-target type. Import from BGP YANG";
  }

  /* Groupings */

  grouping bgp-parameters-grp {
    description "BGP parameters grouping";
    container bgp-parameters {
      description "BGP parameters";
      container common {
        description "Common BGP parameters";
```

```
            leaf route-distinguisher {
              type string;
              description "BGP RD";
            }
            list vpn-targets {
              key rt-value;
              description "Route Targets";
              leaf rt-value {
                type string;
                description "Route-Target value";
              }
              leaf rt-type {
                type bgp-rt-type;
                mandatory true;
                description "Type of RT";
              }
            }
          }
        }
      }

      /* EVPN YANG Model */

      container evpn {
        description "evpn";
        container common {
          description "common epn 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";
            }
            leaf source-bmac {
              type yang:hex-string;
              description "source-bmac";
            }
            leaf evpn-arp-proxy {
              type boolean;
              description "evpn-arp-proxy";
            }
            leaf nd-arp-proxy {
              type boolean;
              description "nd-arp-proxy";
            }
            uses bgp-parameters-grp;
          }
        }
      }
    }
    <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. Acknowledgments

The authors would like to acknowledge TBD for their useful comments.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

[RFC6241] R.Enns et al., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011

[RFC6020] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6242] M. Wasserman, "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6536] A. Bierman et al., "Network Configuration Protocol (NETCONF) Access Control Model" RFC 6536, March 2012.

[RFC7432] Sajassi et al., "BGP MPLS-Based Ethernet VPN", RFC 7432, February 2015.

[RFC7623] Sajassi et al., "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, September 2015

Authors' Addresses

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

Himanshu Shah
Ciena Corporation
EMail: hshah@ciena.com

Zhenbin Li
Huawei Technologies
EMail: lizhenbin@huawei.com

Autumn Liu
Ericsson
EMail: autumn.liu@ericsson.com

Kishore Tiruveedhula
Juniper Networks
EMail: kishoret@juniper.net

Tapraj Singh
Juniper Networks
EMail: tsingh@juniper.net

Iftekar Hussain
Infinera Corporation
EMail: ihussain@infinera.com

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 20, 2017

D. Jain
K. Patel
P. Brissette
Cisco
Z. Li
S. Zhuang
Huawei Technologies
X. Liu
Ericsson
J. Haas
S. Esale
Juniper Networks
B. Wen
Comcast

August 19, 2016

Yang Data Model for BGP/MPLS L3 VPNs
draft-dhjain-bess-bgp-l3vpn-yang-02.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 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 February 20, 2017.

Copyright Notice

Table of Contents

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 sepcific 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 routing-instance container defined in the routing model [I-D.ietf-netmod-routing-cfg] 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.shaikh-idr-bgp-model].

3.2. VRF Specific Configuration

Routing-instance defined in the IETF routing model defines a default instance when routing-instance type is default-routing-instance and named vrf instance when type is vrf-routing-instance. For L3VPN, the VRF specific parameters are defined by augmenting the routing-instance container corresponding to named vrf instance. A new container l3vpn is added for VPN parameters.

3.2.1. VRF interface

To associate a VRF instance with an interface, the interface should be defined in the context of routing-instance representing a VRF. This is covered in base routing model [I-D.ietf-netmod-routing-cfg].

3.2.2. Route distinguisher

Route distinguisher (RD) is an unique identifier used in VPN routes to distinguish prefixes across different VPNs. RD is 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, RDs are defined l3vpn container under routing-instance.

3.2.3. Import and export route target

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. An import rule is modeled as list of RTs or a policy leafref specifying the list of RTs to be matched for importing routes into the VRF. Similarly an export rule is set or RTs or a policy leafref specifying the list of RTs which should be attached to routes exported from this 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 routing-instance as defined in IETF routing model [I-D.ietf-netmod-routing-cfg]. 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

```
augment /rt:routing/rt:routing-instance:
  +--rw l3vpn
     +--rw route-distinguisher
     |  +--rw config
     |  |  +--rw rd?   string
     |  +--ro state
     |     +--ro rd?   string
     +--rw ipv4
     |  +--rw unicast
     |     +--rw import-routes
```

```
|     |  +--rw config
|     |  |  +--rw route-targets
|     |  |  |  +--rw rts* [rt]
|     |  |  |     +--rw rt    string
|     |  |  +--rw route-policy?    string
|     |  +--ro state
|     |     +--ro route-targets
|     |     |  +--ro rts* [rt]
|     |     |     +--ro rt    string
|     |     +--ro route-policy?    string
|     +--rw export-routes
|     |  +--rw config
|     |  |  +--rw route-targets
|     |  |  |  +--rw rts* [rt]
|     |  |  |     +--rw rt    string
|     |  |  +--rw route-policy?    string
|     |  +--ro state
|     |     +--ro route-targets
|     |     |  +--ro rts* [rt]
|     |     |     +--ro rt    string
|     |     +--ro route-policy?    string
|     +--rw import-export-routes
|     |  +--rw config
|     |  |  +--rw route-targets
|     |  |  |  +--rw rts* [rt]
|     |  |  |     +--rw rt    string
|     |  |  +--rw route-policy?    string
|     |  +--ro state
|     |     +--ro route-targets
|     |     |  +--ro rts* [rt]
|     |     |     +--ro rt    string
|     |     +--ro route-policy?    string
|     +--rw import-from-global
|     |  +--rw config
|     |  |  +--rw enable?             boolean
|     |  |  +--rw advertise-as-vpn?   boolean
|     |  |  +--rw route-policy?       string
|     |  |  +--rw bgp-valid-route?    boolean
|     |  |  +--rw protocol?           enumeration
|     |  |  +--rw instance?           string
|     |  +--ro state
|     |     +--ro enable?             boolean
|     |     +--ro advertise-as-vpn?   boolean
|     |     +--ro route-policy?       string
|     |     +--ro bgp-valid-route?    boolean
|     |     +--ro protocol?           enumeration
|     |     +--ro instance?           string
|     +--rw export-to-global
```

```
         |     |  +--rw config
         |     |  |  +--rw enable?   boolean
         |     |  +--ro state
         |     |     +--ro enable?   boolean
         |     +--rw routing-table-limit
         |     |  +--rw config
         |     |  |  +--rw routing-table-limit-number?   uint32
         |     |  |  +--rw (routing-table-limit-action)?
         |     |  |     +--:(enable-alert-percent)
         |     |  |     |  +--rw alert-percent-value?          uint8
         |     |  |     +--:(enable-simple-alert)
         |     |  |        +--rw simple-alert?                boolean
         |     |  +--ro state
         |     |     +--ro routing-table-limit-number?   uint32
         |     |     +--ro (routing-table-limit-action)?
         |     |        +--:(enable-alert-percent)
         |     |        |  +--ro alert-percent-value?          uint8
         |     |        +--:(enable-simple-alert)
         |     |           +--ro simple-alert?                boolean
         |     +--rw tunnel-params
         |        +--rw config
         |        |  +--rw tunnel-policy?   string
         |        +--ro state
         |           +--ro tunnel-policy?   string
```

```
   augment /if:interfaces/if:interface:
      +--rw forwarding-mode
      |  +--rw config
      |  |  +--rw forwarding-mode?   fwd-mode-type
      |  +--ro state
      |     +--ro forwarding-mode?   fwd-mode-type
      +--rw mpls-label-security
         +--rw config
         |  +--rw rpf?   boolean
         +--ro state
            +--ro rpf?   boolean
```

3.3. BGP Specific Configuration

The BGP specific configuration for L3VPNs is defined by augmenting base BGP model [I-D.shaikh-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 default routing-instance. Base BGP draft [I-D.shaikh-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 routing instance realizing a VRF.

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 and rewrite next hop to self, for inter-as VPN peering across ASBRs with option-B method. Similarly next hop unchanged on ASBRs for option-C peering. Appropriate containers under BGP AF and NBR modes are augmented for these parameters. As a note, when a knob is applicable for neighbor, it is also defined under corresponding peer-group container.

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-rts
      +--rw config
```

```
      |  +--rw all?            empty
      |  +--rw route-policy?   string
      +--ro state
         +--ro all?            empty
         +--ro route-policy?   string
   +--rw prefix-limit
      +--rw config
      |  +--rw prefix-limit-number?   uint32
      |  +--rw (prefix-limit-action)?
      |     +--:(enable-alert-percent)
      |     |  +--rw alert-percent-value?   uint8
      |     |  +--rw route-unchanged?       boolean
      |     +--:(enable-simple-alert)
      |        +--rw simple-alert?          boolean
      +--ro state
         +--ro prefix-limit-number?   uint32
         +--ro (prefix-limit-action)?
            +--:(enable-alert-percent)
            |  +--ro alert-percent-value?   uint8
            |  +--ro route-unchanged?       boolean
            +--:(enable-simple-alert)
               +--ro simple-alert?          boolean       ...

augment /bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast:
   +--rw config
   |  +--rw label-mode?   bgp-label-mode
   +--ro state
      +--ro label-mode?   bgp-label-mode
   +--rw routing-table-limit
      +--rw config
      |  +--rw routing-table-limit-number?   uint32
      |  +--rw (routing-table-limit-action)?
      |     +--:(enable-alert-percent)
      |     |  +--rw alert-percent-value?          uint8
      |     +--:(enable-simple-alert)
      |        +--rw simple-alert?                 boolean
      +--ro state
         +--ro routing-table-limit-number?   uint32
         +--ro (routing-table-limit-action)?
            +--:(enable-alert-percent)
            |  +--ro alert-percent-value?          uint8
            +--:(enable-simple-alert)
               +--ro simple-alert?                 boolean
      ...

augment /bgp:bgp/bgp:neighbors/bgp:neighbor:
   +--rw nexthop-options
      +--rw config
```

```
      |  +--rw next-hop-self?        boolean
      |  +--rw next-hop-unchanged?   boolean
      +--rw state
         +--rw next-hop-self?        boolean
         +--rw next-hop-unchanged?   boolean

augment /bgp:bgp/bgp:peer-groups/bgp:peer-group:
   +--rw nexthop-options
      +--rw config
      |  +--rw next-hop-self?        boolean
      |  +--rw next-hop-unchanged?   boolean
      +--rw state
         +--rw next-hop-self?        boolean
         +--rw next-hop-unchanged?   boolean

augment /bgp:bgp/bgp:neighbors/bgp:neighbor/bgp:afi-safis/bgp:afi-safi:
   +--rw nexthop-options
      +--rw config
      |  +--rw next-hop-self?        boolean
      |  +--rw next-hop-unchanged?   boolean
      +--rw state
         +--rw next-hop-self?        boolean
         +--rw next-hop-unchanged?   boolean

augment /bgp:bgp/bgp:peer-groups/bgp:peer-group/bgp:afi-safis/bgp:afi-safi:
   +--rw nexthop-options
      +--rw config
      |  +--rw next-hop-self?        boolean
      |  +--rw next-hop-unchanged?   boolean
      +--rw state
         +--rw next-hop-self?        boolean
         +--rw next-hop-unchanged?   boolean
```

4. BGP Yang Module

<CODE BEGINS> file "ietf-bgp-l3vpn@2016-02-22.yang"

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

  import ietf-routing {
    prefix rt;
```

```
    revision-date 2015-10-16;
  }

  import ietf-interfaces {
    prefix if;
  }

  import ietf-bgp {
    prefix bgp;
    revision-date 2016-01-06;
  }

  organization
     "IETF BGP Enabled Services WG";

  contact
     "draft-dhjain-bess-l3vpn-yang@tools.ietf.org";

  description
    "This YANG module defines a YANG data model to configure and manage BGP Laye
r3 VPNs.
     It augments the IETF bgp yang model and IETF routing model to add L3VPN spe
cific
     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 2016-02-22 {
    description
      "Initial revision.";
    reference
      "RFC XXXX: A YANG Data Model for BGP L3VPN config management";
  }


  grouping bgp-rd-spec {
    description "Route distinguisher specification as per RFC4364";
      leaf rd {
        type string;
        description "Route distinguisher value as per RFC4364";
    }
  }
  grouping bgp-rd {
    description "BGP route distinguisher";
    container route-distinguisher {
      description "Route distinguisher";
        container config {
          description "Configuration parameters for route distinguisher";
          uses bgp-rd-spec ;
        }
        container state {
          config "false" ;
          description "State information for route distinguisher";
          uses bgp-rd-spec ;
        }
     }
  }


  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";
  }

  typedef fwd-mode-type {
    type enumeration {
      enum mpls {
        description "Forwarding mode mpls";
      }
    }
    description "Enable forwarding mode under ASBR facing interface";
  }

  grouping forwarding-mode {
    description "Forwarding mode of interface for ASBR scenario";
    container forwarding-mode {
      description "Forwarding mode of interface for ASBR scenario";
      container config {
        description "Configuration of Forwarding mode";
        leaf forwarding-mode {
          type  fwd-mode-type;
          description "Forwarding mode for this interface";
        }
      }
      container state {
        config "false";
        description "State information of Forwarding mode";
        leaf forwarding-mode {
        type  fwd-mode-type;
          description "Forwarding mode for this interface";
        }
      }
    }
 }

  grouping label-security {
    description "Mpls label security for ASBR option B scenario";
    container mpls-label-security {
      description "MPLS label secruity";
      container config {
        description "Configuration parameters";
        leaf rpf {
          type boolean;
          description "Enable MPLS label security rpf on interface";
        }
```

```
      }
      container state {
        config "false";
        description "State information";
        leaf rpf {
          type boolean;
          description "MPLS label security rpf on interface";
        }
      }
    }
  }


  //per VPN instance table limit under BGP
  grouping prefix-limit {
    description
       "The prefix limit command 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 uint8 {
             range "1..100";
           }
           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  vpn-pfx-limit {
    description "Per VPN instance table limit under BGP";
    container vpn-prefix-limit {
      description "Prefix limit for this table";
      container config {
         description "Config parameters";
         uses prefix-limit;
      }
      container state {
         config "false";
         description "State parameters";
         uses prefix-limit;
     }
   }
 }

  grouping route-target-set {
    description
      "Extended community route-target set ";
    container route-targets {
      description
        "Route-target" ;
      list rts {
        key "rt" ;
        description
```

```
          "List of route-targets" ;
        leaf rt {
          type string {
            pattern '([0-9]+:[0-9]+)';
          }
          description "Route target extended community as per RFC4360";
        }
      }
    }
    leaf route-policy {
      type string;
      description
        "Reference to the policy containing set of routes.
               TBD: leafref to policy entry in IETF policy model";
    }
  }

  grouping import-from-gbl {
    description "Import from global routing table";
    leaf enable {
      type boolean;
        description "Enable";
    }
    leaf advertise-as-vpn {
      when "../from-default-vrf == TRUE" {
        description "This option is valid only when importing from global routin
g table";
      }
      type boolean;
      description "Advertise routes imported from global table as VPN routes";
    }
    leaf route-policy {
      type string;
      description "Policy name or import routes";
    }

    leaf bgp-valid-route {
      type boolean;
      description "Enable all valid routes (including non-best paths) to be cand
idate
                   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:";
        }
        enum INVALID {
          value "9";
          description "INVALID:";
        }
      }
      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-imports {
      description "Grouping for imports from global routing table";
    container import-from-global {
      description "Import from global global routing table";
      container config {
          description "Configuration";
          uses import-from-gbl;
      }
      container state {
          config "false";
          description "State";
          uses import-from-gbl;
      }
    }
  }


  grouping export-to-gbl {
    description "Export routes to default VRF";
    leaf enable {
      type boolean;
        description "Enable";
    }
  }

  grouping global-exports {
    description "Grouping for exports routes to global table";
    container export-to-global {
      description "Export to global routing table";
      container config {
         description "Configuration";
         uses export-to-gbl;
      }
      container state {
        config "false";
        description "State";
        uses export-to-gbl;
      }
    }
  }

  grouping route-import-set {
    description "Grouping to specify rules for route import";
    container import-routes {
      description "Set of route-targets to match to import routes into VRF";
      container config {
         description
               "Configuration parameters for import routes";
```

```
            uses route-target-set ;
        }
        container state {
          config "false" ;
          description
                 "State information for the import routes";
            uses route-target-set ;
        }
      }
    }
    grouping route-export-set {
      description "Grouping to specify rules for route export";
      container export-routes {
        description "Set of route-targets to attach with exported routes from VRF"
;
        container config {
          description
               "Configuration parameters for export routes";
            uses route-target-set ;
        }
        container state {
          config "false" ;
          description
                 "State information for export routes";
            uses route-target-set ;
        }
      }
    }

    grouping route-import-export-set {
      description "Grouping to specify rules for route import/export both";
      container import-export-routes {
        description "Set of route-targets for import/export both";
        container config {
          description "Both import/export routes";
          uses route-target-set;
        }
        container state {
          config "false" ;
            description "Both import/export routes";
            uses route-target-set;
        }
      }
    }

    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 uint8 {
                 range "1..100";
               }
               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.";
          container config {
             description "Config parameters";
             uses route-tbl-limit-params;
          }
          container state {
              config "false";
              description "State parameters";
              uses route-tbl-limit-params;
          }
       }
   }

  // Tunnel policy parameters
  grouping tunnel-params {
     description "Tunnel parameters";
     container tunnel-params {
        description "Tunnel config parameters";
        container config {
           description "configuration parameters";
           leaf tunnel-policy {
              type string;
              description
                  "Tunnel policy name.";
           }
        }
        container state {
           config "false";
           description "state parameters";
           leaf tunnel-policy {
              type string;
              description
                   "Tunnel policy name.";
           }
        }
     }
  }

  // Grouping for the L3vpn specific parameters under VRF (aka routing-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-import-set;
            uses route-export-set;
            uses route-import-export-set;
            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-import-set;
              uses route-export-set;
              uses route-import-export-set;
              uses global-imports;
              uses global-exports;
              uses routing-tbl-limit;
              uses tunnel-params;
          }
      }
  }

  grouping bgp-label-mode {
       description "MPLS/VPN label allocation mode";
       container config {
         description "Configuration parameters for label allocation mode";
         leaf label-mode {
           type bgp-label-mode;
           description "Label allocation mode";
         }
       }
       container state {
         config "false" ;
         description "State information for 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";
```

```
      container config {
         description "Configuration parameters for retaining route targets";
         leaf all {
            type empty;
            description "Disable filtering of all route-targets";
         }
         leaf route-policy {
            type string;
            description "Filter routes as per filter policy name
                         TBD: leafref to IETF routing policy model";
         }
      }
      container state {
         config "false" ;
         description "State information for retaining route targets";
         leaf all {
            type empty;
            description "Disable filtering of all route-targets";
         }
         leaf route-policy {
            type string;
            description "Filter routes as per filter policy name";
         }
      }
    }
  }

  grouping nexthop-opts {
     description "Next hop control options for inter-as route exchange";
     leaf next-hop-self {
        type boolean;
        description "Set nexthop of the route to self when advertising routes";
     }
     leaf next-hop-unchanged {
        type boolean;
        description "Enforce no nexthop change when advertising routes";
     }
  }

  grouping asbr-nexthop-options {
     description "Nexthop parameters for inter-as VPN options ";
     container nexthop-options {
         description "Nexthop related options for inter-as options";
         container config {
             description "Configuration parameters for nexthop options";
             uses nexthop-opts;
         }
         container state {
```

```
            config "false";
            description "State information for nexthop options" ;
            uses nexthop-opts;
          }
      }
    }

    //
    // VRF specific parameters.
    // RD and RTs are added in VRF routing-intance, therefore per per VRF scoped.
    //

    // route import-export rules in VRF context
    // (routing instance container in ietf-routing model).
    augment "/rt:routing/rt:routing-instance" {
       description "Augment routing instance container for per VRF import/export c
onfig";
       container l3vpn {
          when "../type='rt:vrf-routing-instance'" {
             description "This container is only valid for vrf routing instance.";
          }
          description "Configuration of L3VPN specific parameters";

          uses bgp-rd;
          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 sc
enario";
      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 configura
tion";
     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 configura
tion";
     uses bgp-label-mode ;
     uses routing-tbl-limit;
  }


  // Nexthop options for the inter-as ASBR peering.
  augment "/bgp:bgp/bgp:neighbors/bgp:neighbor" {
     description "Augment BGP NBR mode with nexthop options for inter-as ASBRs";
     uses asbr-nexthop-options;
  }

  augment "/bgp:bgp/bgp:peer-groups/bgp:peer-group" {
     description "Augment BGP peer-group mode with nexthop options for inter-as
ASBRs";
     uses asbr-nexthop-options;
  }

  augment "/bgp:bgp/bgp:neighbors/bgp:neighbor/bgp:afi-safis/bgp:afi-safi" {
     description "Augment BGP NBR AF mode with nexthop options for inter-as ASBR
s";
     uses asbr-nexthop-options;
  }

  augment "/bgp:bgp/bgp:peer-groups/bgp:peer-group/bgp:afi-safis/bgp:afi-safi" {
     description "Augment BGP peer-group AF mode with nexthop options for inter-
as ASBRs";
     uses asbr-nexthop-options;
  }

}


<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-netmod-routing-cfg] and [I-D.shaikh-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-netmod-routing-cfg]
Lhotka, L., "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-15 (work in progress), May 2014.

[I-D.shaikh-idr-bgp-model]
Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Alex, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", draft-shaikh-idr-bgp-model-02 (work in progress), June 2015.

[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>.

[RFC2547] Rosen, E. and Y. Rekhter, "BGP/MPLS VPNs", RFC 2547, DOI 10.17487/RFC2547, March 1999, <http://www.rfc-editor.org/info/rfc2547>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <http://www.rfc-editor.org/info/rfc2629>.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <http://www.rfc-editor.org/info/rfc3552>.

[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>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://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, <http://www.rfc-editor.org/info/rfc6241>.

8.2. Informative References

[RFC5492] Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <http://www.rfc-editor.org/info/rfc5492>.

Authors' Addresses

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

Email: dhjain@cisco.com

Keyur Patel
Cisco
170 W. Tasman Drive
San Jose, CA 95134
USA

Email: keyupate@cisco.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
Ericsson
1595 Spring Hill Road, Suite 500
Vienna, VA 22182
USA

Email: xliu@kuatrotech.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

INTERNET-DRAFT
Intended status: Proposed Standard

Donald Eastlake
Weiguo Hao
Lili Wang
Yizhou Li
Shunwan Zhuang
Huawei

Expires: October 9, 2019
April 10, 2019

Centralized EVPN DF Election
draft-hao-bess-evpn-centralized-df-04.txt

Abstract

This document proposes a centralized DF Designated Forwarder election mechanism to be used between an SDN (Software Defined Network) controller and each PE (Provider Edge) device in an EVPN network. Such a mechanism overcomes some issues with the current standalone DF election defined in RFC 7432. A new BGP capability and an additional DF Election Result Route Type are specified to support this centralized DF election mechanism.

Status of this Memo

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

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

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

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

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html. The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

Table of Contents

1. Introduction

[RFC7432] defines a standardized Designated Forwarder (DF) election mechanism in EVPN networks to appoint one Provider Edge (PE) device as the DF from a candidate list of PEs for each VLAN (or VLAN bundle) connecting to a multi-homed Customer Edge (CE) device or access network. The DF PE is responsible for sending broadcast, multicast and unknown unicast traffic (BUM) to the multi-homed CE device or network and non-DF PEs must drop such traffic. This DF based mechanism is used to prevent duplicated packet injection into the multi-homed access network via multiple PEs.

In [RFC7432] the DF is selected according to the VLAN modulus "service-carving" algorithm in order to perform load balancing for multi-destination traffic destined to a given segment. The algorithm can ensure each participating PE independently and unambiguously determines which one of the participating PEs is the DF; however, use of this algorithm has some drawbacks as follows [EVPN-HRW-DF]:

1. Uneven load balancing in some VLAN configuration cases when the Ethernet tags have a non-uniform distribution, for instance when the Ethernet tags in use are all even or all odd.

2. Unnecessary service disruption when PEs join or leave a redundancy group. In Figure 1 below, say v1, v2 and v3 are VLANs configured on ES2 with associated Ethernet tags of value 3, 4 and 5 respectively. So PE1, PE2 and PE3 are also the DFs for v1, v2 and v3 respectively. Now when PE3 goes down, PE2 will become the DF for v1 and v3 while PE1 will become the DF for v2, so needless churn of v1 and v2 occurs causing unnecessary service disruption in v1 and v2.

3. Lack of user control over DF election. In some cases, the user may want to flexibly control the load balancing based on VLAN number, bandwidth consumption, and other factors. The user should be allowed to use some specific DF re-election algorithm to avoid service disruption. The user also should be allowed to specify revertive and non-revertive mode for on-demand DF switchover in order to carry out some maintenance tasks.

This document specifies a centralized DF election method to overcome the issues aforementioned. A physically distributed but logically centralized controller is deployed to perform the DF election calculation for all multi-homed PEs. Each individual multi-homed PE in the redundancy group should disable its own DF election process and listen to the DF election result from the SDN controller. [RFC7432] DF election procedures are extended for the interaction between the SDN Controller and each PE.

2. Conventions used in this document

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

The following terms and acronyms are used:

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

DF: Designated Forwarder.

Ethernet Segment (ES): When a customer site (device or network) is connected to one or more PEs via a set of Ethernet links, then that set of links is referred to as an "Ethernet segment".

ESI: Ethernet Segment Identifier: A unique non-zero identifier that identifies an Ethernet segment.

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

EVPN: Ethernet Virtual Private Network [RFC7432].

PE: Provider Edge device.

NLRI: Network Layer Reachability Information.

SDN: Software Defined Networking.

VLAN: Virtual Local Area Network.

3. Solution Overview

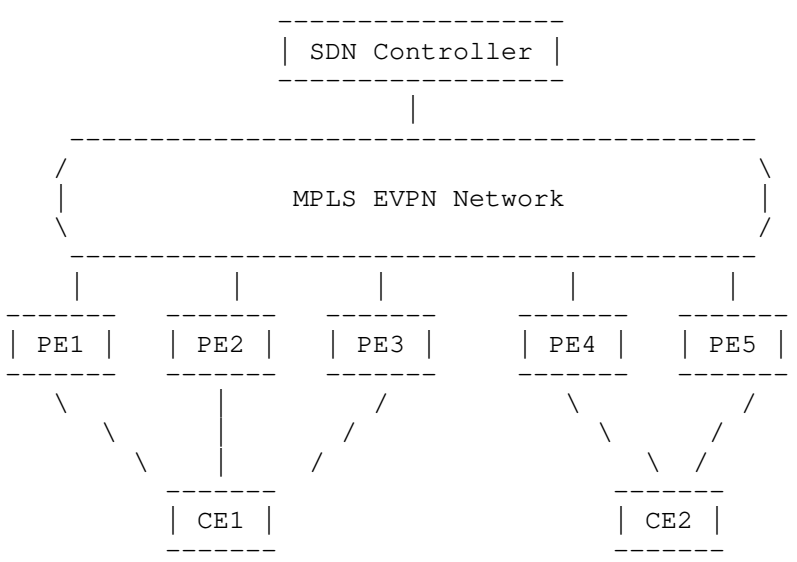


Figure 1. Centralized DF Election Scenario

In Figure 1, CE1 is multi-homed to PE1, PE2 and PE3, the ESI is 1. CE2 is multi-homed to PE4 and PE5, the ESI is 2. The SDN controller will be pre-provisioned with the entire network's ESI related configuration. This includes EVI, the Ethernet Tags on each ESI, redundancy mode of active-active or active-standby for each ESI, <ESI, Ethernet Tag> and EVI correspondence.

Before each PE and the SDN controller exchange BGP route information for DF election, the SDN controller and each PE MUST negotiate a new BGP centralized DF election capability and role when OPEN messages are first exchanged; each PE participating in multi-homing is the client for the DF election information while the SDN controller is the server. For these PEs the regular DF election process as per [RFC7432] will be disabled and each PE listens to the DF/Non-DF result from the SDN controller at the granularity of <ES,VLAN> or <ES, VLAN bundle>. For the DF election server, after it receives Ethernet Segment route from each PE, it will perform DF election calculation based on a local algorithm and will notify each EVPN PE of the election result through a new EVPN route type.

3.1 Centralized DF Election Capability

The centralized DF election capability is a new BGP capability [RFC5492] that can be used by a BGP speaker to indicate its ability

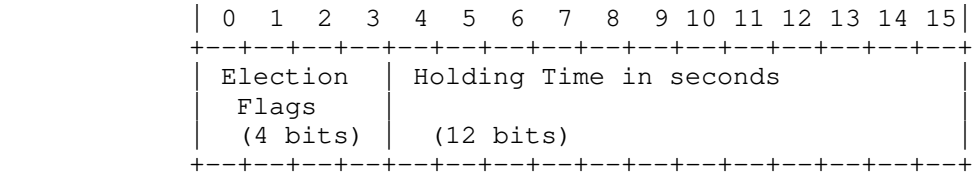
to support for the new DF election process.

This capability is defined as follows:

Capability code: TBD1

Capability length: 2 octets

Capability value: Consists of the "Election Flags" field and "Holding Time" field as follows:



The use and meaning of these fields are as follows:

Election Flags: This field contains bit flags related to restart as follows:

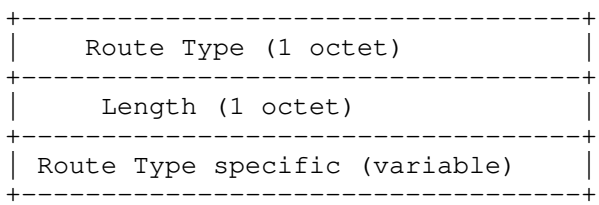
S: The most significant bit is the election Server bit. When set to 1, this bit indicates that the BGP speaker is the Server (Controller) that has the DF election calculation capability for all multi-homed PEs in the entire EVPN network. When set to 0 it indicates the BGP speaker is a Client which will await the DF election result from the Controller (Server).

Resv: Reserved bits that MUST be sent as zero and ignored on receipt.

Holding Time: This is the estimated maximum time in seconds it will take for the client to get DF election results from the controller after the BGP session is established. When no result for the DF election is received after the holding time, PEs will revert to the traditional EVPN DF election process as per [RFC7432].

4. DF Election Result Route Type

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



This document defines an additional Route Type used for the server (SDN Controller) to send DF election results to each client (PE). The Route Type is named the "DF Election Result Route Type".

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

4.1 DF Election Result Route Encoding

The route type specific information for a DF Election Result Route NLRI consists of the following fields:

Route Type specific information:

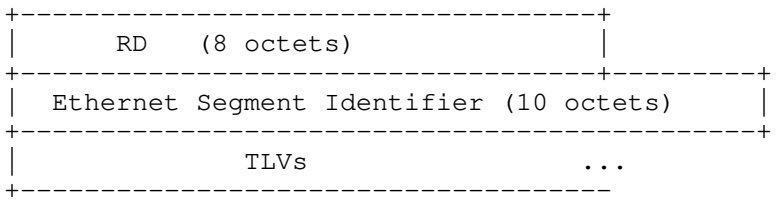


Figure 2: DF Election Result Router Type specific information

RD: The Route Distinguisher (RD) MUST be a Type 1 RD [RFC4364]. The value field comprises an IP address of the Controller (typically, the loopback address) followed by a number unique to the Controller.

ESI: Ethernet Segment Identifier: Is a non-zero 10-octet identifier for an Ethernet Segment.

TLVs: Information in the TLVs field is encoded in Type/Length/Value triplets. Multiple TLVs can be included. This document specifies type 1, the VLAN Bitmap type, whose structure is as follows:

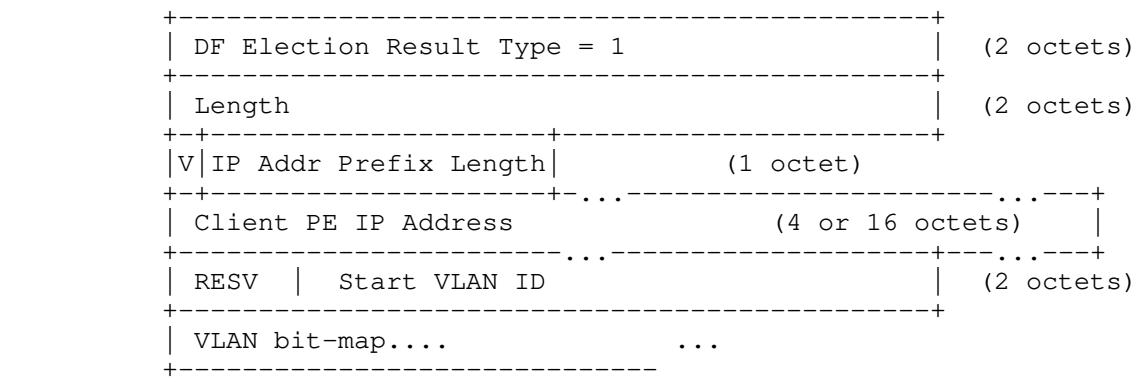


Figure 3. DF Election Result TLV Format

- DF Election Result Type (2 octets): Identifies the type of DF Election result as an unsigned integer in network byte order. This document defines type 1 as the "VLAN Bitmap" Type. TLVs withe unknown types are ignored and skipped upon receipt.

- Length (2 octets): The total number of octets of the value part of the TLV as an unsigned integer in network byte order.

The type and length are followed by the variable length value. This value, for the VLAN Bitmap type, consists of the following fields:

- V: A one bit field that indicates which version of IP the TLV uses. A value of 1 implies ipv6 while 0 implies ipv4.

- The IP Prefix Length can be set to a value between 0 and 32 (bits) for ipv4 and between 0 and 127 for ipv6. If IP Prefix Length is greater than 32 for ipv4, the TLV is corrupt and MUST be ignored.

- The Client PE IP Address will be a 32 or 128-bit field (ipv4 or ipv6 depending on the value of the V field) as PE's identification.

- RESV is a 4-bit reserved field that MUST be sent as zero and ignored on receipt.

- Start VLAN ID: The 12-bit VLAN ID that is represented by the high order bit of the first byte of the VLAN bit-map.

- VLAN bit-map: The highest order bit indicates the VLAN equal to the start VLAN ID, the next highest bit

indicates the VLAN equal to start VLAN ID + 1, continuing to the end of the VLAN bit-map field. A bit value of 1 indicates DF and a bit value of 0 indicates non-DF.

4.2 Centralized DF Election procedures

The controller has all ES related configuration information for the entire EVPN network. After the controller boots up, it can start a boot-timer to allow the establishment of BGP EVPN sessions with all multi-homed EVPN PEs. The controller also needs to receive all ES routes from those PEs before the boot-timer timeout. The controller will preserve all EVPN PE's ES routes.

Based on a local algorithm for each ES, after it has received the above data, it can start to perform the DF election calculation. The default algorithm is the VLAN modulus method defined in section 8.5 [RFC7432] relying on local VLAN configuration for each ES. A user defined algorithm should be allowed.

After the DF election calculation is finished on the controller, it will notify each multi-homed PE using the newly defined DF Election Result Route. The DF Election Result Route is per ES, i.e., the DF election results for all PEs connecting to the same ES are carried in one route. The controller that advertises the Ethernet Segment route MUST carry an ES-Import Route Target. The DF Election Result filtering procedure is the same as the Ethernet Segment route filtering defined in [RFC7432], i.e., the DF Election Result Route filtering MUST be imported only by the PEs that are Multi-homed to the same Ethernet segment. Each Multi-homed PE compares the Client PE IP Address with its local IP Address, if the two IP addresses are same, then it gets the corresponding start VLAN and VLAN Bitmap as the DF election results.

When the failure of a multi-homed PE is detected by the controller, the controller will initiate the DF re-election process. Because it's the controller making decisions as to which PE is DF or non-DF, the controller should ensure that the DF re-election does not cause unnecessary service disruption. In the example above, the controller should only redistribute the DF VLAN on PE3 to PE1 and PE2, the existing DF VLAN on PE1 and PE2 should remain unchanged to avoid service disruption.

When the access link fails on one multi-homed PE, the PE will advertise an Ethernet Segment Withdraw message to the controller, which will trigger the DF re-election on the controller. The re-election principle in this case is same as in the node failure case to minimize service disruption.

5. Security Considerations

Procedures and protocol extensions defined in this document do not affect the BGP security model. The communications between the SDN Controller and EVPN PEs should be protected to ensure security. BGP peerings are not automatic and require configuration, thus it is the responsibility of the network operator to ensure that they are trusted entities.

6. IANA Considerations

Three IANA actions are requested as below.

IANA is requested to assign a new BGP Capability Code in the Capability Code registry as follows:

Value	Description	Reference
TBD1	Centralized DF Election	[this document]

This document requested the assignment of value TBD2 in the "EVPN Route Types" registry created by [RFC7432] and modification of the registry to add the following:

Value	Description	Reference
TBD2	DF Election Result	[this document]

IANA is requested to create a registry for "DF Election Result Types" as follows:

Name: DF Election Result Types
Registration Procedure: First Come First Served
Reference: [this document]

Type	Description	Reference
0	(Reserved)	
1	VLAN Bitmap	[this document]
2-65534	unassigned	
65535	(reserved)	

Normative References

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

[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>.

[RFC5492] - Scudder, J. and R. Chandra, "Capabilities Advertisement with BGP-4", RFC 5492, DOI 10.17487/RFC5492, February 2009, <https://www.rfc-editor.org/info/rfc5492>.

[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>.

[RFC8174] - Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Informative References

[EVPN-HRW-DF] - Mohanty S. et al. "A new Designated Forwarder Election for the EVPN", draft-mohanty-bess-evpn-df-election-02, work-in-progress, October 19, 2015.

Acknowledgments

The authors wish to acknowledge the important contributions of Qiandeng Liang.

Authors' Addresses

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

Email: d3e3e3@gmail.com

Weiguo Hao
Huawei Technologies
101 Software Avenue,
Nanjing 210012, China

Email: haoweiguo@huawei.com

Lili Wang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095, China

Email: lily.wong@huawei.com

Yizhou Li
Huawei Technologies
101 Software Avenue,
Nanjing 210012, China

Email: liyizhou@huawei.com

Shunwan Zhuang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing, 100095 China

Email: zhuangshunwan@huawei.com

BESS WG
Internet-Draft
Intended status: Standards Track
Expires: September 18, 2016

Fangwei. Hu
Ran. Chen
Jie. Yao
ZTE Corporation

March 17, 2016

L2VPN Service YANG Model
draft-hu-bess-l2vpn-service-yang-00.txt

Abstract

This document defines a YANG data model that can be used to deliver a Layer 2 Provider Provisioned VPN service. These services include Virtual Private Wire Service (VPWS) and Virtual Private LAN service (VPLS). This model is intended to be instantiated at management system to deliver the L2VPN service, and is not a configuration model to be used directly on network elements.

This model provides an abstracted view of the Layer 2 VPN service configuration components. It will be up to a management system(orchestrator) to take this as an input and use specific configurations models to configure the different network elements to deliver the service. It is called as north bound L2VPN Service YANG data model. How configuration of network elements is out of scope of the document, and is defined in document[I-D.shah-bess-l2vpn-yang].

Table of Contents

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 [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 defines a YANG data model for Layer 2 Provider Provisioned VPN service configuration. These services include Virtual Private Wire Service (VPWS) and Virtual Private LAN service (VPLS).

2. Terminology

3. Typical Scenario

The idea of the Lay 2 VPN service model is to propose an abstracted interface to manage configuration of components of a Lay 2 VPN service. A typical usage is to use this model as an input for an orchestration layer who will be responsible to translate it to orchestrated configuration of network elements which will be part of the service.

Figure 1 is the typical scenario for the SDN based layer 2 VPN Service. The Layer 2 service YANG data model is used between the orchestration and controller 1 and controller2 and as the input for the orchestrator which is responsible to translate the application service to configure the network elements. The interfaces between controller and orchestration are called north bound interfaces, so the YANG data model defined in this document is also called north bound YANG service data model for Lay 2 VPN. The interfaces between controller and element networks are called south bound interfaces, and the YANG data model as the input of controller to configure the network elements is defined in document [I-D.shah-bess-l2vpn-yang].

There are two network element domains in the figure: domain A and domain C. The network elements in domain A are A11, A12, A21, A22, and A31 and A32. A31 and A32 are ASBRs, which are the edge routers connect to the other domains. Network elements C11, C12, C21, C22, C31, and C32 are in the domain C, and C31 and C32 are ASBRs in domain C.

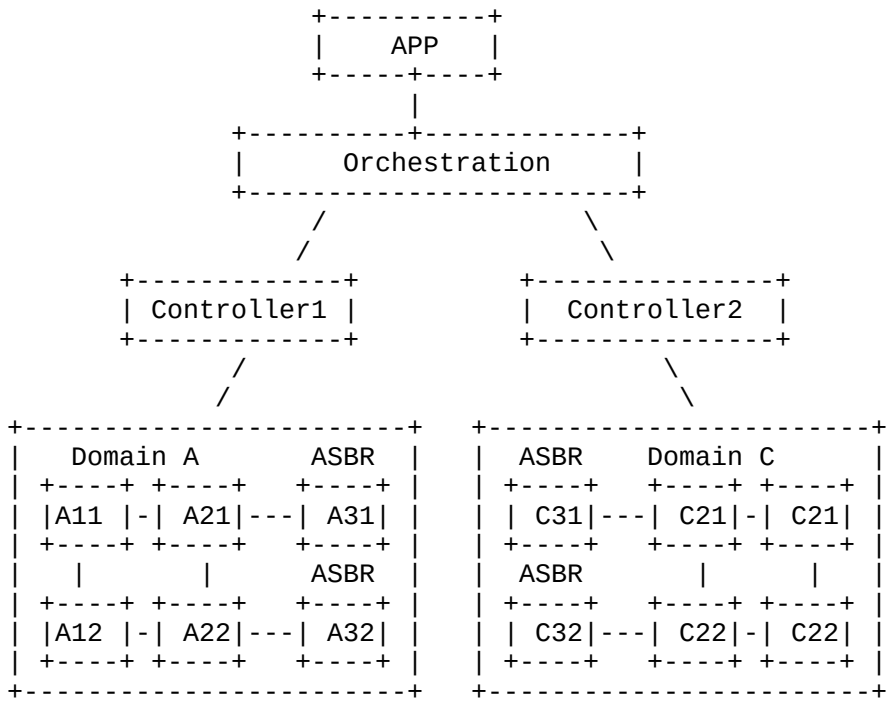


Figure 1 Typical Scenario for SDN based L2VPN Service

4. L2VPN Service Common

4.1. PW-Template

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 negotiation, cc, cv etc.

```
+--rw pw-template* [name]
   +--rw name              string
   +--rw mtu?              uint32
   +--rw cw-negotiation?   cw-negotiation-type
   +--rw cc                cc-type
   +--rw cv                cv-type
```

5. VPWS and VPLS Instance

5.1. PW List

If the L2VPN Service(VPWS service or VPLS service) needs to cross the AS domain, the PW container is configured as the following tree structure. If the L2VPN service is established in one domain, it is no need to configure PW container.

The asbr-id leaf is used to configure the edge ASBR of the domain. As the figure 1 shows, there are two ASBRs in the domain A, so it is required to indicate which exit ASBR it is when the controller establishes the cross PW.

If the L2VPN service is VPLS service, it is necessary to configure the split-horizon-group for the PW list.

```
+--rw pw* [name]
   +--rw name                string
   +--rw asbr-id?            string
   +--rw peer?               inet:ip-address
   +--rw vcid?               uint32
   +--rw type?               pw-type
   +--rw tunnel-policy?      string
   +--rw request-vlanid?     uint16
   +--rw vlan-tpid?          string
   +--ro cw-negotiation?     cw-negotiation-type
   +--ro cc?                 cc-type
   +--ro cv?                 cv-type
```

5.2. AC List

Each VPWS and VPLS instances define a list of AC that are participating members of the given service instance. The leaf ne-id is used to configure the ingress and egress ac of the end to end L2VPN service tunnel. The access type of ac could be port, dot1q, qinq, etc.

The QoS policy is defined in the AC list of the VPWS and VPLS instance. QoS dscp2exp mapping, cos2exp mapping, and QoS CAR parameters are defined.

```
+--rw qos-policy
   +--rw qos-dscp2exp      dscp2exp template
   +--rw qos-cos2exp       cos2exp template
   +--rw qos-if-trust
      +--rw trust-type?    enumeration
      +--rw direct-type?   enumeration
      +--rw ds-name?       string
   +--rw qos-if-cars
      +--rw direction    enumeration
      +--rw cir?         int32
      +--rw pir?         int32
      +--rw cbs?         int32
      +--rw pbs?         int32
```

5.3. Service Policy

Service policy is only used in a single AS domain. The PW type, redundancy tunnel for a Layer 2 VPN service is defined in the service policy container. If the L2VPN service is VPLS,there is no leaf node communicate-unit, it has split-horizon-group leaf node instead. The tree structure of service policy is designed as following:

```
+--rw service-policy* [id]
|  +--rw id                    uint8
|  +--rw communicate-unit?     int32
|  +--rw ne-id?                string
|  +--rw (primary)
|  |  +--:(primary-pw)
|  |  |  +--rw primary-pw* [name]
|  |  |     +--rw name     -> ../../../pw/name
|  |  +--:(primary-ac)
|  |     +--rw primary-ac?          -> ../../ac/name
|  +--rw (backup)?
|  |  +--:(backup-pw)
|  |  |  +--rw backup-pw* [name]
|  |  |     +--rw name           -> ../../../pw/name
|  |  |     +--rw precedence?    uint32
|  |  +--:(backup-ac)
|  |     +--rw backup-ac?           -> ../../ac/name
|  +--rw protect-type?         protect-type
|  +--rw receive-mode?         receive-mode
|  +--rw (revertive-type)?
|     +--:(never)
|     +--:(wtr)
|        +--rw revert-delay?       uint16
```

5.4. Tunnel Policy

The tunnel policy is used to configured the L2VPN underlay network's parameters. The signal type, tunnel mode and protect policy is defined in the container.

5.5. Tree Design for VPWS Instance YANG Data Model

The tree design for VPWS instance is as following:

```
+--rw vpws-instances
   +--rw vpws-instance* [name]
      +--rw name                              string
      +--rw description?                      string
      +--rw service-type?                     l2vpn-service-type
      +--rw signaling-type                    l2vpn-signaling-type
      +--rw pw* [name]
      |  +--rw name                           string
      |  +--rw asbr-id?                       string
      |  +--rw peer?                          inet:ip-address
      |  +--rw vcid?                          uint32
```

```
|  +--rw type?                          pw-type
|  +--rw tunnel-policy?                 string
|  +--rw request-vlanid?                uint16
|  +--rw vlan-tpid?                     string
|  +--ro cw-negotiation?                cw-negotiation-type
|  +--ro cc?                            cc-type
|  +--ro cv?                            cv-type
+--rw ac* [name]
|  +--rw name                           string
|  +--rw ac-nodeid?                     string
|  +--rw link-discovery-protocol-type?  link-discovery-protocol-type
|  +--rw (access-type)?
|  |  +--:(port)
|  |  +--:(dot1q)
|  |  |  +--rw dot1q-vlan-bitmap?       int32
|  |  +--:(qinq)
|  |     +--rw qinq-svlan-bitmap?       int32
|  |     +--rw qinq-cvlan-bitmap?       int32
|  +--rw (access-action)?
|  |  +--:(keep)
|  |  +--:(push)
|  |  |  +--rw push-vlan-id?            int32
|  |  +--:(pop)
|  |  +--:(swap)
|  |     +--rw swap-vlan-id?            int32
|  +--rw qos-policy
|  |  +--rw qos-dscp2exp?               dscp2exp
|  |  +--rw qos-cos2exp?                cos2exp
|  |  +--rw qos-if-cars
|  |     +--rw direction?               uint32
|  |     +--rw cir?                     uint32
|  |     +--rw pir?                     uint32
|  |     +--rw cbs?                     uint32
|  |     +--rw pbs?                     uint32
|  +--rw pipe-type?                     pipe-mode
+--rw service-policy* [id]
|  +--rw id                   uint8
|  +--rw communicate-unit?    int32
|  +--rw ne-id?               string
|  +--rw (primary)
|  |  +--:(primary-pw)
|  |  |  +--rw primary-pw* [name]
|  |  |     +--rw name    -> ../../../pw/name
|  |  +--:(primary-ac)
|  |     +--rw primary-ac?         -> ../../ac/name
|  +--rw (backup)?
|  |  +--:(backup-pw)
|  |  |  +--rw backup-pw* [name]
```

```
|  |  |     +--rw name          -> ../../../pw/name
|  |  |     +--rw precedence?   uint32
|  |  +--:(backup-ac)
|  |     +--rw backup-ac?          -> ../../ac/name
|  +--rw protect-type?       protect-type
|  +--rw receive-mode?       receive-mode
|  +--rw (revertive-type)?
|     +--:(never)
|     +--:(wtr)
|        +--rw revert-delay?       uint16
+--rw tunnel-policy
|  +--rw tunnel-signaling-type?         tunnel-signaling-type
|  +--rw tunnel-mode?                   tunnel-mode
|  +--rw protect-type?                  protect-type
|  +--rw receive-mode?                  receive-mode
|  +--rw (revertive-type)?
|     +--:(never)
|     +--:(wtr)
|        +--rw revert-delay?            uint16
+--rw master-multi-segment-nodes* [multi-segment-node]
|  +--rw multi-segment-node             string
|  +--rw designated-node?               boolean
+--rw slave-multi-segment-nodes* [multi-segment-node]
   +--rw multi-segment-node             string
  +--rw designated-node?                boolean
```

5.6. Tree Design for VPLS Instance YANG Data Model

The tree design for VPLS instance is as following:

```
+--rw vpls-instances
   +--rw vpls-instance* [name]
      +--rw name                                string
      +--rw description?                        string
      +--rw mac-withdraw?                       boolean
      +--rw bgp-parameters
      |  +--rw route-distinguisher?             string
      |  +--rw vpn-targets* [rt-value]
      |  |  +--rw rt-value                      string
      |  |  +--rw rt-type                       bgp-rt-type
      |  +--rw discovery
      |     +--rw vpn-id?                       string
      +--rw service-type?                       l2vpn-service-type
      +--rw signaling-type                      l2vpn-signaling-type
      +--rw pw* [name]
```

```
            |  +--rw name                              string
            |  +--rw asbr-id?                          string
            |  +--rw peer?                             inet:ip-address
            |  +--rw vcid?                             uint32
            |  +--rw type?                             pw-type
            |  +--rw tunnel-policy?                    string
            |  +--rw request-vlanid?                   uint16
            |  +--rw vlan-tpid?                        string
            |  +--ro cw-negotiation?                   cw-negotiation-type
            |  +--ro cc?                               cc-type
            |  +--ro cv?                               cv-type
            |  +--rw hub-spoken?                       hub-spoken
            +--rw ac* [name]
            |  +--rw name                              string
            |  +--rw ac-nodeid?                        string
            |  +--rw link-discovery-protocol-type?   link-discovery-protocol-typ
e
            |  +--rw (access-type)?
            |  |  +--:(port)
            |  |  +--:(dot1q)
            |  |  |  +--rw dot1q-vlan-bitmap?         int32
            |  |  +--:(qinq)
            |  |     +--rw qinq-svlan-bitmap?         int32
            |  |     +--rw qinq-cvlan-bitmap?         int32
            |  +--rw (access-action)?
            |  |  +--:(keep)
            |  |  +--:(push)
            |  |  |  +--rw push-vlan-id?              int32
            |  |  +--:(pop)
            |  |  +--:(swap)
            |  |     +--rw swap-vlan-id?              int32
            |  +--rw qos-policy
            |  |  +--rw qos-dscp2exp?                 dscp2exp
            |  |  +--rw qos-cos2exp?                  cos2exp
            |  |  +--rw qos-if-cars
            |  |     +--rw direction?                 uint32
            |  |     +--rw cir?                       uint32
            |  |     +--rw pir?                       uint32
            |  |     +--rw cbs?                       uint32
            |  |     +--rw pbs?                       uint32
            |  +--rw split-horizon-group?             string
            +--rw service-policy* [id]
            |  +--rw id                                uint8
            |  +--rw split-horizon-group?              string
            |  +--rw pw-type?                          pw-type
            |  +--rw ne-id?                            string
            |  +--rw (primary)
            |  |  +--:(primary-pw)
            |  |  |  +--rw primary-pw* [name]
```

```
            |  |  |     +--rw name    -> ../../../pw/name
            |  |  +--:(primary-ac)
            |  |     +--rw primary-ac?            -> ../../ac/name
            |  +--rw (backup)?
            |  |  +--:(backup-pw)
            |  |  |  +--rw backup-pw* [name]
            |  |  |     +--rw name          -> ../../../pw/name
            |  |  |     +--rw precedence?             uint32
            |  |  +--:(backup-ac)
            |  |     +--rw backup-ac?             -> ../../ac/name
            |  +--rw protect-type?                    protect-type
            |  +--rw receive-mode?                    receive-mode
            |  +--rw (revertive-type)?
            |     +--:(never)
            |     +--:(wtr)
            |        +--rw revert-delay?              uint16
            +--rw tunnel-policy
            |  +--rw tunnel-signaling-type?           tunnel-signaling-type
            |  +--rw tunnel-mode?                     tunnel-mode
            |  +--rw protect-type?                    protect-type
            |  +--rw receive-mode?                    receive-mode
            |  +--rw (revertive-type)?
            |     +--:(never)
            |     +--:(wtr)
            |        +--rw revert-delay?              uint16
            +--rw master-multi-segment-nodes* [multi-segment-node]
            |  +--rw multi-segment-node               string
            |  +--rw designated-node?                 boolean
            +--rw slave-multi-segment-nodes* [multi-segment-node]
               +--rw multi-segment-node               string
               +--rw designated-node?                 boolean
```

6. L2VPN Service YANG Data Model

```
 <CODE BEGINS> file "ietf-mpls-l2vpn-svc@2016-03-17.yang"
module ietf-mpls-l2vpn-svc {
    namespace "urn:ietf:params:xml:ns:yang:ietf-mpls-l2vpn-svc";
    prefix "mpls-l2vpn-svc";

    import ietf-inet-types
    {
        prefix inet;
    }

    organization "ietf";
```

```
    contact "ietf";
    description "mpls-l2vpn-svc";

    revision "2016-03-17" {
        description "Initial revision of mpls-l2vpn-service model.";
        reference "draft-hu-bess-l2vpn-service-yang-00.txt";
    }

    /* identities */

    identity link-discovery-protocol {
        description "Base identiy from which identities describing link discover
y 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 represens BPDU";
    }

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

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

    /* typedefs */
    typedef l2vpn-service-type {
        type enumeration {
            enum ethernet {
                description "Ethernet service";
            }
            enum ATM {
                description "Asynchronous Transfer Mode";
            }
```

```
            enum FR {
                description "Frame-Relay";
            }
            enum TDM {
                description "Time Division Multiplexing";
            }
        }
        description "L2VPN service type";
    }

    typedef l2vpn-signaling-type {
        type enumeration {
            enum static {
                description "Static configuration of labels (no signaling)";
            }
            enum ldp {
                description "Label Distribution Protocol (LDP) signaling";
            }
            enum bgp {
                description "Border Gateway Protocol (BGP) signaling";
            }
            enum mixed {
                description "Mixed";
            }
        }
        description "L2VPN signaling type";
    }

    typedef tunnel-signaling-type {
        type enumeration {
             enum static {
               value 0 ;
               description "static" ;
             }
             enum RSVP-TE {
               value 1 ;
               description "RSVP-TE" ;
             }
            enum LDP {
                value 2 ;
                description "LDP" ;
            }
          }
        description "tunnel signaling type." ;
    }

    typedef bgp-rt-type {
        type enumeration {
```

```
            enum import {
                description "For import";
            }
            enum export {
                description "For export";
            }
            enum both {
                description "For both import and export";
            }
        }
        description "BGP route-target type. Import from BGP YANG";
    }

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

    typedef cc-type {
        type enumeration {
            enum pw-ach {
                description "PWE3 Control Word with 0001b as first nibble (PW-AC
H, see [RFC4385])";
            }
            enum alert-label {
                description "MPLS Router Alert Label";
            }
            enum ttl {
                description "MPLS PW Label with TTL == 1";
            }
        }
        description "The defined values for CC(Control Channel) Types for MPLS P
Ws.";
    }

    typedef cv-type {
```

```
        type enumeration {
            enum ICMP-ping {
                description "ICMP-ping.";
            }
            enum LSP-ping {
                description "LSP-ping";
            }
            enum BFD-basic-ip {
                description "BFD basic ip";
            }
            enum BFD-basic-raw {
                description "BFD basic raw ";
            }
            enum BFD-signalling-ip {
                description "BFD signalling ip";
            }
            enum BFD-signalling-raw {
                description "BFD signalling raw";
            }
        }
        description "The defined values for CV(Connectivity Verification) Types
for MPLS PWs";
    }

    typedef pipe-mode{
        type enumeration {
            enum "pipe" {
                value 0;
                description "regular pipe mode";
            }
            enum "short-pipe" {
                value 1;
                description "short pipe mode";
            }
            enum "uniform" {
                value 2;
                description "uniform pipe mode";
            }
        }
        description " ";
    }

    typedef pw-type {
        type enumeration {
            enum unknown {
                value 0 ;
                description "The PW type is unknown";
            }
            enum dlciOld {
```

```
                value 1 ;
                description "The PW type is dlciOld";
            }
            enum atmSdu {
                value 2 ;
                description "The PW type is atmSdu";
            }
            enum atmCell {
                value 3 ;
                description "The PW type is atmCell";
            }
            enum vlan {
                value 4 ;
                description "The PW type is vlan";
            }
            enum ethernet {
                value 5 ;
                description "The PW type is ethernet";
            }
            enum hdlc {
                value 6 ;
                description "The PW type is hdlc";
            }
            enum ppp {
                value 7 ;
                description "The PW type is ppp";
            }
            enum sdhCESoM {
                value 8 ;
                description "The PW type is sdhCESoM";
            }
            enum atmVCCn {
                value 9 ;
                description "The PW type is atmVCCn";
            }
            enum atmVPCn {
                value 10 ;
                description "The PW type is atmVPCn";
            }
            enum ipL2 {
                value 11 ;
                description "The PW type is ipL2";
            }
            enum atmVCC1 {
                value 12 ;
                description "The PW type is atmVCC1";
            }
            enum atmVPC1 {
```

```
                value 13 ;
                description "The PW type is atmVPC1";
            }
            enum atmPDU {
                value 14 ;
                description "The PW type is atmPDU";
            }
            enum frPort {
                value 15 ;
                description "The PW type is frPort";
            }
            enum sdhCEoP {
                value 16 ;
                description "The PW type is sdhCEoP";
            }
            enum saTopE1 {
                value 17 ;
                description "The PW type is saTopE1";
            }
            enum saTopT1 {
                value 18 ;
                description "The PW type is saTopT1";
            }
            enum saTopE3 {
                value 19 ;
                description "The PW type is saTopE3";
            }
            enum saTopT3 {
                value 20 ;
                description "The PW type is saTopT3";
            }
            enum ceSoPSNB {
                value 21 ;
                description "The PW type is ceSoPSNB";
            }
            enum tdmAAL1 {
                value 22 ;
                description "The PW type is tdmAAL1";
            }
            enum ceSoPSNC {
                value 23 ;
                description "The PW type is ceSoPSNC";
            }
            enum tdmAAL2 {
                value 24 ;
                description "The PW type is tdmAAL2";
            }
            enum dlciNew {
```

```
                value 25 ;
                description "The PW type is dlciNew";
            }
         }
         description "The PW type of the PW.";
    }

    typedef dscp2exp {
       type string;
       description "define the dscp to exp mapping";
       }

    typedef cos2exp {
       type string;
       description "define the cos to exp mapping";
      }

    typedef protect-type {
       type enumeration {
            enum unprotected {
                value 0 ;
                description "unprotected." ;
                }
            enum 1to1 {
                value 1 ;
                description "protect type is 1:1";
            }
            enum 1plus1 {
                value 2 ;
                description "protect type is 1+1";
            }
            enum dni-pw {
                value 3 ;
                description "protect type is dni-pw";
            }
            enum dni-ac {
                value 4 ;
                description "protect type is dni-ac";
            }

         }
         description "define the protect type";
    }

    typedef receive-mode {
       type enumeration {
            enum selective {
                value 0 ;
```

```
                description "receive mode is selective";
            }
            enum both {
                value 1 ;
                description "receive mode is both";
            }
        }
        description "define the receive mode";
    }

    typedef tunnel-mode {
      type enumeration {
           enum static {
              value 2 ;
              description "static tunnel" ;
             }
           enum RSVP-TE {
              value 0 ;
               description "RSVP-TE" ;
           }
           enum LDP {
               value 1 ;
           description "LDP" ;
           }


        }
        description "define the tunnel mode";
    }


    typedef hub-spoken {
       type enumeration {
           enum hub {
              value 0 ;
              description "the hub role in the network" ;
             }
           enum spoken {
              value 1 ;
               description "the spokend role in the network" ;
           }
        }
        description "define the hub spoken type";
    }
    grouping pw-template{
        description "pw-template";
        leaf name {
            type string;
```

```
            description "name";
        }

        leaf cw-negotiation {
            type cw-negotiation-type;
            default "preferred";
            description "control-word negotiation preference";
        }

        leaf cc {
            type cc-type;
            description "Control Channel Types";
        }

        leaf cv {
            type cv-type;
            description "Connectivity Verification Types";
        }
    }

    grouping qos-policy-grp {
       container qos-policy {
         leaf qos-dscp2exp {
                type dscp2exp;
                description "the dscp to exp mapping template";
               }
         leaf  qos-cos2exp {
                type cos2exp ;
                description "the cos to exp mapping template";
               }

         container qos-if-cars {
              leaf direction {
                type uint32;
                description "the direction of qos";
               }

              leaf cir {
                type uint32;
                description "the cir parameter for the car";
               }

              leaf pir {
                type uint32;
                description "the pir parameter for the car";
               }

              leaf cbs {
```

```
                type uint32;
                description "the cbs parameter for the car";
               }

              leaf pbs {
                 type uint32;
                 description "the pbs parameter for the car";
               }
            description "configuration the qos interface car parameters.";
           }
         description "qos policy container.";
       }
      description "qos policy template.";
    }

    grouping pseudowire{
        description "pseudowire";
        leaf name {
            type string;
            description "pseudowire name";
        }

        leaf asbr-id {
            type string;
            description "asbr name for the cross domain lsp";
        }
        leaf peer {
            type inet:ip-address;
            description "pw peer address, use IPv4";
        }

        leaf vcid {
            type uint32;
            description "pseudo-wire vcid";
        }

        leaf type {
            type pw-type;
            description "pseudo-wire type";
        }

        leaf tunnel-policy {
            type string;
            description "Used to override the tunnel policy name specified in th
e pseduowire template";
        }

        leaf request-vlanid {
            type uint16;
```

```
            description "request vlanid";
        }

        leaf vlan-tpid {
            type string;
            description "vlan tpid";
        }

        leaf cw-negotiation {
            type cw-negotiation-type;
            default "preferred";
            config false;
            description "control-word negotiation preference";
        }

        leaf cc {
            type cc-type;
            config false;
            description "Control Channel Types";
        }

        leaf cv {
            type cv-type;
            config false;
            description "Connectivity Verification Types";
        }
    }

    grouping attachment-circuit{
        description "attachment circuit";
        leaf name {
            type string;
            description "name";
        }

        leaf ac-nodeid{
            type string;
            description "The nodeid of the AC.";
        }

        leaf link-discovery-protocol-type{
            type link-discovery-protocol-type;
            description "link discovery protocol";
        }

        choice access-type{
            description "access-type";
            case port{
```

```
                description "port.";
            }

            case dot1q {
                description "Dot1Q";
                leaf dot1q-vlan-bitmap {
                    type int32 {
                        range "1..4094";
                    }
                    description "Dot1Q Vlan Bitmap." ;
                }
            }

            case qinq {
                description "QinQ";
                leaf qinq-svlan-bitmap {
                    type int32 {
                        range "1..4094";
                    }
                    description "QinQ svlan Bitmap." ;
                }

                leaf qinq-cvlan-bitmap {
                    type int32 {
                        range "1..4094";
                    }
                    description "QinQ cvlan Bitmap." ;
                }
            }
        }

        choice access-action {
            description "access type." ;
            case keep {
                description "keep." ;
            }
            case push {
                description "push." ;
                leaf push-vlan-id {
                    type int32 {
                        range "1..4094";
                    }
                    description "action vlan id." ;
                }
            }
            case pop {
                description "pop." ;
            }
```

```
            case swap {
                description "swap." ;
                leaf swap-vlan-id {
                    type int32 {
                        range "1..4094";
                    }
                    description "action vlan id." ;
                }
            }
        }
        uses qos-policy-grp;
    }

    grouping protect-policy-grp {
        leaf protect-type {
               type protect-type;
               description "protection-type";
             }

        leaf receive-mode {
               type receive-mode;
               description "receive-mode";
            }

        choice revertive-type {
            description "revertive-type";
             case never {
              description "never type";
              }
             case wtr {
               leaf revert-delay {
                type uint16;
                description "the revertive type is wtr";
                }
             }
        }

     description "define the group of protect-policy.";
    }

    grouping redundancy-grp {
         leaf ne-id {
            type string;
            description "the name of ne";
           }
         choice primary {
           mandatory true;
           description "primary options";
```

```
            case primary-pw {
              list primary-pw {
                key "name";
                leaf name {
                  type leafref {
                    path "../../../pw/name";
                  }
                  description "Reference a pseudowire";
                }
                description "A list of primary pseudowires";
              }
              description "primary-pw";
            }

            case primary-ac {
              leaf primary-ac {
                type leafref {
                  path "../../ac/name";
                }
                description "Reference an attachment circuit";
              }
              description "primary-ac";
            }
          }

          choice backup {
            description "backup options";
            case backup-pw {
              list backup-pw {
                key "name";
                leaf name {
                  type leafref {
                    path "../../../pw/name";
                  }
                  description "Reference an attachment circuit";
                }
                leaf precedence {
                  type uint32;
                  description "precedence of the pseudowire";
                }
                description "list of backup pseudowires";
              }
            }
            case backup-ac {
              leaf backup-ac {
                type leafref {
                  path "../../ac/name";
                }
```

```
              description "Reference an attachment circuit";
            }
            description "backup-ac";
          }
        }
        uses protect-policy-grp;
       description "define the redundancy group";
      }

    grouping vpws-service-policy-grp {
        list service-policy {
         key "id";
         leaf id {
           type uint8;
           description "the id of service policy";
         }

         uses redundancy-grp;
         description "the service policy list";

         leaf communicate-unit {
             type int32;
             description "ICCP id for dni-pw";
           }
        }
        description "service policy list";
      }

    grouping vpls-service-policy-grp {
        list service-policy {
         key "id";
         leaf id {
           type uint8;
           description "the id of service policy";
         }

         leaf split-horizon-group {
            type string;
            description "split horizon group for the vpls instance";
          }

         leaf pw-type {
           type pw-type;
           description "the pseudowires type";
         }

         uses redundancy-grp;
         description "the service policy list";
```

```
        }
        description "service policy";
    }

    grouping tunnel-policy-grp {
        container tunnel-policy {
           leaf tunnel-signaling-type {
             type tunnel-signaling-type;
             description "signaling-type";
            }
          leaf tunnel-mode {
             type tunnel-mode;
             description "tunnel-mode";
          }
         uses protect-policy-grp;
         description "tunnel policy";
        }
        description "tunnel policy group";
       }

    grouping multi-segment{
        leaf multi-segment-node{
            type string;
            description "The multi-segment nodeid of the PW.";
        }

        leaf designated-node{
            type boolean;
            description "The multi-segment nodeid of the PW must select.";
        }
        description "multi-segment group.";
    }

    container l2vpn {
        description "l2vpn";
        container common {
            description "common l2pn attributes";
            container pw-templates {
                description "pw-templates";
                list pw-template {
                    key "name";
                    uses pw-template;
                    description "pw-templates";
                }
            }
        }

        container vpws-instances {
```

```
            description "configure vpws-instances";
            list vpws-instance {
               key "name";
                leaf name {
                  type string;
                  description "the name of a vpws instance";
                }

               leaf description {
                 type string;
                 description "Description of the vpws instance";
                }

               leaf service-type {
                 type l2vpn-service-type;
                 default ethernet;
                 description "vpws service type";
                }

                leaf signaling-type {
                 type l2vpn-signaling-type;
                 mandatory true;
                 description "vpws signaling type";
                }

                list pw{
                 key "name";
                 uses pseudowire;
                 description "pseudowires";
                }

                list ac{
                 key "name";
                 uses attachment-circuit;
                 leaf pipe-type{
                   type pipe-mode;
                   description "pipe mode";
                }
                 description "attachment-circuit";

                }

                uses vpws-service-policy-grp;
                uses tunnel-policy-grp ;

                list master-multi-segment-nodes{
                  key "multi-segment-node";
                  uses multi-segment;
```

```
                description "The master multi-segment nodeid of the PW.";
              }

              list slave-multi-segment-nodes{
                 key "multi-segment-node";
                 uses multi-segment;
                 description "The slave multi-segment nodeid of the PW.";
              }

              description "vpws service instance list";

          }
        }

        container vpls-instances {
            description "configure vpls-instances";
            list vpls-instance {
               key "name";
                leaf name {
                  type string;
                  description "the name of a vpls instance";
                }

               leaf description {
                 type string;
                 description "Description of the vpls instance";
                }

                leaf mac-withdraw {
                  type boolean;
                  description "Withdraw MAC";
                }

                container bgp-parameters {
                   leaf route-distinguisher {
                     type string;
                     description "BGP RD";
                   }

                   list vpn-targets {
                     key rt-value;
                     description "Route Targets";

                     leaf rt-value {
                        type string;
                        description "Route-Target value";
                      }
```

```
                    leaf rt-type {
                       type bgp-rt-type;
                       mandatory true;
                       description "Type of RT";
                    }
                  }

                  container discovery {
                     description "BGP parameters for discovery";
                     leaf vpn-id {
                       type string;
                       description "VPN ID";
                     }
                  }
                 description "Parameters for BGP";
                }

               leaf service-type {
                 type l2vpn-service-type;
                 default ethernet;
                 description "vpls service type";
                }

                leaf signaling-type {
                 type l2vpn-signaling-type;
                 mandatory true;
                 description "vpls signaling type";
                }

                list pw{
                 key "name";
                 uses pseudowire;
                 leaf hub-spoken {
                   type hub-spoken;
                   description "hub-spoken role";
                  }
                 description "pseudowires";
                }

                list ac{
                 key "name";
                 uses attachment-circuit;
                 leaf split-horizon-group {
                   type string;
                   description "split horizon group for the vpls instance";
                  }
                 description "attachment-circuit";
```

```
                }

                uses vpls-service-policy-grp;
                uses tunnel-policy-grp ;

                list master-multi-segment-nodes{
                  key "multi-segment-node";
                  uses multi-segment;
                  description "The master multi-segment nodeid of the PW.";
                }

                list slave-multi-segment-nodes{
                   key "multi-segment-node";
                   uses multi-segment;
                   description "The slave multi-segment nodeid of the PW.";
                }
                description "vpls service instance list";
            }
        }
    }
}
 <CODE ENDS>
```

7. Security Considerations

8. Acknowledgements

9. IANA Considerations

This document requires no IANA Actions. Please remove this section before RFC publication.

10. Normative References

[I-D.shah-bess-l2vpn-yang]
Shah, H., Brissette, P., Rahman, R., Raza, K., Li, Z., Zhuang, S., Wang, H., Chen, I., Ahmed, S., Bocci, M., Hardwick, J., Esale, S., Tiruveedhula, K., Singh, T., Hussain, I., Wen, B., Walker, J., Delregno, N., Jalil, L., and M. Joecylyn, "YANG Data Model for MPLS-based L2VPN", draft-shah-bess-l2vpn-yang-01 (work in progress), March 2016.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://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, <http://www.rfc-editor.org/info/rfc6241>.

Authors' Addresses

Fangwei Hu
ZTE Corporation
No.889 Bibo Rd
Shanghai 201203
China

Phone: +86 21 68896273
Email: hu.fangwei@zte.com.cn

Ran Chen
ZTE Corporation
No.50 Software Avenue,Yuhuatai District
Nanjing, Jiangsu Province 210012
China

Phone: +86 025 88014636
Email: chen.ran@zte.com.cn

Jie Yao
ZTE Corporation
Zijinghua Rd. Yuhuatai District
Nanjing, Jiangsu Province 210012
China

Email: yao.jie@zte.com.cn

BGP Enabled Services
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2016

Z. Li
China Mobile

March 8, 2016

Connecting IPv4 Islands over IPv6 MPLS Using IPv4 Provider Edge Routers (4PE)
draft-li-bess-4pe-01

Abstract

This document explains how to interconnect IPv4 islands over a Multiprotocol Label Switching (MPLS)-enabled IPv6-only core. This approach relies on IPv4 Provider Edge routers (4PE), which are Dual Stacks in order to connect to IPv4 islands and to the MPLS core. The 4PE routers exchange the IPv4 reachability information transparently over the core using the Multiprotocol Border Gateway Protocol (MP-BGP). MP-BGP is extended to do this. A new Subsequence Address Family Identifier (SAFI) with corresponding new format Network Layer Reachability Information (NLRI), is introduced. The BGP Next Hop field is used to convey the IPv4 address of the 4PE router, a field is added in Network Layer Reachability Information (NLRI) to convey the IPv6 address of the 4PE router, so that dynamically established IPv6-signaled MPLS Label Switched Paths (LSPs) can be used without explicit tunnel configuration.

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/.

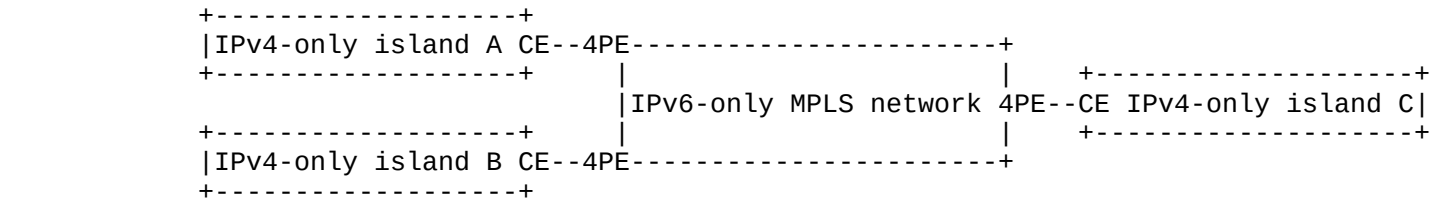
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 9, 2016.

Table of Contents

1. Introduction

After IPv6 [RFC2460] is widely deployed, IPv6-only networks and nodes will become dominate. How to provide the connectivity for the remaining IPv4-only islands through the IPv6-only MPLS network will become a problem, as depicted in the following figure.



IPv6 Provider Edge Routers (6PE), as specified in [RFC4798], are used to connect IPv6 islands over IPv4 MPLS network. However, [RFC7439]

pointed out that there is no solution to address the above problem. So, in this document, 4PE is proposed to meet this gap.

2. Protocol Overview

Each IPv4 island is connected to at least one Provider Edge router that is located on the border of the IPv6-only MPLS network. We call such a router a IPv4 Provider Edge router (4PE). The 4PE router MUST be IPv4 and IPv6 dual stack. At least one IPv4 address MUST be configured for the 4PE to connect the IPv4-only island. And at least one IPv6 address on the IPv6-ony MPLS network side MUST be configured. The configured IPv6 address needs to be routable in the IPv6-only network, and there needs to be a label bound via an IPv6 label distribution protocol to this IPv6 route. The configured IPv4 address MUST be unique among the IPv4 islands.

As a result of this, every considered 4PE router knows which MPLS label (we call it forwarding label) to use to send packets to any other 4PE router. Note that [RFC5036] updated by [RFC7552] fulfills these requirements.

We call the 4PE router receiving IPv4 packets from an IPv4 island an ingress 4PE router (relative to these IPv4 packets). We call a 4PE router forwarding IPv4 packets to an IPv4 island an egress 4PE router (relative to these IPv4 packets).

Interconnecting IPv4 islands over an IPv6 MPLS network mainly includes the following two steps:

1. Exchange IPv4 reachability information among 4PE routers with MP-BGP [RFC4760]

   The IPv4 routes MUST not be injected in the IPv6-only MPLS network.

   The 4PE routers MUST exchange the IPv4 routes over MP-BGP sessions running over IPv6. To do so, a new Subsequence Address Family Identifier (SAFI) with corresponding new format Network Layer Reachability Information (NLRI), is introduced in this document. We call this new SAFI 4PE SAFI, the detail of which is illustrated in section 3. The MP-BGP Address Family Identifier (AFI) used MUST be IPv4 (value 1). The MP-BGP Network Address of Next Hop MUST be the 4PE IPv4 address from which the 4PE receives the IPv4 routes of the IPv4 island. The IPv6 address of the 4PE, the IPv4 routes of the IPv4 island, and the corresponding MPLS label for the IPv4 routes (we call it 4PE label) MUST be encoded in the NLRI.

The reason to allocate MPLS label for the IPv4 route is to reduce the requirements for the IPv6-only MPLS network, as explained in [RFC4798]. Here it is. While this approach could theoretically operate in some situations using a single level of labels, there are significant advantages in using a second level of labels that are bound to IPv4 prefixes via MP-BGP advertisements.

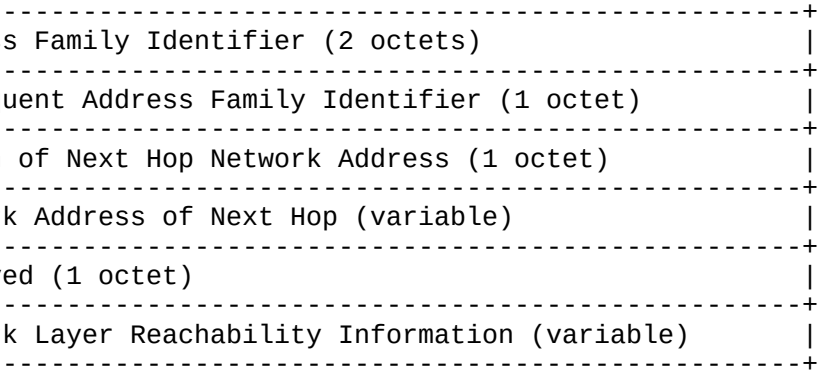
For instance, the use of a second level label allows Penultimate Hop Popping (PHP) on the IPv4 Label Switch Router (LSR) upstream of the egress 4PE router, without any IPv4 capabilities on the penultimate router. This is because it still transmits MPLS packets even after the PHP (instead of having to transmit IPv4 packets and encapsulate them appropriately).

2. Transport IPv4 packets from the ingress 4PE router to the egress 4PE router over the IPv6-signaled LSPs

The ingress 4PE router MUST forward IPv4 data over the IPv6-signaled LSP towards the egress 4PE router identified by the IPv6 address advertised in the new introduced NLRI for the corresponding IPv4 route.

3. 4PE SAFI

As depicted in the following figure, 4PE SAFI is proposed to carry the IPv4 routes for the IPv4-only island, the MPLS label for the IPv4 routes, the IPv4 and IPv6 IP addresses of the 4PE router .



The use and meaning of these fields are as follows:

Address Family Identifier (AFI):

This field MUST be 1, to indicate IPv4 routes are carried in the NLRI. This is in accordance with [RFC4760]

Subsequent Address Family Identifier (SAFI):

This field is used to indicate the new introduced 4PE SAFI. The value for this field is be assigned by IANA.

Length of Next Hop Network Address:

This field MUST be 4, to indicate an IPv4 address is encoded in the "Network Address of Next Hop" field.

Network Address of Next Hop:

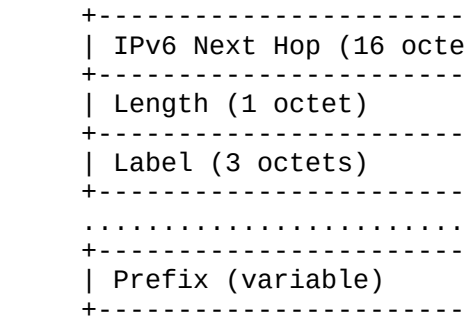
This field MUST be one of the 4PE IPv4 address from which the 4PE receives the IPv4 routes of the IPv4 island.

Reserved:

A 1 octet field that MUST be set to 0, and SHOULD be ignored upon receipt.

Network Layer Reachability Information (NLRI):

The format and meaning of this field is specified in the following.



The "IPv6 Next Hop" field MUST be the IPv6 address of the 4PE, through which the corresponding 4PE can be reached in the IPv6-only MPLS network.

The "Label" field MUST be the MPLS label allocated by the 4PE for the IPv4 routes carried in the Prefix field. This label is called 4PE label.

The "Prefix" field MUST be used to carry the IPv4 routes for the IPv4-only island.

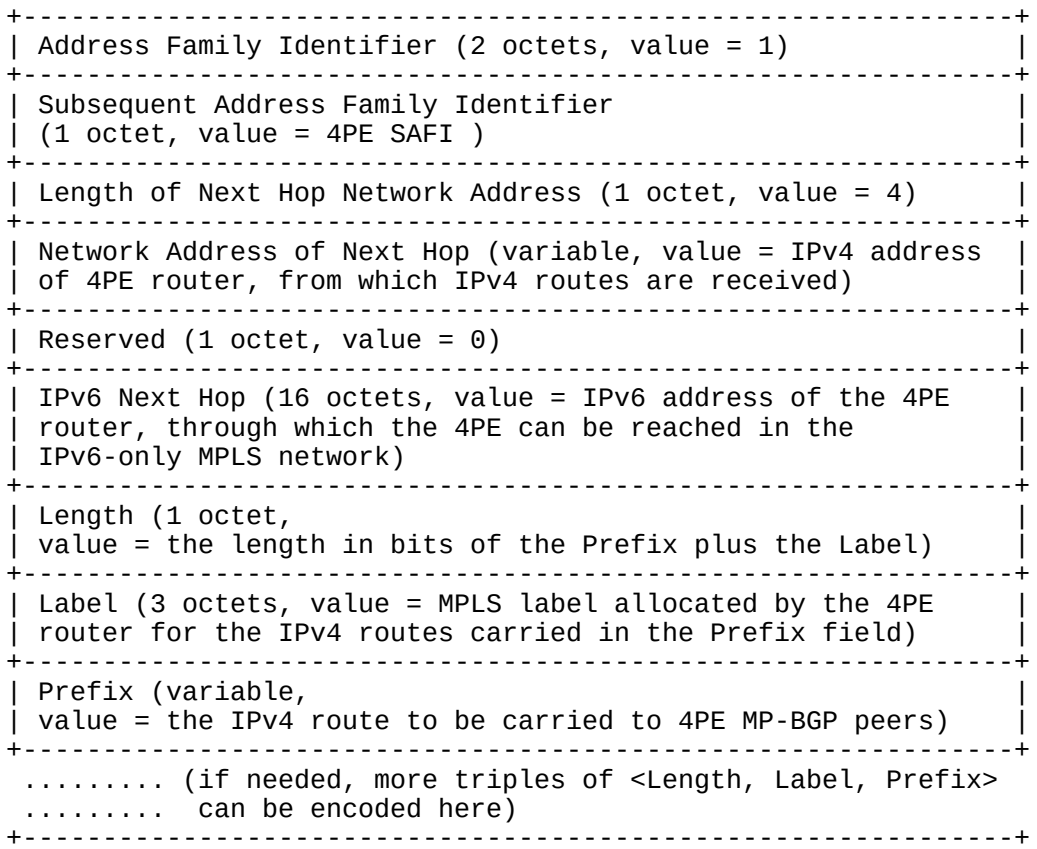
The "Length" field indicates the length in bits of the Prefix plus the Label.

One or more triples of the form <length, label, prefix > can be encoded in this NLRI. The use and meaning of the fields in this NLRI, except IPv6 Next Hop, are in accordance with [RFC3107].

4. Exchange IPv4 reachability information among 4PE routers

4PE routers MUST encode the IPv4 routes for the IPv4-only island in the UPDATE message of [RFC4271]. MP_REACH_NLRI (Type Code 14) and MP_UNREACH_NLRI (Type Code 15) defined in [RFC4760] MUST be used for route advertisement and withdraw respectively. 4PE SAFI defined in this document MUST be used in MP_REACH_NLRI or MP_UNREACH_NLRI to complete the exchange of IPv4 routes.

For advertisement, the 4PE router sets the fields of MP_REACH_NLRI as following, and then send the UPDATE message to its 4PE peers (or Route Reflectors(RR) in the network where RRs are deployed) over the MP-BGP session established on IPv6.

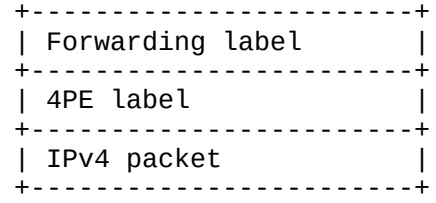


When receiving the UPDATE message, the 4PE MP-BGP peer treats the message as per [RFC4760] and [RFC3107]. Since the 4PE router can learn IPv4 routes from other 4PE routers through 4PE SAFI defined in this document, and from the IPv4-only island directly connected to it, 4PE routers MUST distinguish those two kinds of IPv4 routes. Further, the 4PE MP-BGP peer MUST establish the relation between Network Address of Next Hop and IPv6 Next Hop carried in the 4PE SAFI. Through this relation, 4PE routers can get IPv6 Next Hop using Network Address of Next Hop. The method or data structure used to do this is out of scope of this document.

5. Transport IPv4 packets among 4PE routers

When ingress 4PE router receives IPv4 packet, it treats the IPv4 packet as normal IPv4 router does except for the following steps. If the matching IPv4 route for this packet is learned from other 4PE routers through the 4PE SAFI defined in this document, the 4PE router has further to get the IPv6 Next Hop using the IPv4 next hop of the

matching IPv4 route. Then, ingress 4PE router uses the IPv6 Next Hop to lookup in its IPv6 routing table to get the IPv6-signaled LSP to reach the egress 4PE router. Next, ingress 4PE router encapsulates the received IPv4 packet using two labels as shown in the figure below. Finally, ingress 4PE router forwards the encapsulated packet to egress 4PE router through the IPv6-signaled LSP.



When the packet reaches egress 4PE router, only 4PE label is left since the forwarding label is popped up by the penultimate hop toward egress 4PE router. Egress 4PE router pops up the 4PE label, looks up in the IPv4 routing table using the destination address of the IPv4 packet, and then forwards the IPv4 packets to the IPv4-only island.

6. IANA Requirements

IANA is requested to assign a new SAFI for the 4PE SAFI. 4PE routers use this SAFI to transport IPv4 routes, the corresponding MPLS label, IPv4 and IPv6 next hop addresses among 4PE routers. The following number is suggested.

Value	Meaning
9	4PE SAFI

7. Security Consideration

This document raises no new security issues.

8. References

8.1. Normative References

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <http://www.rfc-editor.org/info/rfc3107>.

[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>.

[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>.

8.2. Informative References

[RFC4798] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)", RFC 4798, DOI 10.17487/RFC4798, February 2007, <http://www.rfc-editor.org/info/rfc4798>.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <http://www.rfc-editor.org/info/rfc5036>.

[RFC7439] George, W., Ed. and C. Pignataro, Ed., "Gap Analysis for Operating IPv6-Only MPLS Networks", RFC 7439, DOI 10.17487/RFC7439, January 2015, <http://www.rfc-editor.org/info/rfc7439>.

[RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R. Papneja, "Updates to LDP for IPv6", RFC 7552, DOI 10.17487/RFC7552, June 2015, <http://www.rfc-editor.org/info/rfc7552>.

Author’s Address

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing 100032
P.R. China

Email: li_zhenqiang@hotmail.com

BGP Enabled Services
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2017

Z. Li
China Mobile

March 9, 2017

Connecting IPv4 Islands over IPv6 MPLS Using IPv4 Provider Edge Routers (4PE)
draft-li-bess-4pe-02

Abstract

This document explains how to interconnect IPv4 islands over a Multiprotocol Label Switching (MPLS)-enabled IPv6-only core. This approach relies on IPv4 Provider Edge routers (4PE), which are Dual Stacks in order to connect to IPv4 islands and to the MPLS core. The 4PE routers exchange the IPv4 reachability information transparently over the core using the Multiprotocol Border Gateway Protocol (MP-BGP). MP-BGP is extended to do this. A new Subsequence Address Family Identifier (SAFI) with corresponding new format Network Layer Reachability Information (NLRI), is introduced. The BGP Next Hop field is used to convey the IPv4 address of the 4PE router, a field is added in Network Layer Reachability Information (NLRI) to convey the IPv6 address of the 4PE router, so that dynamically established IPv6-signaled MPLS Label Switched Paths (LSPs) can be used without explicit tunnel configuration.

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/.

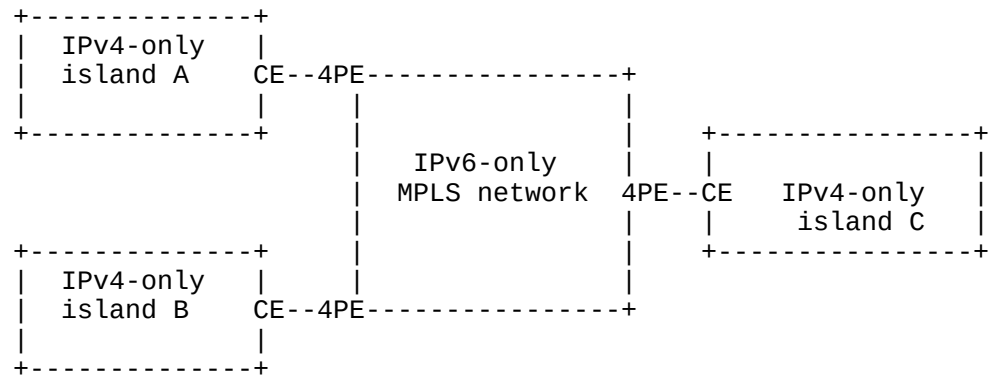
Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2017.

Table of Contents

1. Introduction

After IPv6 [RFC2460] is widely deployed, IPv6-only networks and nodes will become dominate. How to provide the connectivity for the remaining IPv4-only islands through the IPv6-only MPLS network will become a problem, as depicted in the following figure.



IPv6 Provider Edge Routers (6PE), as specified in [RFC4798], are used to connect IPv6 islands over IPv4 MPLS network. However, [RFC7439] pointed out that there is no solution to address the above problem. So, in this document, 4PE is proposed to meet this gap.

2. Protocol Overview

Each IPv4 island is connected to at least one Provider Edge router that is located on the border of the IPv6-only MPLS network. We call such a router a IPv4 Provider Edge router (4PE). The 4PE router MUST be IPv4 and IPv6 dual stack. At least one IPv4 address MUST be configured for the 4PE to connect the IPv4-only island. And at least one IPv6 address on the IPv6-ony MPLS network side MUST be configured. The configured IPv6 address needs to be routable in the IPv6-only network, and there needs to be a label bound via an IPv6 label distribution protocol to this IPv6 route. The configured IPv4 address MUST be unique among the IPv4 islands.

As a result of this, every considered 4PE router knows which MPLS label (we call it forwarding label) to use to send packets to any other 4PE router. Note that [RFC5036] updated by [RFC7552] fulfills these requirements.

We call the 4PE router receiving IPv4 packets from an IPv4 island an ingress 4PE router (relative to these IPv4 packets). We call a 4PE router forwarding IPv4 packets to an IPv4 island an egress 4PE router (relative to these IPv4 packets).

Interconnecting IPv4 islands over an IPv6 MPLS network mainly includes the following two steps:

1. Exchange IPv4 reachability information among 4PE routers with MP-BGP [RFC4760]

The IPv4 routes MUST not be injected in the IPv6-only MPLS network.

The 4PE routers MUST exchange the IPv4 routes over MP-BGP sessions running over IPv6. To do so, a new Subsequence Address Family Identifier (SAFI) with corresponding new format Network Layer Reachability Information (NLRI), is introduced in this document. We call this new SAFI 4PE SAFI, the detail of which is illustrated in section 3. The MP-BGP Address Family Identifier (AFI) used MUST be IPv4 (value 1). The MP-BGP Network Address of Next Hop MUST be the 4PE IPv4 address from which the 4PE receives the IPv4 routes of the IPv4 island. The IPv6 address of the 4PE, the IPv4 routes of the IPv4 island, and the corresponding MPLS label for the IPv4 routes (we call it 4PE label) MUST be encoded in the NLRI.

The reason to allocate MPLS label for the IPv4 route is to reduce the requirements for the IPv6-only MPLS network, as explained in [RFC4798]. Here it is. While this approach could theoretically operate in some situations using a single level of labels, there are significant advantages in using a second level of labels that are bound to IPv4 prefixes via MP-BGP advertisements.

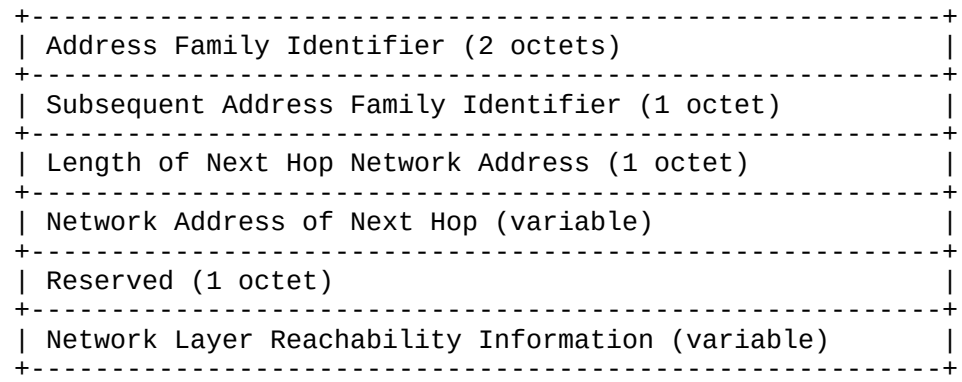
For instance, the use of a second level label allows Penultimate Hop Popping (PHP) on the IPv6 Label Switch Router (LSR) upstream of the egress 4PE router, without any IPv4 capabilities on the penultimate router. This is because it still transmits MPLS packets even after the PHP (instead of having to transmit IPv4 packets and encapsulate them appropriately).

2. Transport IPv4 packets from the ingress 4PE router to the egress 4PE router over the IPv6-signaled LSPs

   The ingress 4PE router MUST forward IPv4 data over the IPv6-signaled LSP towards the egress 4PE router identified by the IPv6 address advertised in the new introduced NLRI for the corresponding IPv4 route.

3. 4PE SAFI

As depicted in the following figure, 4PE SAFI is proposed to carry the IPv4 routes for the IPv4-only island, the MPLS label for the IPv4 routes, the IPv4 and IPv6 IP addresses of the 4PE router.



The use and meaning of these fields are as follows:

Address Family Identifier (AFI):

This field MUST be 1, to indicate IPv4 routes are carried in the NLRI. This is in accordance with [RFC4760]

Subsequent Address Family Identifier (SAFI):

This field is used to indicate the new introduced 4PE SAFI. The value for this field is be assigned by IANA.

Length of Next Hop Network Address:

This field MUST be 4, to indicate an IPv4 address is encoded in the "Network Address of Next Hop" field.

Network Address of Next Hop:

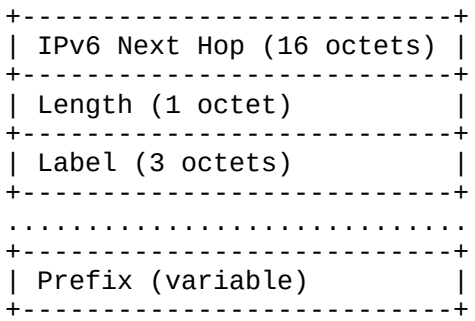
This field MUST be one of the 4PE IPv4 address from which the 4PE receives the IPv4 routes of the IPv4 island.

Reserved:

A 1 octet field that MUST be set to 0, and SHOULD be ignored upon receipt.

Network Layer Reachability Information (NLRI):

The format and meaning of this field is specified in the following.



The "IPv6 Next Hop" field MUST be the IPv6 address of the 4PE, through which the corresponding 4PE can be reached in the IPv6-only MPLS network.

The "Label" field MUST be the MPLS label allocated by the 4PE for the IPv4 routes carried in the Prefix field. This label is called 4PE label.

The "Prefix" field MUST be used to carry the IPv4 routes for the IPv4-only island.

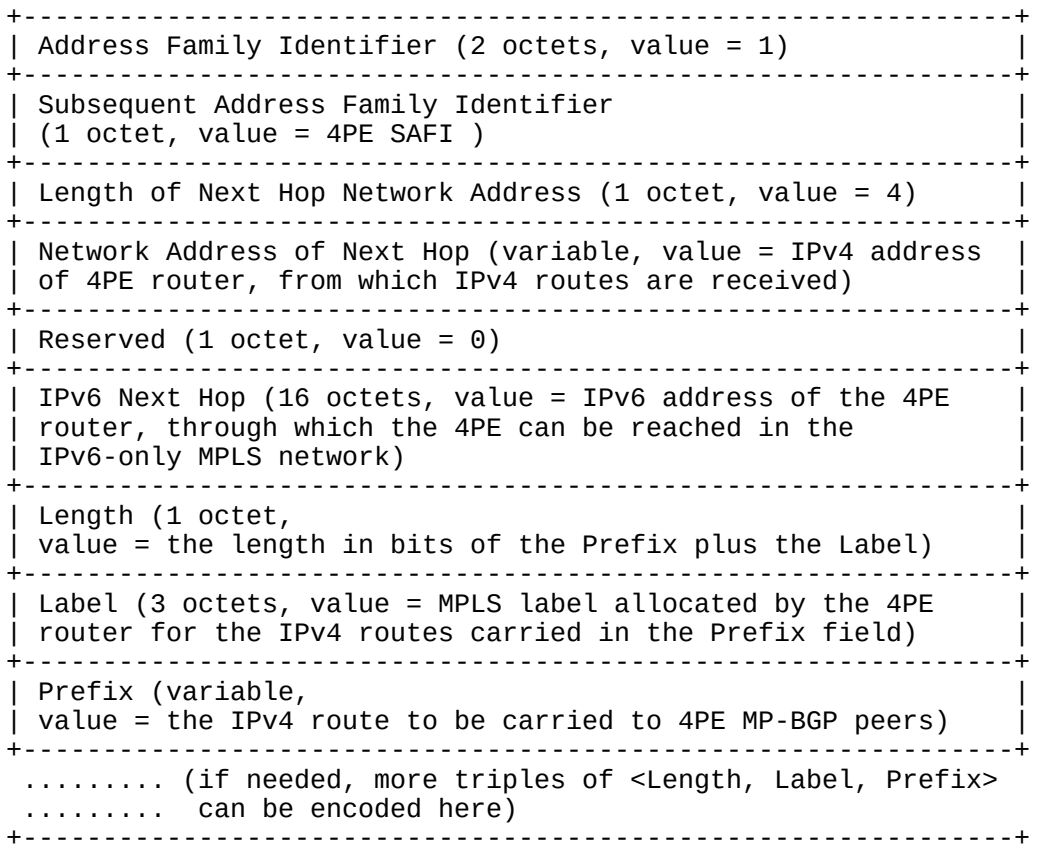
The "Length" field indicates the length in bits of the Prefix plus the Label.

One or more triples of the form <length, label, prefix > can be encoded in this NLRI. The use and meaning of the fields in this NLRI, except IPv6 Next Hop, are in accordance with [RFC3107].

4. Exchange IPv4 reachability information among 4PE routers

4PE routers MUST encode the IPv4 routes for the IPv4-only island in the UPDATE message of [RFC4271]. MP_REACH_NLRI (Type Code 14) and MP_UNREACH_NLRI (Type Code 15) defined in [RFC4760] MUST be used for route advertisement and withdraw respectively. 4PE SAFI defined in this document MUST be used in MP_REACH_NLRI or MP_UNREACH_NLRI to complete the exchange of IPv4 routes.

For advertisement, the 4PE router sets the fields of MP_REACH_NLRI as following, and then send the UPDATE message to its 4PE peers (or Route Reflectors(RR) in the network where RRs are deployed) over the MP-BGP session established on IPv6.

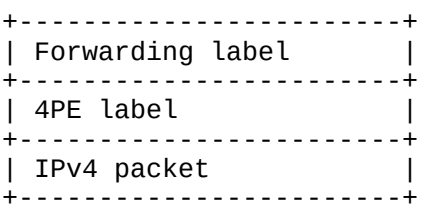


When receiving the UPDATE message, the 4PE MP-BGP peer treats the message as per [RFC4760] and [RFC3107]. Since the 4PE router can learn IPv4 routes from other 4PE routers through 4PE SAFI defined in this document, and from the IPv4-only island directly connected to it, 4PE routers MUST distinguish those two kinds of IPv4 routes. Further, the 4PE MP-BGP peer MUST establish the relation between Network Address of Next Hop and IPv6 Next Hop carried in the 4PE SAFI. Through this relation, 4PE routers can get IPv6 Next Hop using Network Address of Next Hop. The method or data structure used to do this is out of scope of this document.

5. Transport IPv4 packets among 4PE routers

When ingress 4PE router receives IPv4 packet, it treats the IPv4 packet as normal IPv4 router does except for the following steps. If the matching IPv4 route for this packet is learned from other 4PE routers through the 4PE SAFI defined in this document, the 4PE router has further to get the IPv6 Next Hop using the IPv4 next hop of the

matching IPv4 route. Then, ingress 4PE router uses the IPv6 Next Hop to lookup in its IPv6 routing table to get the IPv6-signaled LSP to reach the egress 4PE router. Next, ingress 4PE router encapsulates the received IPv4 packet using two labels as shown in the figure below. Finally, ingress 4PE router forwards the encapsulated packet to egress 4PE router through the IPv6-signaled LSP.



When the packet reaches egress 4PE router, only 4PE label is left since the forwarding label is popped up by the penultimate hop toward egress 4PE router. Egress 4PE router pops up the 4PE label, looks up in the IPv4 routing table using the destination address of the IPv4 packet, and then forwards the IPv4 packets to the IPv4-only island.

6. IANA Requirements

IANA is requested to assign a new SAFI for the 4PE SAFI from range 1-63, the registy of Subsequent Address Family Identifiers (SAFI) Parameters. 4PE routers use this SAFI to transport IPv4 routes, the corresponding MPLS label, IPv4 and IPv6 next hop addresses among 4PE routers.

7. Security Consideration

This document raises no new security issues.

8. References

8.1. Normative References

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

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <http://www.rfc-editor.org/info/rfc2460>.

[RFC3107] Rekhter, Y. and E. Rosen, "Carrying Label Information in BGP-4", RFC 3107, DOI 10.17487/RFC3107, May 2001, <http://www.rfc-editor.org/info/rfc3107>.

[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>.

[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>.

8.2. Informative References

[RFC4798] De Clercq, J., Ooms, D., Prevost, S., and F. Le Faucheur, "Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)", RFC 4798, DOI 10.17487/RFC4798, February 2007, <http://www.rfc-editor.org/info/rfc4798>.

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, DOI 10.17487/RFC5036, October 2007, <http://www.rfc-editor.org/info/rfc5036>.

[RFC7439] George, W., Ed. and C. Pignataro, Ed., "Gap Analysis for Operating IPv6-Only MPLS Networks", RFC 7439, DOI 10.17487/RFC7439, January 2015, <http://www.rfc-editor.org/info/rfc7439>.

[RFC7552] Asati, R., Pignataro, C., Raza, K., Manral, V., and R. Papneja, "Updates to LDP for IPv6", RFC 7552, DOI 10.17487/RFC7552, June 2015, <http://www.rfc-editor.org/info/rfc7552>.

Author’s Address

Zhenqiang Li
China Mobile
No.32 Xuanwumenxi Ave., Xicheng District
Beijing 100032
P.R. China

Email: li_zhenqiang@hotmail.com

BESS
Internet-Draft
Intended status: Standards Track
Expires: September 18, 2016

W. Lin
Z. Zhang
J. Drake
Juniper Networks, Inc.
J. Rabadan
Nokia

March 17, 2016

EVPN Inter-subnet Multicast Forwarding
draft-lin-bess-evpn-irb-mcast-02

Abstract

This document describes inter-subnet multicast forwarding procedures for Ethernet VPNs (EVPN).

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 September 18, 2016.

Copyright Notice

Table of Contents

1. Introduction

EVPN offers an efficient L2 VPN solution with all-active multi-homing support for intra-subnet connectivity over MPLS/IP network. EVPN also provides an integrated L2 and L3 service. When forwarding among Tenant Systems (TS) across different IP subnets is required, Integrated Routing and Bridging (IRB) can be used [ietf-bess-evpn-inter-subnet-forwarding].

An network virtualization endpoint (NVE) device supporting IRB is called a L3 Gateway. In a centralized approach, a centralized gateway provides all L3 routing functionality, and even two tenant systems on two subnets connected to the same NVE need to go through the central gateway, which is inefficient. In a distributed approach, each NVE has IRB configured, and inter-subnet traffic will be locally routed without having to go through a central gateway.

Inter-subnet multicast forwarding is more complicated and not covered in [ietf-bess-evpn-inter-subnet-forwarding]. This document describes the procedures for inter-subnet multicast forwarding.

For multicast traffic sourced from a TS in subnet 1, EVPN Broadcast, Unknow unicast, Multicast (BUM) forwarding based on RFC 7432, will deliver it to all sites in subnet 1. When IRBs in subnet 1 receive the mulitcast traffic, they deliver to other corresponding IRBs in

other subnets at L3. From L3 point of view, those NVEs are routers connected to the subnet via the IRB interfaces and the source is locally attached. Nothing is different from a traditional LAN and regular IGMP/MLD/PIM procedures kick in.

If a TS is a multicast receiver, it uses IGMP/MLD to signal its interest in some multicast flows. One of the gateways is the IGMP/MLD querier for a given subnet. It sends queries out the IRB for that subnet, which in turn causes the queries to be forwarded throughout the subnet following the EVPN BUM procedures. TS's send IGMP/MLD joins via multicast, which are also forwarded throughout the subnet via EVPN BUM procedure. The gateways receive the joins via their IRB interfaces. From layer 3 point of view, again it is nothing different from a traditional LAN.

On a traditional LAN, only one router can send multicast to the LAN. That is either the PIM Designated Router (DR) or IGMP/MLD querier (when PIM is not needed - e.g., the LAN is a stub network). On the source subnet, PIM is typically needed so that traffic can be delivered to other subnets via other routers. For example, in case of PIM-SM, the DR on the source network encapsulates the initial packets for a particular flow in PIM Register messages and unicasts the Register messages to the Rendezvous Point (RP) for that flow, triggering necessary states for that flow to be built throughout the network.

That also works in the EVPN scenario, although not efficiently. Consider the example depicted in Figure 1, where a tenant has two subnets corresponding to two VLANs realized by two EVPN Instances (EVIs) at three sites. The VLAN1 and VLAN 2 shown in Figure 1 correspond to subnet 1 and subnet 2 respectively. A multicast source is located at site 1 on subnet 1 and three receivers are located at site 2 on subnet 1, site 1 and 2 on subnet 2 respectively. On subnet 1, NVE1 is the PIM DR while on subnet 2, NVE3 is the PIM DR. The connection drawn in Figure 1 among NVEs are L3 connections.

Multicast traffic from the source at site 1 on subnet 1 is forwarded to all three sites on VLAN 1 following EVPN BUM procedure. Rcvr1 gets the traffic when NVE2 sends it out of its local Attachment Circuit (AC). The three gateways for EVI1 also receive the traffic on their IRB interfaces to potentially route to other subnets. NVE3 is the DR on subnet 2 so it routes the local traffic (from L3 point of view) to subnet 2 while NVE1/2 is not the DR on subnet 2 so they don't. Once traffic gets onto subnet 2, it is forwarded back to NVE1/2 and delivered to rcvr2/3 following the EVPN BUM procedures.

Notice that both NVE1 and NVE2 receive the multicast traffic from subnet 1 on their IRB interfaces for subnet 1, but they do not route

to subnet 2 where they are not the PIM DRs. Instead, they wait to receive traffic at L2 from NVE3. For example, for receiver 3 connected to NVE1 but on different IP subnet as the multicast source, the multicast traffic from source has to go from NVE1 to NVE3 and then back to NVE1 before it is being delivered to the receiver 3. This is similar to the hairpinning issue with centralized approach, here the multicast forwarding is centralized via the DR, even though distributed approach is being used for unicast (in that each NVE is supporting IRB and routing inter-subnet unicast traffic locally).

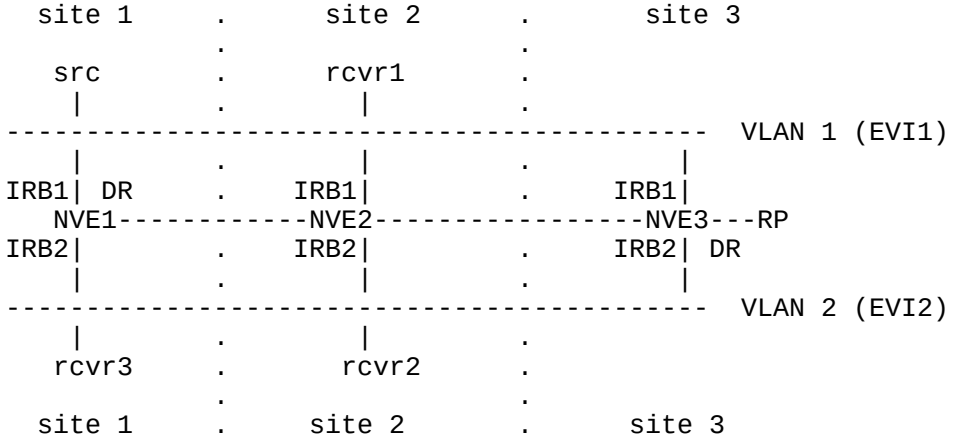


Figure 1 - EVPN IRB multicast scenario

2. EVPN-aware Solution

This multicast hairpinning can be avoided if the following procedures are followed:

- On the IRB interface, the gateway receives multicast traffic from a source subnet, it sends the traffic on its IRB interfaces to any other subnets that have receivers for the traffic, regardless whether the gateway is DR for that subnet or not.

- On the IRB interface, if a gateway receives Membership Reports from one of its ACs, it sends PIM joins towards the RP or source regardless if it is DR/querier or not.

- Multicast data traffic sent out of the IRB interfaces is forwarded to local ACs only and not to other EVPN sites.

Essentially, each router on an IRB interface behaves as a DR/querier for receivers (but only the true DR behaves as a DR for sources), and multicast data traffic from IRB interfaces is limited to local receivers.

Note that link local multicast traffic (e.g. addressed to 224.0.0.x in case of IPv4)), typically use for protocols, is not subject to the above procedures and still forwarded to remote sites following EVPN procedures.

In the example in Figure 1, when NVE1 gets traffic on its IRB1 interface it will route the traffic out of its IRB2 and deliver to local rcvr3. It also sends register messages to the RP, since it is the DR on the source network. Both NVE2 and NVE3 will receive the traffic on IRB1 but neither sends register messages to the RP, since they are not the DR on the source subnet. NVE2 will route the traffic out of its IRB2 and deliver to local rcvr2. NVE3 will also route the traffic out of IRB2 even though there is no receiver at the local site, because the IGMP/MLD joins from rcvr2/3 are also received by NVE3.

2.1. IGMP/MLD Snooping Consideration

In the example in Figure 1, NVE3 receives IGMP/MLD joins from rcvr2/3 and will route packets out of IRB2, even though there are no receivers at the local site. IGMP/MLD snooping on NVE3 can prevent the traffic from actually being sent out of ACs but at L3 there will still be related states and processing/forwarding (e.g., IRB2 will be in the downstream interface list for PIM join states and forwarding routes).

To prevent NVE3 from learning those remote receivers at all, IGMP/MLD snooping on NVE3 could optionally suppress the joins from remote sites being sent to its IRB interface. With that, in the example in Figure 1, NVE3 will not learn of rcvr2/3 on IRB2 and will not try to route packets out of IRB2 at all.

2.2. Receiver sites not connected to a source subnet

In the example in Figure 1, the source subnet is connected to all NVEs that has receiver sites, and there are no receivers outside the EVPN network. As a result, PIM is not really needed and each NVE can just route multicast traffic locally. In that case, IGMP/MLD querier will be responsible to send traffic to a subnet.

If there is a receiver subnet connected to an NVE that is not connected to the source subnet, then there must exist layer 3 multicast paths between them. This could be over an L3VPN core (in

this revision it is assumed that the subnets realized by EVPN are stub only and not transit) and normal PIM and MVPN procedures will be followed.

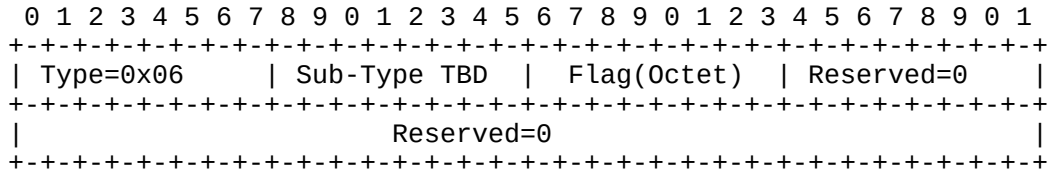
The L3VPN routes can be propagated either per RFC 4364 procedures or per EVPN Type 5 procedures [bess-evpn-prefix-advertisement]. BGP-MVPN [RFC 6514] requires that the routes used for RPF checking carry two extended communities (ECs) - VRF Route Import EC and Source AS EC. That must be applied to EVPN Prefix Advertisement (Type 5) routes as well.

2.3. Receiver sites without IRB

It is possible that a particular NVE may not have an IRB interface for its l2 domain. In that case, for traffic from another l2 domain, receivers need to receive from another NVE following EVPN procedures. The obvious choice is that it receives from the DR of that subnet. Because an NVE does not deliver traffic out of IRBs to remote sites with IRB, the DR needs to use a separate provider tunnel to deliver traffic only to sites that do not have IRB interfaces. The tunnel is advertised in new EVPN route type that is analogous to the MVPN "S-PMSI A-D" route [RFC6514]. This route will carry an EVPN Non-IRB Extended Community, indicating that a PE attached to the EVI identified in the route should join the advertised tunnel only if it does not have an IRB for that EVI. The routes could be either be a (*,*) wildcard S-PMSI A-D routes if an inclusive tunnel is used (but only for all sites without IRBs), or individual (*,g)/(s,*) S-PMSI A-D routes if selective tunnels are used per [draft-zzhang-bess-evpn-bum-procedure-updates]. The (*,*) wildcard S-PMSI A-D route may be advertised by the NVE carrying Non-IRB Site extended community for each of its BD to deliver multicast traffic routed out of the IRB interface to remote sites that do not have IRBs. Different RDs MUST be used for this (*, *) S-PMSI A-D route in the following case: if instead of an inclusive multicast Ethernet tag route, the NVE also uses (*,*) S-PMSI to deliver BUM traffic received from local ACs to remote PEs.

If [draft-sajassi-bess-evpn-igmp-mld-proxy] procedures are used, then routes from those non-IRB sites MUST also carry the EVPN non-IRB extended community, so that the DR will only forward traffic to those non-IRB NVEs.

The EVPN non-IRB Extended Community is a new EVPN extended community. EVPN extended communities are transitive extended community with a Type field of 6. The subtype of this new EVPN extended community will be assigned by IANA, and with the following 8-octet encoding:



The lower-order bit of the Flag is defined as non-IRB bit. A value one indicates non-IRB NVE. The rest of the undefined bits are set to zero.

2.4. Multi-homing Support

The solution works equally well in multi-homing situations, as long as the multi-homed PEs attached to the same Ethernet segment have the same IRB capability, which is expected to be the normal deployment.

As shown in Figure 2, both rcvr4 and rcvr5 are active-active multi-homed to NVE2 and NVE3. Receiver 4 is on subnet VLAN 1 and receiver 5 is on VLAN 2. When IRBs on NVE1 and NVE2 forward multicast traffic to its local attached access interface(s) based on EVPN BUM procedure, only DF for the ES deliveries multicast traffic to its multi-homed receiver. Hence no duplicated multicast traffic will be forwarded to receiver 4 or receiver 5.

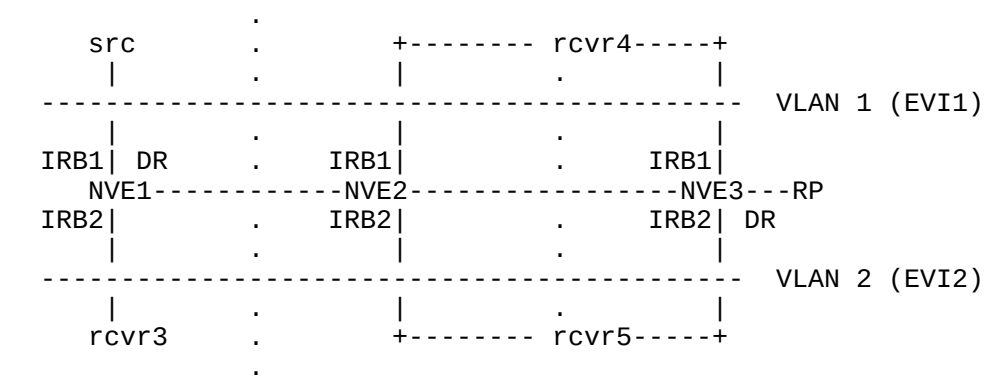


Figure 2 - EVPN IRB multicast and multi-homing

3. IANA Considerations

This document requests the following IANA assignments:

- A "Non-IRB Site" Sub-Type in "EVPN Extended Community Sub-Types" registry for the EVPN Non-IRB Extended Community.
- A "non-IRB" flag bit in the EVPN Non-IRB Extended Community.

4. Security Considerations

This document does not introduce new security risks.

5. Acknowledgements

The authors would like to thank Eric Rosen for his detailed review and valuable comments.

6. References

6.1. Normative References

[I-D.sajassi-bess-evpn-igmp-mld-proxy]
Sajassi, A., Patel, K., Thoria, S., Yeung, D., Drake, J., and W. Lin, "IGMP and MLD Proxy for EVPN", draft-sajassi-bess-evpn-igmp-mld-proxy-00 (work in progress), October 2015.

[I-D.zzhang-bess-evpn-bum-procedure-updates]
Zhang, J., Lin, W., Rabadan, J., and K. Patel, "Updates on EVPN BUM Procedures", draft-zzhang-bess-evpn-bum-procedure-updates-01 (work in progress), December 2015.

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

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

6.2. Informative References

[I-D.ietf-bess-evpn-inter-subnet-forwarding]
Sajassi, A., Salam, S., Thoria, S., Rekhter, Y., Drake, J., Yong, L., and L. Dunbar, "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-00 (work in progress), November 2014.

[I-D.ietf-bess-evpn-prefix-advertisement]
Rabadan, J., Henderickx, W., Palislamovic, S., Balus, F., and A. Isaac, "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-01 (work in progress), March 2015.

[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>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <http://www.rfc-editor.org/info/rfc6514>.

Authors' Addresses

Wen Lin
Juniper Networks, Inc.

EMail: wlin@juniper.net

Zhaohui Zhang
Juniper Networks, Inc.

EMail: zzhang@juniper.net

John Drake
Juniper Networks, Inc.

EMail: jdrake@juniper.net

Jorge Rabadan
Nokia

EMail: jorge.rabadan@nokia.com

BESS
Internet-Draft
Intended status: Standards Track
Expires: April 27, 2018

W. Lin
Z. Zhang
J. Drake
E. Rosen, Ed.
Juniper Networks, Inc.
J. Rabadan
Nokia
A. Sajassi
Cisco Systems

October 24, 2017

EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding
draft-lin-bess-evpn-irb-mcast-04

Abstract

Ethernet VPN (EVPN) provides a service that allows a single Local Area Network (LAN), i.e., a single IP subnet, to be distributed over multiple sites. The sites are interconnected by an IP or MPLS backbone. Intra-subnet traffic (either unicast or multicast) always appears to the endusers to be bridged, even when it is actually carried over the IP backbone. When a single "tenant" owns multiple such LANs, EVPN also allows IP unicast traffic to be routed between those LANs. This document specifies new procedures that allow inter-subnet IP multicast traffic to be routed among the LANs of a given tenant, while still making intra-subnet IP multicast traffic appear to be bridged. These procedures can provide optimal routing of the inter-subnet multicast traffic, and do not require any such traffic to leave a given router and then reenter that same router. These procedures also accommodate IP multicast traffic that needs to travel to or from systems that are outside the EVPN domain.

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 27, 2018.

Table of Contents

1. Introduction

1.1. Background

Ethernet VPN (EVPN) [RFC7432] provides a Layer 2 VPN (L2VPN) solution, which allows IP backbone provider to offer ethernet service to a set of customers, known as "tenants".

In this section (as well as in [EVPN-IRB]), we provide some essential background information on EVPN.

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].

1.1.1. Segments, Broadcast Domains, and Tenants

One of the key concepts of EVPN is the Broadcast Domain (BD). A BD is essentially an emulated ethernet. Each BD belongs to a single tenant. A BD typically consists of multiple ethernet "segments", and each segment may be attached to a different EVPN Provider Edge (EVPN-PE) router. EVPN-PE routers are often referred to as "Network Virtualization Endpoints" or NVEs. However, this document will use the term "EVPN-PE", or, when the context is clear, just "PE".

In this document, we use the term "segment" to mean the same as "Ethernet Segment" or "ES" in [RFC7432].

Attached to each segment are "Tenant Systems" (TSes). A TS may be any type of system, physical or virtual, host or router, etc., that can attach to an ethernet.

When two TSes are on the same segment, traffic between them does not pass through an EVPN-PE. When two TSes are on different segments of the same BD, traffic between them does pass through an EVPN-PE.

When two TSes, say TS1 and TS2 are on the same BD, then:

- If TS1 knows the MAC address of TS2, TS1 can send unicast ethernet frames to TS2. TS2 will receive the frames unaltered. That is, TS1's MAC address will be in the MAC Source Address field. If the frame contains an IP datagram, the IP header is not modified in any way during the transmission.

- If TS1 broadcasts an ethernet frame, TS2 will receive the unaltered frame.

- If TS1 multicasts an ethernet frame, TS2 will receive the unaltered frame, as long as TS2 has been provisioned to receive ethernet multicasts.

When we say that TS2 receives an unaltered frame from TS1, we mean that the frame still contains TS1's MAC address, and that no alteration of the frame's payload has been done.

EVPN allows a single segment to be attached to multiple PE routers. This is known as "EVPN multi-homing". EVPN has procedures to ensure that a frame from a given segment, arriving at a particular PE router, cannot be returned to that segment via a different PE router. This is particularly important for multicast, because a frame arriving at a PE from a given segment will already have been seen by all systems on the segment that need to see it. If the frame were sent back to the originating segment, receivers on that segment would receive the packet twice. Even worse, the frame might be sent back to a PE, which could cause an infinite loop.

1.1.2. Inter-BD (Inter-Subnet) IP Traffic

If a given tenant has multiple BDs, the tenant may wish to allow IP communication among these BDs. Such a set of BDs is known as an "EVPN Tenant Domain" or just a "Tenant Domain".

If tenant systems TS1 and TS2 are not in the same BD, then they do not receive unaltered ethernet frames from each other. In order for TS1 to send traffic to TS2, TS1 encapsulates an IP datagram inside an ethernet frame, and uses ethernet to send these frames to an IP router. The router decapsulates the IP datagram, does the IP processing, and re-encapsulates the datagram for ethernet. The MAC source address field now has the MAC address of the router, not of TS1. The TTL field of the IP datagram should be decremented by exactly 1; this hides the structure of the provider's IP backbone from the tenants.

EVPN accommodates the need for inter-BD communication within a Tenant Domain by providing an integrated L2/L3 service for unicast IP traffic. EVPN's Integrated Routing and Bridging (IRB) functionality is specified in [EVPN-IRB]. Each BD in a Tenant Domain is assumed to be a single IP subnet, and each IP subnet within a a given Tenant Domain is assumed to be a single BD. EVPN's IRB functionality allows IP traffic to travel from one BD to another, and ensures that proper IP processing (e.g., TTL decrement) is done.

A brief overview of IRB, including the notion of an "IRB interface", can be found in Appendix A. As explained there, an IRB interface is a sort of virtual interface connecting an L3 routing instance to a BD. A BD may have multiple attachment circuits (ACs) to a given PE, where each AC connects to a different ethernet segment of the BD. However, these ACs are not visible to the L3 routing function; from the perspective of an L3 routing instance, a PE has just one interface to each BD, viz., the IRB interface for that BD.

The "L3 routing instance" depicted in Appendix A is associated with a single Tenant Domain, and may be thought of as an IP-VRF for that Tenant Domain.

1.1.3. EVPN and IP Multicast

[EVPN-IRB] and [EVPN_IP_Prefix] cover inter-subnet (inter-BD) IP unicast forwarding, but they do not cover inter-subnet IP multicast forwarding.

[RFC7432] covers intra-subnet (intra-BD) ethernet multicast. The intra-subnet ethernet multicast procedures of [RFC7432] are used for ethernet Broadcast traffic, for ethernet unicast traffic whose MAC Destination Address field contains an Unknown address, and for ethernet traffic whose MAC Destination Address field contains an ethernet Multicast MAC address. These three classes of traffic are known collectively as "BUM traffic" (Broadcast/UnknownUnicast/ Multicast), and the procedures for handling BUM traffic are known as "BUM procedures".

[IGMP-Proxy] extends the intra-subnet ethernet multicast procedures by adding procedures that are specific to, and optimized for, the use of IP multicast within a subnet. However,that document does not cover inter-subnet IP multicast.

The purpose of this document is to specify procedures for EVPN that provide optimized IP multicast functionality within an EVPN tenant domain. This document also specifies procedures that allow IP multicast packets to be sourced from or destined to systems outside the Tenant Domain. We refer to the entire set of these procedures as "OISM" (Optimized Inter-Subnet Multicast) procedures.

In order to support the OISM procedures specified in this document, an EVPN-PE MUST also support [EVPN-IRB] and [IGMP-Proxy].

1.1.4. BDs, MAC-VRFS, and EVPN Service Models

[RFC7432] defines the notion of "MAC-VRF". A MAC-VRF contains one or more "Bridge Tables" (see section 3 of [RFC7432] for a discussion of this terminology), each of which represents a single Broadcast Domain.

In the IRB model (outlined in Appendix A) a L3 routing instance has one IRB interface per BD, NOT one per MAC-VRF. The procedures of this document are intended to work with all the EVPN service models. This document does not distinguish between a "Broadcast Domain" and a "Bridge Table", and will use the terms interchangeably (or will use the acronym "BD" to refer to either). The way the BDs are grouped into MAC-VRFs is not relevant to the procedures specified in this document.

Section 6 of [RFC7432] also defines several different EVPN service models:

- In the "vlan-based service", each MAC-VRF contains one "bridge table", where the bridge table corresponds to a particular Virtual LAN (VLAN). (See section 3 of [RFC7432] for a discussion of this terminology.) Thus each VLAN is treated as a BD.

- In the "vlan bundle service", each MAC-VRF contains one bridge table, where the bridge table corresponds to a set of VLANs. Thus a set of VLANs are treated as constituting a single BD.

- In the "vlan-aware bundle service", each MAC-VRF may contain multiple bridge tables, where each bridge table corresponds to one BD. If a MAC-VRF contains several bridge tables, then it corresponds to several BDs.

The procedures of this document are intended to work for all these service models.

1.2. Need for EVPN-aware Multicast Procedures

Inter-subnet IP multicast among a set of BDs can be achieved, in a non-optimal manner, without any specific EVPN procedures. For instance, if a particular tenant has n BDs among which he wants to send IP multicast traffic, he can simply attach a conventional multicast router to all n BDs. Or more generally, as long as each BD has at least one IP multicast router, and the IP multicast routers communicate multicast control information with each other, conventional IP multicast procedures will work normally, and no special EVPN functionality is needed.

However, that technique does not provide optimal routing for multicast. In conventional multicast routing, for a given multicast flow, there is only one multicast router on each BD that is permitted to send traffic of that flow to the BD. If that BD has receivers for a given flow, but the source of the flow is not on that BD, then the flow must pass through that multicast router. This leads to the "hair-pinning" problem described (for unicast) in Appendix A.

For example, consider an (S,G) flow that is sourced by a TS S and needs to be received by TSes R1 and R2. Suppose S is on a segment of BD1, R1 is on a segment of BD2, but both are attached to PE1. Suppose also that the tenant has a multicast router, attached to a segment of BD1 and to a segment of BD2. However, the segments to which that router is attached are both attached to PE2. Then the flow from S to R would have to follow the path: S-->PE1-->PE2-->Tenant Multicast Router-->PE2-->PE1-->R1. Obviously, the path S-->PE1-->R would be preferred.

Now suppose that there is a second receiver, R2. R2 is attached to a third BD, BD3. However, it is attached to a segment of BD3 that is attached to PE1. And suppose also that the Tenant Multicast Router is attached to a segment of BD3 that attaches to PE2. In this case, the Tenant Multicast Router will make two copies of the packet, one for BD2 and one for BD3. PE2 will send both copies back to PE1. Not only is the routing sub-optimal, but PE2 sends multiple copies of the same packet to PE1. This is a further sub-optimality.

This is only an example; many more examples of sub-optimal multicast routing can easily be given. To eliminate sub-optimal routing and extra copies, it is necessary to have a multicast solution that is EVPN-aware, and that can use its knowledge of the internal structure of a Tenant Domain to ensure that multicast traffic gets routed optimally. The procedures of this document allow us to avoid all such sub-optimalities when routing inter-subnet multicasts within a Tenant Domain.

1.3. Additional Requirements That Must be Met by the Solution

In addition to providing optimal routing of multicast flows within a Tenant Domain, the EVPN-aware multicast solution is intended to satisfy the following requirements:

- The solution must integrate well with the procedures specified in [IGMP-Proxy]. That is, an integrated set of procedures must handle both intra-subnet multicast and inter-subnet multicast.

- With regard to intra-subnet multicast, the solution MUST maintain the integrity of multicast ethernet service. This means:

    * If a source and a receiver are on the same subnet, the MAC source address (SA) of the multicast frame sent by the source will not get rewritten.

    * If a source and a receiver are on the same subnet, no IP processing of the ethernet payload is done. The IP TTL is not decremented, the header checksum is not changed, no fragmentation is done, etc.

- On the other hand, if a source and a receiver are on different subnets, the frame received by the receiver will not have the MAC Source address of the source, as the frame will appear to have come from a multicast router. Also, proper processing of the IP header is done, e.g., TTL decrement by 1, header checksum modification, possibly fragmentation, etc.

- If a Tenant Domain contains several BDs, it MUST be possible for a multicast flow (even when the multicast group address is an "any source multicast" (ASM) address), to have sources in one of those BDs and receivers in one or more of the other BDs, without requiring the presence of any system performing PIM Rendezvous Point (RP) functions ([RFC7761]). Multicast throughout a Tenant Domain must not require the tenant systems to be aware of any underlying multicast infrastructure.

- Sometimes a MAC address used by one TS on a particular BD is also used by another TS on a different BD. Inter-subnet routing of multicast traffic MUST NOT make any assumptions about the uniqueness of a MAC address across several BDs.

- If two EVPN-PEs attached to the same Tenant Domain both support the OISM procedures, each may receive inter-subnet multicasts from the other, even if the egress PE is not attached to any segment of the BD from which the multicast packets are being sourced. It MUST NOT be necessary to provision the egress PE with knowledge of the ingress BD.

- There must be a procedure that that allows EVPN-PE routers supporting OISM procedures to send/receive multicast traffic to/ from EVPN-PE routers that support only [RFC7432], but that do not support the OISM procedures or even the procedures of [EVPN-IRB]. However, when interworking with such routers (which we call "non-OISM PE routers"), optimal routing may not be achievable.

- It MUST be possible to support scenarios in which multicast flows with sources inside a Tenant Domain have "external" receivers, i.e., receivers that are outside the domain. It must also be possible to support scenarios where multicast flows with external

sources (sources outside the Tenant Domain) have receivers inside the domain.

This presupposes that unicast routes to multicast sources outside the domain can be distributed to EVPN-PEs attached to the domain, and that unicast routes to multicast sources within the domain can be distributed outside the domain.

Of particular importance are the scenario in which the external sources and/or receivers are reachable via L3VPN/MVPN, and the scenario in which external sources and/or receivers are reachable via IP/PIM.

The solution for external interworking MUST allow for deployment scenarios in which EVPN does not need to export a host route for every multicast source.

- The solution for external interworking must not presuppose that the same tunneling technology is used within both the EVPN domain and the external domain. For example, MVPN interworking must be possible when MVPN is using MPLS P2MP tunneling, and EVPN is using Ingress Replication or VXLAN tunneling.

- The solution must not be overly dependent on the details of a small set of use cases, but must be adaptable to new use cases as they arise. (That is, the solution must be robust.)

1.4. Terminology

In this document we make frequent use of the following terminology:

- OISM: Optimized Inter-Subnet Multicast. EVPN-PEs that follow the procedures of this document will be known as "OISM" PEs. EVPN-PEs that do not follow the procedures of this document will be known as "non-OISM" PEs.

- IP Multicast Packet: An IP packet whose IP Destination Address field is a multicast address that is not a link-local address. (Link-local addresses are IPv4 addresses in the 224/8 range and IPv6 address in the FF02/16 range.)

- IP Multicast Frame: An ethernet frame whose payload is an IP multicast packet (as defined above).

- (S,G) Multicast Packet: An IP multicast packet whose IP Source Address field contains S and whose IP Destination Address field contains G.

- (S,G) Multicast Frame: An IP multicast frame whose payload contains S in its IP Source Address field and G in its IP Destination Address field.

- Broadcast Domain (BD): an emulated ethernet, such that two systems on the same BD will receive each other's link-local broadcasts.

  Note that EVPN supports models in which a single EVPN Instance (EVI) contains only one BD, and models in which a single EVI contains multiple BDs. Both models are supported by this draft. However, a given BD belongs to only one EVI.

- Designated Forwarder (DF). As defined in [RFC7432], an ethernet segment may be multi-homed (attached to more than one PE). An ethernet segment may also contain multiple BDs, of one or more EVIs. For each such EVI, one of the PEs attached to the segment becomes that EVI's DF for that segment. Since a BD may belong to only one EVI, we can speak unambiguously of the BD's DF for a given segment.

  When the text makes it clear that we are speaking in the context of a given BD, we will frequently use the term "a segment's DF" to mean the given BD's DF for that segment.

- AC: Attachment Circuit. An AC connects the bridging function of an EVPN-PE to an ethernet segment of a particular BD. ACs are not visible at the router (L3) layer.

- L3 Gateway: An L3 Gateway is a PE that connects an EVPN tenant domain to an external multicast domain by performing both the OISM procedures and the Layer 3 multicast procedures of the external domain.

- PEG (PIM/EVPN Gateway): A L3 Gateway that connects an EVPN tenant domain to an external multicast domain whose Layer 3 multicast procedures are those of PIM ([RFC7761]).

- MEG (MVPN/EVPN Gateway): A L3 Gateway that connects an EVPN tenant domain to an external multicast domain whose Layer 3 multicast procedures are those of MVPN ([RFC6513], [RFC6514]).

- IPMG (IP Multicast Gateway): A PE that is used for interworking OISM EVPN-PEs with non-OISM EVPN-PEs.

- DR (Designated Router): A PE that has special responsibilities for handling multicast on a given BD.

- o Use of the "C-" prefix. In many documents on VPN multicast, the prefix "C-" appears before any address or wildcard that refers to an address or addresses in a tenant's address space, rather than to an address of addresses in the address space of the backbone network. This document omits the "C-" prefix in many cases where it is clear from the context that the reference is to the tenant's address space.

This document also assumes familiarity with the terminology of [RFC4364], [RFC6514], [RFC7432], [RFC7761], [IGMP-Proxy], [EVPN_IP_Prefix] and [EVPN-BUM].

1.5. Model of Operation: Overview

1.5.1. Control Plane

In this section, and in the remainder of this document, we assume the reader is familiar with the procedures of IGMP/MLD (see [RFC2236] and [RFC2710]), by which hosts announce their interest in receiving particular multicast flows.

Consider a Tenant Domain consisting of a set of k BDs: BD1, ..., BDk. To support the OISM procedures, each Tenant Domain must also be associated with a "Supplementary Broadcast Domain" (SBD). An SBD is treated in the control plane as a real BD, but it does not have any ACs. The SBD has several uses, that will be described later in this document. (See Section 2.1.)

Each PE that attaches to one or more of the BDs in a given tenant domain will be provisioned to recognize that those BDs are part of the same Tenant Domain. Note that a given PE does not need to be configured with all the BDs of a given Tenant Domain. In general, a PE will only be attached to a subset of the BDs in a given Tenant Domain, and will be configured only with that subset of BDs. However, each PE attached to a given Tenant Domain must be configured with the SBD for that Tenant Domain.

Suppose a particular segment of a particular BD is attached to PE1. [RFC7432] specifies that PE1 must originate an Inclusive Multicast Ethernet Tag (IMET) route for that BD, and that the IMET must be propagated to all other PEs attached to the same BD. If the given segment contains a host that has interest in receiving a particular multicast flow, either an (S,G) flow or a (*,G) flow, PE1 will learn of that interest by participating in the IGMP/MLD procedures, as specified in [IGMP-Proxy]. In this case, we will say that:

- PE1 is interested in receiving the flow;
- The AC attaching the interested host to PE1 is also said to be interested in the flow;
- The BD containing an AC that is interested in a particular flow is also said to be interested in that flow.

Once PE1 determines that it has interest in receiving a particular flow or set of flows, it uses the procedures of [IGMP-Proxy] to advertise its interest in those flows. It advertises its interest in a given flow by originating a Selective Multicast Ethernet Tag (SMET) route. An SMET route is propagated to the other PEs that attach to the same BD.

OISM PEs MUST follow the procedures of [IGMP-Proxy]. In this document, we extend the procedures of [IGMP-Proxy] so that IMET and SMET routes for a particular BD are distributed not just to PEs that attach to that BD, but to PEs that attach to any BD in the Tenant Domain.

In this way, each PE attached to a given Tenant Domain learns, from each other PE attached to the same Tenant Domain, the set of flows that are of interest to each of those other PEs.

An OISM PE that is provisioned with several BDs in the same Tenant Domain may originate an IMET route for each such BD. To indicate its support of [IGMP-Proxy], it MUST attach the EVPN Multicast Flags Extended Community to each such IMET route.

Suppose PE1 is provisioned with both BD1 and BD2, and is provisioned to consider them to be part of the same Tenant Domain. It is possible that PE1 will receive from PE2 both an IMET route for BD1 and an IMET route for BD2. If either of these IMET routes has the EVPN Multicast Flags Extended Community, PE1 MUST assume that PE2 is supporting the procedures of [IGMP-Proxy] for ALL BDs in the Tenant Domain.

If a PE supports OISM functionality, it MUST indicate that by attaching an "OISM-supported" flag or Extended Community (EC) to all its IMET routes. (Details to be specified in next revision.) An OISM PE SHOULD attach this flag or EC to all the IMET routes it originates. However, if PE1 imports IMET routes from PE2, and at least one of PE2’s IMET routes indicates that PE2 is an OISM PE, PE1 will assume that PE2 is following OISM procedures.

1.5.2. Data Plane

Suppose PE1 has an AC to a segment in BD1, and PE1 receives from that AC an (S,G) multicast frame (as defined in Section 1.4).

There may be other ACs of PE1 on which TSes have indicated an interest (via IGMP/MLD) in receiving (S,G) multicast packets. PE1 is responsible for sending the received multicast packet out those ACs. There are two cases to consider:

- Intra-Subnet Forwarding: In this case, an attachment AC with interest in (S,G) is connected to a segment that is part of the source BD, BD1. If the segment is not multi-homed, or if PE1 is the Designated Forwarder (DF) (see [RFC7432]) for that segment, PE1 sends the multicast frame on that AC without changing the MAC SA. The IP header is not modified at all; in particular, the TTL is not decremented.

- Inter-Subnet Forwarding: An AC with interest in (S,G) is connected to a segment of BD2, where BD2 is different than BD1. If PE1 is the DF for that segment (or if the segment is not multi-homed), PE1 decapsulates the IP multicast packet, performs any necessary IP processing (including TTL decrement), then re-encapsulates the packet appropriately for BD2. PE1 then sends the packet on the AC. Note that after re-encapsulation, the MAC SA will be PE1’s MAC address on BD2. The IP TTL will have been decremented by 1.

In addition, there may be other PEs that are interested in (S,G) traffic. Suppose PE2 is such a PE. Then PE1 tunnels a copy of the IP multicast frame (with its original MAC SA, and with no alteration of the payload’s IP header). The tunnel encapsulation contains information that PE2 can use to associate the frame with a source BD. If the source BD is BD1:

- If PE2 is attached to BD1, the tunnel encapsulation used to send the frame to PE2 will cause PE2 to identify BD1 as the source BD.

- If PE2 is not attached to BD1, the tunnel encapsulation used to send the frame to PE2 will cause PE2 to identify the SBD as the source BD.

The way in which the tunnel encapsulation identifies the source BD is of course dependent on the type of tunnel that is used. This will be specified later in this document.

When PE2 receives the tunneled frame, it will forward it on any of its ACs that have interest in (S,G).

If PE2 determines from the tunnel encapsulation that the source BD is BD1, then

- For those ACs that connect PE2 to BD1, the intra-subnet forwarding procedure described above is used, except that it is now PE2, not PE1, carrying out that procedure. Unmodified EVPN procedures from [RFC7432] are used to ensure that a packet originating from a multi-homed segment is never sent back to that segment.

- For those ACs that do not connect to BD1, the inter-subnet forwarding procedure described above is used, except that it is now PE2, not PE1, carrying out that procedure.

If the tunnel encapsulation identifies the source BD as the SBD, PE2 applies the inter-subnet forwarding procedures described above to all of its ACs that have interest in the flow.

These procedures ensure that an IP multicast frame travels from its ingress PE to all egress PEs that are interested in receiving it. While in transit, the frame retains its original MAC SA, and the payload of the frame retains its original IP header. Note that in all cases, when an IP multicast packet is sent from one BD to another, these procedures cause its TTL to be decremented by 1.

So far we have assumed that an IP multicast packet arrives at its ingress PE over an AC that belongs to one of the BDs in a given Tenant Domain. However, it is possible for a packet to arrive at its ingress PE in other ways. Since an EVPN-PE supporting IRB has an IP-VRF, it is possible that the IP-VRF will have a "VRF interface" that is not an IRB interface. For example, there might be a VRF interface that is actually a physical link to an external ethernet switch, or to a directly attached host, or to a router. When an EVPN-PE, say PE1, receives a packet through such means, we will say that the packet has an "external" source (i.e., a source "outside the tenant domain"). There are also other scenarios in which a multicast packet might have an external source, e.g., it might arrive over an MVPN tunnel from an L3VPN PE. In such cases, we will still refer to PE1 as the "ingress EVPN-PE".

When an EVPN-PE, say PE1, receives an externally sourced multicast packet, and there are receivers for that packet inside the Tenant Domain, it does the following:

- Suppose PE1 has an AC in BD1 that has interest in (S,G). Then PE1 encapsulates the packet for BD1, filling in the MAC SA field with the MAC address of PE1 itself on BD1. It sends the resulting frame on the AC.

- o Suppose some other EVPN-PE, say PE2, has interest in (S,G). PE1 encapsulates the packet for ethernet, filling in the MAC SA field with PE1's own MAC address on the SBD. PE1 then tunnels the packet to PE2. The tunnel encapsulation will identify the source BD as the SBD. Since the source BD is the SBD, PE2 will know to treat the frame as an inter-subnet multicast.

When ingress replication is used to transmit IP multicast frames from an ingress EVPN-PE to a set of egress PEs, then of course the ingress PE has to send multiple copies of the frame. Each copy is the original ethernet frame; decapsulation and IP processing take place only at the egress PE.

If a Point-to-Multipoint (P2MP) tree or BIER ([EVPN-BIER]) is used to transmit an IP multicast frame from an ingress PE to a set of egress PEs, then the ingress PE only has to send one copy of the frame to each of its next hops. Again, each egress PE receives the original frame and does any necessary IP processing.

2. Detailed Model of Operation

The model described in Section 1.5.2 can be expressed more precisely using the notion of "IRB interface" (see Appendix A). However, this requires that the semantics of the IRB interface be modified for multicast packets. It is also necessary to have an IRB interface that connects the L3 routing instance of a particular Tenant Domain (in a particular PE) to the SBD of that Tenant Domain.

In this section we assume that PIM is not enabled on the IRB interfaces. In general, it is not necessary to enable PIM on the IRB interfaces unless there are PIM routers on one of the Tenant Domain's BDs, or unless there is some other scenario requiring a Tenant Domain's L3 routing instance to become a PIM adjacency of some other system. These cases will be discussed in Section 7.

2.1. Supplementary Broadcast Domain

Suppose a given Tenant Domain contains three BDs (BD1, BD2, BD3) and two PEs (PE1, PE2). PE1 attaches to BD1 and BD2, while PE2 attaches to BD2 and BD3.

To carry out the procedures described above, all the PEs attached to the Tenant Domain must be provisioned to have the SBD for that tenant domain. An RT must be associated with the SBD, and provisioned on each of those PEs. We will refer to that RT as the "SBD-RT".

A Tenant Domain is also configured with an IP-VRF ([EVPN-IRB]), and the IP-VRF is associated with an RT. This RT MAY be the same as the SBD-RT.

Suppose an (S,G) multicast frame originating on BD1 has a receiver on BD3. PE1 will transmit the packet to PE2 as a frame, and the encapsulation will identify the frame's source BD as BD1. Since PE2 is not provisioned with BD1, it will treat the packet as if its source BD were the SBD. That is, a packet can be transmitted from BD1 to BD3 even though its ingress PE is not configured for BD3, and/or its egress PE is not configured for BD1.

EVPN supports service models in which a given EVPN Instance (EVI) can contain only one BD. It also supports service models in which a given EVI can contain multiple BDs. The SBD can be treated either as its own EVI, or it can be treated as one BD within an EVI that contains multiple BDs. The procedures specified in this document accommodate both cases.

2.2. When is a Route About/For/From a Particular BD

In this document, we will frequently say that a particular route is "about" a particular BD, or is "from" a particular BD, or is "for" a particular BD or is "related to" a particular BD. These terms are used interchangeably. In this section, we explain exactly what that means.

In EVPN, each BD is assigned an RT. In some service models, each BD is assigned a unique RT. In other service models, a set of BDs (all in the same Tenant Domain) may be assigned the same RT. (An RT is actually assigned to a MAC-VRF, and hence is shared by all the BDs that share the MAC-VRF.) The RT is a BGP extended community that may be attached to the BGP routes used by the EVPN control plane.

In those service models that allow a set of BDs to share a single RT, each BD is assigned a non-zero Tag ID. The Tag ID appears in the Network Layer Reachability Information (NLRI) of many of the BGP routes that are used by the EVPN control plane.

A route is about a particular BD if it carries the RT that has been assigned to that BD, and its NLRI contains the Tag ID that has been assigned to that BD.

Note that a route that is about a particular BD may also carry additional RTs.

2.3. Use of IRB Interfaces at Ingress PE

When an (S,G) multicast frame is received from an AC belonging to a particular BD, say BD1:

1. The frame is sent unchanged to other EVPN-PEs that are interested in (S,G) traffic. The encapsulation used to send the frame to the other EVPN-PEs depends on the tunnel type being used for multicast transmission. (For our purposes, we consider Ingress Replication (IR), Assisted Replication (AR) and BIER to be "tunnel types", even though IR, AR and BIER do not actually use P2MP tunnels.) At the egress PE, the source BD of the frame can be inferred from the tunnel encapsulation. If the egress PE is not attached to the real source BD, it will infer that the source BD is the SBD.

   Note that the the inter-PE transmission of a multicast frame among EVPN-PEs of the same Tenant Domain does NOT involve the IRB interfaces, as long as the multicast frame was received over an AC attached to one of the Tenant Domain’s BDs.

2. The frame is also sent up the IRB interface that attaches BD1 to the Tenant Domain’s L3 routing instance in this PE. That is, the L3 routing instance, behaving as if it were a multicast router, receives the IP multicast frames that arrive at the PE from its local ACs. The L3 routing instance decapsulates the frame’s payload to extract the IP multicast packet, decrements the IP TTL, adjusts the header checksum, and does any other necessary IP processing (e.g., fragmentation).

3. The L3 routing instance keeps track of which BDs have local receivers for (S,G) traffic. (A "local receiver" is a tenant system, reachable via a local attachment circuit that has expressed interest in (S,G) traffic.) If the L3 routing instance has an IRB interface to BD2, and it knows that BD2 has a LOCAL receiver interested in (S,G) traffic, it encapsulates the packet in an ethernet header for BD2, putting its own MAC address in the MAC SA field. Then it sends the packet down the IRB interface to BD2.

If a packet is sent from the L3 routing instance to a particular BD via the IRB interface (step 3 in the above list), and if the BD in question is NOT the SBD, the packet is sent ONLY to LOCAL ACs of that BD. If the packet needs to go to other PEs, it has already been sent to them in step 1. Note that this is a change in the IRB interface semantics from what is described in [EVPN-IRB] and Figure 2.

Existing EVPN procedures ensure that a packet is not sent by a given PE to a given locally attached segment unless the PE is the DF for that segment. Those procedures also ensure that a packet is never sent by a PE to its segment of origin. Thus EVPN segment multi-homing is fully supported; duplicate delivery to a segment or looping on a segment are thereby prevented, without the need for any new procedures to be defined in this document.

What if an IP multicast packet is received from outside the tenant domain? For instance, perhaps PE1's IP-VRF for a particular tenant domain also has a physical interface leading to an external switch, host, or router, and PE1 receives an IP multicast packet or frame on that interface. Or perhaps the packet is from an L3VPN, or a different EVPN Tenant Domain.

Such a packet is first processed by the L3 routing instance, which decrements TTL and does any other necessary IP processing. Then the packet is sent into the Tenant Domain by sending it down the IRB interface to the SBD of that Tenant Domain. This requires encapsulating the packet in an ethernet header, with the PE's own MAC address, on the SBD, in the MAC SA field.

An IP multicast packet sent by the L3 routing instance down the IRB interface to the SBD is treated as if it had arrived from a local AC, and steps 1-3 are applied. Note that the semantics of sending a packet down the IRB interface to the SBD are thus slightly different than the semantics of sending a packet down other IRB interfaces. IP multicast packets sent down the SBD's IRB interface may be distributed to other PEs, but IP multicast packets sent down other IRB interfaces are distributed only to local ACs.

If a PE sends a link-local multicast packet down the SBD IRB interface, that packet will be distributed (as an ethernet frame) to other PEs of the Tenant Domain, but will not appear on any of the actual BDs.

2.4. Use of IRB Interfaces at an Egress PE

Suppose an egress EVPN-PE receives an (S,G) multicast frame from the frame's ingress EVPN-PE. As described above, the packet will arrive as an ethernet frame over a tunnel from the ingress PE, and the tunnel encapsulation will identify the source BD of the ethernet frame.

We define the notion of the frame's "inferred source BD" as follows. If the egress PE is attached to the actual source BD, the actual source BD is the inferred source BD. If the egress PE is not attached to the actual source BD, the inferred source BD is the SBD.

The egress PE now takes the following steps:

1. If the egress PE has ACs belonging to the inferred source BD of the frame, it sends the frame unchanged to any ACs of that BD that have interest in (S,G) packets. The MAC SA of the frame is not modified, and the IP header of the frame's payload is not modified in any way.

2. The frame is also sent to the L3 routing instance by being sent up the IRB interface that attaches the L3 routing instance to the inferred source BD. Steps 2 and 3 of Section 2.3 are then applied.

2.5. Announcing Interest in (S,G)

[IGMP-Proxy] defines the procedures used by an egress PE to announce its interest in a multicast flow or set of flows. This is done by originating an SMET route. If an egress PE determines it has LOCAL receivers in a particular BD that are interested in a particular set of flows, it originates one or more SMET routes for that BD. The SMET route specifies a flow or set of flows, and identifies the egress PE. The SMET route is specific to a particular BD. A PE that originates an SMET route is announcing "I have receivers for (S,G) or (*,G) in BD-x".

In [IGMP-Proxy], an SMET route for a particular BD carries a Route Target (RT) that ensures it will be distributed to all PEs that are attached to that BD. In this document, it is REQUIRED that an SMET route also carry the RT that is assigned to the SBD. This ensures that every ingress PE attached to a particular Tenant Domain will learn of all other PEs (attached to the same Tenant Domain) that have interest in a particular set of flows. Note that it is not necessary for the ingress PE to have any BDs other than the SBD in common with the egress PEs.

Since the SMET routes from any BD in a given Tenant Domain are propagated to all PEs of that Tenant Domain, an (S,G) receiver on one BD can receive (S,G) packets that originate in a different BD. Within an EVPN domain, a given IP source address can only be on one BD. Therefore inter-subnet multicasting can be done, within the Tenant Domain, without requiring any Rendezvous Points, shared trees, or other complex aspects of multicast routing infrastructure. (Note that while the MAC addresses do not have to be unique across all the BDs in a Tenant Domain, the IP addresses to have to be unique across all those BDs.)

If some PE attached to the Tenant Domain does not support [IGMP-Proxy], it will be assumed to be interested in all flows. Whether a

particular remote PE supports [IGMP-Proxy] is determined by the presence of the Multicast Flags Extended Community in its IMET route; this is specified in [IGMP-Proxy].)

2.6. Tunneling Frames from Ingress PE to Egress PEs

[RFC7432] specifies the procedures for setting up and using "BUM tunnels". A BUM tunnel is a tunnel used to carry traffic on a particular BD if that traffic is (a) broadcast traffic, or (b) unicast traffic with an unknown MAC DA, or (c) ethernet multicast traffic.

This document allows the BUM tunnels to be used as the default tunnels for transmitting intra-subnet IP multicast frames. It also allows a separate set of tunnels to be used, instead of the BUM tunnels, as the default tunnels for carrying intra-subnet IP multicast frames. Let’s call these "IP Multicast Tunnels".

When the tunneling is done via Ingress Replication or via BIER, this difference is of no significance. However, when P2MP tunnels are used, there is a significant advantages to having separate IP multicast tunnels.

It is desirable for an ingress PE to transmit a copy of a given (S,G) multicast frame on only one tunnel. All egress PEs interested in (S,G) packets must then join that tunnel. If the source BD/PE for an (S,G) packet is BD1/PE1, and PE2 has receivers for (S,G) on BD2, PE2 must join the P2MP LSP on which PE1 transmits the frame. PE2 must join this P2MP LSP even if PE2 is not attached to the source BD (BD1). If PE1 were transmitting the multicast frame on its BD1 BUM tunnel, then PE2 would have to join the BD1 BUM tunnel, even though PE2 has no BD1 attachment circuits. This would cause PE2 to pull all the BUM traffic from BD1, most of which it would just have to discard. Thus we RECOMMEND that the default IP multicast tunnels be distinct from the BUM tunnels.

Whether or not the default IP multicast tunnels are distinct from the BUM tunnels, selective tunnels for particular multicast flows can still be used. Traffic sent on a selective tunnel would not be sent on the default tunnel.

Notwithstanding the above, link local IP multicast traffic MUST always be carried on the BUM tunnels, and ONLY on the BUM tunnels. Link local IP multicast traffic consists of IPv4 traffic with a destination address prefix of 224/8 and IPv6 traffic with a destination address prefix of FF02/16. In this document, the terms "IP multicast packet" and "IP multicast frame" are defined in Section 1.4 so as to exclude the link-local traffic.

2.7. Advanced Scenarios

There are some deployment scenarios that require special procedures:

1. Some multicast sources or receivers are attached to PEs that support [RFC7432], but do not support this document or [EVPN-IRB]. To interoperate with these "non-OISM PEs", it is necessary to have one or more gateway PEs that interface the tunnels discussed in this document with the BUM tunnels of the legacy PEs. This is discussed in Section 5.

2. Sometimes multicast traffic originates from outside the EVPN domain, or needs to be sent outside the EVPN domain. This is discussed in Section 6. An important special case of this, integration with MVPN, is discussed in Section 6.1.2.

3. In some scenarios, one or more of the tenant systems is a PIM router, and the Tenant Domain is used for as a transit network that is part of a larger multicast domain. This is discussed in Section 7.

3. EVPN-aware Multicast Solution Control Plane

3.1. Supplementary Broadcast Domain (SBD) and Route Targets

Every Tenant Domain is associated with a single Supplementary Broadcast Domain (SBD), as discussed in Section 2.1. Recall that a Tenant Domain is defined to be a set of BDs that can freely send and receive IP multicast traffic to/from each other. If an EVPN-PE has one or more ACs in a BD of a particular Tenant Domain, and if the EVPN-PE supports the procedures of this document, that EVPN-PE must be provisioned with the SBD of that Tenant Domain.

At each EVPN-PE attached to a given Tenant Domain, there is an IRB interface leading from the L3 routing instance of that Tenant Domain and the SBD. However, the SBD has no ACs.

The SBD may be in an EVPN Instance (EVI) of its own, or it may be one of several BDs (of the same Tenant Domain) in an EVI.

Each SBD is provisioned with a Route Target (RT). All the EVPN-PEs supporting a given SBD are provisioned with that RT as an import RT.

Each SBD is also provisioned with a "Tag ID" (see Section 6 of [RFC7432]).

- If the SBD is the only BD in its EVI, the mapping from RT to SBD is one-to-one. The Tag ID is zero.

- o If the SBD is one of several BDs in its EVI, it may have its own RT, or it may share an RT with one or more of those other BDs. In either case, it must be assigned a non-zero Tag ID. The mapping from <RT, Tag ID> is always one-to-one.

We will use the term "SBD-RT" to denote the RT has has been assigned to an SBD. Routes carrying this RT will be propagated to all EVPN-PEs in the same Tenant Domain as the originator.

An EVPN-PE that receives a route can always determine whether a received route "belongs to" a particular SBD, by seeing if that route carries the SBD-RT and has the Tag ID of the SBD in its NLRI.

If the VLAN-based service model is being used for a particular Tenant Domain, and thus each BD is in a distinct EVI, it is natural to have the SBD be in a distinct EVI as well. If the VLAN-aware bundle service is being used, it is natural to include the SBD in the same EVI that contains the other BDs. However, it is not required to do so; the SBD can still be placed in an EVI of its own, if that is desired.

Note that an SBD, just like any other BD, is associated on each EVPN-PE with a MAC-VRF. Per [RFC7432], each MAC-VRF is associated with a Route Distinguisher (RD). When constructing a route that is "about" an SBD, an EVPN-PE will place the RD of the associated MAC-VRF in the "Route Distinguisher" field of the NLRI. (If the Tenant Domain has several MAC-VRFs on a given PE, the EVPN-PE has a choice of which RD to use.)

If Assisted Replication (AR, see [EVPN-AR]) is used, each AR-REPLICATOR for a given Tenant Domain must be provisioned with the SBD of that Tenant Domain, even if the AR-REPLICATOR does not have any L3 routing instance.

3.2. Advertising the Tunnels Used for IP Multicast

The procedures used for advertising the tunnels that carry IP multicast traffic depend upon the type of tunnel being used. If the tunnel type is neither Ingress Replication, Assisted Replication, nor BIER, there are procedures for advertising both "inclusive tunnels" and "selective tunnels".

When IR, AR or BIER are used to transmit IP multicast packets across the core, there are no P2MP tunnels. Once an ingress EVPN-PE determines the set of egress EVPN-PEs for a given flow, the IMET routes contain all the information needed to transport packets of that flow to the egress PEs.

If AR is used, the ingress EVPN-PE is also an AR-LEAF and the IMET route coming from the selected AR-REPLICATOR contains the information needed. The AR-REPLICATOR will behave as an ingress EVPN-PE when sending a flow to the egress EVPN-PEs.

If the tunneling technique requires P2MP tunnels to be set up (e.g., RSVP-TE P2MP, mLDP, PIM), some of the tunnels may be selective tunnels and some may be inclusive tunnels.

Selective tunnels are always advertised by the ingress PE using S-PMSI A-D routes ([EVPN-BUM]).

For inclusive tunnels, there is a choice between using a BD's ordinary "BUM tunnel" [RFC7432] as the default inclusive tunnel for carrying IP multicast traffic, or using a separate IP multicast tunnel as the default inclusive tunnel for carrying IP multicast. In the former case, the inclusive tunnel is advertised in an IMET route. In the latter case, the inclusive tunnel is advertised in a (C-*,C-*) S-PMSI A-D route ([EVPN-BUM]). Details may be found in subsequent sections.

3.2.1. Constructing SBD Routes

3.2.1.1. Constructing an SBD-IMET Route

In general, an EVPN-PE originates an IMET route for each real BD. Whether an EVPN-PE has to originate an IMET route for the SBD (of a particular Tenant Domain) depends upon the type of tunnels being used to carry EVPN multicast traffic across the backbone. In some cases, an IMET route does not need to be originated for the SBD, but the other IMET routes have to carry the SBD-RT as well as any other RTs they would ordinarily carry (per [RFC7432].

Subsequent sections will specify when it is necessary for an EVPN-PE to originate an IMET route for the SBD. We will refer to such a route as an "SBD-IMET route".

When an EVPN-PE needs to originate an SBD-IMET route that is "for" the SBD, it constructs the route as follows:

- the RD field of the route's NLRI is set to the RD of the MAC-VRF that is associated with the SBD;
- a Route Target Extended Community containing the value of the SBD-RT is attached to that route;
- the "Tag ID" field of the NLRI is set to the Tag ID that has been assigned to the SBD. This is most likely 0 if a VLAN-based or

VLAN-bundle service is being used and non-zero if a VLAN-aware bundle service is being used.

3.2.1.2. Constructing an SBD-SMET Route

An EVPN-PE can originate an SMET route to indicate that it has receivers, on a specified BD, for a specified multicast flow. In some scenarios, an EVPN-PE must originate an SMET route that is for the SBD, which we will call an "SBD-SMET route". Whether an EVPN-PE has to originate an SMET route for the SBD (of a particular tenant domain) depends upon various factors, detailed in subsequent sections.

When an EVPN-PE needs to originate an SBD-SMET route that is "for" the SBD, it constructs the route as follows:

- the RD field of the route’s NLRI is set to the RD of the MAC-VRF that is associated with the SBD;
- a Route Target Extended Community containing the value of the SBD-RT is attached to that route;
- the "Tag ID" field of the NLRI is set to the Tag ID that has been assigned to the SBD. This is most likely 0 if a VLAN-based or VLAN-bundle service is being used and non-zero if a VLAN-aware bundle service is being used.

3.2.1.3. Constructing an SBD-SPMSI Route

An EVPN-PE can originate an S-PMSI A-D route (see [EVPN-BUM]) to indicate that it is going to use a particular P2MP tunnel to carry the traffic of particular IP multicast flows. In general, an S-PMSI A-D route is specific to a particular BD. In some scenarios, an EVPN-PE must originate an S-PMSI A-D route that is for the SBD, which we will call an "SBD-SPMSI route". Whether an EVPN-PE has to originate an SBD-SPMSI route for (of a particular Tenant Domain) depends upon various factors, detailed in subsequent sections.

When an EVPN-PE needs to originate an SBD-SPMSI route that is "for" the SBD, it constructs the route as follows:

- the RD field of the route’s NLRI is set to the RD of the MAC-VRF that is associated with the SBD;
- a Route Target Extended Community containing the value of the SBD-RT is attached to that route;

- o the "Tag ID" field of the NLRI is set to the Tag ID that has been assigned to the SBD. This is most likely 0 if a VLAN-based or VLAN-bundle service is being used and non-zero if a VLAN-aware bundle service is being used.

3.2.2. Ingress Replication

When Ingress Replication (IR) is used to transport IP multicast frames of a given Tenant Domain, each EVPN-PE attached to that Tenant Domain MUST originate an SBD-IMET route, as described in Section 3.2.1.1.

The SBD-IMET route MUST carry a PMSI Tunnel attribute (PTA), and the MPLS label field of the PTA MUST specify a downstream-assigned MPLS label that maps uniquely (in the context of the originating EVPN-PE) to the SBD.

An EVPN-PE MUST also originate an IMET route for each BD to which it is attached, following the procedures of [RFC7432]. Each of these IMET routes carries a PTA that specifying a downstream-assigned label that maps uniquely (in the context of the originating EVPN-PE) to the BD in question. These IMET routes need not carry the SBD-RT.

When an ingress EVPN-PE needs to use IR to send an IP multicast frame from a particular source BD to an egress EVPN-PE, the ingress PE determines whether the egress PE has originated an IMET route for that BD. If so, that IMET route contains the MPLS label that the egress PE has assigned to the source BD. The ingress PE uses that label when transmitting the packet to the egress PE. Otherwise, the ingress PE uses the label that the egress PE has assigned to the SBD (in the SBD-IMET route originated by the egress).

Note that the set of IMET routes originated by a given egress PE, and installed by a given ingress PE, will change over time. If the egress PE withdraws its IMET route for the source BD, the ingress PE must stop using the label carried in that IMET route, and start using the label carried in the SBD-IMET route from that egress PE.

3.2.3. Assisted Replication

When Assisted Replication is used to transport IP multicast frames of a given Tenant Domain, each EVPN-PE (including the AR-REPLICATOR) attached to the Tenant Domain MUST originate an SBD-IMET route, as described in Section 3.2.1.1.

An AR-REPLICATOR attached to a given Tenant Domain is considered to be an EVPN-PE of that Tenant Domain. It is attached to all the BDs in the Tenant Domain, but it has no IRB interfaces.

As with Ingress Replication, the SBD-IMET route carries a PTA where the MPLS label field specifies the downstream-assigned MPLS label that identifies the SBD. However, the AR-REPLICATOR and AR-LEAF EVPN-PEs will set the PTA's flags differently, as per [EVPN-AR].

In addition, each EVPN-PE originates an IMET route for each BD to which it is attached. As in the case of Ingress Replication, these routes carry the downstream-assigned MPLS labels that identify the BDs and do not carry the SBD-RT.

When an ingress EVPN-PE, acting as AR-LEAF, needs to send an IP multicast frame from a particular source BD to an egress EVPN-PE, the ingress PE determines whether there is any AR-REPLICATOR that originated an IMET route for that BD. After the AR-REPLICATOR selection (if there are more than one), the AR-LEAF uses the label contained in the IMET route of the AR-REPLICATOR when transmitting packets to it. The AR-REPLICATOR receives the packet and, based on the procedures specified in [EVPN-AR], transmits the packets to the egress EVPN-PEs using the labels contained in the IMET routes received from the egress PEs.

If an ingress AR-LEAF for a given BD has not received any IMET route for that BD from an AR-REPLICATOR, the ingress AR-LEAF follows the procedures in Section 3.2.2.

3.2.4. BIER

When BIER is used to transport multicast packets of a given Tenant Domain, each EVPN-PE attached to that Tenant Domain MUST originate an SBD-IMET route, as described in Section 3.2.1.1.

In addition, IMET routes that are originated for other BDs in the Tenant Domain MUST carry the SBD-RT.

Each IMET route (including but not limited to the SBD-IMET route) MUST carry a PMSI Tunnel attribute (PTA). The MPLS label field of the PTA MUST specify an upstream-assigned MPLS label that maps uniquely (in the context of the originating EVPN-PE) to the BD for which the route is originated.

When an ingress EVPN-PE uses BIER to send an IP multicast packet (inside an ethernet frame) from a particular source BD to a set of egress EVPN-PEs, the ingress PE follows the BIER encapsulation with the upstream-assigned label it has assigned to the source BD. (This label will come from the originated SBD-IMET route ONLY if the traffic originated from outside the Tenant Domain.) An egress PE can determine from that label whether the packet's source BD is one of the BDs to which the egress PE is attached.

Further details on the use of BIER to support EVPN can be found in [EVPN-BIER].

3.2.5. Inclusive P2MP Tunnels

3.2.5.1. Using the BUM Tunnels as IP Multicast Inclusive Tunnels

The procedures in this section apply only when it is desired to use the BUM tunnels to carry IP multicast traffic across the backbone. In this cases, an IP multicast frame (whether inter-subnet or intra-subnet) will be carried across the backbone in the BUM tunnel belonging to its source BD. An EVPN-PE attached to a given Tenant Domain will then need to join the BUM tunnels for each BD in the Tenant Domain, even if the EVPN-PE is not attached to all of those BDs. The reason is that an IP multicast packet from any source BD might be needed by an EVPN-PE that is not attached to that source domain.

Note that this will cause BUM traffic from a given BD in a Tenant Domain to be sent to all PEs that attach to that tenant domain, even the PEs that don’t attach to the given BD. To avoid this, it is RECOMMENDED that the BUM tunnels not be used as IP Multicast inclusive tunnels, and that the procedures of Section 3.2.5.2 be used instead.

3.2.5.1.1. RSVP-TE P2MP

When BUM tunnels created by RSVP-TE P2MP are used to transport IP multicast frames of a given Tenant Domain, each EVPN-PE attached to that Tenant Domain MUST originate an SBD-IMET route, as described in Section 3.2.1.1.

In addition, IMET routes that are originated for other BDs in the Tenant Domain MUST carry the SBD-RT.

Each IMET route (including but not limited to the SBD-IMET route) MUST carry a PMSI Tunnel attribute (PTA).

If received IMET route is not the SBD-IMET route, it will also be carrying the RT for its source BD. The route’s NLRI will carry the Tag ID for the source BD. From the RT and the Tag ID, any PE receiving the route can determine the route’s source BD.

If the MPLS label field of the PTA contains zero, the specified RSVP-TE P2MP tunnel is used only to carry frames of a single source BD.

If the MPLS label field of the PTA does not contain zero, it MUST contain an upstream-assigned MPLS label that maps uniquely (in the context of the originating EVPN-PE) to the source BD (or, in the case of an SBD-IMET route, the SBD). The tunnel may be used to carry frames of multiple source BDs, and the source BD for a particular packet is inferred from the label carried by the packet.

IP multicast traffic originating outside the Tenant Domain is transmitted with the label corresponding to the SBD, as specified in the ingress EVPN-PE's SBD-IMET route.

3.2.5.1.2. mLDP or PIM

When either mLDP or PIM is used to transport multicast packets of a given Tenant Domain, an EVPN-PE attached to that tenant domain originates an SBD-IMET route only if it is the ingress PE for IP multicast traffic originating outside the tenant domain. Such traffic is treated as having the SBD as its source BD.

An EVPN-PE MUST originate an IMET routes for each BD to which it is attached. These IMET routes MUST carry the SBD-RT of the Tenant Domain to which the BD belongs. Each such IMET route must also carry the RT of the BD to which it belongs.

When an IMET route (other than the SBD-IMET route) is received by an egress PE, the route will be carrying the RT for its source BD and the route's NLRI will contain the Tag ID for that source BD. This allows any PE receiving the route to determine the source BD associated with the route.

If the MPLS label field of the PTA contains zero, the specified mLDP or PIM tunnel is used only to carry frames of a single source BD.

If the MPLS label field of the PTA does not contain zero, it MUST contain an upstream-assigned MPLS label that maps uniquely (in the context of the originating EVPN-PE) to the source BD. The tunnel may be used to carry frames of multiple source BDs, and the source BD for a particular packet is inferred from the label carried by the packet.

The EVPN-PE advertising these IMET routes is specifying the default tunnel that it will use (as ingress PE) for transmitting IP multicast packets. The upstream-assigned label allows an egress PE to determine the source BD of a given packet.

The procedures of this section apply whenever the tunnel technology is based on the construction of the multicast trees in a "receiver-driven" manner; mLDP and PIM are two ways of constructing trees in a receiver-driven manner.

3.2.5.2. Using Wildcard S-PMSI A-D Routes to Advertise Inclusive Tunnels Specific to IP Multicast

The procedures of this section apply when (and only when) it is desired to transmit IP multicast traffic on an inclusive tunnel, but not on the same tunnel used to transmit BUM traffic.

However, these procedures do NOT apply when the tunnel type is Ingress Replication or BIER, EXCEPT in the case where it is necessary to interwork between non-OISM PEs and OISM PEs, as specified in Section 5.

Each EVPN-PE attached to the given Tenant Domain MUST originate an SBD-SPMSI A-D route. The NLRI of that route MUST contain (C-*,C-*) (see [RFC6625]). Additional rules for constructing that route are given in Section 3.2.1.3.

In addition, an EVPN-PE MUST originate an S-PMSI A-D route containing (C-*,C-*) in its NLRI for each of the other BDs in the Tenant Domain to which it is attached. All such routes MUST carry the SBD-RT. This ensures that those routes are imported by all EVPN-PEs attached to the Tenant Domain.

The route carrying the PTA will also be carrying the RT for that source BD, and the route’s NLRI will contain the Tag ID for that source BD. This allows any PE receiving the route to determine the source BD associated with the route.

If the MPLS label field of the PTA contains zero, the specified tunnel is used only to carry frames of a single source BD.

If the MPLS label field of the PTA does not contain zero, it MUST specify an upstream-assigned MPLS label that maps uniquely (in the context of the originating EVPN-PE) to the source BD. The tunnel may be used to carry frames of multiple source BDs, and the source BD for a particular packet is inferred from the label carried by the packet.

The EVPN-PE advertising these S-PMSI A-D route routes is specifying the default tunnel that it will use (as ingress PE) for transmitting IP multicast packets. The upstream-assigned label allows an egress PE to determine the source BD of a given packet.

3.2.6. Selective Tunnels

An ingress EVPN-PE for a given multicast flow or set of flows can always assign the flow to a particular P2MP tunnel by originating an S-PMSI A-D route whose NLRI identifies the flow or set of flows. The NLRI of the route could be (C-*,C-G), or (C-S,C-G). The S-PMSI A-D

route MUST carry the SBD-RT, so that it is imported by all EVPN-PEs attached to the Tenant Domain.

An S-PMSI A-D route is "for" a particular source BD. It MUST carry the RT associated with that BD, and it MUST have the Tag ID for that BD in its NLRI.

Each such route MUST contain a PTA, as specified in Section 3.2.5.2.

An egress EVPN-PE interested in the specified flow or flows MUST join the specified tunnel. Procedures for joining the specified tunnel are specific to the tunnel type. (Note that if the tunnel type is RSVP-TE P2MP LSP, the Leaf Information Required (LIR) flag of the PTA SHOULD NOT be set. An ingress OISM PE knows which OISM EVPN PEs are interested in any given flow, and hence can add them to the RSVP-TE P2MP tunnel that carries such flows.)

When an EVPN-PE imports an S-PMSI A-D route, it infers the source BD from the RTs and the Tag ID. If the EVPN-PE is not attached to the source BD, the tunnel it specifies is treated as belonging to the SBD. That is, packets arriving on that tunnel are treated as having been sourced in the SBD. Note that a packet is only considered to have arrived on the specified tunnel if the packet carries the upstream-assigned label specified in in the PTA, or if there is no upstream-assigned label specified in the PTA.

It should be noted that when either IR or BIER is used, there is no need for an ingress PE to use S-PMSI A-D routes to assign specific flows to selective tunnels. The procedures of Section 3.3, along with the procedures of Section 3.2.2, Section 3.2.3, or Section 3.2.4, provide the functionality of selective tunnels without the need to use S-PMSI A-D routes.

3.3. Advertising SMET Routes

[IGMP-Proxy] allows an egress EVPN-PE to express its interest in a particular multicast flow or set of flows by originating an SMET route. The NLRI of the SMET route identifies the flow or set of flows as (C-*,C-*) or (C-*,C-G) or (C-S,C-G).

Each SMET route belongs to a particular BD. The Tag ID for the BD appears in the NLRI of the route, and the route carries the RT associated that that BD. From this <RT, tag> pair, other EVPN-PEs can identify the BD to which a received SMET route belongs. (Remember though that the route may be carrying multiple RTs.)

There are two cases to consider:

1. Case 1: When it is known that no BD of a Tenant Domain contains a multicast router.

   In this case, an egress PE can advertise its interest in a flow or set of flows by originating a single SMET route. The SMET route will belong to the SBD. We refer to this as an SBD-SMET route. The SBD-SMET route carries the SBD-RT, and has the Tag ID for the SBD in its NLRI. SMET routes for the individual BDs are not needed.

2. Case 2: When it is possible that a BD of a Tenant Domain contains a multicast router.

   Suppose that an egress PE is attached to a BD on which there might be a tenant multicast router. (The tenant router is not necessarily on a segment that is attached to that PE.) And suppose that the PE has one or more ACs attached to that BD which are interested in a given multicast flow. In this case, IN ADDITION to the SMET route for the SBD, the egress PE MUST originate an SMET route for that BD. This will enable the ingress PE(s) to send IGMP/MLD messages on ACs for the BD, as specified in [IGMP-Proxy].

   If an SMET route is not an SBD-SMET route, and if the SMET route is for (C-S,C-G) (i.e., no wildcard source), and if the EVPN-PE originating it knows the source BD of C-S, it MAY put only the RT for that BD on the route. Otherwise, the route MUST carry the SBD-RT, so that it gets distributed to all the EVPN-PEs attached to the tenant domain.

As detailed in [IGMP-Proxy], an SMET route carries flags saying whether it is to result in the propagation of IGMP v1, v2, or v3 messages on the ACs of the BD to which the SMET route belongs. These flags SHOULD be set to zero in an SBD-SMET route.

Note that a PE only needs to originate the set SBD-SMET routes that are needed to pull in all the traffic in which it is interested. Suppose PE1 has ACs attached to BD1 that are interested in (C-*,C-G) traffic, and ACs attached to BD2 that are interested in (C-S,C-G) traffic. A single SBD-SMET route specifying (C-*,C-G) will pull in all the necessary flows.

As another example, suppose the ACs attached to BD1 are interested in (C-*,C-G) but not in (C-S,C-G), while the ACs attached to BD2 are interested in (C-S,C-G). A single SBD-SMET route specifying (C-*,C-G) will pull in all the necessary flows.

In other words, to determine the set of SBD-SMET routes that have to be sent for a given C-G, the PE has to merge the IGMP/MLD state for all the BDs (of the given Tenant Domain) to which it is attached.

Per [IGMP-Proxy], importing an SMET route for a particular BD will cause IGMP/MLD state to be instantiated for the IRB interface to that BD. This applies as well when the BD is the SBD.

However, traffic originating in a BD of a particular Tenant Domain MUST NOT be sent down the IRB interface that connects the L3 routing instance of that Tenant Domain to the SBD of that Tenant Domain. That would cause duplicate delivery of traffic, since traffic arriving at L3 over the IRB interface from the SBD has already been distributed throughout the Tenant Domain. When setting up the IGMP/MLD state based on SBD-SMET routes, care must be taken to ensure that the IRB interface to the SBD is not added to the Outgoing Interface (OIF) list if the traffic originates within the Tenant Domain.

4. Constructing Multicast Forwarding State

4.1. Layer 2 Multicast State

An EVPN-PE maintains "layer 2 multicast state" for each BD to which it is attached.

Let PE1 be an EVPN-PE, and BD1 be a BD to which it is attached. At PE1, BD1’s layer 2 multicast state for a given (C-S,C-G) or (C-*,C-G) governs the disposition of an IP multicast packet that is received by BD1’s layer 2 multicast function on an EVPN-PE.

An IP multicast (S,G) packet is considered to have been received by BD1’s layer 2 multicast function in PE1 in the following cases:

- The packet is the payload of an ethernet frame received by PE1 from an AC that attaches to BD1.
- The packet is the payload of an ethernet frame whose source BD is BD1, and which is received by the PE1 over a tunnel from another EVPN-PE.
- The packet is received from BD1’s IRB interface (i.e., has been transmitted by PE1’s L3 routing instance down BD1’s IRB interface).

According to the procedures of this document, all transmission of IP multicast packets from one EVPN-PE to another is done at layer 2. That is, the packets are transmitted as ethernet frames, according to the layer 2 multicast state.

Each layer 2 multicast state (S,G) or (*,G) contains a set "output interfaces" (OIF list). The disposition of an (S,G) multicast frame received by BD1's layer 2 multicast function is determined as follows:

- The OIF list is taken from BD1's layer 2 (S,G) state, or if there is no such (S,G) state, then from BD1's (*,G) state. (If neither state exists, the OIF list is considered to be null.)

- The rules of Section 4.1.2 are applied to the OIF list. This will generally result in the frame being transmitted to some, but not all, elements of the OIF list.

Note that there is no RPF check at layer 2.

4.1.1. Constructing the OIF List

In this document, we have extended the procedures of [IGMP-Proxy] so that IMET and SMET routes for a particular BD are distributed not just to PEs that attach to that BD, but to PEs that attach to any BD in the Tenant Domain. In this way, each PE attached to a given Tenant Domain learns, from each other PE attached to the same Tenant Domain, the set of flows that are of interest to each of those other PEs. (If some PE attached to the Tenant Domain does not support [IGMP-Proxy], it will be assumed to be interested in all flows. Whether a particular remote PE supports [IGMP-Proxy] is determined by the presence of an Extended Community in its IMET route; this is specified in [IGMP-Proxy].) If a set of remote PEs are interested in a particular flow, the tunnels used to reach those PEs are added to the OIF list of the multicast states corresponding to that flow.

An EVPN-PE may run IGMP/MLD procedures on each of its ACs, in order to determine the set of flows of interest to each AC. (An AC is said to be interested in a given flow if it connects to a segment that has tenant systems interested in that flow.) If IGMP/MLD procedures are not being run on a given AC, that AC is considered to be interested in all flows. For each BD, the set of ACs interested in a given flow is determined, and the ACs of that set are added to the OIF list of that BD's multicast state for that flow.

The OIF list for each multicast state must also contain the IRB interface for the BD to which the state belongs.

Implementors should note that the OIF list of a multicast state will change from time to time as ACs and/or remote PEs either become interested in, or lose interest in, particular multicast flows.

4.1.2. Data Plane: Applying the OIF List to an (S,G) Frame

When an (S,G) multicast frame is received by the layer 2 multicast function of a given EVPN-PE, say PE1, its disposition depends (a) the way it was received, (b) upon the OIF list of the corresponding multicast state (see Section 4.1.1), (c) upon the "eligibility" of an AC to receive a given frame (see Section 4.1.2.1 and (d) upon its source BD (see Section 3.2 for information about determining the source BD of a frame received over a tunnel from another PE).

4.1.2.1. Eligibility of an AC to Receive a Frame

A given (S,G) multicast frame is eligible to be transmitted by a given PE, say PE1, on a given AC, say AC1, only if one of the following conditions holds:

1. ESI labels are being used, PE1 is the DF for the segment to which AC1 is connected, and the frame did not originate from that same segment (as determined by the ESI label), or

2. The ingress PE for the frame is a remote PE, say PE2, local bias is being used, and PE2 is not connected to the same segment as AC1.

4.1.2.2. Applying the OIF List

Assume a given (S,G) multicast frame has been received by a given PE, say PE1. PE1 determines the source BD of the frame, finds the layer 2 (S,G) state for the source BD (or the (*,G) state if there is no (S,G) state), and takes the OIF list from that state. Note that if PE1 is not attached to the actual source BD, it will treat the frame as if its source BD is the SBD.

Suppose PE1 has determined the frame’s source BD to be BD1 (which may or may not be the SBD.) There are the following cases to consider:

1. The frame was received by PE1 from a local AC, say AC1, that attaches to BD1.

    a. The frame MUST be sent out all local ACs of BD1 that appear in the OIF list, except for AC1 itself.

    b. The frame MUST also be delivered to any other EVPN-PEs that have interest in it. This is achieved as follows:

        i. If (a) AR is being used, and (b) PE1 is an AR-LEAF, and (c) the OIF list is non-null, PE1 MUST send the frame to the AR-REPLICATOR.

ii. Otherwise the frame MUST be sent on all tunnels in the OIF list.

c. The frame MUST be sent to the local L3 routing instance by being sent up the IRB interface of BD1. It MUST NOT be sent up any other IRB interfaces.

2. The frame was received by PE1 over a tunnel from another PE. (See Section 3.2 for the rules to determine the source BD of a packet received from another PE. Note that if PE1 is not attached to the source BD, it will regard the SBD as the source BD.)

   a. The frame MUST be sent out all local ACs in the OIF list that connect to BD1 and that are eligible (per Section 4.1.2.1) to receive the frame.

   b. The frame MUST be sent up the IRB interface of the source BD. (Note that this may be the SBD.) The frame MUST NOT be sent up any other IRB interfaces.

   c. If PE1 is not an AR-REPLICATOR, it MUST NOT send the frame to any other EVPN-PEs. However, if PE1 is an AR-REPLICATOR, it MUST send the frame to all tunnels in the OIF list, except for the tunnel over which the frame was received.

3. The frame was received by PE1 from the BD1 IRB interface (i.e., the frame has been transmitted by PE1’s L3 routing instance down the BD1 IRB interface), and BD1 is NOT the SBD.

   a. The frame MUST be sent out all local ACs in the OIF list that are eligible (per Section 4.1.2.1 to receive the frame.

   b. The frame MUST NOT be sent to any other EVPN-PEs.

   c. The frame MUST NOT be sent up any IRB interfaces.

4. The frame was received from the SBD IRB interface (i.e., has been transmitted by PE1’s L3 routing instance down the SBD IRB interface).

   a. The frame MUST be sent on all tunnels in the OIF list. This causes the frame to be delivered to any other EVPN-PEs that have interest in it.

   b. The frame MUST NOT be sent on any local ACs.

   c. The frame MUST NOT be sent up any IRB interfaces.

4.2. Layer 3 Forwarding State

If an EVPN-PE is performing IGMP/MLD procedures on the ACs of a given BD, it processes those messages at layer 2 to help form the layer 2 multicast state. If also sends those messages up that BD's IRB interface to the L3 routing instance of a particular tenant domain. This causes layer 2 (C-S,C-G) or (C-*,C-G) L3 state to be created/ updated.

A layer 3 multicast state has both an Input Interface (IIF) and an OIF list.

To set the IIF of an (C-S,C-G) state, the EVPN-PE must determine the source BD of C-S. This is done by looking up S in the local MAC-VRF(s) of the given Tenant Domain.

If the source BD is present on the PE, the IIF is set to the IRB interface that attaches to that BD. Otherwise the IIF is set to the SBD IRB interface.

For (C-*,C-G) states, traffic can arrive from any BD, so the IIF needs to be set to a wildcard value meaning "any IRB interface".

The OIF list of these states includes one or more of the IRB interfaces of the Tenant Domain. In general, maintenance of the OIF list does not require any EVPN-specific procedures. However, there is one EVPN-specific rule:

> If the IIF is one of the IRB interfaces (or the wild card meaning "any IRB interface"), then the SBD IRB interface MUST NOT be added to the OIF list. Traffic originating from within a particular EVPN Tenant Domain must not be sent down the SBD IRB interface, as such traffic has already been distributed to all EVPN-PEs attached to that Tenant Domain.

Please also see Section 6.1.1, which states a modification of this rule for the case where OISM is interworking with external Layer 3 multicast routing.

5. Interworking with non-OISM EVPN-PEs

It is possible that a given Tenant Domain will be attached to both OISM PEs and non-OISM PEs. Inter-subnet IP multicast should be possible and fully functional even if not all PEs attaching to a Tenant Domain can be upgraded to support OISM functionality.

Note that the non-OISM PEs are not required to have IRB support, or support for [IGMP-Proxy]. It is however advantageous for the non-OISM PEs to support [IGMP-Proxy].

In this section, we will use the following terminology:

- PE-S: the ingress PE for an (S,G) flow.
- PE-R: an egress PE for an (S,G) flow.
- BD-S: the source BD for an (S,G) flow. PE-S must have one or more ACs attached BD-S, at least one of which attaches to host S.
- BD-R: a BD that contains a host interested in the flow. The host is attached to PE-R via an AC that belongs to BD-R.

To allow OISM PEs to interwork with non-OISM PEs, a given Tenant Domain needs to contain one or more "IP Multicast Gateways" (IPMGs). An IPMG is an OISM PE with special responsibilities regarding the interworking between OISM and non-OISM PEs.

If a PE is functioning as an IPMG, it MUST signal this fact by attaching a particular flag or EC (details to be determined) to its IMET routes. An IPMG SHOULD attach this flag or EC to all IMET routes it originates. However, if PE1 imports any IMET route from PE2 that has the "IPMG" flag or EC present, then the PE1 will assume that PE2 is an IPMG.

An IPMG Designated Forwarder (IPMG-DF) selection procedure is used to ensure that, at any given time, there is exactly one active IPMG-DF for any given BD. Details of the IPMG-DF selection procedure are in Section 5.1. The IPMG-DF for a given BD, say BD-S, has special functions to perform when it receives (S,G) frames on that BD:

- If the frames are from a non-OISM PE-S:
  - The IPMG-DF forwards them to OISM PEs that do not attach to BD-S but have interest in (S,G).

    Note that OISM PEs that do attach to BD-S will have received the frames on the BUM tunnel from the non-OISM PE-S.
  - The IPMG-DF forwards them to non-OISM PEs that have interest in (S,G) on ACs that do not belong to BD-S.

    Note that if a non-OISM PE has multiple BDs other than BD-S with interest in (S,G), it will receive one copy of the frame

for each such BD. This is necessary because the non-OISM PEs cannot move IP multicast traffic from one BD to another.

- If the frames are from an OISM PE, the IPMG-DF forwards them to non-OISM PEs that have interest in (S,G) on ACs that do not belong to BD-S.

  If a non-OISM PE has interest in (S,G) on an AC belonging to BD-S, it will have received a copy of the (S,G) frame, encapsulated for BD-S, from the OISM PE-S. (See Section 3.2.2.) If the non-OISM PE has interest in (S,G) on one or more ACs belonging to BD-R1,...,BD-Rk where the BD-Ri are distinct from BD-S, the IPMG-DF needs to send it a copy of the frame for BD-Ri.

If an IPMG receives a frame on a BD for which it is not the IPMG-DF, it just follows normal OISM procedures.

This section specifies several sets of procedures:

- the procedures that the IPMG-DF for a given BD needs to follow when receiving, on that BD, an IP multicast frame from a non-OISM PE;

- the procedures that the IPMG-DF for a given BD needs to follow when receiving, on that BD, an IP multicast frame from an OISM PE;

- the procedures that an OISM PE needs to follow when receiving, on a given BD, an IP multicast frame from a non-OISM PE, when the OISM PE is not the IPMG-DF for that BD.

To enable OISM/non-OISM interworking in a given Tenant Domain, the Tenant Domain MUST have some EVPN-PEs that can function as IPMGs. An IPMG must be configured with the SBD. It must also be configured with every BD of the Tenant Domain that exists on any of the non-OISM PEs of that domain. (Operationally, it may be simpler to configure the IPMG with all the BDs of the Tenant Domain.)

A non-OISM PE of course only needs to be configured with BDs for which it has ACs. An OISM PE that is not an IPMG only needs to be configured with the SBD and with the BDs for which it has ACs.

An IPMG MUST originate a wildcard SMET route (with (C-*,C-*) in the NLRI) for each BD in the Tenant Domain. This will cause it to receive all the IP multicast traffic that is sourced in the Tenant Domain. Note that non-OISM nodes that do not support [IGMP-Proxy] will send all the multicast traffic from a given BD to all PEs attached to that BD, even if those PEs do not originate an SMET route.

The interworking procedures vary somewhat depending upon whether packets are transmitted from PE to PE via Ingress Replication (IR) or via Point-to-Multipoint (P2MP) tunnels. We do not consider the use of BIER in this section, due to the low likelihood of there being a non-OISM PE that supports BIER.

5.1. IPMG Designated Forwarder

Each IPMG MUST be configured with an "IPMG dummy ethernet segment" that has no ACs.

EVPN supports a number of procedures that can be used to select the Designated Forwarder (DF) for a particular BD on a particular ethernet segment. Some of the possible procedures can be found, e.g., in [RFC7432], [EVPN-DF-NEW], and [EVPN-DF-WEIGHTED]. Whatever procedure is in use in a given deployment can be adapted to select an IPMG-DF for a given BD, as follows.

Each IPMG will originate an Ethernet Segment route for the IPMG dummy ethernet segment. It MUST carry a Route Target derived from the corresponding Ethernet Segment Identifier. Thus only IPMGs will import the route.

Once the set of IPMGs is known, it is also possible to determine the set of BDs supported by each IPMG. The DF selection procedure can then be used to choose a DF for each BD. (The conditions under which the IPMG-DF for a given BD changes depends upon the DF selection algorithm that is in use.)

5.2. Ingress Replication

The procedures of this section are used when Ingress Replication is used to transmit packets from one PE to another.

When a non-OISM PE-S transmits a multicast frame from BD-S to another PE, PE-R, PE-S will use the encapsulation specified in the BD-S IMET route that was originated by PE-R. This encapsulation will include the label that appears in the "MPLS label" field of the PMSI Tunnel attribute (PTA) of the IMET route. If the tunnel type is VXLAN, the "label" is actually a Virtual Network Identifier (VNI); for other tunnel types, the label is an MPLS label. In either case, we will speak of the transmitted frames as carrying a label that was assigned to a particular BD by the PE-R to which the frame is being transmitted.

To support OISM/non-OISM interworking, an OISM PE-R MUST originate, for each of its BDs, both an IMET route and an S-PMSI (C-*,C-*) A-D route. Note that even when IR is being used, interworking between

OISM and non-OISM PEs requires the OISM PEs to follow the rules of Section 3.2.5.2, as modified below.

Non-OISM PEs will not understand S-PMSI A-D routes. So when a non-OISM PE-S transmits an IP multicast frame with a particular source BD to an IPMG, it encapsulates the frame using the label specified in that IPMG's BD-S IMET route. (This is just the procedure of [RFC7432].)

The (C-*,C-*) S-PMSI A-D route originated by a given OISM PE will have a PTA that specifies IR.

- If MPLS tunneling is being used, the MPLS label field SHOULD contain a non-zero value, and the LIR flag SHOULD be zero. (The case where the MPLS label field is zero or the LIR flag is set is outside the scope of this document.)

- If the tunnel encapsulation is VXLAN, the MPLS label field MUST contain a non-zero value, and the LIR flag MUST be zero.

When an OISM PE-S transmits an IP multicast frame to an IPMG, it will use the label specified in that IPMG's (C-*,C-*) S-PMSI A-D route.

When a PE originates both an IMET route and a (C-*,C-*) S-PMSI A-D route, the values of the MPLS label field in the respective PTAs must be distinct. Further, each MUST map uniquely (in the context of the originating PE) to the route's BD.

As a result, an IPMG receiving an MPLS-encapsulated IP multicast frame can always tell by the label whether the frame's ingress PE is an OISM PE or a non-OISM PE. When an IPMG receives a VXLAN-encapsulated IP multicast frame it may need to determine the identity of the ingress PE from the outer IP encapsulation; it can then determine whether the ingress PE is an OISM PE or a non-OISM PE by looking the IMET route from that PE.

Suppose an IPMG receives an IP multicast frame from another EVPN-PE in the Tenant Domain, and the IPMG is not the IPMG-DF for the frame's source BD. Then the IPMG performs only the ordinary OISM functions; it does not perform the IPMG-specific functions for that frame. In the remainder of this section, when we discuss the procedures applied by an IPMG when it receives an IP multicast frame, we are presuming that the source BD of the frame is a BD for which the IPMG is the IPMG-DF.

We have two basic cases to consider: (1) a frame's ingress PE is a non-OISM node, and (2) a frame's ingress PE is an OISM node.

5.2.1. Ingress PE is non-OISM

In this case, a non-OISM PE, PE-S, has received an (S,G) multicast frame over an AC that is attached to a particular BD, BD-S. By virtue of normal EVPN procedures, PE-S has sent a copy of the frame to every PE-R (both OISM and non-OISM) in the Tenant Domain that is attached to BD-S. If the non-OISM node supports [IGMP-Proxy], only PEs that have expressed interest in (S,G) receive the frame. The IPMG will have expressed interest via a (C-*,C-*) SMET route and thus receives the frame.

Any OISM PE (including an IPMG) receiving the frame will apply normal OISM procedures. As a result it will deliver the frame to any of its local ACs (in BD-S or in any other BD) that have interest in (S,G).

An OISM PE that is also the IPMG-DF for a particular BD, say BD-S, has additional procedures that it applies to frames received on BD-S from non-OISM PEs:

1. When the IPMG-DF for BD-S receives an (S,G) frame from a non-OISM node, it MUST forward a copy of the frame to every OISM PE that is NOT attached to BD-S but has interest in (S,G). The copy sent to a given OISM PE-R must carry the label that PE-R has assigned to the SBD in an S-PMSI A-D route. The IPMG MUST NOT do any IP processing of the frame’s IP payload. TTL decrement and other IP processing will be done by PE-R, per the normal OISM procedures. There is no need for the IPMG to include an ESI label in the frame’s tunnel encapsulation, because it is already known that the frame’s source BD has no presence on PE-R. There is also no need for the IPMG to modify the frame’s MAC SA.

2. In addition, when the IPMG-DF for BD-S receives an (S,G) frame from a non-OISM node, it may need to forward copies of the frame to other non-OISM nodes. Before it does so, it MUST decapsulate the (S,G) packet, and do the IP processing (e.g., TTL decrement). Suppose PE-R is a non-OISM node that has an AC to BD-R, where BD-R is not the same as BD-S, and that AC has interest in (S,G). The IPMG must then encapsulate the (S,G) packet (after the IP processing has been done) in an ethernet header. The MAC SA field will have the MAC address of the IPMG’s IRB interface to BD-R. The IPMG then sends the frame to PE-R. The tunnel encapsulation will carry the label that PE-R advertised in its IMET route for BD-R. There is no need to include an ESI label, as the source and destination BDs are known to be different.

Note that if a non-OISM PE-R has several BDs (other than BD-S) with local ACs that have interest in (S,G), the IPMG will send it one copy for each such BD. This is necessary because the non-OISM PE cannot move packets from one BD to another.

There may be deployment scenarios in which every OISM PE is configured with every BD that is present on any non-OISM PE. In such scenarios, the procedures of item 1 above will not actually result in the transmission of any packets. Hence if it is known a priori that this deployment scenario exists for a given tenant domain, the procedures of item 1 above can be disabled.

5.2.2. Ingress PE is OISM

In this case, an OISM PE, PE-S, has received an (S,G) multicast frame over an AC that attaches to a particular BD, BD-S.

By virtue of receiving all the IMET routes about BD-S, PE-S will know all the PEs attached to BD-S. By virtue of normal OISM procedures:

- PE-S will send a copy of the frame to every OISM PE-R (including the IPMG) in the Tenant Domain that is attached to BD-S and has interest in (S,G). The copy sent to a given PE-R carries the label that that the PE-R has assigned to BD-S in its (C-*,C-*) S-PMSI A-D route.

- PE-S will also transmit a copy of the (S,G) frame to every OISM PE-R that has interest in (S,G) but is not attached to BD-S. The copy will contain the label that the PE-R has assigned to the SBD. (As in Section 5.2.1, an IPMG is assumed to have indicated interest in all multicast flows.)

- PE-S will also transmit a copy of the (S,G) frame to every non-OISM PE-R that is attached to BD-S. It does this using the label advertised by that PE-R in its IMET route for BD-S.

The PE-Rs follow their normal procedures. An OISM PE that receives the (S,G) frame on BD-S applies the OISM procedures to deliver the frame to its local ACs, as necessary. A non-OISM PE that receives the (S,G) frame on BD-S delivers the frame only to its local BD-S ACs, as necessary.

Suppose that a non-OISM PE-R has interest in (S,G) on a BD, BD-R, that is different than BD-S. If the non-OISM PE-R is attached to BD-S, the OISM PE-S will send forward it the original (S,G) multicast frame, but the non-OISM PE-R will not be able to send the frame to ACs that are not in BD-S. If PE-R is not even attached to BD-S, the OISM PE-S will not send it a copy of the frame at all, because PE-R

is not attached to the SBD. In these cases, the IPMG needs to relay the (S,G) multicast traffic from OISM PE-S to non-OISM PE-R.

When the IPMG-DF for BD-S receives an (S,G) frame from an OISM PE-S, it has to forward it to every non-OISM PE-R that that has interest in (S,G) on a BD-R that is different than BD-S. The IPMG MUST decapsulate the IP multicast packet, do the IP processing, re-encapsulate it for BD-R (changing the MAC SA to the IPMG's own MAC address on BD-R), and send a copy of the frame to PE-R. Note that a given non-OISM PE-R will receive multiple copies of the frame, if it has multiple BDs on which there is interest in the frame.

5.3. P2MP Tunnels

When IR is used to distribute the multicast traffic among the EVPN-PEs, the procedures of Section 5.2 ensure that there will be no duplicate delivery of multicast traffic. That is, no egress PE will ever send a frame twice on any given AC. If P2MP tunnels are being used to distribute the multicast traffic, it is necessary have additional procedures to prevent duplicate delivery.

At the present time, it is not clear that there will be a use case in which OISM nodes need to interwork with non-OISM nodes that use P2MP tunnels. If it is determined that there is such a use case, procedures for it will be included in a future revision of this document.

6. Traffic to/from Outside the EVPN Tenant Domain

In this section, we discuss scenarios where a multicast source outside a given EVPN Tenant Domain sends traffic to receivers inside the domain (as well as, possibly, to receivers outside the domain). This requires the OISM procedures to interwork with various layer 3 multicast routing procedures.

We assume in this section that the Tenant Domain is not being used as an intermediate transit network for multicast traffic; that is, we do not consider the case where the Tenant Domain contains multicast routers that will receive traffic from sources outside the domain and forward the traffic to receivers outside the domain. The transit scenario is considered in Section 7.

We can divide the non-transit scenarios into two classes:

1. One or more of the EVPN PE routers provide the functionality needed to interwork with layer 3 multicast routing procedures.

2. One BD in the Tenant Domain contains external multicast routers ("tenant multicast routers") that are used to interwork the entire Tenant Domain with layer 3 multicast routing procedures.

6.1. Layer 3 Interworking via EVPN OISM PEs

6.1.1. General Principles

Sometimes it is necessary to interwork an EVPN Tenant Domain with an external layer 3 multicast domain (the "external domain"). This is needed to allow EVPN tenant systems to receive multicast traffic from sources ("external sources") outside the EVPN Tenant Domain. It is also needed to allow receivers ("external receivers") outside the EVPN Tenant Domain to receive traffic from sources inside the Tenant Domain.

In order to allow interworking between an EVPN Tenant Domain and an external domain, one or more OISM PEs must be "L3 Gateways". An L3 Gateway participates both in the OISM procedures and in the L3 multicast routing procedures of the external domain.

An L3 Gateway that has interest in receiving (S,G) traffic must be able to determine the best route to S. If an L3 Gateway has interest in (*,G), it must be able to determine the best route to G's RP. In these interworking scenarios, the L3 Gateway must be running a layer 3 unicast routing protocol. Via this protocol, it imports unicast routes (either IP routes or VPN-IP routes) from routers other than EVPN PEs. And since there may be multicast sources inside the EVPN Tenant Domain, the EVPN PEs also need to export, either as IP routes or as VPN-IP routes (depending upon the external domain), unicast routes to those sources.

When selecting the best route to a multicast source or RP, an L3 Gateway might have a choice between an EVPN route and an IP/VPN-IP route. When such a choice exists, the L3 Gateway SHOULD always prefer the EVPN route. This will ensure that when traffic originates in the Tenant Domain and has a receiver in the tenant domain, the path to that receiver will remain within the EVPN tenant domain, even if the source is also reachable via a routed path. This also provides protection against sub-optimal routing that might occur if two EVPN PEs export IP/VPN-IP routes and each imports the other's IP/VPN-IP routes.

Section 4.2 discusses the way layer 3 multicast states are constructed by OISM PEs. These layer 3 multicast states have IRB interfaces as their IIF and OIF list entries, and are the basis for interworking OISM with other layer 3 multicast procedures such as MVPN or PIM. From the perspective of the layer 3 multicast

procedures running in a given L3 Gateway, an EVPN Tenant Domain is a set of IRB interfaces.

When interworking an EVPN Tenant Domain with an external domain, the L3 Gateway’s layer 3 multicast states will not only have IRB interfaces as IIF and OIF list entries, but also other "interfaces" that lead outside the Tenant Domain. For example, when interworking with MVPN, the multicast states may have MVPN tunnels as well as IRB interfaces as IIF or OIF list members. When interworking with PIM, the multicast states may have PIM-enabled non-IRB interfaces as IIF or OIF list members.

As long as a Tenant Domain is not being used as an intermediate transit network for IP multicast traffic, it is not necessary to enable PIM on its IRB interfaces.

In general, an L3 Gateway has the following responsibilities:

- It exports, to the external domain, unicast routes to those multicast sources in the EVPN Tenant Domain that are locally attached to the L3 Gateway.

- It imports, from the external domain, unicast routes to multicast sources that are in the external domain.

- It executes the procedures necessary to draw externally sourced multicast traffic that is of interest to locally attached receivers in the EVPN Tenant Domain. When such traffic is received, the traffic is sent down the IRB interfaces of the BDs on which the locally attached receivers reside.

One of the L3 Gateways in a given Tenant Domain becomes the "DR" for the SBD.(See Section 6.1.2.4.) This L3 gateway has the following additional responsibilities:

- It exports, to the external domain, unicast routes to multicast sources that in the EVPN Tenant Domain that are not locally attached to any L3 gateway.

- It imports, from the external domain, unicast routes to multicast sources that are in the external domain.

- It executes the procedures necessary to draw externally sourced multicast traffic that is of interest to receivers in the EVPN Tenant Domain that are not locally attached to an L3 gateway. When such traffic is received, the traffic is sent down the SBD IRB interface. OISM procedures already described in this document will then ensure that the IP multicast traffic gets distributed

throughout the Tenant Domain to any EVPN PEs that have interest in it. Thus to an OISM PE that is not an L3 gateway the externally sourced traffic will appear to have been sourced on the SBD.

In order for this to work, some special care is needed when an L3 gateway creates or modifies a layer 3 (*,G) multicast state. Suppose group G has both external sources (sources outside the EVPN Tenant Domain) and internal sources (sources inside the EVPN tenant domain). Section 4.2 states that when there are internal sources, the SBD IRB interface must not be added to the OIF list of the (*,G) state. Traffic from internal sources will already have been delivered to all the EVPN PEs that have interest in it. However, if the OIF list of the (*,G) state does not contain its SBD IRB interface, then traffic from external sources will not get delivered to other EVPN PEs.

One way of handling this is the following. When a L3 gateway receives (S,G) traffic from other than an IRB interface, and the traffic corresponds to a layer 3 (*,G) state, the L3 gateway can create (S,G) state. The IIF will be set to the external interface over which the traffic is expected. The OIF list will contain the SBD IRB interface, as well as the IRB interfaces of any other BDs attached to the PEG DR that have locally attached receivers with interest in the (S,G) traffic. The (S,G) state will ensure that the external traffic is sent down the SBD IRB interface. The following text will assume this procedure; however other implementation techniques may also be possible.

If a particular BD is attached to several L3 Gateways, one of the L3 Gateways becomes the DR for that BD. (See Section 6.1.2.4.) If the interworking scenario requires FHR functionality, it is generally the DR for a particular BD that is responsible for performing that functionality on behalf of the source hosts on that BD. (E.g., if the interworking scenario requires that PIM Register messages be sent by a FHR, the DR for a given BD would send the PIM Register messages for sources on that BD.) Note though that the DR for the SBD does not perform FHR functionality on behalf of external sources.

An optional alternative is to have each L3 gateway perform FHR functionality for locally attached sources. Then the DR would only have to perform FHR functionality on behalf of sources that are locally attached to itself AND sources that are not attached to any L3 gateway.

6.1.2. Interworking with MVPN

In this section, we specify the procedures necessary to allow EVPN PEs running OISM procedures to interwork with L3VPN PEs that run BGP-based MVPN ([RFC6514]) procedures. More specifically, the procedures

herein allow a given EVPN Tenant Domain to become part of an L3VPN/MVPN, and support multicast flows where either:

- The source of a given multicast flow is attached to an ethernet segment whose BD is part of an EVPN Tenant Domain, and one or more receivers of the flow are attached to the network via L3VPN/MVPN. (Other receivers may be attached to the network via EVPN.)

- The source of a given multicast flow is attached to the network via L3VPN/MVPN, and one or more receivers of the flow are attached to an ethernet segment that is part of an EVPN tenant domain. (Other receivers may be attached via L3VPN/MVPN.)

In this interworking model, existing L3VPN/MVPN PEs are unaware that certain sources or receivers are part of an EVPN Tenant Domain. The existing L3VPN/MVPN nodes run only their standard procedures and are entirely unaware of EVPN. Interworking is achieved by having some or all of the EVPN PEs function as L3 Gateways running L3VPN/MVPN procedures, as detailed in the following sub-sections.

In this section, we assume that there are no tenant multicast routers on any of the EVPN-attached ethernet segments. (There may of course be multicast routers in the L3VPN.) Consideration of the case where there are tenant multicast routers is deferred till Section 7.)

To support MVPN/EVPN interworking, we introduce the notion of an MVPN/EVPN Gateway, or MEG.

A MEG is an L3 Gateway (see Section 6.1.1), hence is both an OISM PE and an L3VPN/MVPN PE. For a given EVPN Tenant Domain it will have an IP-VRF. If the Tenant Domain is part of an L3VPN/MVPN, the IP-VRF also serves as an L3VPN VRF ([RFC4364]). The IRB interfaces of the IP-VRF are considered to be "VRF interfaces" of the L3VPN VRF. The L3VPN VRF may also have other local VRF interfaces that are not EVPN IRB interfaces.

The VRF on the MEG will import VPN-IP routes ([RFC4364]) from other L3VPN Provider Edge (PE) routers. It will also export VPN-IP routes to other L3VPN PE routers. In order to do so, it must be appropriately configured with the Route Targets used in the L3VPN to control the distribution of the VPN-IP routes. These Route Targets will in general be different than the Route Targets used for controlling the distribution of EVPN routes, as there is no need to distribute EVPN routes to L3VPN-only PEs and no reason to distribute L3VPN/MVPN routes to EVPN-only PEs.

Note that the RDs in the imported VPN-IP routes will not necessarily conform to the EVPN rules (as specified in [RFC7432]) for creating

RDs. Therefore a MEG MUST NOT expect the RDs of the VPN-IP routes to be of any particular format other than what is required by the L3VPN/ MVPN specifications.

The VPN-IP routes that a MEG exports to L3VPN are subnet routes and/ or host routes for the multicast sources that are part of the EVPN tenant domain. The exact set of routes that need to be exported is discussed in Section 6.1.2.2.

Each IMET route originated by a MEG SHOULD carry a flag or Extended Community (to be determined) indicating that the originator of the IMET route is a MEG. However, PE1 will consider PE2 to be a MEG if PE1 imports at least one IMET route from PE2 that carries the flag or EC.

All the MEGs of a given Tenant Domain attach to the SBD of that domain, and one of them is selected to be the SBD's Designated Router (DR) for the domain. The selection procedure is discussed in Section 6.1.2.4.

In this model of operation, MVPN procedures and EVPN procedures are largely independent. In particular, there is no assumption that MVPN and EVPN use the same kind of tunnels. Thus no special procedures are needed to handle the common scenarios where, e.g., EVPN uses VXLAN tunnels but MVPN uses MPLS P2MP tunnels, or where EVPN uses Ingress Replication but MVPN uses MPLS P2MP tunnels.

Similarly, no special procedures are needed to prevent duplicate data delivery on ethernet segments that are multi-homed.

The MEG does have some special procedures (described below) for interworking between EVPN and MVPN; these have to do with selection of the Upstream PE for a given multicast source, with the exporting of VPN-IP routes, and with the generation of MVPN C-multicast routes triggered by the installation of SMET routes.

6.1.2.1. MVPN Sources with EVPN Receivers

6.1.2.1.1. Identifying MVPN Sources

Consider a multicast source S. It is possible that a MEG will import both an EVPN unicast route to S and a VPN-IP route (or an ordinary IP route), where the prefix length of each route is the same. In order to draw (S,G) multicast traffic for any group G, the MEG SHOULD use the EVPN route rather than the VPN-IP or IP route to determine the "Upstream PE" (see section 5 of [RFC6513]).

Doing so ensures that when an EVPN tenant system desires to receive a multicast flow from another EVPN tenant system, the traffic from the source to that receiver stays within the EVPN domain. This prevents problems that might arise if there is a unicast route via L3VPN to S, but no multicast routers along the routed path. This also prevents problem that might arise as a result of the fact that the MEGs will import each others' VPN-IP routes.

In the Section 6.1.2.1.2, we describe the procedures to be used when the selected route to S is a VPN-IP route.

6.1.2.1.2. Joining a Flow from an MVPN Source

Suppose a tenant system R wants to receive (S,G) multicast traffic, where source S is not attached to any PE in the EVPN Tenant Domain, but is attached to an MVPN PE.

- Suppose R is on a singly homed ethernet segment of BD-R, and that segment is attached to PE1, where PE1 is a MEG. PE1 learns via IGMP/MLD listening that R is interested in (S,G). PE1 determines from its VRF that there is no route to S within the Tenant Domain (i.e., no EVPN RT-2 route with S's IP address), but that there is a route to S via L3VPN (i.e., the VRF contains a subnet or host route to S that was received as a VPN-IP route). PE1 thus originates (if it hasn't already) an MVPN C-multicast Source Tree Join(S,G) route. The route is constructed according to normal MVPN procedures.

  The layer 2 multicast state is constructed as specified in Section 4.1.

  In the layer 3 multicast state, the IIF is the appropriate MVPN tunnel, and the IRB interface to BD-R is added to the OIF list.

  When PE1 receives (S,G) traffic from the appropriate MVPN tunnel, it performs IP processing of the traffic, and then sends the traffic down its IRB interface to BD-R. Following normal OISM procedures, the (S,G) traffic will be encapsulated for ethernet and sent out the AC to which R is attached.

- Suppose R is on a singly homed ethernet segment of BD-R, and that segment is attached to PE1, where PE1 is an OISM PE but is NOT a MEG. PE1 learns via IGMP/MLD listening that R is interested in (S,G). PE1 follows normal OISM procedures, originating an SMET route in BD-R for (S,G). Since this route will carry the SBD-RT, it will be received by the MEG that is the DR for the Tenant Domain. The MEG DR can determine from PE1's IMET route whether PE1 is itself a MEG. If PE1 is not a MEG, the MEG DR will

originate (if it hasn't already) an MVPN C-multicast Source Tree Join(S,G) route. This will cause the DR MEG to receive (S,G) traffic on an MVPN tunnel.

The layer 2 multicast state is constructed as specified in Section 4.1.

In the layer 3 multicast state, the IIF is the appropriate MVPN tunnel, and the IRB interface to the SBD is added to the OIF list.

When the DR MEG receives (S,G) traffic on an MVPN tunnel, it performs IP processing of the traffic, and the sends the traffic down its IRB interface to the SBD. Following normal OISM procedures, the traffic will be encapsulated for ethernet and delivered to all PEs in the Tenant Domain that have interest in (S,G), including PE1.

- o If R is on a multi-homed ethernet segment of BD-R, one of the PEs attached to the segment will be its DF (following normal EVPN procedures), and the DF will know (via the procedures of [IGMP-Proxy] that a tenant system reachable via one of its local ACs to BD-R is interested in (S,G) traffic. The DF is responsible for originating an SMET route for (S,G), following normal OISM procedures. If the DF is a MEG, it will originate the corresponding MVPN C-multicast Source Tree Join(S,G) route; if the DF is not a MEG, the MEG that is the DR will originate the C-multicast route when it receives the SMET route.

- o If R is attached to a non-OISM PE, it will receive the traffic via an IPMG, as specified in Section 5.

If an EVPN-attached receiver is interested in (*,G) traffic, and if it is possible for there to be sources of (*,G) traffic that are attached only to L3VPN nodes, the MEGs will have to know the group-to-RP mappings. That will enable them to originate MVPN C-multicast Shared Tree Join(*,G) routes and to send them towards the RP. (Since we are assuming in this section that there are no tenant multicast routers attached to the EVPN Tenant Domain, the RP must be attached via L3VPN. Alternatively, the MEG itself could be configured to function as an RP for group G.)

The layer 2 multicast states are constructed as specified in Section 4.1.

In the layer 3 (*,G) multicast state, the IIF is the appropriate MVPN tunnel. A MEG will add to the (*,G) OIF list its IRB interfaces for any BDs containing locally attached receivers. If there are receivers attached to other EVPN PEs, then whenever (S,G) traffic

from an external source matches a (*,G) state, the MEG will create (S,G) state, with the MVPN tunnel as the IIF, the OIF list copied from the (*,G) state, and the SBD IRB interface added to the OIF list. (Please see the discussion in Section 6.1.1 regarding the inclusion of the SBD IRB interface in a (*,G) state; the SBD IRB interface is used in the OIF list only for traffic from external sources.)

Normal MVPN procedures will then result in the MEG getting the (*,G) traffic from all the multicast sources for G that are attached via L3VPN. This traffic arrives on MVPN tunnels. When the MEG removes the traffic from these tunnels, it does the IP processing. If there are any receivers on a given BD, BD-R, that are attached via local EVPN ACs, the MEG sends the traffic down its BD-R IRB interface. If there are any other EVPN PEs that are interested in the (*,G) traffic, the MEG sends the traffic down the SBD IRB interface. Normal OISM procedures then distribute the traffic as needed to other EVPN-PEs.

6.1.2.2. EVPN Sources with MVPN Receivers

6.1.2.2.1. General procedures

Consider the case where an EVPN tenant system S is sending IP multicast traffic to group G, and there is a receiver R for the (S,G) traffic that is attached to the L3VPN, but not attached to the EVPN Tenant Domain. (We assume in this document that the L3VPN/MVPN-only nodes will not have any special procedures to deal with the case where a source is inside an EVPN domain.)

In this case, an L3VPN PE through which R can be reached has to send an MVPN C-multicast Join(S,G) route to one of the MEGs that is attached to the EVPN Tenant Domain. For this to happen, the L3VPN PE must have imported a VPN-IP route for S (either a host route or a subnet route) from a MEG.

If a MEG determines that there is multicast source transmitting on one of its ACs, the MEG SHOULD originate a VPN-IP host route for that source. This determination SHOULD be made by examining the IP multicast traffic that arrives on the ACs. (It MAY be made by provisioning.) A MEG SHOULD NOT export a VPN-IP host route for any IP address that is not known to be a multicast source (unless it has some other reason for exporting such a route). The VPN-IP host route for a given multicast source MUST be withdrawn if the source goes silent for a configurable period of time, or if it can be determined that the source is no longer reachable via a local AC.

A MEG SHOULD also originate a VPN-IP subnet route for each of the BDs in the Tenant Domain.

VPN-IP routes exported by a MEG must carry any attributes or extended communities that are required by L3VPN and MVPN. In particular, a VPN-IP route exported by a MEG must carry a VRF Route Import Extended Community corresponding to the IP-VRF from which it is imported, and a Source AS Extended Community.

As a result, if S is attached to a MEG, the L3VPN nodes will direct their MVPN C-multicast Join routes to that MEG. Normal MVPN procedures will cause the traffic to be delivered to the L3VPN nodes. The layer 3 multicast state for (S,G) will have the MVPN tunnel on its OIF list. The IIF will be the IRB interface leading to the BD containing S.

If S is not attached to a MEG, the L3VPN nodes will direct their C-multicast Join routes to whichever MEG appears to be on the best route to S’s subnet. Upon receiving the C-multicast Join, that MEG will originate an EVPN SMET route for (S,G). As a result, the MEG will receive the (S,G) traffic at layer 2 via the OISM procedures. The (S,G) traffic will be sent up the appropriate IRB interface, and the layer 3 MVPN procedures will ensure that the traffic is delivered to the L3VPN nodes that have requested it. The layer 3 multicast state for (S,G) will have the MVPN tunnel in the OIF list, and the IIF will be one of the following:

- If S belongs to a BD that is attached to the MEG, the IIF will be the IRB interface to that BD;
- Otherwise the IIF will be the SBD IRB interface.

Note that this works even if S is attached to a non-OISM PE, per the procedures of Section 5.

6.1.2.2.2. Any-Source Multicast (ASM) Groups

Suppose the MEG DR learns that one of the PEs in its Tenant Domain is interested in (*,G), traffic, where G is an Any-Source Multicast (ASM) group. If there are no tenant multicast routers, the MEG DR SHOULD perform the "First Hop Router" (FHR) functionality for group G on behalf of the Tenant Domain, as described in [RFC7761]. This means that the MEG DR must know the identity of the Rendezvous Point (RP) for each group, must send Register messages to the Rendezvous Point, etc.

If the MEG DR is to be the FHR for the Tenant Domain, it must see all the multicast traffic that is sourced from within the domain and

destined to an ASM group address. The MEG can ensure this by originating an SBD-SMET route for (*,*). As an optimization, an SBD-SMET route for (*, "any ASM group"), or even (*, "any ASM group that might have MVPN sources") can be defined.

In some deployment scenarios, it may be preferred that the MEG that receives the (S,G) traffic over an AC be the one provides the FHR functionality. In that case, the MEG DR wold not need to provide the FHR functionality for (S,G) traffic that is attached to another MEG.

Other deployment scenarios are also possible. For example, one might want to configure the MEGs to themselves be RPs. In this case, the RPs would have to exchange with each other information about which sources are active. The method exchanging such information is outside the scope of this document.

6.1.2.2.3. Source on Multihomed Segment

Suppose S is attached to a segment that is all-active multi-homed to PEl and PE2. If S is transmitting to two groups, say G1 and G2, it is possible that PE1 will receive the (S,G1) traffic from S while PE2 receives the (S,G2) traffic from S.

This creates an issue for MVPN/EVPN interworking, because there is no way to cause L3VPN/MVPN nodes to select PE1 as the ingress PE for (S,G1) traffic while selecting PE2 as the ingress PE for (S,G2) traffic.

However, the following procedure ensures that the IP multicast traffic will still flow, even if the L3VPN/MVPN nodes picks the "wrong" EVPN-PE as the Upstream PE for (say) the (S,G1) traffic.

Suppose S is on an ethernet segment, belonging to BD1, that is multi-homed to both PE1 and PE2, where PE1 is a MEG. And suppose that IP multicast traffic from S to G travels over the AC that attaches the segment to PE2 . If PE1 receives a C-multicast Source Tree Join (S,G) route, it MUST originate an SMET route for (S,G). Normal OISM procedures will then cause PE2 to send the (S,G) traffic to PE1 on an EVPN IP multicast tunnel. Normal OISM procedures will also cause PE1 to send the (S,G) traffic up its BD1 IRB interface. Normal MVPN procedures will then cause PE1 to forward the traffic on an MVPN tunnel. In this case, the routing is not optimal, but the traffic does flow correctly.

6.1.2.3. Obtaining Optimal Routing of Traffic Between MVPN and EVPN

The routing of IP multicast traffic between MVPN nodes and EVPN nodes will be optimal as long as there is a MEG along the optimal route. There are various deployment strategies that can be used to obtain optimal routing between MVPN and EVPN.

In one such scenario, a Tenant Domain will have a small number of strategically placed MEGs. For example, a Data Center may have a small number of MEGs that connect it to a wide-area network. Then the optimal route into or out of the Data Center would be through the MEGs.

In this scenario, the MEGs do not need to originate VPN-IP host routes for the multicast sources, they only need to originate VPN-IP subnet routes. The internal structure of the EVPN is completely hidden from the MVPN node. EVPN actions such as MAC Mobility and Mass Withdrawal ([RFC7432]) have zero impact on the MVPN control plane.

While this deployment scenario provides the most optimal routing and has the least impact on the installed based of MVPN nodes, it does complicate network planning considerations.

Another way of providing routing that is close to optimal is to turn each EVPN PE into a MEG. Then routing of MVPN-to-EVPN traffic is optimal. However, routing of EVPN-to-MVPN traffic is not guaranteed to be optimal when a source host is on a multi-homed ethernet segment (as discussed in Section 6.1.2.2.)

The obvious disadvantage of this method is that it requires every EVPN PE to be a MEG.

The procedures specified in this document allow an operator to add MEG functionality to any subset of his EVPN OISM PEs. This allows an operator to make whatever trade-offs he deems appropriate between optimal routing and MEG deployment.

6.1.2.4. DR Selection

Each MEG MUST be configured with an "MEG dummy ethernet segment" that has no ACs.

EVPN supports a number of procedures that can be used to select the Designated Forwarder (DF) for a particular BD on a particular ethernet segment. Some of the possible procedures can be found, e.g., in [RFC7432], [EVPN-DF-NEW], and [EVPN-DF-WEIGHTED]. Whatever

procedure is in use in a given deployment can be adapted to select a MEG DR for a given BD, as follows.

Each MEG will originate an Ethernet Segment route for the MEG dummy ethernet segment. It MUST carry a Route Target derived from the corresponding Ethernet Segment Identifier. Thus only MEGs will import the route.

Once the set of MEGs is known, it is also possible to determine the set of BDs supported by each MEG. The DF selection procedure can then be used to choose a MEG DR for the SBD. (The conditions under which the MEG DR changes depends upon the DF selection algorithm that is in use.)

These procedures can also be used to select a DR for each BD.

6.1.3. Interworking with 'Global Table Multicast'

If multicast service to the outside sources and/or receivers is provided via the BGP-based "Global Table Multicast" (GTM) procedures of [RFC7716], the procedures of Section 6.1.2 can easily be adapted for EVPN/GTM interworking. The way to adapt the MVPN procedures to GTM is explained in [RFC7716].

6.1.4. Interworking with PIM

As we have been discussing, there may be receivers in an EVPN tenant domain that are interested in multicast flows whose sources are outside the EVPN Tenant Domain. Or there may be receivers outside an EVPN Tenant Domain that are interested in multicast flows whose sources are inside the Tenant Domain.

If the outside sources and/or receivers are part of an MVPN, interworking procedures are covered in Section 6.1.2.

There are also cases where an external source or receiver are attached via IP, and the layer 3 multicast routing is done via PIM. In this case, the interworking between the "PIM domain" and the EVPN tenant domain is done at L3 Gateways that perform "PIM/EVPN Gateway" (PEG) functionality. A PEG is very similar to a MEG, except that its layer 3 multicast routing is done via PIM rather than via BGP.

If external sources or receivers for a given group are attached to a PEG via a layer 3 interface, that interface should be treated as a VRF interface attached to the Tenant Domain's L3VPN VRF. The layer 3 multicast routing instance for that Tenant Domain will either run PIM on the VRF interface or will listen for IGMP/MLD messages on that interface. If the external receiver is attached elsewhere on an IP

network, the PE has to enable PIM on its interfaces to the backbone network. In both cases, the PE needs to perform PEG functionality, and its IMET routes must carry a flag or EC identifying it as a PEG.

For each BD on which there is a multicast source or receiver, one of the PEGs will becomes the PEG DR. DR selection can be done using the same procedures specified in Section 6.1.2.4.

As long as there are no tenant multicast routers within the EVPN Tenant Domain, the PEGs do not need to run PIM on their IRB interfaces.

6.1.4.1. Source Inside EVPN Domain

If a PEG receives a PIM Join(S,G) from outside the EVPN tenant domain, it may find it necessary to create (S,G) state. The PE needs to determine whether S is within the Tenant Domain. If S is not within the EVPN Tenant Domain, the PE carries out normal layer 3 multicast routing procedures. If S is within the EVPN tenant domain, the IIF of the (S,G) state is set as follows:

- if S is on a BD that is attached to the PE, the IIF is the PE's IRB interface to that BD;
- if S is not on a BD that is attached to the PE, the IIF is the PE's IRB interface to the SBD.

When the PE creates such an (S,G) state, it MUST originate (if it hasn't already) an SBD-SMET route for (S,G). This will cause it to pull the (S,G) traffic via layer 2. When the traffic arrives over an EVPN tunnel, it gets sent up an IRB interface where the layer 3 multicast routing determines the packet's disposition. The SBD-SMET route is withdrawn when the (S,G) state no longer exists (unless there is some other reason for not withdrawing it).

If there are no tenant multicast routers with the EVPN tenant domain, there cannot be an RP in the Tenant Domain, so a PEG does not have to handle externally arriving PIM Join(*,G) messages.

The PEG DR for a particular BD MUST act as the a First Hop Router for that BD. It will examine all (S,G) traffic on the BD, and whenever G is an ASM group, the PEG DR will send Register messages to the RP for G. This means that the PEG DR will need to pull all the (S,G) traffic originating on a given BD, by originating an SMET (*,*) route for that BD. If a PEG DR is the DR for all the BDS, in SHOULD originate just an SBD-SMET (*,*) route rather than an SMET (*,*) route for each BD.

The rules for exporting IP routes to multicast sources are the same as those specified for MEGs in Section 6.1.2.2, except that the exported routes will be IP routes rather than VPN-IP routes, and it is not necessary to attach the VRF Route Import EC or the Source AS EC.

When a source is on a multi-homed segment, the same issue discussed in Section 6.1.2.2.3 exists. Suppose S is on an ethernet segment, belonging to BD1, that is multi-homed to both PE1 and PE2, where PE1 is a PEG. And suppose that IP multicast traffic from S to G travels over the AC that attaches the segment to PE2. If PE1 receives an external PIM Join (S,G) route, it MUST originate an SMET route for (S,G). Normal OISM procedures will cause PE2 to send the (S,G) traffic to PE1 on an EVPN IP multicast tunnel. Normal OISM procedures will also cause PE1 to send the (S,G) traffic up its BD1 IRB interface. Normal PIM procedures will then cause PE1 to forward the traffic along a PIM tree. In this case, the routing is not optimal, but the traffic does flow correctly.

6.1.4.2. Source Outside EVPN Domain

By means of normal OISM procedures, a PEG learns whether there are receivers in the Tenant Domain that are interested in receiving (*,G) or (S,G) traffic. The PEG must determine whether S (or the RP for G) is outside the EVPN Tenant Domain. If so, and if there is a receiver on BD1 interested in receiving such traffic, the PEG DR for BD1 is responsible for originating a PIM Join(S,G) or Join(*,G) control message.

An alternative would be to allow any PEG that is directly attached to a receiver to originate the PIM Joins. Then the PEG DR would only have to originate PIM Joins on behalf of receivers that are not attached to a PEG. However, if this is done, it is necessary for the PEGs to run PIM on all their IRB interfaces, so that the PIM Assert procedures can be used to prevent duplicate delivery to a given BD.

The IIF for the layer 3 (S,G) or (*,G) state is determined by normal PIM procedures. If a receiver is on BD1, and the PEG DR is attached to BD1, its IRB interface to BD1 is added to the OIF list. This ensures that any receivers locally attached to the PEG DR will receive the traffic. If there are receivers attached to other EVPN PEs, then whenever (S,G) traffic from an external source matches a (*,G) state, the PEG will create (S,G) state. The IIF will be set to whatever external interface the traffic is expected to arrive on (copied from the (*,G) state), the OIF list is copied from the (*,G) state, and the SBD IRB interface added to the OIF list.

6.2. Interworking with PIM via an External PIM Router

Section 6.1 describes how to use an OISM PE router as the gateway to a non-EVPN multicast domain, when the EVPN tenant domain is not being used as an intermediate transit network for multicast. An alternative approach is to have one or more external PIM routers (perhaps operated by a tenant) on one of the BDs of the tenant domain. We will refer to this BD as the "gateway BD".

In this model:

- The EVPN Tenant Domain is treated as a stub network attached to the external PIM routers.
- The external PIM routers follow normal PIM procedures, and provide the FHR and LHR functionality for the entire Tenant Domain.
- The OISM PEs do not run PIM.
- If an OISM PE not attached to the gateway BD has interest in a given multicast flow, it conveys that interest to the OISM PEs that are attached to the gateway BD. This is done by following normal OISM procedures. As a result, IGMP/MLD messages will seen by the external PIM routers on the gateway BD, and those external PIM routers will send PIM Join messages externally as required. Traffic of the given multicast flow will then be received by one of the external PIM routers, and that traffic will be forwarded by that router to the gateway BD.

  The normal OISM procedures will then cause the given multicast flow to be tunneled to any PEs of the EVPN Tenant Domain that have interest in the flow. PEs attached to the gateway BD will see the flow as originating from the gateway BD, other PEs will see the flow as originating from the SBD.
- An OISM PE attached to a gateway BD MUST set its layer 2 multicast state to indicate that each AC to the gateway BD has interest in all multicast flows. It MUST also originate an SMET route for (*,*). The procedures for originating SMET routes are discussed in Section 2.5.
- This will cause the OISM PEs attached to the gateway BD to receive all the IP multicast traffic that is sourced within the EVPN tenant domain, and to transmit that traffic to the gateway BD, where the external PIM routers will see it. (Of course, if the gateway BD has a multi-homed segment, only the PE that is the DF for that segment will transmit the multicast traffic to the segment.)

7. Using an EVPN Tenant Domain as an Intermediate (Transit) Network for Multicast traffic

In this section, we consider the scenario where one or more BDs of an EVPN Tenant Domain are being used to carry IP multicast traffic for which the source and at least one receiver are not part the tenant domain. That is, one or more BDs of the Tenant Domain are intermediate "links" of a larger multicast tree created by PIM.

We define a "tenant multicast router" as a multicast router, running PIM, that is:

attached to one or more BDs of the Tenant Domain, but

is not an EVPN PE router.

In order an EVPN Tenant Domain to be used as a transit network for IP multicast, one or more of its BDs must have tenant multicast routers, and an OISM PE that attaching to such a BD MUST be provisioned to enable PIM on its IRB interface to that BD. (This is true even if none of the tenant routers is on a segment attached to the PE.) Further, all the OISM PEs (even ones not attached to a BD with tenant multicast routers) MUST be provisioned to enable PIM on their SBD IRB interfaces.

If PIM is enabled on a particular BD, the DR Selection procedure of Section 6.1.2.4 MUST be replaced by the normal PIM DR Election procedure of [RFC7761]. Note that this may result in one of the tenant routers being selected as the DR, rather than one of the OISM PE routers. In this case, First Hop Router and Last Hop Router functionality will not be performed by any of the EVPN PEs.

A PIM control message on a particular BD is considered to be a link-local multicast message, and as such is sent transparently from PE to PE via the BUM tunnel for that BD. This is true whether the control message was received from an AC, or whether it was received from the local layer 3 routing instance via an IRB interface.

A PIM Join/Prune message contains three fields that are relevant to the present discussion:

- Upstream Neighbor
- Group Address (G)
- Source Address (S), omitted in the case of (*,G) Join/Prune messages.

We will generally speak of a PIM Join as a "Join(S,G)" or a "Join(*,G)" message, and will use the term "Join(X,G)" to mean "either Join(S,G) or Join(*,G)". In the context of a Join(X,G), we will use the term "X" to mean "S in the case of (S,G), or G's RP in the case of (*,G)".

Suppose BD1 contains two tenant multicast routers, C1 and C2. Suppose C1 is on a segment attached to PE1, and C2 is on a segment attached to PE2. When C1 sends a PIM Join(X,G) to BD1, the Upstream Neighbor field might be set to either PE1, PE2, or C2. C1 chooses the Upstream Neighbor based on its unicast routing. Typically, it will choose as the Upstream Neighbor the PIM router on BD1 that is "closest" (according to the unicast routing) to X. Note that this will not necessarily be PE1. PE1 may not even be visible to the unicast routing algorithm used by the tenant routers. Even if it is, it is unlikely to be the PIM router that is closest to X. So we need to consider the following two cases:

C1 sends a PIM Join(X,G) to BD1, with PE1 as the Upstream Neighbor.

PE1's PIM routing instance will see the Join arrive on the BD1 IRB interface. If X is not within the Tenant Domain, PE1 handles the Join according to normal PIM procedures. This will generally result in PE1 selecting an Upstream Neighbor and sending it a Join(X,G).

If X is within the Tenant Domain, but is attached to some other PE, PE1 sends (if it hasn't already) an SBD-SMET route for (X,G). The IIF of the layer 3 (X,G) state will be the SBD IRB interface, and the OIF list will include the IRB interface to BD1.

The SBD-SMET route will pull the (X,G) traffic to PE1, and the (X,G) state will result in the (X,G) traffic being forwarded to C1.

If X is within the Tenant Domain, but is attached to PE1 itself, no SBD-SMET route is sent. The IIF of the layer 3 (X,G) state will be the IRB interface to X's BD, and the OIF list will include the IRB interface to BD1.

C1 sends a PIM Join(X,G) to BD1, with either PE2 or C2 as the Upstream Neighbor.

PE1's PIM routing instance will see the Join arrive on the BD1 IRB interface. If neither X nor Upstream Neighbor is within the

tenant domain, PE1 handles the Join according to normal PIM procedures. This will NOT result in PE1 sending a Join(X,G).

If either X or Upstream Neighbor is within the Tenant Domain, PE1 sends (if it hasn't already) an SBD-SMET route for (X,G). The IIF of the layer 3 (X,G) state will be the SBD IRB interface, and the OIF list will include the IRB interface to BD1.

The SBD-SMET route will pull the (X,G) traffic to PE1, and the (X,G) state will result in the (X,G) traffic being forwarded to C1.

8. IANA Considerations

To be supplied.

9. Security Considerations

This document uses protocols and procedures defined in the normative references, and inherits the security considerations of those references.

This document adds flags or Extended Communities (ECs) to a number of BGP routes, in order to signal that particular nodes support the OISM, IPMG, MEG, and/or PEG functionalities that are defined in this document. Incorrect addition, removal, or modification of those flags and/or ECs will cause the procedures defined herein to malfunction, in which case loss or diversion of data traffic is possible.

10. Acknowledgements

The authors thank Vikram Nagarajan and Princy Elizabeth for their work on Section 6.2. The authors also benefited tremendously from discussions with Aldrin Isaac on EVPN multicast optimizations.

11. References

11.1. Normative References

[EVPN-AR] Rabadan, J., Ed., "Optimized Ingress Replication solution for EVPN", internet-draft ietf-bess-evpn-optimized-ir-02.txt, August 2017.

[EVPN-BUM]
Zhang, Z., Lin, W., Rabadan, J., and K. Patel, "Updates on EVPN BUM Procedures", internet-draft ietf-bess-evpn-bum-procedure-updates-01.txt, December 2016.

[EVPN-IRB]
Sajassi, A., Salam, S., Thoria, S., Drake, J., Rabadan, J., and L. Yong, "Integrated Routing and Bridging in EVPN", internet-draft draft-ietf-bess-evpn-inter-subnet-forwarding-03.txt, February 2017.

[EVPN_IP_Prefix]
Rabadan, J., Henderickx, W., Drake, J., Lin, W., and A. Sajassi, "IP Prefix Advertisement in EVPN", internet-draft ietf-bess-evpn-prefix-advertisement-05.txt, July 2017.

[IGMP-Proxy]
Sajassi, A., Thoria, S., Patel, K., Yeung, D., Drake, J., and W. Lin, "IGMP and MLD Proxy for EVPN", internet-draft draft-ietf-bess-evpn-igmp-mld-proxy-00.txt, March 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>.

[RFC2236] Fenner, W., "Internet Group Management Protocol, Version 2", RFC 2236, DOI 10.17487/RFC2236, November 1997, <https://www.rfc-editor.org/info/rfc2236>.

[RFC2710] Deering, S., Fenner, W., and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, DOI 10.17487/RFC2710, October 1999, <https://www.rfc-editor.org/info/rfc2710>.

[RFC6625] Rosen, E., Ed., Rekhter, Y., Ed., Hendrickx, W., and R. Qiu, "Wildcards in Multicast VPN Auto-Discovery Routes", RFC 6625, DOI 10.17487/RFC6625, May 2012, <https://www.rfc-editor.org/info/rfc6625>.

[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>.

11.2. Informative References

[EVPN-BIER]
Zhang, Z., Przygienda, A., Sajassi, A., and J. Rabadan, "Updates on EVPN BUM Procedures", internet-draft ietf-zzhang-bier-evpn-00.txt, June 2017.

[EVPN-DF-NEW]
Mohanty, S., Patel, K., Sajassi, A., Drake, J., and T. Przygienda, "A new Designated Forwarder Election for the EVPN", internet-draft ietf-bess-evpn-df-election-02.txt, April 2017.

[EVPN-DF-WEIGHTED]
Rabadan, J., Sathappan, S., Przygienda, T., Lin, W., Drake, J., Sajassi, A., and S. Mohanty, "Preference-based EVPN DF Election", internet-draft ietf-bess-evpn-pref-df-00.txt, June 2017.

[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>.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <https://www.rfc-editor.org/info/rfc6513>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <https://www.rfc-editor.org/info/rfc6514>.

[RFC7716] Zhang, J., Giuliano, L., Rosen, E., Ed., Subramanian, K., and D. Pacella, "Global Table Multicast with BGP Multicast VPN (BGP-MVPN) Procedures", RFC 7716, DOI 10.17487/RFC7716, December 2015, <https://www.rfc-editor.org/info/rfc7716>.

[RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <https://www.rfc-editor.org/info/rfc7761>.

Appendix A. Integrated Routing and Bridging

This Appendix provides a short tutorial on the interaction of routing and bridging. First it shows the traditional model, where bridging and routing are performed in separate boxes. Then it shows the model specified in [EVPN-IRB], where a single box contains both routing and bridging functions. The latter model is presupposed in the body of this document.

Figure 1 shows a "traditional" router that only does routing and has no L2 bridging capabilities. There are two LANs, LAN1 and LAN2. LAN1 is realized by switch1, LAN2 by switch2. The router has an interface, "lan1" that attaches to LAN1 (via switch1) and an interface "lan2" that attachs to LAN2 (via switch2). Each intreface is configured, as an IP interface, with an IP address and a subnet mask.

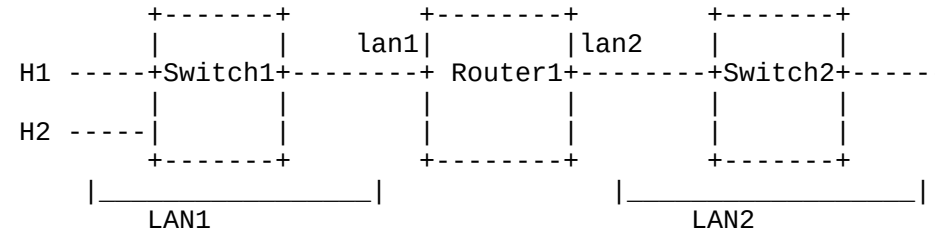


Figure 1: Conventional Router with LAN Interfaces

IP traffic (unicast or multicast) that remains within a single subnet never reaches the router. For instance, if H1 emits an ethernet frame with H2's MAC address in the ethernet destination address field, the frame will go from H1 to Switch1 to H2, without ever reaching the router. Since the frame is never seen by a router, the IP datagram within the frame remains entirely unchanged; e.g., its TTL is not decremented. The ethernet Source and Destination MAC addresses are not changed either.

If H1 wants to send a unicast IP datagram to H3, which is on a different subnet, H1 has to be configured with the IP address of a "default router". Let's assume that H1 is configured with an IP address of Router1 as its default router address. H1 compares H3's IP address with its own IP address and IP subnet mask, and determines that H3 is on a different subnet. So the packet has to be routed. H1 uses ARP to map Router1's IP address to a MAC address on LAN1. H1 then encapsulates the datagram in an ethernet frame, using router1's MAC address as the destination MAC address, and sends the frame to Router1.

Router1 then receives the frame over its lan1 interface. Router1 sees that the frame is addressed to it, so it removes the ethernet encapsulation and processes the IP datagram. The datagram is not addressed to Router1, so it must be forwarded further. Router1 does a lookup of the datagram's IP destination field, and determines that the destination (H3) can be reached via Router1's lan2 interface. Router1 now performs the IP processing of the datagram: it decrements the IP TTL, adjusts the IP header checksum (if present), may fragment the packet is necessary, etc. Then the datagram (or its fragments) are encapsulated in an ethernet header, with Router1's MAC address on LAN2 as the MAC Source Address, and H3's MAC address on LAN2 (which Router1 determines via ARP) as the MAC Destination Address. Finally the packet is sent out the lan2 interface.

If H1 has an IP multicast datagram to send (i.e., an IP datagram whose Destination Address field is an IP Multicast Address), it encapsulates it in an ethernet frame whose MAC Destination Address is computed from the IP Destination Address.

If H2 is a receiver for that multicast address, H2 will receive a copy of the frame, unchanged, from H1. The MAC Source Address in the ethernet encapsulation does not change, the IP TTL field does not get decremented, etc.

If H3 is a receiver for that multicast address, the datagram must be routed to H3. In order for this to happen, Router1 must be configured as a multicast router, and it must accept traffic sent to ethernet multicast addresses. Router1 will receive H1's multicast frame on its lan1 interface, will remove the ethernet encapsulation, and will determine how to dispatch the IP datagram based on Router1's multicast forwarding states. If Router1 knows that there is a receiver for the multicast datagram on LAN2, makes a copy of the datagram, decrements the TTL (and performs any other necessary IP processing), then encapsulates the datagram in ethernet frame for LAN2. The MAC Source Address for this frame will be Router1's MAC Source Address on LAN2. The MAC Destination Address is computed from the IP Destination Address. Finally, the frame is sent out Router1's LAN2 interface.

Figure 2 shows an Integrated Router/Bridge that supports the routing/ bridging integration model of [EVPN-IRB].

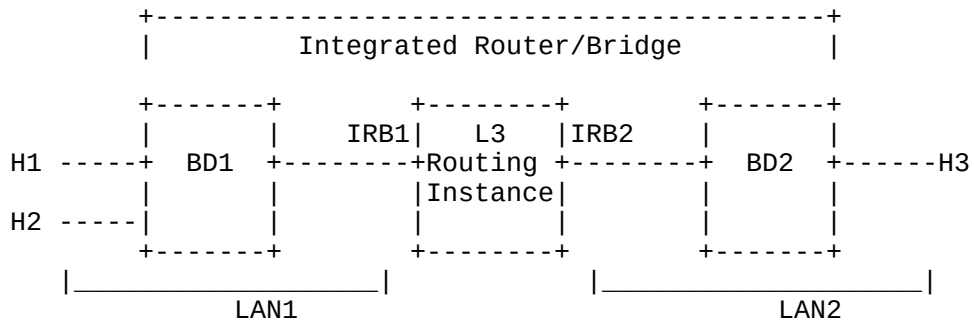


Figure 2: Integrated Router/Bridge

In Figure 2, a single box consists of one or more "L3 Routing Instances". The routing/forwarding tables of a given routing instance is known as an IP-VRF ([EVPN-IRB]). In the context of EVPN, it is convenient to think of each routing instance as representing the routing of a particular tenant. Each IP-VRF is attached to one or more interfaces.

When several EVPN PEs have a routing instance of the same tenant domain, those PEs advertise IP routes to the attached hosts. This is done as specified in [EVPN-IRB].

The integrated router/bridge shown in Figure 2 also attaches to a number of "Broadcast Domains" (BDs). Each BD performs the functions that are performed by the bridges in Figure 1. To the L3 routing instance, each BD appears to be a LAN. The interface attaching a particular BD to a particular IP-VRF is known as an "IRB Interface". From the perspective of L3 routing, each BD is a subnet. Thus each IRB interface is configured with a MAC address (which is the router's MAC address on the corresponding LAN), as well as an IP address and subnet mask.

The integrated router/bridge shown in Figure 2 may have multiple ACs to each BD. These ACs are visible only to the bridging function, not to the routing instance. To the L3 routing instance, there is just one "interface" to each BD.

If the L3 routing instance represents the IP routing of a particular tenant, the BDs attached to that routing instance are BDs belonging to that same tenant.

Bridging and routing now proceed exactly as in the case of Figure 1, except that BD1 replaces Switch1, BD2 replaces Switch2, interface IRB1 replaces interface lan1, and interface IRB2 replaces interface lan2.

It is important to understand that an IRB interface connects an L3 routing instance to a BD, NOT to a "MAC-VRF". (See [RFC7432] for the definition of "MAC-VRF".) A MAC-VRF may contain several BDs, as long as no MAC address appears in more than one BD. From the perspective of the L3 routing instance, each individual BD is an individual IP subnet; whether each BD has its own MAC-VRF or not is irrelevant to the L3 routing instance.

Figure 3 illustrates IRB when a pair of BDs (subnets) are attached to two different PE routers. In this example, each BD has two segments, and one segment of each BD is attached to one PE router.

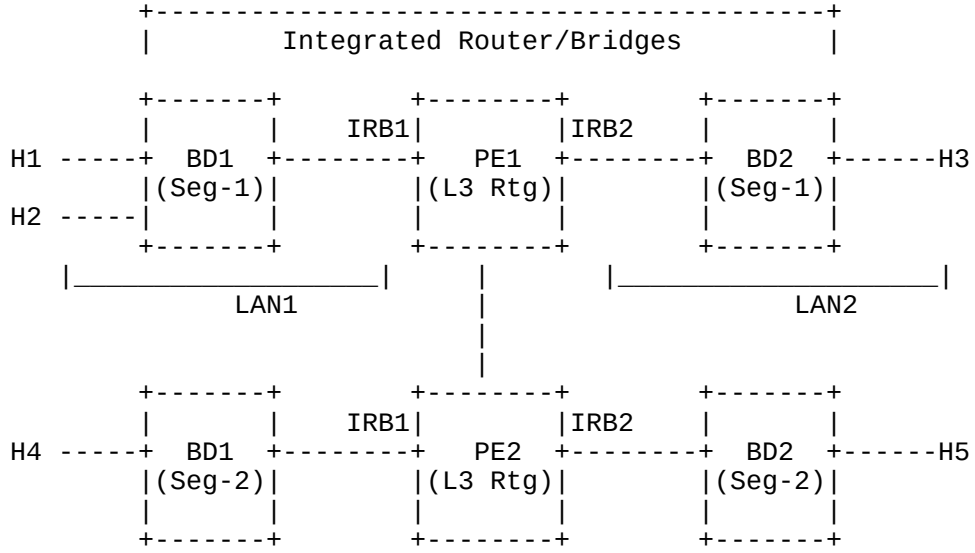


Figure 3: Integrated Router/Bridges with Distributed Subnet

If H1 needs to send an IP packet to H4, it determines from its IP address and subnet mask that H4 is on the same subnet as H1. Although H1 and H4 are not attached to the same PE router, EVPN provides ethernet communication among all hosts that are on the same BD. H1 thus uses ARP to find H4's MAC address, and sends an ethernet frame with H4's MAC address in the Destination MAC address field. The frame is received at PE1, but since the Destination MAC address is not PE1's MAC address, PE1 assumes that the frame is to remain on BD1. Therefore the packet inside the frame is NOT decapsulated, and is NOT send up the IRB interface to PE1's routing instance. Rather, standard EVPN intra-subnet procedures (as detailed in [RFC7432] are used to deliver the frame to PE2, which then sends it to H4.

If H1 needs to send an IP packet to H5, it determines from its IP address and subnet mask that H5 is NOT on the same subnet as H1. Assuming that H1 has been configured with the IP address of PE1 as its default router, H1 sends the packet in an ethernet frame with PE1's MAC address in its Destination MAC Address field. PE1 receives the frame, and sees that the frame is addressed to it. PE1 thus sends the frame up its IRB1 interface to the L3 routing instance. Appropriate IP processing is done (e.g., TTL decrement). The L3 routing instance determines that the "next hop" for H5 is PE2, so the packet is encapsulated (e.g., in MPLS) and sent across the backbone to PE2's routing instance. PE2 will see that the packet's destination, H5, is on BD2 segment-2, and will send the packet down its IRB2 interface. This causes the IP packet to be encapsulated in an ethernet frame with PE2's MAC address (on BD2) in the Source Address field and H5's MAC address in the Destination Address field.

Note that if H1 has an IP packet to send to H3, the forwarding of the packet is handled entirely within PE1. PE1's routing instance sees the packet arrive on its IRB1 interface, and then transmits the packet by sending it down its IRB2 interface.

Often, all the hosts in a particular Tenant Domain will be provisioned with the same value of the default router IP address. This IP address can be assigned, as an "anycast address", to all the EVPN PEs attached to that Tenant Domain. Thus although all hosts are provisioned with the same "default router address", the actual default router for a given host will be one of the PEs that is attached to the same ethernet segment as the host. This provisioning method ensures that IP packets from a given host are handled by the closest EVPN PE that supports IRB.

In the topology of Figure 3, one could imagine that H1 is configured with a default router address that belongs to PE2 but not to PE1. Inter-subnet routing would still work, but IP packets from H1 to H3 would then follow the non-optimal path H1-->PE1-->PE2-->PE1-->H3. Sending traffic on this sort of path, where it leaves a router and then comes back to the same router, is sometimes known as "hairpinning". Similarly, if PE2 supports IRB but PE1 dos not, the same non-optimal path from H1 to H3 would have to be followed. To avoid hairpinning, each EVPN PE needs to support IRB.

It is worth pointing out the way IRB interfaces interact with multicast traffic. Referring again to Figure 3, suppose PE1 and PE2 are functioning as IP multicast routers. Suppose also that H3 transmits a multicast packet, and both H1 and H4 are interested in receiving that packet. PE1 will receive the packet from H3 via its IRB2 interface. The ethernet encapsulation from BD2 is removed, the IP header processing is done, and the packet is then reencapsulated

for BD1, with PE1's MAC address in the MAC Source Address field. Then the packet is sent down the IRB1 interface. Layer 2 procedures (as defined in [RFC7432] would then be used to deliver a copy of the packet locally to H1, and remotely to H4.

Please be aware that his document modifies the semantics, described in the previous paragraph, of sending/receiving multicast traffic on an IRB interface. This is explained in Section 1.5.1 and subsequent sections.

Authors' Addresses

Wen Lin
Juniper Networks, Inc.

EMail: wlin@juniper.net

Zhaohui Zhang
Juniper Networks, Inc.

EMail: zzhang@juniper.net

John Drake
Juniper Networks, Inc.

EMail: jdrake@juniper.net

Eric C. Rosen (editor)
Juniper Networks, Inc.

EMail: erosen@juniper.net

Jorge Rabadan
Nokia

EMail: jorge.rabadan@nokia.com

Ali Sajassi
Cisco Systems

EMail: sajassi@cisco.com

BESS Workgroup
Internet Draft

Intended status: Informational

J. Rabadan
K. Nagaraj
S. Sathappan
V. Prabhu
W. Henderickx
Alcatel-Lucent

A. Liu
Ericsson

Expires: July 14, 2016

January 11, 2016

AC-influenced Designated Forwarder Election for EVPN
draft-rabadan-bess-evpn-ac-df-03

Abstract

The Designated Forwarder (DF) in EVPN networks is the PE responsible for sending multicast, broadcast and unknown unicast traffic to a multi-homed CE, on a given Ethernet Tag on a particular Ethernet Segment (ES). The DF is selected based on the list of PEs that advertise the Ethernet Segment Identifier (ESI) to the EVPN network. While PE node or link failures trigger the DF re-election for a given <ESI, EVI>, individual Attachment Circuit (AC) or MAC-VRF failures do not trigger such DF re-election and the traffic may therefore be permanently impacted, even though there is an alternative path. This document improves the DF election algorithm so that the AC status can influence the result of the election and this type of "logical" failures can be protected too.

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."

This Internet-Draft will expire on July 14, 2016.

Table of Contents

1. Problem Statement

[RFC7432] defines the Designated Forwarder (DF) as the EVPN PE responsible for:

- o Flooding Broadcast, Unknown unicast and Multicast traffic (BUM), on a given Ethernet Tag on a particular Ethernet Segment (ES), to the CE. This is valid for single-active and all-active EVPN multi-homing.

- o Sending unicast traffic on a given Ethernet Tag on a particular ES to the CE. This is valid for single-active multi-homing.

The default DF election algorithm defined by [RFC7432] is called service-carving and, for a given ESI, is based on a (V mod N)= i function that provides a local DF election of a PEi at <ESI, EVI> level. V is the Ethernet Tag associated to the EVI (the numerically lowest Ethernet Tag value in case of multiple Ethernet Tags), whereas N is the number of PEs for which ES routes have been successfully imported. In other words, EVPN’s service-carving takes into account only two variables in the DF election for a given ESI: the existence of the PE’s IP address on the candidate list and the locally provisioned Ethernet Tags.

If the DF for an <ESI, EVI> fails (due to physical link/node failures) an ES route withdrawn will make the Non-DF (NDF) PEs re-elect the DF for that <ESI, EVI> and the service will be recovered.

However the current DF election procedure does not provide a protection against "logical" failures or human errors that may occur at service level on the DF, while the list of active PEs for a given ES does not change. These failures may have an impact not only on the local PE where the issue happens, but also on the rest of the PEs of the ES. Some examples of such logical failures are listed below:

a) A given individual Attachment Circuit (AC) defined in an ES is accidentally shutdown or even not provisioned yet (hence the Attachment Circuit Status - ACS - is DOWN), while the ES is operationally active (since the ES route is active).

b) A given MAC-VRF - with an ES defined - is shutdown or not provisioned yet, while the ES is operationally active (since the ES route is active). In this case, the ACS of all the AC defined in that MAC-VRF is considered to be DOWN.

Neither (a) nor (b) will trigger the DF re-election on the remote PEs for a given ES since the ACS is not taken into account in the DF election procedures. While the ACS is used as a DF election tie-breaker and trigger in [VPLS-MH], there is no procedure defined in [RFC7432] to trigger the DF re-election based on the ACS change on the DF.

This document improves the [RFC7432] service-carving procedure so

that the ACS may be taken into account as a variable in the DF election, and therefore EVPN can provide protection against logical failures.

2. Solution description

The ACS for a given Ethernet Tag on an ESI is implicitly conveyed in the corresponding EVPN A-D per EVI route for that given <ESI, Ethernet Tag>. This section describes how to use the A-D per EVI routes to improve the DF election algorithm.

Figure 1 illustrates an example EVPN network that will be used to describe the proposed solution.

EVI-1 is defined in PE-1, PE-2, PE-3 and PE-4. CE12 is a multi-homed CE connected to ESI12 in PE-1 and PE-2. Similarly CE23 is multi-homed to PE-2 and PE-3 using ESI23. CE12-VID 1 (VLAN ID 1 on CE12) is associated to AC1 and AC2 in EVI-1, whereas CE23-VID 1 is associated to AC3 and AC4 in EVI-1. Note that there are other ACs defined on these ESIs mapped to different EVIs.

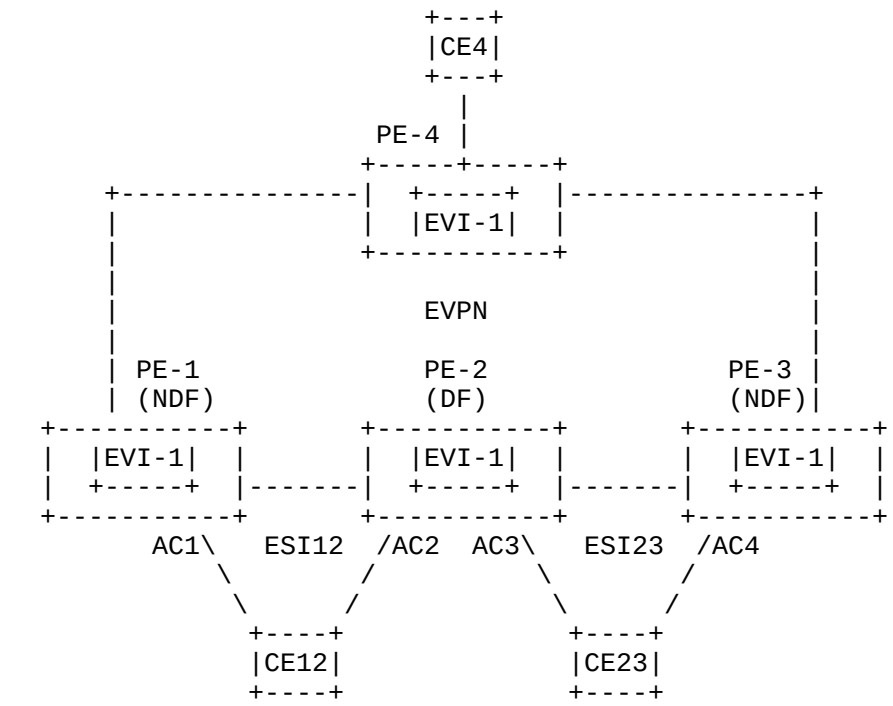


Figure 1 EVPN network example

2.1. Current DF election procedure and AC failures

After running the service-carving DF election algorithm, PE-2 turns out to be the DF for ESI12 and ESI23 in EVI-1. The following two examples illustrate the issues with the existing defined procedure in [RFC7432]:

a) If AC2 is accidentally shutdown or even not configured, CE12 traffic will be impacted. In case of all-active multi-homing, only the BUM traffic to CE12 will be impacted, whereas for single-active multi-homing all the traffic to/from CE12 will be discarded. This is due to the fact that a logical failure in PE-2 AC2 will not trigger an ES route withdrawn for ESI12 (since there are still other ACs active on ESI12) and therefore PE-1 will not re-run the DF election procedures.

b) If EVI-1 is administratively shutdown or even not configured yet on PE-2, CE12 and CE23 will both be impacted: BUM traffic to both CEs will be discarded in case of all-active multi-homing and all traffic will be discarded to/from the CEs in case of single-active multi-homing. This is due to the fact that PE-1 and PE-3 will not re-run the DF election procedures and will keep assuming PE-2 is the DF.

According to [RFC7432], "when an Ethernet tag is decommissioned on an Ethernet segment, then the PE MUST withdraw the Ethernet A-D per EVI route(s) announced for the <ESI, Ethernet tags> that are impacted by the decommissioning", however, while this A-D per EVI route withdrawal is used at the remote PEs performing aliasing or backup procedures, it is not used to influence the DF election for the affected EVIs.

2.2. The Attachment Circuit (AC) influenced DF election

Modifying the service-carving DF election procedure in the following way solves the issue:

1. When PE-1 and PE-2 discover ESI12, they advertise an ES route for ESI12 with the associated ES-import extended community, starting a timer at the same time. Likewise, PE-2 and PE-3 advertise an ES route for ESI23 and start a timer.

2. Similarly, PE-1 and PE-2 advertise an Ethernet A-D per ES route for ESI12, and PE-2/PE-3 advertise an Ethernet A-D per ES route for ESI23.

3. In addition, PE-1/PE-2/PE-3 advertise an Ethernet A-D per EVI

route for AC1, AC2, AC3 and AC4 as soon as the ACs are enabled. Note that the AC can be associated to a single customer VID (e.g. VLAN-based interfaces) or a bundle of customer VIDs (e.g. VLAN-bundle interfaces).

4. When the timer expires, each PE builds an ordered "candidate" list of the IP addresses of all the PE nodes connected to the Ethernet Segment (including itself), in increasing numeric order. The candidate list is based on the Originator Router's IP addresses of the ES routes, excluding all the PEs for which no Ethernet A-D per ES route has been received.

5. When electing the DF for a given EVI, a PE will not be considered candidate until an Ethernet A-D per EVI route has been received from that PE. In other words, the ACS on the ESI for a given PE must be UP so that the PE is considered as candidate for a given EVI. For example, PE-1 will not consider PE-2 as candidate for DF election for <ESI12, EVI-1> until an Ethernet A-D per EVI route is not received from PE-2 for <ESI12, EVI-1>.

6. Once the PEs with ACS = DOWN for a given EVI have been eliminated from the candidate list, the (V mod N) = i function can be applied for the remaining N candidates, as per [RFC7432].

Note that this procedure does not modify the existing EVPN control plane whatsoever. It only modifies the candidate list of PEs taken into account for the DF election algorithm defined in [RFC7432].

In addition to the procedure described above, the following events SHALL modify the candidate PE list and trigger the DF re-election in a PE for a given <ESI,EVI>:

a) Local ES going DOWN due to a physical failure or reception of an ES route withdraw for that ESI.

b) Local ES going UP due to its detection/configuration or reception of a new ES route update for that ESI.

c) Local AC going DOWN/UP.

d) Reception of a new Ethernet A-D per EVI update/withdraw for the <ESI, EVI>.

e) Reception of a new Ethernet A-D per ES update/withdraw for the ESI.

This procedure is backwards compatible with the DF election procedures described in [RFC7432].

3. Solution benefits

The solution described in this document provides the following benefits:

a) Improves the DF election procedures for EVPN so that failures due to human errors, logical failures or even delay in provisioning of Attachment Circuits can be protected by multi-homing.

b) It does not modify or add any BGP new attributes or NLRI changes.

c) It is backwards compatible with the procedures defined in RFC7432.

4. Conventions used in this document

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

5. Security Considerations

The same Security Considerations described in [RFC7432] are valid for this document.

6. IANA Considerations

There are no new IANA considerations in this document.

7. References

7.1. Normative References

[RFC4684] Marques, P., Bonica, R., Fang, L., Martini, L., Raszuk, R., Patel, K., and J. Guichard, "Constrained Route Distribution for Border Gateway Protocol/MultiProtocol Label Switching (BGP/MPLS) Internet Protocol (IP) Virtual Private Networks (VPNs)", RFC 4684, DOI 10.17487/RFC4684, November 2006, <http://www.rfc-

editor.org/info/rfc4684>.

[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>.

7.2. Informative References

[VPLS-MH] Kothari, Henderickx et al., "BGP based Multi-homing in Virtual Private LAN Service", draft-ietf-bess-vpls-multihoming-01.txt, work in progress, January, 2016.

8. Acknowledgments

Will be added.

Authors' Addresses

Jorge Rabadan
Alcatel-Lucent
777 E. Middlefield Road
Mountain View, CA 94043 USA
Email: jorge.rabadan@alcatel-lucent.com

Kiran Nagaraj
Alcatel-Lucent
Email: kiran.nagaraj@alcatel-lucent.com

Senthil Sathappan
Alcatel-Lucent
Email: senthil.sathappan@alcatel-lucent.com

Vinod Prabhu
Alcatel-Lucent
Email: vinod.prabhu@alcatel-lucent.com

Wim Henderickx
Alcatel-Lucent
Email: wim.henderickx@alcatel-lucent.com

Autumn Liu
Ericsson
Email: autumn.liu@ericsson.com

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
S. Sathappan
Alcatel-Lucent

T. Przygienda
Ericsson

W. Lin
T. Singh
Juniper Networks

A. Sajassi
S. Mohanty
Cisco Systems

Expires: May 28, 2016

November 25, 2015

Preference-based EVPN DF Election
draft-rabadan-bess-evpn-pref-df-00

Abstract

RFC7432 defines the Designated Forwarder (DF) in (PBB-)EVPN networks as the PE responsible for sending broadcast, multicast and unknown unicast traffic (BUM) to a multi-homed device/network in the case of an all-active multi-homing ES, or BUM and unicast in the case of single-active multi-homing.

The DF is selected out of a candidate list of PEs that advertise the Ethernet Segment Identifier (ESI) to the EVPN network, according to the 'service-carving' algorithm.

While 'service-carving' provides an efficient and automated way of selecting the DF across different EVIs or ISIDs in the ES, there are some use-cases where a more 'deterministic' and user-controlled method is required. At the same time, Service Providers require an easy way to force an on-demand DF switchover in order to carry out some maintenance tasks on the existing DF or control whether a new active PE can preempt the existing DF PE.

This document proposes an extension to the current RFC7432 DF election procedures so that the above requirements can be met.

Status of this Memo

Copyright Notice

Table of Contents

1. Problem Statement

RFC7432 defines the Designated Forwarder (DF) in (PBB-)EVPN networks as the PE responsible for sending broadcast, multicast and unknown unicast traffic (BUM) to a multi-homed device/network in the case of an all-active multi-homing ES or BUM and unicast traffic to a multi-homed device or network.

The DF is selected out of a candidate list of PEs that advertise the Ethernet Segment Identifier (ESI) to the EVPN network and according to the 'service-carving' algorithm.

While 'service-carving' provides an efficient and automated way of selecting the DF across different EVIs or ISIDs in the ES, there are some use-cases where a more 'deterministic' and user-controlled method is required. At the same time, Service Providers require an easy way to force an on-demand DF switchover in order to carry out some maintenance tasks on the existing DF or control whether a new active PE can preempt the existing DF PE.

This document proposes an extension to the current RFC7432 DF election procedures so that the above requirements can be met.

2. Solution requirements

This document proposes an extension of the RFC7432 'service-carving' DF election algorithm motivated by the following requirements:

a) The solution MUST provide an administrative preference option so that the user can control in what order the candidate PEs may become DF, assuming they are all operationally ready to take over.

b) This extension MUST work for RFC7432 Ethernet Segments (ES) and

virtual ES, as defined in [vES].

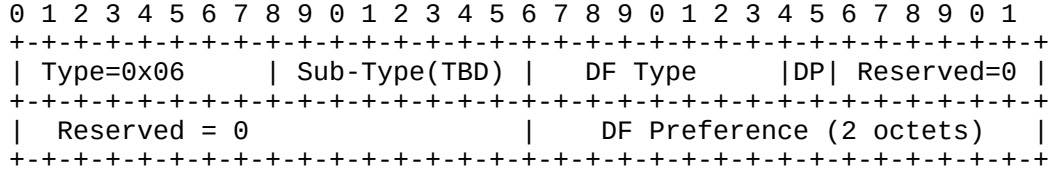
c) The user MUST be able to force a PE to preempt the existing DF for a given EVI/ISID without re-configuring all the PEs in the ES.

d) The solution SHOULD allow an option to NOT preempt the current DF, even if the former DF PE comes back up after a failure. This is also known as "non-revertive" behavior, as opposed to the RFC7432 DF election procedures that are always revertive.

e) The solution MUST work for single-active and all-active multi-homing Ethernet Segments.

3. EVPN BGP Attributes for Deterministic DF Election

This solution reuses and extends the DF Election Extended Community defined in [EVPN-HRW-DF] that is advertised along with the ES route:



Where the following fields are re-defined as follows:

o DF Type can have the following values:

- Type 0 - Default, mod based DF election as per RFC7432.
- Type 1 - HRW algorithm as per [EVPN-HRW-DF]
- Type 2 - Preference algorithm (this document)

o DP or ’Don’t Preempt’ bit, determines if the PE advertising the ES route requests the remote PEs in the ES not to preempt it as DF. The default value is DP=0, which is compatible with the current ’preempt’ or ’revertive’ behavior in RFC7432. The DP bit SHOULD be ignored if the DF Type is different than 2.

o DF Preference defines a 2-octet value that indicates the PE preference to become the DF in the ES. The default value MUST be 32767. This value is the midpoint in the allowed Preference range of values, which gives the operator the flexibility of choosing a significant number of values, above or below the default Preference.

4. Solution description

Figure 1 illustrates an example that will be used in the description of the solution.

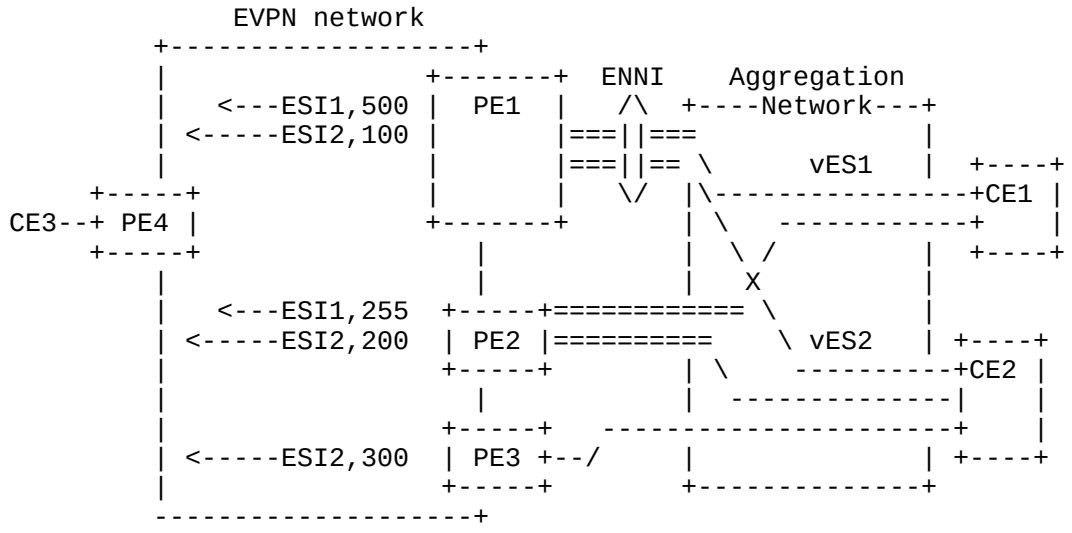


Figure 1 ES and Deterministic DF Election

Figure 1 shows three PEs that are connecting EVCs coming from the Aggregation Network to their EVIs in the EVPN network. CE1 is connected to vES1 - that spans PE1 and PE2 - and CE2 is connected to vES2, that is defined in PE1, PE2 and PE3.

If the algorithm chosen for vES1 and vES2 is type 2, i.e. Preference-based, the PEs may become DF irrespective of their IP address and based on an administrative Preference value. The following sections provide some examples of the new defined procedures and how they are applied in the use-case in Figure 1.

4.1 Use of the Preference algorithm

Assuming the operator wants to control - in a flexible way - what PE becomes the DF for a given vES and the order in which the PEs become DF in case of multiple failures, the following procedure may be used:

a) vES1 and vES2 are now configurable with three optional parameters that are signaled in the DF Election extended community. These parameters are the Preference, Preemption option (or "Don’t

Preempt Me" option) and DF algorithm type. We will represent these parameters as [Pref,DP,type]. Let’s assume vES1 is configured as [500,0,Pref] in PE1, and [255,0,Pref] in PE2. vES2 is configured as [100,0,Pref], [200,0,Pref] and [300,0,Pref] in PE1, PE2 and PE3 respectively.

b) The PEs will advertise an ES route for each vES, including the 3 parameters in the DF Election Extended Community.

c) According to RFC7432, each PE will wait for the DF timer to expire before running the DF election algorithm. After the timer expires, each PE runs the Preference-based DF election algorithm as follows:

   o The PE will check the DF type in each ES route, and assuming all the ES routes are consistent in this DF type and the value is 2 (Preference-based), the PE will run the new extended procedure. Otherwise, the procedure will fall back to RFC7432 ’service-carving’.

   o In this extended procedure, each PE builds a list of candidate PEs, ordered based on the Preference. E.g. PE1 will build a list of candidate PEs for vES1 ordered by the Preference, from high to low: PE1>PE2. Hence PE1 will become the DF for vES1. In the same way, PE3 becomes the DF for vES2.

d) Note that, by default, the Highest-Preference is chosen for each ES or vES, however the ES configuration can be changed to the Lowest-Preference algorithm as long as this option is consistent in all the PEs in the ES. E.g. vES1 could have been explicitly configured as type Preference-based with Lowest-Preference, in which case, PE2 would have been the DF.

e) Assuming some maintenance tasks had to be executed on PE3, the operator could set vES2’s preference to e.g. 50 so that PE2 is forced to take over as DF for vES2. Once the maintenance on PE3 is over, the operator could decide to leave the existing preference or configure the old preference back.

f) In case of equal Preference in two or more PEs in the ES, the tie-breakers will be the DP bit and the lowest IP PE in that order. For instance:

   o If vES1 parameters were [500,0,Pref] in PE1 and [500,1,Pref] in PE2, PE2 would be elected due to the DP bit.

   o If vES1 parameters were [500,0,Pref] in PE1 and [500,0,Pref] in PE2, PE1 would be elected, assuming PE1’s IP address is lower

than PE2's.

g) The Preference is an administrative option that MUST be configured on a per-ES basis from the management plane, but MAY also be dynamically changed based on the use of local policies. For instance, on PE1, ES1's Preference can be lowered from 500 to 100 in case the bandwidth on the ENNI port is decreased a 50% (that could happen if e.g. the 2-port LAG between PE1 and the Aggregation Network loses one port). Policies MAY also trigger dynamic Preference changes based on the PE's bandwidth availability in the core, of specific ports going operationally down, etc. The definition of the actual local policies is out of scope of this document. The default Preference value is 32767.

4.2 Use of the Preference algorithm in RFC7432 Ethernet-Segments

While the Preference-based DF type described in section 4.1 is typically used in virtual ES scenarios where there is normally an individual EVI per vES, the existing RFC7432 definition of ES allows potentially up to thousands of EVIs on the same ES. If this is the case, and the operator still wants to control who the DF is for a given EVI, the use of the Preference-based DF type can also provide the desired level of load balancing.

In this type of scenarios, the ES is configured with an administrative Preference value, but then a range of EVI/ISIDs can be defined to use the Highest-Preference or the Lowest-Preference depending on the desired behavior. With this option, the PE will build a list of candidate PEs ordered by the Preference, however the DF for a given EVI/ISID will be determined by the local configuration.

For instance:

- Assuming ES3 is defined in PE1 and PE2, PE1 may be configured as [500,0,Preference] for ES3 and PE2 as [100,0,Preference].
- In addition, assuming vlan-based service interfaces, the PEs will be configured with (vlan/ISID-range,high_or_low), e.g. (1-2000,high) and (2001-4000, low).
- This will result in PE1 being DF for EVI/ISIDs 1-2000 and PE2 being DF for EVI/ISIDs 2001-4000.

4.3 The Non-Revertive option

As discussed in section 2(d), an option to NOT preempt the existing

DF for a given EVI/ISID is required and therefore added to the DF Election extended community. This option will allow a non-revertive behavior in the DF election.

Note that, when a given PE in an ES is taken down for maintenance operations, before bringing it back, the Preference may be changed in order to provide a non-revertive behavior. The DP bit and the mechanism explained in this section will be used for those cases when a former DF comes back up without any controlled maintenance operation, and the non-revertive option is desired in order to avoid service impact.

In Figure 1, we assume that based on the Highest-Pref, PE3 is the DF for ESI2.

If PE3 has a link, EVC or node failure, PE2 would take over as DF. If/when PE3 comes back up again, PE3 will take over, causing some unnecessary packet loss in the ES.

The following procedure avoids preemption upon failure recovery (please refer to Figure 1):

1) A new "Don’t Preempt Me" parameter is defined on a per-PE per-ES basis. If "Don’t Preempt Me" is disabled (default behavior) the advertised DP bit will be 0. If "Don’t Preempt Me" is enabled on a PE, the ES route will be advertised with DP=1 ("Don’t Preempt Me") once the DF timer is expired and the DF elected.

2) Assuming we want to avoid ’preemption’, the three PEs are configured with the "Don’t Preempt Me" option. Note that each PE individually MAY be configured with different preemption value. In this example, we assume ESI2 is configured as ’DP=disabled’ in PE1 but ’DP=enabled’ in PE2 and PE3.

3) When ES2 is enabled in the three PEs, and after the DF timer, the PEs (due to the Highest-Pref type) select PE3 as DF for EVI1. Only after the timer and the DF election, the PEs will check the ’DP’ configuration and since it is enabled on PE2 and PE3, these two PEs will send an ES route update, now with DP=1. This update will not cause any change in the existing DFs since there is no change in the Preference value.

4) If PE3’s vES2 goes down (due to EVC failure - detected by OAM, or port failure or node failure), PE2 will become the DF for ESI2/EVI1.

5) When PE3’s vES2 comes back up, PE3 will start a boot-timer (if booting up) or hold-timer (if the port or EVC recovers). That

timer will allow some time for PE3 to receive the ES routes from PE1 and PE2. PE3 will then check its own [Pref,DP,type]=[300,1,Pref] and if its Pref is higher than any of the other PE's Pref, then PE3 will send the ES route with an 'in-use' Preference equal to the highest received Preference. In this case, since PE2 advertised [Pref,DP,type]=[200,1,Pref], PE3 will then send [200,0,Pref].

Note that, a PE will always send DP=0 the first time it advertises an ES route after the ES becomes active, and irrespective of the configuration. Also a PE will always send DP=0 as long as the advertised Pref is the 'in-use' Pref (as opposed to the 'admin' Pref).

This ES route update sent by PE3 (with [200,0,Pref]) will not cause any changes in the DF election and PE2 will continue being DF. This is because the DP bit will be used as a tie-breaker in the DF election. That is, if a PE has two candidate PEs with the same Pref, it will pick up the one with DP=1.

6) Only in case of PE2's failure, PE3 will become DF again (assuming it wins the DF election to PE1), and will resend the ES route with the admin Pref (as opposed to the 'in-use' Pref) and the DP bit that corresponds to its configuration.

5. Conclusions

Service Providers are seeking for options where the DF election can be controlled by the user in a deterministic way and with a non-revertive behavior. This document defines the use of a Preference algorithm that can be configured and used in a flexible manner to achieve those objectives.

11. Conventions used in this document

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying

or finding the explicit compliance requirements of this RFC.

12. Security Considerations

This section will be added in future versions.

13. IANA Considerations

This document solicits the allocation of DF type = 2 in the registry created by [vES] for the DF type field.

15. References

15.1 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>.

15.2 Informative References

[vES] Sajassi et al. "EVPN Virtual Ethernet Segment", draft-sajassi-bess-evpn-virtual-eth-segment-01, work-in-progress, July 6, 2015.

[EVPN-HRW-DF] Mohanty S. et al. "A new Designated Forwarder Election for the EVPN", draft-mohanty-bess-evpn-df-election-02, work-in-progress, October 19, 2015.

16. Acknowledgments

17. Contributors

In addition to the authors listed, the following individuals also contributed to this document:

Vinod Prabhu, ALU
Kiran Nagaraj, ALU
John Drake, Juniper
Selvakumar Sivaraj, Juniper

17. Authors’ Addresses

Jorge Rabadan
Alcatel-Lucent
777 E. Middlefield Road
Mountain View, CA 94043 USA
Email: jorge.rabadan@alcatel-lucent.com

Senthil Sathappan
Alcatel-Lucent
Email: senthil.sathappan@alcatel-lucent.com

Tony Przygienda
Ericsson
Email: antoni.przygienda@ericsson.com

Tapraj Singh
Juniper Networks, Inc.
Email: tsingh@juniper.net

Wen Lin
Juniper Networks, Inc.
Email: wlin@juniper.net

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

Satya Ranjan Mohanty
Cisco Systems, Inc.
Email: satyamoh@cisco.com

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
S. Sathappan
Nokia

S. Boutros
VMware

T. Przygienda
W. Lin
J. Drake
Juniper Networks

A. Sajassi
S. Mohanty
Cisco Systems

Expires: June 22, 2017

December 19, 2016

Preference-based EVPN DF Election
draft-rabadan-bess-evpn-pref-df-02

Abstract

RFC7432 defines the Designated Forwarder (DF) in (PBB-)EVPN networks as the PE responsible for sending broadcast, multicast and unknown unicast traffic (BUM) to a multi-homed device/network in the case of an all-active multi-homing ES, or BUM and unicast in the case of single-active multi-homing.

The DF is selected out of a candidate list of PEs that advertise the Ethernet Segment Identifier (ESI) to the EVPN network, according to the 'service-carving' algorithm.

While 'service-carving' provides an efficient and automated way of selecting the DF across different EVIs or ISIDs in the ES, there are some use-cases where a more 'deterministic' and user-controlled method is required. At the same time, Service Providers require an easy way to force an on-demand DF switchover in order to carry out some maintenance tasks on the existing DF or control whether a new active PE can preempt the existing DF PE.

This document proposes an extension to the current RFC7432 DF election procedures so that the above requirements can be met.

Status of this Memo

This Internet-Draft is submitted in full conformance with the

Copyright Notice

Table of Contents

1. Problem Statement

RFC7432 defines the Designated Forwarder (DF) in (PBB-)EVPN networks as the PE responsible for sending broadcast, multicast and unknown unicast traffic (BUM) to a multi-homed device/network in the case of an all-active multi-homing ES or BUM and unicast traffic to a multi-homed device or network in case of single-active multi-homing.

The DF is selected out of a candidate list of PEs that advertise the Ethernet Segment Identifier (ESI) to the EVPN network and according to the 'service-carving' algorithm.

While 'service-carving' provides an efficient and automated way of selecting the DF across different EVIs or ISIDs in the ES, there are some use-cases where a more 'deterministic' and user-controlled method is required. At the same time, Service Providers require an easy way to force an on-demand DF switchover in order to carry out some maintenance tasks on the existing DF or control whether a new active PE can preempt the existing DF PE.

This document proposes an extension to the current RFC7432 DF election procedures so that the above requirements can be met.

2. Solution requirements

This document proposes an extension of the RFC7432 'service-carving' DF election algorithm motivated by the following requirements:

a) The solution MUST provide an administrative preference option so that the user can control in what order the candidate PEs may become DF, assuming they are all operationally ready to take over.

b) This extension MUST work for RFC7432 Ethernet Segments (ES) and virtual ES, as defined in [vES].

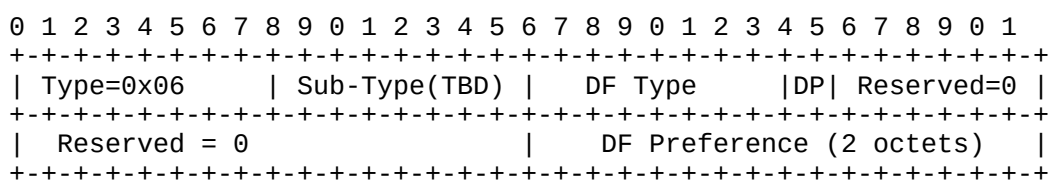
c) The user MUST be able to force a PE to preempt the existing DF for a given EVI/ISID without re-configuring all the PEs in the ES.

d) The solution SHOULD allow an option to NOT preempt the current DF, even if the former DF PE comes back up after a failure. This is also known as "non-revertive" behavior, as opposed to the RFC7432 DF election procedures that are always revertive.

e) The solution MUST work for single-active and all-active multi-homing Ethernet Segments.

3. EVPN BGP Attributes for Deterministic DF Election

This solution reuses and extends the DF Election Extended Community defined in [EVPN-HRW-DF] that is advertised along with the ES route:



Where the following fields are re-defined as follows:

o DF Type can have the following values:

- Type 0 - Default, mod based DF election as per RFC7432.
- Type 1 - HRW algorithm as per [EVPN-HRW-DF]
- Type 2 - Preference algorithm (this document)

o DP or ’Don’t Preempt’ bit, determines if the PE advertising the ES route requests the remote PEs in the ES not to preempt it as DF. The default value is DP=0, which is compatible with the current ’preempt’ or ’revertive’ behavior in RFC7432. The DP bit SHOULD be ignored if the DF Type is different than 2.

o DF Preference defines a 2-octet value that indicates the PE preference to become the DF in the ES. The default value MUST be 32767. This value is the midpoint in the allowed Preference range of values, which gives the operator the flexibility of choosing a significant number of values, above or below the default Preference.

4. Solution description

Figure 1 illustrates an example that will be used in the description of the solution.

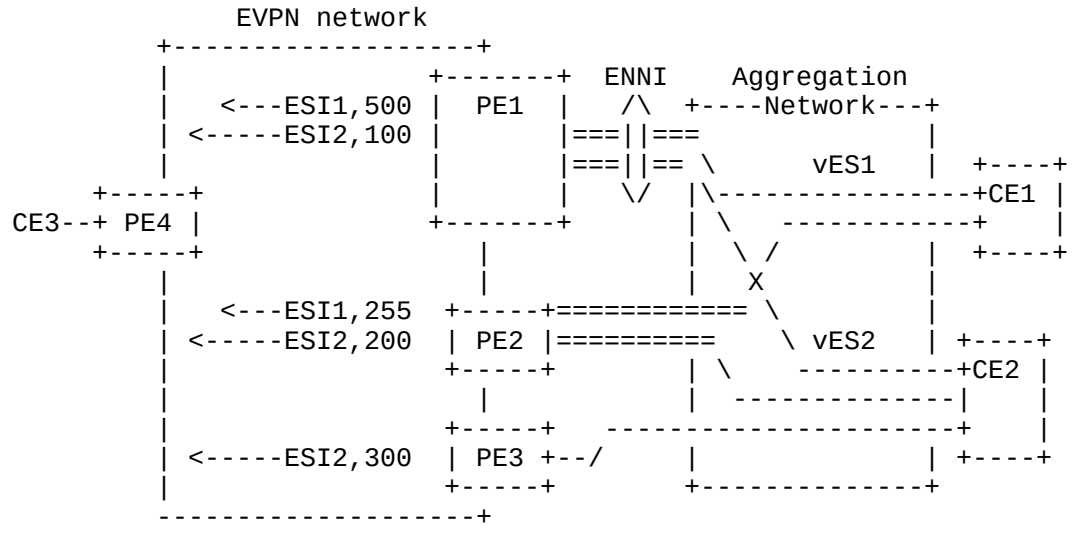


Figure 1 ES and Deterministic DF Election

Figure 1 shows three PEs that are connecting EVCs coming from the Aggregation Network to their EVIs in the EVPN network. CE1 is connected to vES1 - that spans PE1 and PE2 - and CE2 is connected to vES2, that is defined in PE1, PE2 and PE3.

If the algorithm chosen for vES1 and vES2 is type 2, i.e. Preference-based, the PEs may become DF irrespective of their IP address and based on an administrative Preference value. The following sections provide some examples of the new defined procedures and how they are applied in the use-case in Figure 1.

4.1 Use of the Preference algorithm

Assuming the operator wants to control - in a flexible way - what PE becomes the DF for a given vES and the order in which the PEs become DF in case of multiple failures, the following procedure may be used:

a) vES1 and vES2 are now configurable with three optional parameters that are signaled in the DF Election extended community. These parameters are the Preference, Preemption option (or "Don’t

Preempt Me" option) and DF algorithm type. We will represent these parameters as [Pref,DP,type]. Let’s assume vES1 is configured as [500,0,Pref] in PE1, and [255,0,Pref] in PE2. vES2 is configured as [100,0,Pref], [200,0,Pref] and [300,0,Pref] in PE1, PE2 and PE3 respectively.

b) The PEs will advertise an ES route for each vES, including the 3 parameters in the DF Election Extended Community.

c) According to RFC7432, each PE will wait for the DF timer to expire before running the DF election algorithm. After the timer expires, each PE runs the Preference-based DF election algorithm as follows:

   o The PE will check the DF type in each ES route, and assuming all the ES routes are consistent in this DF type and the value is 2 (Preference-based), the PE will run the new extended procedure. Otherwise, the procedure will fall back to RFC7432 ’service-carving’.

   o In this extended procedure, each PE builds a list of candidate PEs, ordered based on the Preference. E.g. PE1 will build a list of candidate PEs for vES1 ordered by the Preference, from high to low: PE1>PE2. Hence PE1 will become the DF for vES1. In the same way, PE3 becomes the DF for vES2.

d) Note that, by default, the Highest-Preference is chosen for each ES or vES, however the ES configuration can be changed to the Lowest-Preference algorithm as long as this option is consistent in all the PEs in the ES. E.g. vES1 could have been explicitly configured as type Preference-based with Lowest-Preference, in which case, PE2 would have been the DF.

e) Assuming some maintenance tasks had to be executed on PE3, the operator could set vES2’s preference to e.g. 50 so that PE2 is forced to take over as DF for vES2. Once the maintenance on PE3 is over, the operator could decide to leave the existing preference or configure the old preference back.

f) In case of equal Preference in two or more PEs in the ES, the tie-breakers will be the DP bit and the lowest IP PE in that order. For instance:

   o If vES1 parameters were [500,0,Pref] in PE1 and [500,1,Pref] in PE2, PE2 would be elected due to the DP bit.

   o If vES1 parameters were [500,0,Pref] in PE1 and [500,0,Pref] in PE2, PE1 would be elected, assuming PE1’s IP address is lower

than PE2's.

g) The Preference is an administrative option that MUST be configured on a per-ES basis from the management plane, but MAY also be dynamically changed based on the use of local policies. For instance, on PE1, ES1's Preference can be lowered from 500 to 100 in case the bandwidth on the ENNI port is decreased a 50% (that could happen if e.g. the 2-port LAG between PE1 and the Aggregation Network loses one port). Policies MAY also trigger dynamic Preference changes based on the PE's bandwidth availability in the core, of specific ports going operationally down, etc. The definition of the actual local policies is out of scope of this document. The default Preference value is 32767.

4.2 Use of the Preference algorithm in RFC7432 Ethernet-Segments

While the Preference-based DF type described in section 4.1 is typically used in virtual ES scenarios where there is normally an individual EVI per vES, the existing RFC7432 definition of ES allows potentially up to thousands of EVIs on the same ES. If this is the case, and the operator still wants to control who the DF is for a given EVI, the use of the Preference-based DF type can also provide the desired level of load balancing.

In this type of scenarios, the ES is configured with an administrative Preference value, but then a range of EVI/ISIDs can be defined to use the Highest-Preference or the Lowest-Preference depending on the desired behavior. With this option, the PE will build a list of candidate PEs ordered by the Preference, however the DF for a given EVI/ISID will be determined by the local configuration.

For instance:

- Assuming ES3 is defined in PE1 and PE2, PE1 may be configured as [500,0,Preference] for ES3 and PE2 as [100,0,Preference].
- In addition, assuming vlan-based service interfaces, the PEs will be configured with (vlan/ISID-range,high_or_low), e.g. (1-2000,high) and (2001-4000, low).
- This will result in PE1 being DF for EVI/ISIDs 1-2000 and PE2 being DF for EVI/ISIDs 2001-4000.

4.3 The Non-Revertive option

As discussed in section 2(d), an option to NOT preempt the existing

DF for a given EVI/ISID is required and therefore added to the DF Election extended community. This option will allow a non-revertive behavior in the DF election.

Note that, when a given PE in an ES is taken down for maintenance operations, before bringing it back, the Preference may be changed in order to provide a non-revertive behavior. The DP bit and the mechanism explained in this section will be used for those cases when a former DF comes back up without any controlled maintenance operation, and the non-revertive option is desired in order to avoid service impact.

In Figure 1, we assume that based on the Highest-Pref, PE3 is the DF for ESI2.

If PE3 has a link, EVC or node failure, PE2 would take over as DF. If/when PE3 comes back up again, PE3 will take over, causing some unnecessary packet loss in the ES.

The following procedure avoids preemption upon failure recovery (please refer to Figure 1):

1) A new "Don’t Preempt Me" parameter is defined on a per-PE per-ES basis, as described in section 3. If "Don’t Preempt Me" is disabled (default behavior) the advertised DP bit will be 0. If "Don’t Preempt Me" is enabled, the ES route will be advertised with DP=1 ("Don’t Preempt Me").

2) Assuming we want to avoid ’preemption’, the three PEs are configured with the "Don’t Preempt Me" option. Note that each PE individually MAY be configured with different preemption value. In this example, we assume ESI2 is configured as ’DP=enabled’ in the three PEs.

3) Assuming EVI1 uses Highest-Pref in vES2 and EVI2 uses Lowest-Pref, when vES2 is enabled in the three PEs, the PEs will exchange the ES routes and select PE3 as DF for EVI1 (due to the Highest-Pref type), and PE1 as DF for EVI2 (due to the Lowest-Pref).

4) If PE3’s vES2 goes down (due to EVC failure - detected by OAM, or port failure or node failure), PE2 will become the DF for EVI1. No changes will occur for EVI2.

5) When PE3’s vES2 comes back up, PE3 will start a boot-timer (if booting up) or hold-timer (if the port or EVC recovers). That timer will allow some time for PE3 to receive the ES routes from PE1 and PE2. PE3 will then:

o Select two "reference-PEs" among the ES routes in the vES, the "Highest-PE" and the "Lowest-PE":

  - The Highest-PE is the PE with higher Preference, using the DP bit first (with DP=1 being better) and, after that, the lower PE-IP address as tie-breakers. PE3 will select PE2 as Highest-PE over PE1, since, when comparing [Pref,DP,PE-IP], [200,1,PE2-IP] wins over [100,1,PE1-IP].

  - The Lowest-PE is the PE with lower Preference, using the DP bit first (with DP=1 being better) and, after that, the lower PE-IP address as tie-breakers. PE3 will select PE1 as Lowest-PE over PE2, since [100,1,PE1-IP] wins over [200,1,PE2-IP].

  - Note that if there were only one remote PE in the ES, Lowest and Highest PE would be the same PE.

o Check its own administrative Pref and compares it with the one of the Highest-PE and Lowest-PE that have DP=1 in their ES routes. Depending on this comparison PE3 will send the ES route with a [Pref,DP] that may be different from its administrative [Pref,DP]:

  - If PE3’s Pref value is higher than the Highest-PE’s, PE3 will send the ES route with an ’in-use’ operational Pref equal to the Highest-PE’s and DP=0.

  - If PE3’s Pref value is lower than the Lowest-PE’s, PE3 will send the ES route with an ’in-use’ operational Preference equal to the Lowest-PE’s and DP=0.

  - If PE3’s Pref value is neither higher nor lower than the Highest-PE’s or the Lowest-PE’s respectively, PE3 will send the ES route with its administrative [Pref,DP]=[300,1].

  - In this example, PE3’s administrative Pref=300 is higher than the Highest-PE with DP=1, that is, PE2 (Pref=200). Hence PE3 will inherit PE2’s preference and send the ES route with an operational ’in-use’ [Pref,DP]=[200,0].

Note that, a PE will always send DP=0 as long as the advertised Pref is the ’in-use’ operational Pref (as opposed to the ’administrative’ Pref).

This ES route update sent by PE3 (with [200,0,PE3-IP]) will not cause any DF switchover for any EVI/ISID. PE2 will continue being DF for EVI1. This is because the DP bit will be used as a tie-

breaker in the DF election. That is, if a PE has two candidate PEs with the same Pref, it will pick up the one with DP=1. There are no DF changes for EVI2 either.

6) Subsequently, if PE2 fails, upon receiving PE2's ES route withdrawal, PE3 and PE1 will go through the process described in (5) to select new Highest and Lowest-PEs (considering their own active ES route) and then they will run the DF Election.

   - If a PE selects itself as new Highest or Lowest-PE and it was not before, the PE will then compare its operational 'in-use' Pref with its administrative Pref. If different, the PE will send an ES route update with its administrative Pref and DP values. In the example, PE3 will be the new Highest-PE, therefore it will send an ES route update with [Pref,DP]=[300,1].

   - After running the DF Election, PE3 will become the new DF for EVI1. No changes will occur for EVI2.

5. Conclusions

Service Providers are seeking for options where the DF election can be controlled by the user in a deterministic way and with a non-revertive behavior. This document defines the use of a Preference algorithm that can be configured and used in a flexible manner to achieve those objectives.

11. Conventions used in this document

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

12. Security Considerations

This section will be added in future versions.

13. IANA Considerations

This document solicits the allocation of DF type = 2 in the registry created by [EVPN-HRW-DF] for the DF type field.

15. References

15.1 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>.

15.2 Informative References

[vES] Sajassi et al. "EVPN Virtual Ethernet Segment", draft-sajassi-bess-evpn-virtual-eth-segment-01, work-in-progress, July 6, 2015.

[EVPN-HRW-DF] Mohanty S. et al. "A new Designated Forwarder Election for the EVPN", draft-mohanty-bess-evpn-df-election-02, work-in-progress, October 19, 2015.

16. Acknowledgments

17. Contributors

In addition to the authors listed, the following individuals also contributed to this document:

Kiran Nagaraj, Nokia
Vinod Prabhu, Nokia
Selvakumar Sivaraj, Juniper

17. Authors’ Addresses

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

Senthil Sathappan
Alcatel-Lucent
Email: senthil.sathappan@nokia.com

Tony Przygienda
Juniper Networks, Inc.
Email: prz@juniper.net

John Drake
Juniper Networks, Inc.
Email: jdrake@juniper.net

Wen Lin
Juniper Networks, Inc.
Email: wlin@juniper.net

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

Satya Ranjan Mohanty
Cisco Systems, Inc.
Email: satyamoh@cisco.com

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

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
M. Vigoureux
M. Gautam
Nokia

S. Dindorkar
Nuage Networks

Expires: September 20, 2016

March 19, 2016

EVPN Generic Route Type
draft-rabadan-bess-vendor-evpn-route-00

Abstract

RFC7432 defines Ethernet VPN as a BGP address family that makes use of Typed NLRIs. IANA has a registry called "EVPN Route Types" that allocates values to Route Types. The purpose of this document is to solicit IANA the registration of a route type value for a vendor specific usage, as well as the definition of the EVPN NLRI for that route.

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/1id-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 September 20, 2016.

Table of Contents

1. Introduction

RFC7432 creates an IANA managed registry called "EVPN Route Types" and makes the initial registrations for different NLRIs. The ability to define Typed NLRIs makes EVPN a flexible and extensible technology that can be used for multiple purposes. This document solicits the value 255 for a new Route Type that will be called "EVPN Vendor Specific" Route Type.

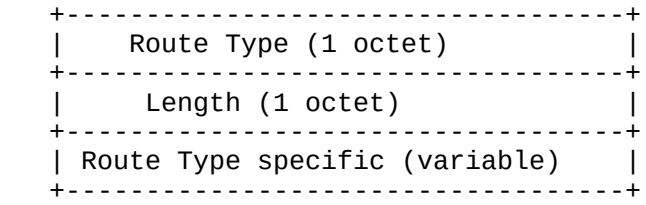
The intend of this new Type is to allow operators and vendors to design rapidly new EVPN applications/prototypes and experiment with them in deployed networks before standardizing the specific application. Software Defined Networks (SDN) are evolving fast and the flexibility allowed by this new Route Type will contribute to the SDN control plane evolution.

Another motivation for this new Route Type is the exchange of vendor specific information that may be relevant only for the vendor using it. Other vendors may convey the information in a different way, or they simply don't need to exchange it.

In order to allow multiple applications, the new NLRI contains a Organizational Unique Identifier (OUI) field for which the IEEE registers and maintains values.

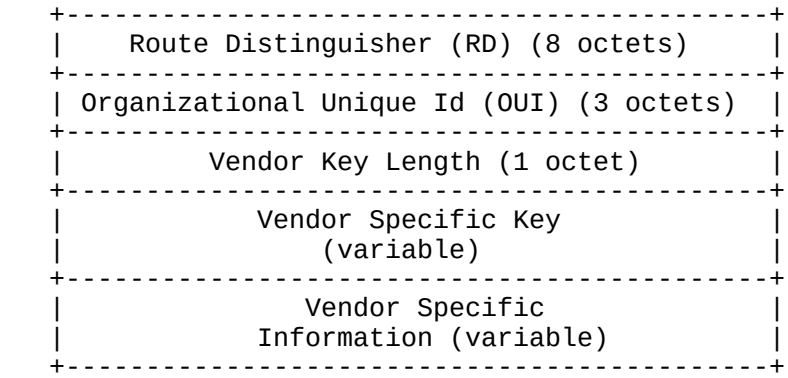
2. The EVPN Generic Route Type

[RFC7432] defines the EVPN NLRI with the following format:



Where Route Type can be a value between 0 and 255. IANA maintains a registry called "EVPN Route Types" where the different values are assigned. This document solicits a new Route Type with value 255.

When the Route Type field includes the value 255, the Route Type specific field will include the following information:



Where Route Distinguisher, OUI, Vendor Key Length and Vendor Specific Key are considered part of the route key for BGP processing. The Vendor Key Length field indicates the length in octets of the Vendor

Specific Key field of the NLRI.

The OUI values are owned and assigned by the IEEE Registration Authority.

As per [RFC7606] section 5.4, a BGP speaker advertising support for EVPN address family MUST handle routes with unrecognized NLRI types within that address family by discarding them unless the relevant specification for that address family specifies otherwise. However, a BGP speaker supporting this new Route Type MUST accept the route even if the OUI and Vendor fields are unrecognized.

3. Conventions used in this document

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

4. Security Considerations

The relevant Security Considerations described in [RFC7432] apply to the new Route Type defined in this document.

5. IANA Considerations

IANA is requested to allocate a new value in the "EVPN Route Types" registry:

255	EVPN Vendor Specific	[This document]

6. References

6.1 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>.

[RFC7606] Chen E., Ed., Scudder J., Mohapatra P. and Patel K., "Revised Error Handling for BGP UPDATE Messages", RFC 7606, August 2015, <http://www.rfc-editor.org/info/rfc7606>.

7. Acknowledgments

The authors want to thank Suresh Boddapati and Senthil Sathappan for their ideas and contributions.

8. Authors' Addresses

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

Martin Vigoureux
Nokia
Email: martin.vigoureux@nokia.com

Siddhesh Dindorkar
Nuage Networks
Email: siddhesh.dindorkar@nuagenetworks.net

Mallika Gautam
Nokia
Email: mallika.gautam@nokia.com

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
M. Vigoureux
M. Gautam
Nokia

S. Dindorkar
Nuage Networks

Expires: October 23, 2019

April 21, 2019

EVPN Vendor-Specific Route Type
draft-rabadan-bess-vendor-evpn-route-07

Abstract

RFC7432 defines Ethernet VPN as a BGP address family that makes use of Typed NLRIs. IANA has a registry called "EVPN Route Types" that allocates values to Route Types. The purpose of this document is to solicit IANA the registration of a route type value for a vendor specific usage, as well as the definition of the EVPN NLRI for that route.

This Internet-Draft will expire on October 23, 2019.

Table of Contents

1. Introduction

RFC7432 creates an IANA managed registry called "EVPN Route Types" and makes the initial registrations for different NLRIs. The ability to define Typed NLRIs makes EVPN a flexible and extensible technology that can be used for multiple purposes. This document solicits the value 255 for a new Route Type that will be called "EVPN Vendor Specific" Route Type.

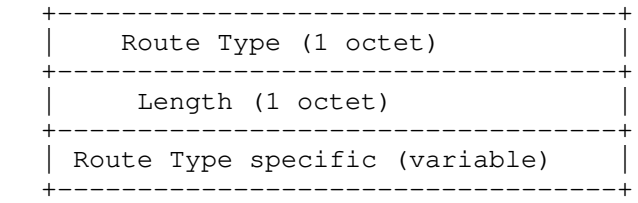
The intend of this new Type is to allow operators and vendors to design rapidly new EVPN applications/prototypes and experiment with them in deployed networks before standardizing the specific application. Software Defined Networks (SDN) are evolving fast and the flexibility allowed by this new Route Type will contribute to the SDN control plane evolution.

Another motivation for this new Route Type is the exchange of vendor specific information that may be relevant only for the vendor using it. Other vendors may convey the information in a different way, or they simply don't need to exchange it.

In order to allow multiple applications, the new NLRI contains a Organizational Unique Identifier (OUI) field for which the IEEE registers and maintains values.

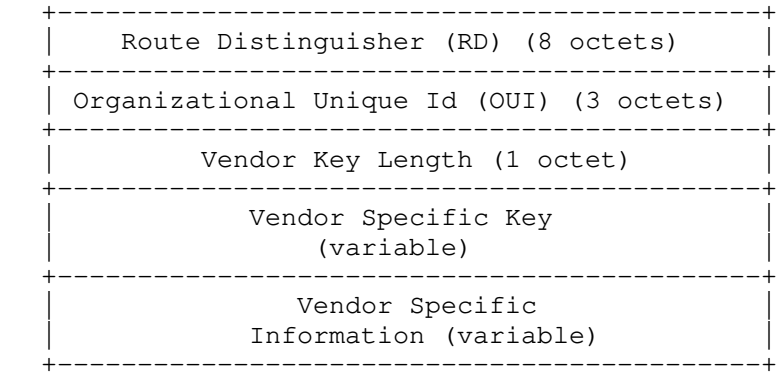
2. The EVPN Vendor-Specific Route Type

[RFC7432] defines the EVPN NLRI with the following format:



Where Route Type can be a value between 0 and 255. IANA maintains a registry called "EVPN Route Types" where the different values are assigned. This document solicits a new Route Type with value 255.

When the Route Type field includes the value 255, the Route Type specific field will include the following information:



Where OUI, Vendor Key Length and Vendor Specific Key are considered part of the route key for BGP processing. The Vendor Key Length field indicates the length in octets of the Vendor Specific Key field of

the NLRI.

The OUI values are owned and assigned by the IEEE Registration Authority.

As per [RFC7606] section 5.4, a BGP speaker advertising support for EVPN address family MUST handle routes with unrecognized NLRI types within that address family by discarding them unless the relevant specification for that address family specifies otherwise. However, a BGP speaker supporting this new Route Type MUST accept the route even if the OUI and Vendor fields are unrecognized. Specifically, a Route Reflector MUST forward this new route type to its BGP peers, even if the receiver does not understand or cannot process the route.

3. Conventions used in this document

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

4. Security Considerations

The relevant Security Considerations described in [RFC7432] apply to the new Route Type defined in this document.

5. IANA Considerations

IANA is requested to allocate a new value in the "EVPN Route Types" registry:

255	EVPN Vendor Specific	[This document]

6. References

6.1 Normative References

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

[RFC7606] Chen, E., Ed., Scudder, J., Ed., Mohapatra, P., and K.

Patel, "Revised Error Handling for BGP UPDATE Messages", RFC 7606, DOI 10.17487/RFC7606, August 2015, <https://www.rfc-editor.org/info/rfc7606>.

[RFC7606] Chen E., Ed., Scudder J., Mohapatra P. and Patel K., "Revised Error Handling for BGP UPDATE Messages", RFC 7606, August 2015, <http://www.rfc-editor.org/info/rfc7606>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

7. Acknowledgments

The authors want to thank Suresh Boddapati and Senthil Sathappan for their ideas and contributions.

8. Authors' Addresses

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

Martin Vigoureux
Nokia
Email: martin.vigoureux@nokia.com

Siddhesh Dindorkar
Nuage Networks
Email: siddhesh.dindorkar@nuagenetworks.net

Mallika Gautam
Nokia
Email: mallika.gautam@nokia.com

BESS Workgroup
INTERNET-DRAFT
Intended Status: Standards Track

Ali Sajassi
Samer Salam
Dennis Cai
Cisco

Sami Boutros
VmWare

John Drake
Juniper

Luay Jalil
Verizon

Expires: October 21, 2016 March 21, 2016

Multi-homed L3VPN Service with Single IP peer to CE
draft-sajassi-bess-evpn-l3vpn-multihoming-01

Abstract

This document describes how EVPN can be used to offer a multi-homed L3VPN service leveraging EVPN Layer 2 access redundancy. The solution offers single IP peering to the Customer Edge (CE) nodes, rapid failure detection, minimal fail-over time and make-before-break paradigm for maintenance.

Table of Contents

1 Introduction

[RFC7432] defines EVPN, a solution for multipoint Layer 2 Virtual Private Network (L2VPN) services, with advanced multi-homing capabilities, using BGP for distributing customer/client MAC address reachability information over the core MPLS/IP network. [EVPN-IRB] and [EVPN-PREFIX] discuss how EVPN can be used to support inter-subnet forwarding among hosts across different IP subnets, while maintaining the redundancy capabilities of the original solution.

In this document, we discuss how EVPN can be used to offer a multi-homed L3VPN service leveraging its Layer 2 access redundancy. The solution offers single IP peering to the Customer Edge (CE) nodes, rapid failure detection, minimal fail-over time and make-before-break paradigm for maintenance.

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

The network topology in question comprises of three domains: the customer network, the MPLS access network and the MPLS core network, as shown in the figure below.

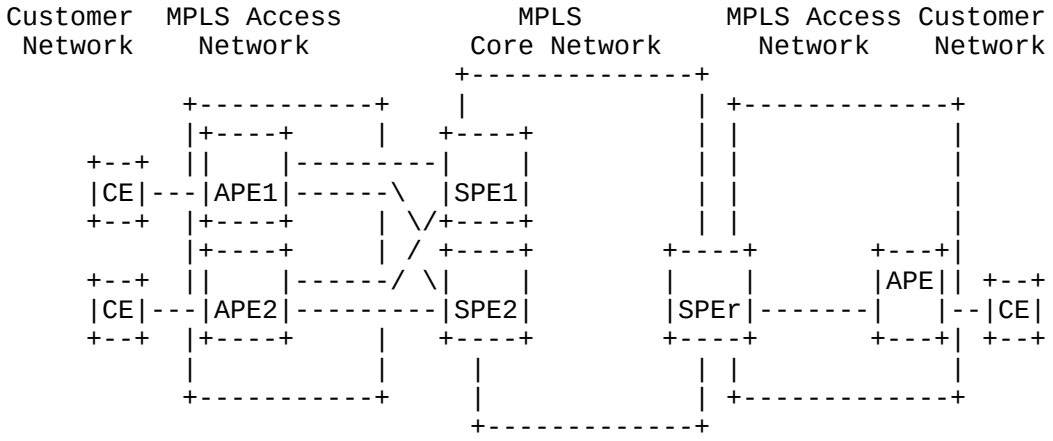


Figure 1: Network Topology

The customer network connects via Customer Edge (CE) nodes to the

MPLS Access Network. The MPLS Access Network includes Access PEs (A-PEs) and MPLS P nodes (not shown for simplicity). The A-PEs provide a Virtual Private Wire Service (VPWS) to the connected CEs using Ethernet over MPLS (EoMPLS) pseudowires per [RFC5462]. The access pseudowires terminate on the service PEs (S-PE1, S-PE2,..., S-PEr). The Service PEs (S-PEs) provide inter-subnet forwarding between the CEs, i.e. L3VPN service between them. To provide redundancy, pseudowires from a given A-PE can terminate on two or more S-PEs forming a Redundancy Group. This provide multi-homed interconnect of A-PEs to S-PEs.

The solution MUST support the following requirements:

- The S-PEs in a redundancy group must provide single-active redundancy to the CEs, i.e. only one S-PE is actively forwarding traffic at any given point of time.

- The SPEs in a redundancy group must appear as a single IP peer to the CE, and a single eBGP session will be established between a given CE and its associated S-PEs.

- In the case of S-PE failure, pseudowire failure or S-PE isolation from access network, the fail-over time should be minimized by optimizing both the backup pseudowire establishment as well as the BGP convergence time. This reduces the amount of traffic loss as the active path reroutes to one of the backup S-PEs.

- The active S-PE must be able to quickly detect pseudowire failures or its isolation from the access MPLS network by means of a proactive monitoring mechanism.

- For system maintenance, it should be possible to support a make-before-break paradigm, where the backup path is in warm standby state before a given active S-PE is taken offline for service.

3 Challenges with L3VPN Multi-homing

The requirements depicted in section 2 above, especially the requirement to maintain a single eBGP session between the CE and the S-PEs, introduce challenges for standard L3VPN multi-homing solutions. In particular, the BGP prefix independent convergence (PIC) solution [BGP-PIC] cannot be used here because the backup S-PEs have no means of learning the IP prefixes from the CE: recall that the CE will only have an active eBGP session with the active S-PE. As a result, when the primary S-PE fails, the backup S-PE will have no alternate paths to the prefixes advertised by the CE. Therefore, with BGP PIC it is not possible to address the fast fail-over requirement.

4 Solution

4.1 Using Pseudowires in Access Network

The solution involves running EVPN on the S-PEs in single-active redundancy mode albeit for inter-subnet forwarding (i.e. Layer 3 forwarding). All pseudowires associated with a given CE are considered collectively as a Virtual Ethernet Segment (vES) [Virtual-ES] from the EVPN PEs perspective.

In the MPLS access network, pseudowire redundancy mechanisms are used [RFC6718][RFC6870] in either the Independent mode or the Master/Slave mode, with the S-PEs acting as the Master. The EVPN Designated Forwarder (DF) election mechanism is used to identify the active and standby S-PEs, and the pseudowire Preferential Forwarding Status Bit [RFC6870], for the access pseudowires, is derived from the outcome of the DF election, as follows:

- The S-PE that is elected as DF for a given vES MUST advertise Active in the Preferential Forwarding Status bit over the pseudowire corresponding to the vES.

- The SPE that is elected as non-DF for a given vES MUST advertise Standby in the Preferential Forwarding Status bit over the pseudowire corresponding to the vES.

On the S-PEs, the pseudowires from the Access PEs are terminated onto VRFs, such that all pseudowires within a given redundancy set terminate on a single IP endpoint on the S-PEs. To achieve this, the S-PEs in a given Redundancy Group are configured with the same Anycast IP and MAC addresses on the virtual (sub)interface corresponding to the VRF termination point.

Since the S-PEs are running in EVPN single-active redundancy mode, the S-PEs would advertise an Ethernet AD route per vES with the single-active flag set per [RFC7432]. Furthermore, the DF PE sets the Primary bit in the L2 extended community and the backup PE set the Backup bit in that extended community. Since only the DF S-PE has its access pseudowire in Active state, only that device would establish an eBGP session with the CE and receive control and data traffic. The DF S-PE advertises host prefixes that it receives, from the CE over the eBGP session, to other PEs in the EVI using EVPN route type-5, with the proper ESI set. Remote PEs learn the host prefixes and associate them with the ESI, using the advertising PE as the next-hop for forwarding.

Other S-PEs in the same Redundancy Group as the advertising PE will receive the same EVPN route type-5 advertisement, and will recognize

the associated ESI as a locally attached vES. This information will be used in the case of failure to provide a backup path to the CE. In other words, the S-PEs in the same Redundancy Group, use EVPN Aliasing procedure to synchronzie their IP-VRFs among themselves. It is worth noting here that the S-PEs in the Redundancy Group will have their ARP caches synchronized through the EVPN route type-2 advertisements from the DF PE.

4.2 Using EVPN-VPWS in Access Network

[EVPN-VPWS] can be used instead of pseudo wires in the MPLS access network, in that case all EVPN-VPWS service instances associated with a given CE are considered collectively as a Virtual Ethernet Segment (vES) [Virtual-ES].

The elected DF S-PE MUST set the Primary bit in the L2 attributes extended community associated with the EVPN-VPWS service instance Ethernet A-D route, corresponding to the vES. The non-DF S-PEs MUST set the Backup bit in the L2 attributes extended community associated with the EVPN-VPWS service instance Ethernet A-D route, corresponding to the vES.

Just as with pseudowires described in previous section, only the DF S-PE has its access EVPN-VPWS service instance in Active state, and thus establishes an eBGP session with the CE and receive control and data traffic. Just as before, the DF S-PE advertises host prefixes that it receives, from the CE over the eBGP session, to other PEs in the EVI using EVPN route type-5, with the proper ESI set. Remote PEs learn the host prefixes and associate them with the ESI, using the advertising PE as the next-hop for forwarding.

5 Failure Scenarios

5.1 Pseudowire Failure

The active (DF) S-PE can proactively monitor the health of the primary pseudowire by using a pseudowire OAM mechanism such as VCCV-BFD. As such, the S-PE can detect the failure of the primary pseudowire, and react by withdrawing both the Ethernet Segment route as well as the Ethernet A-D route associated with the vES. Note that the S-PE advertises the Ethernet A-D route per vES granularity as well as the Ethernet A-D per EVI. The withdrawal of the Ethernet Segment route serves as an indication to the backup S-PE to go active (i.e. act as a backup DF), and activate its pseudowires to the Access PE. The withdrawal of the Ethernet A-D route triggers a "mass withdraw" on the remote PEs: these PEs adjust their next-hop associated with the prefixes that were originally advertised by the

failed PE to point to the "backup path" per [RFC7432]. This provides relatively fast convergence because only a single message per Ethernet Segment is required for the remote PEs to switch over to the backup path irrespective of how many prefixes were learnt from the CE over the pseudowire. Also, note that no synchronization of VRF or ARP tables is required between the primary S-PE and its backup S-PE during the fail-over, because these tables were populated ahead of time during the original EVPN route advertisements.

As a result of the pseudowire failure, the eBGP session between the CE and the original DF PE will time out. This will cause said S-PE to start a timer in order to defer withdrawing the EVPN type-5 and type-2 routes that it had advertised for the prefixes learnt over the session from the CE. As the backup pseudowire to the backup DF PE goes active, the eBGP session will be re-established by the CE with the backup PE. Since both PEs share the same Anycast IP and MAC addresses, the CE does not recognize that it is in communication with a different PE.

To minimize disruption in data forwarding on the CE and the backup PE, the non-stop forwarding feature such as BGP Graceful Restart is used. Since the end-point IP address has not changed, this eBGP session handover between the primary S-PE and the backup S-PE, looks like a eBGP session flap with respect to the CE. Thus, the CE continues its packet forwarding operation in data-plane while synchronizing its control-plane with the backup S-PE.

5.2 EVPN VPWS Service Instance Failure

The failure scenario for an EVPN VPWS in similar to PW failure scenario described in the previous section. The failure detection of an EVPN service instance 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.

5.3 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.

6 Security Considerations

TBD.

7 IANA Considerations

TBD

8 References

8.1 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.

8.2 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

Ali Sajassi
Cisco
EMail: sajassi@cisco.com

Samer Salam
Cisco
EMail: ssalam@cisco.com

Dennis Cai
Cisco
EMail: dcai@cisco.com

John Drake
Juniper
EMail: jdrake@juniper.net

Luay Jalil
Verizon
EMail: luayjalil@gmail.com

Sami Boutros
VmWare
EMail: boutros.sami@gmail.com

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2016

H. Shah
Ciena Corporation
P. Brissette
R. Rahman
K. Raza
Cisco Systems, Inc.
Z. Li
Z. Shunwan
W. Haibo
Huawei Technologies
I. Chen
S. Ahmed
Ericsson
M. Bocci
Alcatel-Lucent
J. Hardwick
Metaswitch
S. Esale
K. Tiruveedhula
T. Singh
Juniper Networks
I. Hussain
Infinera Corporation
B. Wen
J. Walker
Comcast
N. Delregno
L. Jalil
M. Joecylyn
Verizon

March 14, 2016

YANG Data Model for MPLS-based L2VPN
draft-shah-bess-l2vpn-yang-01.txt

Abstract

This document describes a YANG data model for Layer 2 VPN services over MPLS networks. These services include Virtual Private Wire Service (VPWS) and Virtual Private LAN service (VPLS) that uses LDP and BGP signaled Pseudowires. The current version of the document expands the L2VPN object model to include VPLS services in addition to the VPWS services described in the last revision. This is a living document and contains aspects of object models that have been discussed extensively in the working group with consensus. The intention is to continue to seek input from larger audience during evolution of the L2VPN service model through this document.

Status of This Memo

Copyright Notice

Table of Contents

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 MPLS based Layer 2 VPN services (L2VPN) [RFC4664] as well as switching between the local attachment circuits. The L2VPN services include point-to-point VPWS and Multipoint VPLS services. These services are realized by signaling Pseudowires across MPLS networks using LDP [RFC4447][RFC4762] or BGP[RFC4761].

The YANG data model in this document defines Ethernet based Layer 2 services. 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 Ethernet based Layer 2 services will leverage the definitions used in other standards organizations such as IEEE 802.1 and Metro Ethernet Forum (MEF).

The goal is to propose a data object model consisting of building blocks that can be assembled in different order to realize different services. The definition work is undertaken initially by a smaller working group with members representing various vendors and service providers. The VPWS service definitions were covered first in the last revision of the document. The current version documents VPLS services that build on the data blocks defined for VPWS.

In the current version of this document, refinements to the configuration objects and Operational State objects for the same are added.

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

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

The document is organized to first define the data model for the configuration of all the L2VPN services followed by definition of operational state, actions and notifications for the same. The L2VPN data object model defined in this document uses the instance centric approach. The attributes of each service, VPWS, VPLS, etc are specified for a given service instance.

2. Updates in this revision

The organization of the configuration objects has been updated. The ac-templates in the common container is removed and a new redundancy-group-templates is added.

The vpls-instances container is removed and replaced with bridge-table-instances container to include the PBB, BGP parameters. This revision also introduces a reference to EVPN instance. This revision removes the definition of Attachment Circuits, "ac". Instead, the L2VPN data object model will rely on standard definitions of Attachment Circuits that IEEE and IETF are coordinating to define. Thus, this revision uses a string as a placeholder for an Attachment Circuit within the ac-or-pw-redundancy-grp within the respective endpoints list of bridge-table-instances or VPWS instances, and expect to update this field once the standard definitions of Attachment Circuits are available.

3. 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].

4. L2VPN YANG Model

4.1. Overview

One single top level container, l2vpn, is defined as a parent for three different second level containers that are vpws-instances, bridge-table-instances, and common building blocks of redundancy-grp templates and pseudowire-templates. The current version of the document is extended to include refinements to configuration of vpws-instance and bridge-table-instances. The operations state object has been added to hold read-only information of objects that has either been configured or dynamically created.

The L2VPN services have been defined in the IETF L2VPN working group but leverages the pseudowire technologies that were defined in 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:

- Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3) [RFC3916]
- Pseudo-wire Emulation Edge-to-Edge (PWE3) Architecture [RFC3985]
- IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3) [RFC4446]
- Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) [RFC4447]
- Encapsulation Methods for Transport of Ethernet over MPLS Networks [RFC4448]
- Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN [RFC4385]
- Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3) [RFC5254]
- An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge [RFC5659]
- Segmented Pseudowire [RFC6073]
- Framework for Layer 2 Virtual Private Networks [RFC4664]

- Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks [RFC4665]
- Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling [RFC4761]
- Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling [RFC4762]
- Attachment Individual Identifier (AII) Types for Aggregation [RFC5003]
- Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs) [RFC6074]
- Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network [RFC6391]
- Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling [RFC6624]
- Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging [RFC7041]
- LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS) [RFC7361]
- Using the generic associated channel label for Pseudowire in the MPLS Transport Profile [RFC6423]
- Pseudowire status for static pseudowire [RFC6478]

Note that while pseudowire over MPLS-TP related work is in scope, the initial effort will only address definitions of object models for services 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.

```
 template-ref PW // PW
                    template
                    attributes

 template-ref Redundancy-Group // redundancy-group
```

```
                    template
                    attributes


 bridge-table-instance name // container

         common attributes

         PBB-parameters // container
                           pbb specific attributes

         BGP-parameters // container
                           common attributes
                           auto-discovery attributes
                           signaling attributes

         evpn-instance // reference

         // list of PWs being used
         PW // container
                 template-ref PW
                 attribute-override

         // List of endpoints, where each member endpoint container is -
                 PW // reference
                 redundancy-grp // container
                         AC // eventual reference to standard AC
                         PW // reference

 vpws-instance name // 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
                 pw type
                    static-or-ldp
                    bgp-pw
                    bgp-ad-pw

         // ONLY 2 endpoints!!!
```

```
         endpoint-A // container
                 redundancy-grp // container
                         AC // eventual reference to standard AC
                         PW // reference


         endpoint-Z // container
                 redundancy-grp // container
                         AC // eventual reference to standard AC
                         PW // reference

 l2vpn-state // read-only container
```

Figure 1

4.2. L2VPN Common

4.2.1. ac-templates

The ac-templates container is removed. The AC will eventually reference standard AC definitions defined with coordination between the IEEE and IETF, and will inherit all the attributes defined in that reference.

4.2.2. 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.

4.3. VPWS and Bridge-Table-Instance (formerly referred as VPLS)

4.3.1. ac list

AC resides within endpoint container as member of ac-or-pw-or-redundandancy-grp.

4.3.2. pw list

Each VPWS and Bridge-Table-Instance defines a list of PWs which are participating members of the given service instance. Each entry of the PW consists of one pw-template with pre-defined attributes and values, but also defines attributes that override those defined in referenced pw-template.

No restrictions are placed on type of signaling (i.e. LDP or BGP) used for a given PW. It is entirely possible to define two PWs, one signaled by LDP and other by BGP.

The VPLS specific attribute(s) are present in the definition of the PW that are member of VPLS instance only and not applicable to VPWS service.

4.3.3. 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 -

- primary and backup AC
- primary and backup PW
- primary AC and backup PW
- primary PW and backup AC

4.3.4. endpoint container

The endpoint container in general holds 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 container for the VPLS service holds references to a list of ACs, a list of PWs or a redundancy group that contains a list of ACs and/or a list of PWs. This differs from the VPWS instance where an endpoint contains exactly one member; AC or PW or redundancy group and not a list.

4.3.5. vpws-instances and bridge-table-instances container

The vpws-instances container contains a list of vpws-instance. Each entry of the vpws-instance represents a layer-2 cross-connection of two endpoints. This model defines three possible types of endpoints, ac, pw, and redundancy-grp, and allows a vpws-instance to cross-connect any one type of endpoint to all other types of endpoint.

The bridge-table-instances container contains a list of bridge-table-instance. Each entry of the bridge-table-instance represent a list of endpoints that are member of the broadcast/bridge domain. The

bridge-table-instance endpoints introduces an additional forwarding characteristics to a list of PWs and/or ACs. This split-horizon forwarding behavior is typical in bridge-table instance.

The augmentation of ietf-l2vpn module is TBD. All IP addresses defined in this module are currently scoped under global VRF/table.

4.4. Operational State

The operational state of L2VPN can be queried and obtained from the read-only container defined in this document as "l2vpn-state". This container holds the runtime information of the bridge-table-instance and vpws-instance.

4.5. Open items

The design team has identified several attributes that need to be included in the YANG tree. These attributes are listed here for the purpose of keeping track and are candidates for future revisions.

These attributes are :

- configuration - vccv configuration knobs
- operational state - vccv and cw-negotiation

This list is not exhaustive and expected to grow. The list will shrink as items are processed and included in the YANG tree.

4.6. Yang tree

```
module: ietf-l2vpn
   +--rw l2vpn
   |  +--rw common
   |  |  +--rw pw-templates
   |  |  |  +--rw pw-template* [name]
   |  |  |     +--rw name              string
   |  |  |     +--rw mtu?              uint32
   |  |  |     +--rw cw-negotiation?   cw-negotiation-type
   |  |  |     +--rw tunnel-policy?    string
   |  |  +--rw redundancy-group-templates
   |  |     +--rw redundancy-group-template* [name]
   |  |        +--rw name               string
   |  |        +--rw protection-mode?   enumeration
   |  |        +--rw reroute-mode?      enumeration
   |  |        +--rw reroute-delay?     uint16
   |  |        +--rw dual-receive?      boolean
   |  |        +--rw revert?            boolean
```

```
   |  |        +--rw revert-delay?      uint16
   |  +--rw bridge-table-instances
   |  |  +--rw bridge-table-instance* [name]
   |  |     +--rw name               string
   |  |     +--rw mtu?               uint32
   |  |     +--rw mac-aging-timer?   uint32
   |  |     +--rw pbb-parameters
   |  |     |  +--rw (component-type)?
   |  |     |     +--:(i-component)
   |  |     |     |  +--rw i-tag?              uint32
   |  |     |     |  +--rw backbone-src-mac?   yang:mac-address
   |  |     |     +--:(b-component)
   |  |     |        +--rw bind-b-component?   bridge-table-instance-ref
   |  |     +--rw bgp-parameters
   |  |     |  +--rw common
   |  |     |  |  +--rw route-distinguisher?   string
   |  |     |  |  +--rw vpn-targets* [rt-value]
   |  |     |  |     +--rw rt-value    string
   |  |     |  |     +--rw rt-type     bgp-rt-type
   |  |     |  +--rw discovery
   |  |     |  |  +--rw vpn-id?   string
   |  |     |  +--rw signaling
   |  |     |     +--rw site-id?      uint16
   |  |     |     +--rw site-range?   uint16
   |  |     +--rw evpn-instance?     string
   |  |     +--rw pw* [name]
   |  |     |  +--rw name              string
   |  |     |  +--rw template?         pw-template-ref
   |  |     |  +--rw mtu?              uint32
   |  |     |  +--rw mac-withdraw?     boolean
   |  |     |  +--rw cw-negotiation?   cw-negotiation-type
   |  |     |  +--rw discovery-type?   l2vpn-discovery-type
   |  |     |  +--rw signaling-type?   l2vpn-signaling-type
   |  |     |  +--rw peer-ip?          inet:ip-address
   |  |     |  +--rw pw-id?            uint32
   |  |     |  +--rw transmit-label?   uint32
   |  |     |  +--rw receive-label?    uint32
   |  |     |  +--rw tunnel-policy?    string
   |  |     +--rw endpoint* [id]
   |  |        +--rw id                      uint16
   |  |        +--rw split-horizon-group?   string
   |  |        +--rw (ac-or-pw-or-redundancy-grp)?
   |  |           +--:(ac)
   |  |           |  +--rw ac* [name]
   |  |           |     +--rw name    string
   |  |           +--:(pw)
   |  |           |  +--rw pw* [name]
   |  |           |     +--rw name    -> ../../../pw/name
```

```
   |  |           +--:(redundancy-grp)
   |  |              +--rw (primary)
   |  |              |  +--:(primary-pw)
   |  |              |  |  +--rw primary-pw* [name]
   |  |              |  |     +--rw name    -> ../../../pw/name
   |  |              |  +--:(primary-ac)
   |  |              |     +--rw primary-ac?            string
   |  |              +--rw (backup)?
   |  |              |  +--:(backup-pw)
   |  |              |  |  +--rw backup-pw* [name]
   |  |              |  |     +--rw name          -> ../../../pw/name
   |  |              |  |     +--rw precedence?   uint32
   |  |              |  +--:(backup-ac)
   |  |              |     +--rw backup-ac?            string
   |  |              +--rw template?              -> /l2vpn/common/redundancy-gr
oup-templates/redundancy-group-template/name
   |  |              +--rw protection-mode?       enumeration
   |  |              +--rw reroute-mode?          enumeration
   |  |              +--rw reroute-delay?         uint16
   |  |              +--rw dual-receive?          boolean
   |  |              +--rw revert?                boolean
   |  |              +--rw revert-delay?          uint16
   |  +--rw vpws-instances
   |     +--rw vpws-instance* [name]
   |        +--rw name               string
   |        +--rw description?       string
   |        +--rw mtu?               uint32
   |        +--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 common
   |        |  |  +--rw route-distinguisher?   string
   |        |  |  +--rw vpn-targets* [rt-value]
   |        |  |     +--rw rt-value    string
   |        |  |     +--rw rt-type     bgp-rt-type
   |        |  +--rw discovery
   |        |  |  +--rw vpn-id?   string
   |        |  +--rw signaling
   |        |     +--rw site-id?      uint16
   |        |     +--rw site-range?   uint16
   |        +--rw pw* [name]
   |        |  +--rw name              string
   |        |  +--rw template?         pw-template-ref
   |        |  +--rw mtu?              uint32
   |        |  +--rw mac-withdraw?     boolean
   |        |  +--rw cw-negotiation?   cw-negotiation-type
   |        |  +--rw vccv-ability?     boolean
```

```
   |        |  +--rw tunnel-policy?    string
   |        |  +--rw request-vlanid?   uint16
   |        |  +--rw vlan-tpid?        string
   |        |  +--rw ttl?              uint8
   |        |  +--rw (pw-type)?
   |        |     +--:(ldp-or-static-pw)
   |        |     |  +--rw peer-ip?          inet:ip-address
   |        |     |  +--rw pw-id?            uint32
   |        |     |  +--rw icb?              boolean
   |        |     |  +--rw transmit-label?   uint32
   |        |     |  +--rw receive-label?    uint32
   |        |     +--:(bgp-pw)
   |        |     |  +--rw remote-pe-id?     inet:ip-address
   |        |     +--:(bgp-ad-pw)
   |        |        +--rw remote-ve-id?     uint16
   |        +--rw endpoint-a
   |        |  +--rw (ac-or-pw-or-redundancy-grp)?
   |        |     +--:(ac)
   |        |     |  +--rw ac?                string
   |        |     +--:(pw)
   |        |     |  +--rw pw?                -> ../../pw/name
   |        |     +--:(redundancy-grp)
   |        |        +--rw (primary)
   |        |        |  +--:(primary-pw)
   |        |        |  |  +--rw primary-pw?        -> ../../pw/name
   |        |        |  +--:(primary-ac)
   |        |        |     +--rw primary-ac?        string
   |        |        +--rw (backup)
   |        |        |  +--:(backup-pw)
   |        |        |  |  +--rw backup-pw?         -> ../../pw/name
   |        |        |  +--:(backup-ac)
   |        |        |     +--rw backup-ac?         string
   |        |        +--rw template?           -> /l2vpn/common/redundancy-group-
templates/redundancy-group-template/name
   |        |        +--rw protection-mode?    enumeration
   |        |        +--rw reroute-mode?       enumeration
   |        |        +--rw reroute-delay?      uint16
   |        |        +--rw dual-receive?       boolean
   |        |        +--rw revert?             boolean
   |        |        +--rw revert-delay?       uint16
   |        +--rw endpoint-z
   |           +--rw (ac-or-pw-or-redundancy-grp)?
   |              +--:(ac)
   |              |  +--rw ac?                string
   |              +--:(pw)
   |              |  +--rw pw?                -> ../../pw/name
   |              +--:(redundancy-grp)
   |                 +--rw (primary)
   |                 |  +--:(primary-pw)
```

```
   |                |  |  +--rw primary-pw?        -> ../../pw/name
   |                |  +--:(primary-ac)
   |                |     +--rw primary-ac?        string
   |                +--rw (backup)
   |                |  +--:(backup-pw)
   |                |  |  +--rw backup-pw?         -> ../../pw/name
   |                |  +--:(backup-ac)
   |                |     +--rw backup-ac?         string
   |                +--rw template?          -> /l2vpn/common/redundancy-group-
templates/redundancy-group-template/name
   |                +--rw protection-mode?   enumeration
   |                +--rw reroute-mode?      enumeration
   |                +--rw reroute-delay?     uint16
   |                +--rw dual-receive?      boolean
   |                +--rw revert?            boolean
   |                +--rw revert-delay?      uint16
   +--ro l2vpn-state
      +--ro bridge-table-instances-state
      |  +--ro bridge-table-instance-state* [name]
      |     +--ro name                  string
      |     +--ro mtu?                  uint32
      |     +--ro mac-aging-timer?      uint32
      |     +--ro pbb-parameters
      |     |  +--ro (component-type)?
      |     |     +--:(i-component)
      |     |     |  +--ro i-tag?              uint32
      |     |     |  +--ro backbone-src-mac?   yang:mac-address
      |     |     +--:(b-component)
      |     |        +--ro bind-b-component?   string
      |     +--ro bgp-parameters
      |     |  +--ro common
      |     |  |  +--ro route-distinguisher?   string
      |     |  |  +--ro vpn-targets* [rt-value]
      |     |  |     +--ro rt-value    string
      |     |  |     +--ro rt-type     bgp-rt-type
      |     |  +--ro discovery
      |     |  |  +--ro vpn-id?   string
      |     |  +--ro signaling
      |     |     +--ro site-id?      uint16
      |     |     +--ro site-range?   uint16
      |     +--ro evpn-instance-name?   string
      |     +--ro endpoint* [id]
      |        +--ro id                     uint16
      |        +--ro split-horizon-group?   string
      |        +--ro (ac-or-pw-or-redundancy-grp)?
      |           +--:(ac)
      |           |  +--ro ac* [name]
      |           |     +--ro name     string
      |           |     +--ro state?   operational-state-type
```

```
|           +--:(pw)
|           |  +--ro pw* [name]
|           |     +--ro name              string
|           |     +--ro state?            operational-state-type
|           |     +--ro mtu?              uint32
|           |     +--ro mac-withdraw?     boolean
|           |     +--ro cw-negotiation?   cw-negotiation-type
|           |     +--ro discovery-type?   l2vpn-discovery-type
|           |     +--ro signaling-type?   l2vpn-signaling-type
|           |     +--ro peer-ip?          inet:ip-address
|           |     +--ro pw-id?            uint32
|           |     +--ro transmit-label?   uint32
|           |     +--ro receive-label?    uint32
|           |     +--ro tunnel-policy?    string
|           +--:(redundancy-grp)
|              +--ro (primary)
|              |  +--:(primary-pw)
|              |  |  +--ro primary-pw* [name]
|              |  |     +--ro name              string
|              |  |     +--ro state?            operational-state-type
|              |  |     +--ro mtu?              uint32
|              |  |     +--ro mac-withdraw?     boolean
|              |  |     +--ro cw-negotiation?   cw-negotiation-type
|              |  |     +--ro discovery-type?   l2vpn-discovery-type
|              |  |     +--ro signaling-type?   l2vpn-signaling-type
|              |  |     +--ro peer-ip?          inet:ip-address
|              |  |     +--ro pw-id?            uint32
|              |  |     +--ro transmit-label?   uint32
|              |  |     +--ro receive-label?    uint32
|              |  |     +--ro tunnel-policy?    string
|              |  +--:(primary-ac)
|              |     +--ro primary-ac
|              |        +--ro name?    string
|              |        +--ro state?   operational-state-type
|              +--ro (backup)?
|              |  +--:(backup-pw)
|              |  |  +--ro backup-pw* [name]
|              |  |     +--ro name              string
|              |  |     +--ro state?            operational-state-type
|              |  |     +--ro mtu?              uint32
|              |  |     +--ro mac-withdraw?     boolean
|              |  |     +--ro cw-negotiation?   cw-negotiation-type
|              |  |     +--ro discovery-type?   l2vpn-discovery-type
|              |  |     +--ro signaling-type?   l2vpn-signaling-type
|              |  |     +--ro peer-ip?          inet:ip-address
|              |  |     +--ro pw-id?            uint32
|              |  |     +--ro transmit-label?   uint32
|              |  |     +--ro receive-label?    uint32
```

```
|              |  |     +--ro tunnel-policy?    string
|              |  |     +--ro precedence?       uint32
|              |  +--:(backup-ac)
|              |     +--ro backup-ac
|              |        +--ro name?    string
|              |        +--ro state?   operational-state-type
|              +--ro protection-mode?       enumeration
|              +--ro reroute-mode?          enumeration
|              +--ro reroute-delay?         uint16
|              +--ro dual-receive?          boolean
|              +--ro revert?                boolean
|              +--ro revert-delay?          uint16
+--ro vpws-instances-state
   +--ro vpws-instance-state* [name]
      +--ro name               string
      +--ro mtu?               uint32
      +--ro mac-aging-timer?   uint32
      +--ro service-type?      l2vpn-service-type
      +--ro discovery-type?    l2vpn-discovery-type
      +--ro signaling-type     l2vpn-signaling-type
      +--ro bgp-parameters
      |  +--ro common
      |  |  +--ro route-distinguisher?   string
      |  |  +--ro vpn-targets* [rt-value]
      |  |     +--ro rt-value    string
      |  |     +--ro rt-type     bgp-rt-type
      |  +--ro discovery
      |  |  +--ro vpn-id?   string
      |  +--ro signaling
      |     +--ro site-id?      uint16
      |     +--ro site-range?   uint16
      +--ro endpoint-a
      |  +--ro (ac-or-pw-or-redundancy-grp)?
      |     +--:(ac)
      |     |  +--ro ac
      |     |     +--ro name?    string
      |     |     +--ro state?   operational-state-type
      |     +--:(pw)
      |     |  +--ro pw
      |     |     +--ro name?             string
      |     |     +--ro state?            operational-state-type
      |     |     +--ro mtu?              uint32
      |     |     +--ro mac-withdraw?     boolean
      |     |     +--ro cw-negotiation?   cw-negotiation-type
      |     |     +--ro vccv-ability?     boolean
      |     |     +--ro tunnel-policy?    string
      |     |     +--ro request-vlanid?   uint16
      |     |     +--ro vlan-tpid?        string
```

```
|     |     +--ro ttl?              uint8
|     |     +--ro (pw-type)?
|     |        +--:(ldp-or-static-pw)
|     |        |  +--ro peer-ip?          inet:ip-address
|     |        |  +--ro pw-id?            uint32
|     |        |  +--ro icb?              boolean
|     |        |  +--ro transmit-label?   uint32
|     |        |  +--ro receive-label?    uint32
|     |        +--:(bgp-pw)
|     |        |  +--ro remote-pe-id?     inet:ip-address
|     |        +--:(bgp-ad-pw)
|     |           +--ro remote-ve-id?     uint16
|     +--:(redundancy-grp)
|        +--ro (primary)
|        |  +--:(primary-pw)
|        |  |  +--ro primary-pw
|        |  |     +--ro name?             string
|        |  |     +--ro state?            operational-state-type
|        |  |     +--ro mtu?              uint32
|        |  |     +--ro mac-withdraw?     boolean
|        |  |     +--ro cw-negotiation?   cw-negotiation-type
|        |  |     +--ro vccv-ability?     boolean
|        |  |     +--ro tunnel-policy?    string
|        |  |     +--ro request-vlanid?   uint16
|        |  |     +--ro vlan-tpid?        string
|        |  |     +--ro ttl?              uint8
|        |  |     +--ro (pw-type)?
|        |  |        +--:(ldp-or-static-pw)
|        |  |        |  +--ro peer-ip?          inet:ip-address
|        |  |        |  +--ro pw-id?            uint32
|        |  |        |  +--ro icb?              boolean
|        |  |        |  +--ro transmit-label?   uint32
|        |  |        |  +--ro receive-label?    uint32
|        |  |        +--:(bgp-pw)
|        |  |        |  +--ro remote-pe-id?     inet:ip-address
|        |  |        +--:(bgp-ad-pw)
|        |  |           +--ro remote-ve-id?     uint16
|        |  +--:(primary-ac)
|        |     +--ro primary-ac-name?   string
|        +--ro (backup)
|        |  +--:(backup-pw)
|        |  |  +--ro backup-pw
|        |  |     +--ro name?             string
|        |  |     +--ro state?            operational-state-type
|        |  |     +--ro mtu?              uint32
|        |  |     +--ro mac-withdraw?     boolean
|        |  |     +--ro cw-negotiation?   cw-negotiation-type
|        |  |     +--ro vccv-ability?     boolean
```

```
|        |  |     +--ro tunnel-policy?    string
|        |  |     +--ro request-vlanid?   uint16
|        |  |     +--ro vlan-tpid?        string
|        |  |     +--ro ttl?              uint8
|        |  |     +--ro (pw-type)?
|        |  |        +--:(ldp-or-static-pw)
|        |  |        |  +--ro peer-ip?          inet:ip-address
|        |  |        |  +--ro pw-id?            uint32
|        |  |        |  +--ro icb?              boolean
|        |  |        |  +--ro transmit-label?   uint32
|        |  |        |  +--ro receive-label?    uint32
|        |  |        +--:(bgp-pw)
|        |  |        |  +--ro remote-pe-id?     inet:ip-address
|        |  |        +--:(bgp-ad-pw)
|        |  |           +--ro remote-ve-id?     uint16
|        |  +--:(backup-ac)
|        |     +--ro backup-ac-name?    string
|        +--ro protection-mode?   enumeration
|        +--ro reroute-mode?      enumeration
|        +--ro reroute-delay?     uint16
|        +--ro dual-receive?      boolean
|        +--ro revert?            boolean
|        +--ro revert-delay?      uint16
+--ro endpoint-z
   +--ro (ac-or-pw-or-redundancy-grp)?
      +--:(ac)
      |  +--ro ac
      |     +--ro name?    string
      |     +--ro state?   operational-state-type
      +--:(pw)
      |  +--ro pw
      |     +--ro name?             string
      |     +--ro state?            operational-state-type
      |     +--ro mtu?              uint32
      |     +--ro mac-withdraw?     boolean
      |     +--ro cw-negotiation?   cw-negotiation-type
      |     +--ro vccv-ability?     boolean
      |     +--ro tunnel-policy?    string
      |     +--ro request-vlanid?   uint16
      |     +--ro vlan-tpid?        string
      |     +--ro ttl?              uint8
      |     +--ro (pw-type)?
      |        +--:(ldp-or-static-pw)
      |        |  +--ro peer-ip?          inet:ip-address
      |        |  +--ro pw-id?            uint32
      |        |  +--ro icb?              boolean
      |        |  +--ro transmit-label?   uint32
      |        |  +--ro receive-label?    uint32
```

```
|        +--:(bgp-pw)
|        |  +--ro remote-pe-id?     inet:ip-address
|        +--:(bgp-ad-pw)
|           +--ro remote-ve-id?     uint16
+--:(redundancy-grp)
   +--ro (primary)
   |  +--:(primary-pw)
   |  |  +--ro primary-pw
   |  |     +--ro name?             string
   |  |     +--ro state?            operational-state-type
   |  |     +--ro mtu?              uint32
   |  |     +--ro mac-withdraw?     boolean
   |  |     +--ro cw-negotiation?   cw-negotiation-type
   |  |     +--ro vccv-ability?     boolean
   |  |     +--ro tunnel-policy?    string
   |  |     +--ro request-vlanid?   uint16
   |  |     +--ro vlan-tpid?        string
   |  |     +--ro ttl?              uint8
   |  |     +--ro (pw-type)?
   |  |        +--:(ldp-or-static-pw)
   |  |        |  +--ro peer-ip?          inet:ip-address
   |  |        |  +--ro pw-id?            uint32
   |  |        |  +--ro icb?              boolean
   |  |        |  +--ro transmit-label?   uint32
   |  |        |  +--ro receive-label?    uint32
   |  |        +--:(bgp-pw)
   |  |        |  +--ro remote-pe-id?     inet:ip-address
   |  |        +--:(bgp-ad-pw)
   |  |           +--ro remote-ve-id?     uint16
   |  +--:(primary-ac)
   |     +--ro primary-ac-name?   string
   +--ro (backup)
   |  +--:(backup-pw)
   |  |  +--ro backup-pw
   |  |     +--ro name?             string
   |  |     +--ro state?            operational-state-type
   |  |     +--ro mtu?              uint32
   |  |     +--ro mac-withdraw?     boolean
   |  |     +--ro cw-negotiation?   cw-negotiation-type
   |  |     +--ro vccv-ability?     boolean
   |  |     +--ro tunnel-policy?    string
   |  |     +--ro request-vlanid?   uint16
   |  |     +--ro vlan-tpid?        string
   |  |     +--ro ttl?              uint8
   |  |     +--ro (pw-type)?
   |  |        +--:(ldp-or-static-pw)
   |  |        |  +--ro peer-ip?          inet:ip-address
   |  |        |  +--ro pw-id?            uint32
```

```
                    |  |        |  +--ro icb?              boolean
                    |  |        |  +--ro transmit-label?   uint32
                    |  |        |  +--ro receive-label?    uint32
                    |  |        +--:(bgp-pw)
                    |  |        |  +--ro remote-pe-id?     inet:ip-address
                    |  |        +--:(bgp-ad-pw)
                    |  |           +--ro remote-ve-id?     uint16
                    |  +--:(backup-ac)
                    |     +--ro backup-ac-name?    string
                    +--ro protection-mode?   enumeration
                    +--ro reroute-mode?      enumeration
                    +--ro reroute-delay?     uint16
                    +--ro dual-receive?      boolean
                    +--ro revert?            boolean
                    +--ro revert-delay?      uint16
```

Figure 2

5. YANG Module

The L2VPN configuration container is logically divided into following high level config areas:

```
<CODE BEGINS> file "ietf-l2vpn@2016-03-07.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";
  }

  organization  "ietf";
  contact       "ietf";
  description   "l2vpn";

  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 link-discovery-protocol {
    description "Base identiy 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 represens BPDU";
  }

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

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

```
  /* typedefs */

  typedef l2vpn-service-type {
    type enumeration {
      enum ethernet {
        description "Ethernet service";
      }
      enum ATM {
        description "Asynchronous Transfer Mode";
      }
      enum FR {
        description "Frame-Relay";
      }
      enum TDM {
        description "Time Division Multiplexing";
      }
    }
    description "L2VPN service type";
  }

  typedef l2vpn-discovery-type {
    type enumeration {
      enum manual {
        description "Manual configuration";
      }
      enum bgp-ad {
        description "Border Gateway Protocol (BGP) auto-discovery";
      }
      enum ldp {
        description "Label Distribution Protocol (LDP)";
      }
      enum mixed {
        description "Mixed";
      }
    }
    description "L2VPN discovery type";
  }

  typedef l2vpn-signaling-type {
    type enumeration {
      enum static {
        description "Static configuration of labels (no signaling)";
      }
      enum ldp {
        description "Label Distribution Protocol (LDP) signaling";
      }
      enum bgp {
        description "Border Gateway Protocol (BGP) signaling";
```

```
      }
      enum mixed {
        description "Mixed";
      }
    }
    description "L2VPN signaling type";
  }

  typedef bgp-rt-type {
    type enumeration {
      enum import {
        description "For import";
      }
      enum export {
        description "For export";
      }
      enum both {
        description "For both import and export";
      }
    }
    description "BGP route-target type. Import from BGP YANG";
  }

  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 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 pw-template-ref {
    type leafref {
      path "/l2vpn/common/pw-templates/pw-template/name";
    }
    description "pw-template-ref";
  }

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

  typedef bridge-table-instance-ref {
    type leafref {
      path "/l2vpn/bridge-table-instances" +
           "/bridge-table-instance/name";
    }
    description "bridge-table-instance-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";
  }

  /* 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-tag {
            type uint32;
            description "i-tag";
          }
          leaf backbone-src-mac {
            type yang:mac-address;
            description "backbone-src-mac";
          }
        }
        case b-component {
          leaf bind-b-component {
            type bridge-table-instance-ref;
            description "Reference to 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-tag {
            type uint32;
            description "i-tag";
          }
          leaf backbone-src-mac {
            type yang:mac-address;
            description "backbone-src-mac";
          }
        }
        case b-component {
          leaf bind-b-component {
            type string;
            description "Name of the associated b-component";
          }
        }
      }
    }
  }
```

```
  grouping bgp-parameters-grp {
    description "BGP parameters grouping";
    container bgp-parameters {
      description "Parameters for BGP";
      container common {
        when "../../discovery-type = 'bgp-ad'" {
          description "Check discovery type: " +
                      "Can only configure BGP discovery if " +
                      "discovery type is BGP-AD";
        }
        description "Common BGP parameters";
        leaf route-distinguisher {
          type string;
          description "BGP RD";
        }
        list vpn-targets {
          key rt-value;
          description "Route Targets";
          leaf rt-value {
            type string;
            description "Route-Target value";
          }
          leaf rt-type {
            type bgp-rt-type;
            mandatory true;
            description "Type of RT";
          }
        }
      }
      container discovery {
        when "../../discovery-type = 'bgp-ad'" {
          description "BGP parameters for discovery: " +
                      "Can only configure BGP discovery if " +
                      "discovery type is BGP-AD";
        }
        description "BGP parameters for discovery";
        leaf vpn-id {
          type string;
          description "VPN ID";
        }
      }
      container signaling {
        when "../../signaling-type = 'bgp'" {
          description "Check signaling type: " +
                      "Can only configure BGP signaling if " +
                      "signaling type is BGP";
        }
        description "BGP parameters for signaling";
```

```
        leaf site-id {
          type uint16;
          description "Site ID";
        }
        leaf site-range {
          type uint16;
          description "Site Range";
        }
      }
    }
  }

  grouping pw-type-grp {
    description "pseudowire type grouping";
    choice pw-type {
      description "A choice of pseudowire type";
      case ldp-or-static-pw {
        leaf peer-ip {
          type inet:ip-address;
          description "peer IP address";
        }
        leaf pw-id {
          type uint32;
          description "pseudowire id";
        }
        leaf icb {
          type boolean;
          description "inter-chassis backup";
        }
        leaf transmit-label {
          type uint32;
          description "transmit lable";
        }
        leaf receive-label {
          type uint32;
          description "receive label";
        }
      }
      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";
```

```
        }
      }
    }
  }

  grouping bridge-table-instance-pw-list-grp {
    description "bridge-table-instance-pw-list-grp";
    list pw {
      key "name";
      leaf name {
        type leafref {
          path "../../../pw/name";
        }
        description "name of pseudowire";
      }
      description "A bridge table instance's pseudowire list";
    }
  }

  grouping bridge-table-instance-ac-list-grp {
    description "bridge-table-instance-ac-list-grp";
    list ac {
      key "name";
      leaf name {
        type string;
        description "Name of attachment circuit.  This field " +
                    "is intended to reference standardized " +
                    "layer-2 definitions.";
      }
      description "A bridge table instance's " +
                  "attachment circuit list";
    }
  }

  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 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 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";
      /* This is called "revertive" during the discussion. */
    }
    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 ot time to wait to revert to primary";
```

```
      /* This is called "wtr" during discussion. */
    }
  }

  grouping bridge-table-instance-endpoint-grp {
    description "A bridge table instance's endpoint";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        uses bridge-table-instance-ac-list-grp;
        description "reference to attachment circuits";
      }
      case pw {
        uses bridge-table-instance-pw-list-grp;
        description "reference to pseudowires";
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw {
            description "primary-pw";
            list primary-pw {
              key "name";
              leaf name {
                type leafref {
                  path "../../../pw/name";
                }
                description "Reference a pseudowire";
              }
              description "A list of primary pseudowires";
            }
          }
          case primary-ac {
            description "primary-ac";
            leaf primary-ac {
              type string;
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          description "backup options";
          case backup-pw {
            list backup-pw {
```

```
              key "name";
              leaf name {
                type leafref {
                  path "../../../pw/name";
                }
                description "Reference an attachment circuit";
              }
              leaf precedence {
                type uint32;
                description "precedence of the pseudowire";
              }
              description "A list of backup pseudowires";
            }
          }
          case backup-ac {
            leaf backup-ac {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
            description "backup-ac";
          }
        }
        leaf template {
          type leafref {
            path "/l2vpn/common/redundancy-group-templates" +
                 "/redundancy-group-template/name";
          }
          description "Reference a redundancy group " +
                      "properties template";
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpws-endpoint-grp {
    description
      "A vpws-endpoint could either be an ac or a pw";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        leaf ac {
          type string;
          description "Name of attachment circuit.  This " +
                      "field is intended to reference " +
```

```
                    "standardized layer-2 definitions.";
        }
      }
      case pw {
        leaf pw {
          type leafref {
            path "../../pw/name";
          }
          description "reference to a pseudowire";
        }
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw  {
            leaf primary-pw {
              type leafref {
                path "../../pw/name";
              }
              description "primary pseudowire";
            }
          }
          case primary-ac {
            leaf primary-ac {
              type string;
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          mandatory true;
          description "backup options";
          case backup-pw {
            leaf backup-pw {
              type leafref {
                path "../../pw/name";
              }
              description "backup pseudowire";
            }
          }
          case backup-ac {
            leaf backup-ac {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
```

```
                          "standardized layer-2 definitions.";
            }
          }
        }
        leaf template {
          type leafref {
            path "/l2vpn/common/redundancy-group-templates" +
                 "/redundancy-group-template/name";
          }
          description "Reference a redundancy group " +
                      "properties template";
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpws-endpoint-state-grp {
    description
      "A vpws-endpoint could either be an ac or a pw";
    choice ac-or-pw-or-redundancy-grp {
      description "A choice ofattachment circuit or " +
                  "pseudowire or redundancy group";
      case ac {
        container ac {
          description "ac";
          uses ac-state-grp;
        }
      }
      case pw {
        container pw {
          description "pw";
          uses vpws-pw-state-grp;
        }
      }
      case redundancy-grp {
        choice primary {
          mandatory true;
          description "primary options";
          case primary-pw  {
            container primary-pw {
              description "primary pseudowire";
              uses vpws-pw-state-grp;
            }
          }
          case primary-ac {
            leaf primary-ac-name {
              type string;
```

```
              description "Name of primary attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        choice backup {
          mandatory true;
          description "backup options";
          case backup-pw {
            container backup-pw {
              description "backup pseudowire";
              uses vpws-pw-state-grp;
            }
          }
          case backup-ac {
            leaf backup-ac-name {
              type string;
              description "Name of backup attachment circuit.  " +
                          "This field is intended to reference " +
                          "standardized layer-2 definitions.";
            }
          }
        }
        uses redundancy-group-properties-grp;
      }
    }
  }

  grouping vpls-pw-state-grp {
    description "vpls-pw-state-grp";
    leaf name {
      type string;
      description "pseudowire name";
    }
    leaf state {
      type operational-state-type;
      description "pseudowire up/down state";
    }
    leaf mtu {
      type uint32;
      description "pseudowire mtu";
    }
    leaf mac-withdraw {
      type boolean;
      description "MAC withdraw is enabled (true) or disabled (false)";
    }
    leaf cw-negotiation {
```

```
      type cw-negotiation-type;
      description "cw-negotiation";
    }
    leaf discovery-type {
      type l2vpn-discovery-type;
      description "VPLS discovery type";
    }
    leaf signaling-type {
      type l2vpn-signaling-type;
      description "VPLS signaling type";
    }
    leaf peer-ip {
      type inet:ip-address;
      description "peer IP address";
    }
    leaf pw-id {
      type uint32;
      description "pseudowire id";
    }
    leaf transmit-label {
      type uint32;
      description "transmit lable";
    }
    leaf receive-label {
      type uint32;
      description "receive label";
    }
    leaf tunnel-policy {
      type string;
      description "tunnel policy name";
    }
  }

  grouping ac-state-grp {
    description "vpls-ac-state-grp";
    leaf name {
      type string;
      description "attachment circuit name";
    }
    leaf state {
      type operational-state-type;
      description "attachment circuit up/down state";
    }
  }

  grouping vpws-pw-state-grp {
    description "vpws-pw-state-grp";
    leaf name {
```

```
      type string;
      description "pseudowire name";
    }
    leaf state {
      type operational-state-type;
      description "pseudowire operation state up/down";
    }
    leaf mtu {
      type uint32;
      description "PW MTU";
    }
    leaf mac-withdraw {
      type boolean;
      description "MAC withdraw is enabled (ture) or disabled (false)";
    }
    leaf cw-negotiation {
      type cw-negotiation-type;
      description "Override the control-word negotiation " +
                  "preference specified in the " +
                  "pseudowire template.";
    }
    leaf vccv-ability {
      type boolean;
      description "vccv-ability";
    }
    leaf tunnel-policy {
      type string;
      description "Used to override the tunnel policy name " +
                  "specified in the pseduowire template";
    }
    leaf request-vlanid {
      type uint16;
      description "request vlanid";
    }
    leaf vlan-tpid {
      type string;
      description "vlan tpid";
    }
    leaf ttl {
      type uint8;
      description "time-to-live";
    }
    uses pw-type-grp;
  }

  /* L2VPN YANG Model */

  container l2vpn {
```

```
    description "l2vpn";
    container common {
      description "common l2pn attributes";
      container pw-templates {
        description "pw-templates";
        list pw-template {
          key "name";
          description "pw-template";
          leaf name {
            type string;
            description "name";
          }
          leaf mtu {
            type uint32;
            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";
          }
        }
      }
      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;
        }
      }
    }
    container bridge-table-instances {
      /* To be fleshed out in future revisions */
      description "bridge-table-instances";
      list bridge-table-instance {
        key "name";
        description "A bridge table instance";
        leaf name {
```

```
          type string;
          description "Name of a bridge table instance";
        }
        leaf mtu {
          type uint32;
          description "Bridge MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        uses pbb-parameters-grp;
        uses bgp-parameters-grp;
        leaf evpn-instance {
          type string;
          description "Eventual reference to standard EVPN instance";
        }
        list pw {
          key "name";
          description "pseudowire";
          leaf name {
            type string;
            description "pseudowire name";
          }
          leaf template {
            type pw-template-ref;
            description "pseudowire template";
          }
          leaf mtu {
            type uint32;
            description "PW MTU";
          }
          leaf mac-withdraw {
            type boolean;
            default false;
            description "Enable (true) or disable (false) MAC withdraw";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
            description "cw-negotiation";
          }
          leaf discovery-type {
            type l2vpn-discovery-type;
            description "VPLS discovery type";
          }
          leaf signaling-type {
            type l2vpn-signaling-type;
            description "VPLS signaling type";
```

```
          }
          leaf peer-ip {
            type inet:ip-address;
            description "peer IP address";
          }
          leaf pw-id {
            type uint32;
            description "pseudowire id";
          }
          leaf transmit-label {
            type uint32;
            description "transmit lable";
          }
          leaf receive-label {
            type uint32;
            description "receive label";
          }
          leaf tunnel-policy {
            type string;
            description "tunnel policy name";
          }
        }
        list endpoint {
          key "id";
          leaf id {
            type uint16;
            description "endpoint ID";
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          uses bridge-table-instance-endpoint-grp;
          description "List of endpoints";
        }
      }
    }
    container vpws-instances {
      description "vpws-instances";
      list vpws-instance {
        key "name";
        description "A VPWS instance";
        leaf name {
          type string;
          description "Name of VPWS instance";
        }
        leaf description {
          type string;
```

```
          description "Description of the VPWS instance";
        }
        leaf mtu {
          type uint32;
          description "VPWS MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        leaf service-type {
          type l2vpn-service-type;
          default ethernet;
          description "VPWS service type";
        }
        leaf discovery-type {
          type l2vpn-discovery-type;
          default manual;
          description "VPWS discovery type";
        }
        leaf signaling-type {
          type l2vpn-signaling-type;
          mandatory true;
          description "VPWS signaling type";
        }
        uses bgp-parameters-grp;
        list pw {
          key "name";
          description "pseudowire";
          leaf name {
            type string;
            description "pseudowire name";
          }
          leaf template {
            type pw-template-ref;
            description "pseudowire template";
          }
          leaf mtu {
            type uint32;
            description "PW MTU";
          }
          leaf mac-withdraw {
            type boolean;
            default false;
            description "Enable (true) or disable (false) MAC withdraw";
          }
          leaf cw-negotiation {
            type cw-negotiation-type;
```

```
            default "preferred";
            description "Override the control-word negotiation " +
                        "preference specified in the " +
                        "pseudowire template.";
          }
          leaf vccv-ability {
            type boolean;
            description "vccvability";
          }
          leaf tunnel-policy {
            type string;
            description "Used to override the tunnel policy name " +
                        "specified in the pseduowire template";
          }
          leaf request-vlanid {
            type uint16;
            description "request vlanid";
          }
          leaf vlan-tpid {
            type string;
            description "vlan tpid";
          }
          leaf ttl {
            type uint8;
            description "time-to-live";
          }
          uses pw-type-grp;
        }
        container endpoint-a {
          description "endpoint-a";
          uses vpws-endpoint-grp;
        }
        container endpoint-z {
          description "endpoint-z";
          uses vpws-endpoint-grp;
        }
      }
    }
  }

  container l2vpn-state {
    config false;
    description "l2vpn state";
    container bridge-table-instances-state {
      /* To be fleshed out in future revisions */
      description "bridge-table-instances-state";
      list bridge-table-instance-state {
        key "name";
```

```
        description "A bridge table instance's state data";
        leaf name {
          type string;
          description "Name of a bridge table instance";
        }
        leaf mtu {
          type uint32;
          description "Bridge MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        uses pbb-parameters-state-grp;
        uses bgp-parameters-grp;
        leaf evpn-instance-name {
          type string;
          description "Name of associated an EVPN instance";
        }
        list endpoint {
          key "id";
          leaf id {
            type uint16;
            description "endpoint ID";
          }
          leaf split-horizon-group {
            type string;
            description "Identify a split horizon group";
          }
          choice ac-or-pw-or-redundancy-grp {
            description "A choice ofattachment circuit or " +
                        "pseudowire or redundancy group";
            case ac {
              list ac {
                key "name";
                uses ac-state-grp;
                description "A list of attachment circuits";
              }
              description "attachment circuit endpoint state";
            }
            case pw {
              list pw {
                key "name";
                uses vpls-pw-state-grp;
                description "A list of pseudowires";
              }
              description "pseudowire endpoint state";
            }
```

```
            case redundancy-grp {
              choice primary {
                mandatory true;
                description "primary options";
                case primary-pw {
                  description "primary-pw";
                  list primary-pw {
                    key "name";
                    uses vpls-pw-state-grp;
                    description "A list of primary pseudowires";
                  }
                }
                case primary-ac {
                  description "primary-ac";
                  container primary-ac {
                    description "primary-ac";
                    uses ac-state-grp;
                  }
                }
              }
              choice backup {
                description "backup options";
                case backup-pw {
                  list backup-pw {
                    key "name";
                    uses vpls-pw-state-grp;
                    leaf precedence {
                      type uint32;
                      description "precedence of the pseudowire";
                    }
                    description "A list of backup pseudowires";
                  }
                }
                case backup-ac {
                  description "backup-ac";
                  container backup-ac {
                    description "primary-ac";
                    uses ac-state-grp;
                  }
                }
              }
              uses redundancy-group-properties-grp;
            }
          }
          description "List of endpoints";
        }
      }
    }
```

```
    container vpws-instances-state {
      description "vpws-instances-state";
      list vpws-instance-state {
        key "name";
        description "A VPWS instance's state data";
        leaf name {
          type string;
          description "Name of VPWS instance";
        }
        leaf mtu {
          type uint32;
          description "VPWS MTU";
        }
        leaf mac-aging-timer {
          type uint32;
          description "mac-aging-timer";
        }
        leaf service-type {
          type l2vpn-service-type;
          default ethernet;
          description "VPWS service type";
        }
        leaf discovery-type {
          type l2vpn-discovery-type;
          default manual;
          description "VPWS discovery type";
        }
        leaf signaling-type {
          type l2vpn-signaling-type;
          mandatory true;
          description "VPWS signaling type";
        }
        uses bgp-parameters-grp;
        container endpoint-a {
          description "endpoint-a";
          uses vpws-endpoint-state-grp;
        }
        container endpoint-z {
          description "endpoint-z";
          uses vpws-endpoint-state-grp;
        }
      }
    }
  }
}
```

<CODE ENDS>

Figure 3

6. 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]

7. IANA Considerations

None.

8. Acknowledgments

The authors would like to acknowledge TBD for their useful comments.

9. References

9.1. Normative References

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

9.2. 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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc4446>.

[RFC4447] Martini, L., Ed., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, DOI 10.17487/RFC4447, April 2006, <http://www.rfc-editor.org/info/rfc4447>.

[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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc5659>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6073] Martini, L., Metz, C., Nadeau, T., Bocci, M., and M. Aissaoui, "Segmented Pseudowire", RFC 6073, DOI 10.17487/RFC6073, January 2011, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://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, <http://www.rfc-editor.org/info/rfc7361>.

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Reshad Rahman
Cisco Systems, Inc.

Email: rrahman@cisco.com

Kamran Raza
Cisco Systems, Inc.

Email: skraza@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

Ing-When Chen
Ericsson

Email: ichen@kuatrotech.com

Sajjad Ahmed
Ericsson

Email: sajjad.ahmed@ericsson.com

Mathew Bocci
Alcatel-Lucent

Email: mathew.bocci@alcatel-lucent.com

Jonathan Hardwick
Metaswitch

Email: jonathan.hardwick@metaswitch.com

Santosh Esale
Juniper Networks

Email: sesale@juniper.net

Kishore Tiruveedhula
Juniper Networks

Email: kishoret@juniper.net

Tapraj Singh
Juniper Networks

Email: tsingh@juniper.net

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Jason Walker
Comcast

Email: jason_walker2@cable.comcast.com

Nick Delregno
Verizon

Email: nick.deregno@verizon.com

Luay Jalil
Verizon

Email: luay.jalil@verizon.com

Maria Joecylyn
Verizon

Email: joecylyn.malit@verizon.com

BESS
Internet-Draft
Updates: 7432 (if approved)
Intended status: Standards Track
Expires: June 20, 2016

Z. Zhang
W. Lin
Juniper Networks, Inc.
J. Rabadan
Alcatel-Lucent
K. Patel
Cisco Systems

December 18, 2015

Updates on EVPN BUM Procedures
draft-zzhang-bess-evpn-bum-procedure-updates-01

Abstract

This document specifies procedure updates for broadcast, unknown unicast, and multicast (BUM) traffic in Ethernet VPNs (EVPN), including selective multicast, and provider tunnel segmentation.

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 June 20, 2016.

Table of Contents

1. Terminology

To be added

2. Introduction

RFC 7432 specifies procedures to handle broadcast, unknown unicast, and multicast (BUM) traffic in Section 11, 12 and 16, using Inclusive Multicast Ethernet Tag Route. A lot of details are referred to RFC 7117 (VPLS Multicast). In particular, selective multicast is briefly mentioned for Ingress Replication but referred to RFC 7117.

RFC 7117 specifies procedures for using both inclusive tunnels and selective tunnels, similar to MVPN procedures specified in RFC 6513 and RFC 6514. A new SAFI "MCAST-VPLS" is introduced, with two types of NLRIs that match MVPN's S-PMSI A-D routes and Leaf A-D routes. The same procedures can be applied to EVPN selective multicast for both Ingress Replication and other tunnel types, but new route types need to be defined under the same EVPN SAFI.

MVPN uses terms I-PMSI and S-PMSI A-D Routes. For consistency and convenience, this document will use the same I/S-PMSI terms for VPLS and EVPN. In particular, EVPN's Inclusive Multicast Ethernet Tag Route and VPLS's VPLS A-D route carrying PTA (PMSI Tunnel Attribute) for BUM traffic purpose will all be referred to as I-PMSI A-D routes. Depending on the context, they may be used interchangeably.

MVPN provider tunnels and EVPN/VPLS BUM provider tunnels, which are referred to as MVPN/EVPN/VPLS provider tunnels in this document for simplicity, can be segmented for technical or administrative reasons, which are summarized in Section 2.1 of this document. RFC 6513/6514 cover MVPN inter-as segmentation, RFC 7117 covers VPLS multicast inter-as segmentation, and RFC 7524 (Seamless MPLS Multicast) covers inter-area segmentation for both MVPN and VPLS.

There is a difference between MVPN and VPLS multicast inter-as segmentation. For simplicity, EVPN uses the same procedures as in MVPN. All ASBRs can re-advertise their choice of the best route. Each can become the root of its intra-AS segment and inject traffic it receives from its upstream, while each downstream PE/ASBR will only pick one of the upstream ASBRs as its upstream. This is also the behavior even for VPLS in case of inter-area segmentation.

For inter-area segmentation, RFC 7524 requires the use of Inter-area P2MP Segmented Next-Hop Extended Community (S-NH-EC), and the setting of "Leaf Information Required" (LIR) flag in PTA in certain situations. Either of these could be optional in case of EVPN. Removing these requirements would make the segmentation procedures transparent to ingress and egress PEs.

RFC 7524 assumes that segmentation happens at area borders. However, it could be at "regional" borders, where a region could be a sub-

area, or even an entire AS plus its external links (Section 6). That would allow for more flexible deployment scenarios (e.g. for single-area provider networks).

This document specifies/clarifies/redefines certain/additional EVPN BUM procedures, with a salient goal that they're better aligned among MVPN, EVPN and VPLS. For brevity, only changes/additions to relevant RFC 7117 and RFC 7524 procedures are specified, instead of repeating the entire procedures. Note that these are to be applied to EVPN only, even though sometimes they may sound to be updates to RFC 7117/7524.

2.1. Reasons for Tunnel Segmentation

Tunnel segmentation may be required and/or desired because of administrative and/or technical reasons.

For example, an MVPN/VPLS/EVPN network may span multiple providers and Inter-AS Option-B has to be used, in which the end-to-end provider tunnels have to be segmented at and stitched by the ASBRs. Different providers may use different tunnel technologies (e.g., provider A uses Ingress Replication, provider B uses RSVP-TE P2MP while provider C uses mLDP). Even if they use the same tunnel technology like RSVP-TE P2MP, it may be impractical to set up the tunnels across provider boundaries.

The same situations may apply between the ASes and/or areas of a single provider. For example, the backbone area may use RSVP-TE P2MP tunnels while non-backbone areas may use mLDP tunnels.

Segmentation can also be used to divide an AS/area to smaller regions, so that control plane state and/or forwarding plane state/ burden can be limited to that of individual regions. For example, instead of Ingress Replicating to 100 PEs in the entire AS, with inter-area segmentation [RFC 7524] a PE only needs to replicate to local PEs and ABRs. The ABRs will further replicate to their downstream PEs and ABRs. This not only reduces the forwarding plane burden, but also reduces the leaf tracking burden in the control plane. This inter-region segmentation can be further extended to intra-region as an alternative way to achieve Assisted Replication as proposed in [draft-rabadan-bess-evpn-optimized-ir], and it works for MPLS encapsulation.

Smaller regions also have the benefit that, in case of tunnel aggregation, it is easier to find congruence among the segments of different constituent (service) tunnels and the resulting aggregation (base) tunnel in a region. This leads to better bandwidth efficiency, because the more congruent they are, the fewer leaves of

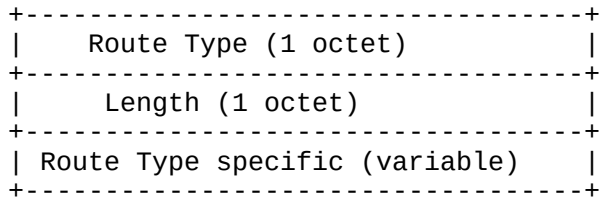
the base tunnel need to discard traffic when a service tunnel's segment does not need to receive the traffic (yet it is receiving the traffic due to aggregation).

Another advantage of the smaller region is smaller BIER sub-domains. In this new multicast architecture BIER, packets carry a BitString, in which the bits correspond to edge routers that needs to receive traffic. Smaller sub-domains means smaller BitStrings can be used without having to send multiple copies of the same packet.

Finally, EVPN tunnel segmentation can be used for EVPN DCIs, as discussed in Section 9. It follows the same concepts discussed above.

3. Additional Route Types of EVPN NLRI

RFC 7432 defines the format of EVPN NLRI as the following:



So far five types have been defined:

+ 1 - Ethernet Auto-Discovery (A-D) route
+ 2 - MAC/IP Advertisement route
+ 3 - Inclusive Multicast Ethernet Tag route
+ 4 - Ethernet Segment route
+ 5 - IP Prefix Route

This document defines three additional route types:

+ 6 - Per-Region I-PMSI A-D route
+ 7 - S-PMSI A-D route
+ 8 - Leaf A-D route

The "Route Type specific" field of the type 6 and type 7 EVPN NLRIs starts with a type 1 RD, whose Administrative sub-field MUST match that of the RD in all the EVPN routes from the same advertising router for a given EVI, except the Leaf A-D route (Section 3.3).

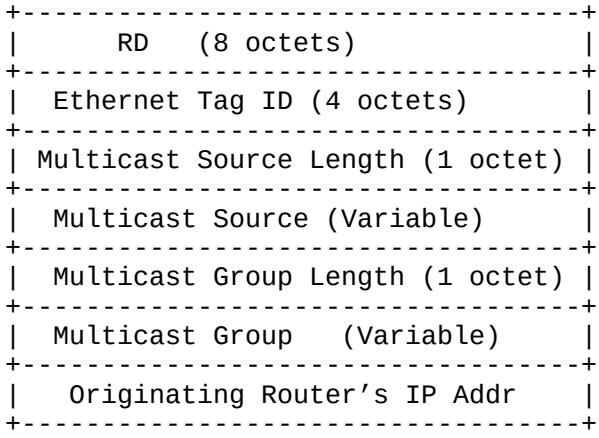
3.1. Per-Region I-PMSI A-D route

The Per-region I-PMSI A-D route has the following format. Its usage is discussed in Section 6.2.

After Ethernet Tag ID, an Extended Community (EC) is used to identify the region. Various types and sub-types of ECs provide maximum flexibility. Note that this is not an EC Attribute, but an 8-octet field embedded in the NLRI itself, following EC encoding scheme.

3.2. S-PMSI A-D route

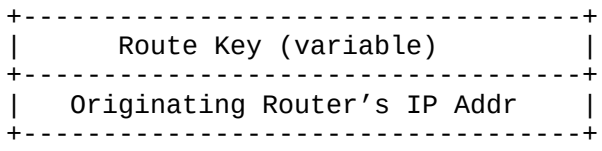
The S-PMSI A-D route has the following format:



Other than the addition of Ethernet Tag ID, it is identical to the S-PMSI A-D route as defined in RFC 7117. The procedures in RFC 7117 also apply (including wildcard functionality), except that the granularity level is per Ethernet Tag.

3.3. Leaf-AD route

The Route Type specific field of a Leaf A-D route consists of the following:



A Leaf A-D route is originated in response to a PMSI route, which could be an Inclusive Multicast Tag route, a per-region I-PMSI A-D route, an S-PMSI A-D route, or some other types of routes that may be defined in the future that triggers Leaf A-D routes. The Route Key is the "Route Type Specific" field of the route for which this Leaf A-D route is generated.

The general procedures of Leaf A-D route are first specified in RFC 6514 for MVPN. The principles apply to VPLS and EVPN as well. RFC 7117 has details for VPLS Multicast, and this document points out some specifics for EVPN, e.g. in Section 5.

4. Selective Multicast

RFC 7117 specifies Selective Multicast for VPLS. Other than that different route types and formats are specified with EVPN SAFI for S-PMSI A-D and Leaf A-D routes (Section 3), all procedures in RFC 7117 with respect to Selective Multicast apply to EVPN as well, including wildcard procedures.

5. Inter-AS Segmentation

5.1. Changes to Section 7.2.2 of RFC 7117

The first paragraph of Section 7.2.2.2 of RFC 7117 says:

> "... The best route procedures ensure that if multiple ASBRs, in an AS, receive the same Inter-AS A-D route from their EBGP neighbors, only one of these ASBRs propagates this route in Internal BGP (IBGP). This ASBR becomes the root of the intra-AS segment of the inter-AS tree and ensures that this is the only ASBR that accepts traffic into this AS from the inter-AS tree."

The above VPLS behavior requires complicated VPLS specific procedures for the ASBRs to reach agreement. For EVPN, a different approach is used and the above quoted text is not applicable to EVPN.

The Leaf A-D based procedure is used for each ASBR who re-advertises into the AS to discover the leaves on the segment rooted at itself. This is the same as the procedures for S-PMSI in RFC 7117 itself.

The following text at the end of the second bullet:

> "..... If, in order to instantiate the segment, the ASBR needs to know the leaves of the tree, then the ASBR obtains this information from the A-D routes received from other PEs/ASBRs in the ASBR's own AS."

is changed to the following:

> "..... If, in order to instantiate the segment, the ASBR needs to know the leaves of the tree, then the ASBR MUST set the LIR flag to 1 in the PTA to trigger Leaf A-D routes from egress PEs and downstream ASBRs. It MUST be (auto-)configured with an import RT, which controls acceptance of leaf A-D routes by the ASBR."

Accordingly, the following paragraph in Section 7.2.2.4:

> "If the received Inter-AS A-D route carries the PMSI Tunnel attribute with the Tunnel Identifier set to RSVP-TE P2MP LSP, then the ASBR that originated the route MUST establish an RSVP-TE P2MP LSP with the local PE/ASBR as a leaf. This LSP MAY have been established before the local PE/ASBR receives the route, or it MAY be established after the local PE receives the route."

is changed to the following:

> "If the received Inter-AS A-D route has the LIR flag set in its PTA, then a receiving PE must originate a corresponding Leaf A-D route, and a receiving ASBR must originate a corresponding Leaf A-D route if and only if it received and imported one or more corresponding Leaf A-D routes from its downstream IBGP or EBGP peers, or it has non-null downstream forwarding state for the PIM/mLDP tunnel that instantiates its downstream intra-AS segment. The ASBR that (re-)advertised the Inter-AS A-D route then establishes a tunnel to the leaves discovered by the Leaf A-D routes."

5.2. I-PMSI Leaf Tracking

An ingress PE does not set the LIR flag in its I-PMSI's PTA, even with Ingress Replication or RSVP-TE P2MP tunnels. It does not rely on the Leaf A-D routes to discover leaves in its AS, and Section 11.2 of RFC 7432 explicitly states that the LIR flag must be set to zero.

An implementation of RFC 7432 might have used the Originating Router's IP Address field of the Inclusive Multicast Ethernet Tag routes to determine the leaves, or might have used the Next Hop field instead. Within the same AS, both will lead to the same result.

With segmentation, an ingress PE MUST determine the leaves in its AS from the BGP next hops in all its received I-PMSI A-D routes, so it does not have to set the LIR bit set to request Leaf A-D routes. PEs within the same AS will all have different next hops in their I-PMSI A-D routes (hence will all be considered as leaves), and PEs from other ASes will have the next hop in their I-PMSI A-D routes set to addresses of ASBRs in this local AS, hence only those ASBRs will be considered as leaves (as proxies for those PEs in other ASes). Note that in case of Ingress Replication, when an ASBR re-advertises IBGP I-PMSI A-D routes, it MUST advertise the same label for all those for the same Ethernet Tag ID and the same EVI. When an ingress PE builds its flooding list, multiple routes may have the same (nexthop, label) tuple and they will only be added as a single branch in the flooding list.

5.3. Backward Compatibility

The above procedures assume that all PEs are upgraded to support the segmentation procedures:

- An ingress PE uses the Next Hop instead of Originating Router’s IP Address to determine leaves for the I-PMSI tunnel.

- An egress PE sends Leaf A-D routes in response to I-PMSI routes, if the PTA has the LIR flag set (by the re-advertising ASBRs).

- In case of Ingress Replication, when an ingress PE builds its flooding list, multiple I-PMSI routes may have the same (nexthop, label) tuple and only a single branch for those will be added in the flooding list.

If a deployment has legacy PEs that does not support the above, then a legacy ingress PE would include all PEs (including those in remote ASes) as leaves of the inclusive tunnel and try to send traffic to them directly (no segmentation), which is either undesired or not possible; a legacy egress PE would not send Leaf A-D routes so the ASBRs would not know to send external traffic to them.

To address this backward compatibility problem, the following procedure can be used (see Section 6.2 for per-PE/AS/region I-PMSI A-D routes):

- An upgraded PE indicates in its per-PE I-PMSI A-D route that it supports the new procedures. Details will be provided in a future revision.

- All per-PE I-PMSI A-D routes are restricted to the local AS and not propagated to external peers.

- The ASBRs in an AS originate per-region I-PMSI A-D routes and advertise to their external peers to advertise tunnels used to carry traffic from the local AS to other ASes. Depending on the types of tunnels being used, the LIR flag in the PTA may be set, in which case the downstream ASBRs and upgraded PEs will send Leaf A-D routes to pull traffic from their upstream ASBRs. In a particular downstream AS, one of the ASBRs is elected, based on the per-region I-PMSI A-D routes for a particular source AS, to send traffic from that source AS to legacy PEs in the downsrream AS. The traffic arrives at the elected ASBR on the tunnel announced in the best per-region I-PMSI A-D route for the source AS, that the ASBR has selected of all those that it received over EBGP or IBGP sessions. Details of the election procedure will be provided in a future revision.

- In an ingress AS, if and only if an ASBR has active downstream receivers (PEs and ASBRs), which are learned either explicitly via Leaf AD routes or implicitly via PIM join or mLDP label mapping, the ASBR originates a per-PE I-PMSI A-D route (i.e., regular Inclusive Multicast Ethernet Tag route) into the local AS, and stitches incoming per-PE I-PMSI tunnels into its per-region I-PMSI tunnel. With this, it gets traffic from local PEs and send to other ASes via the tunnel announced in its per-region I-PMSI A-D route.

Note that, even if there is no backward compatibility issue, the above procedures has the benefit of keeping all per-PE I-PMSI A-D routes in their local ASes, greatly reducing the flooding of the routes and their corresponding Leaf A-D routes (when needed), and the number of inter-as tunnels.

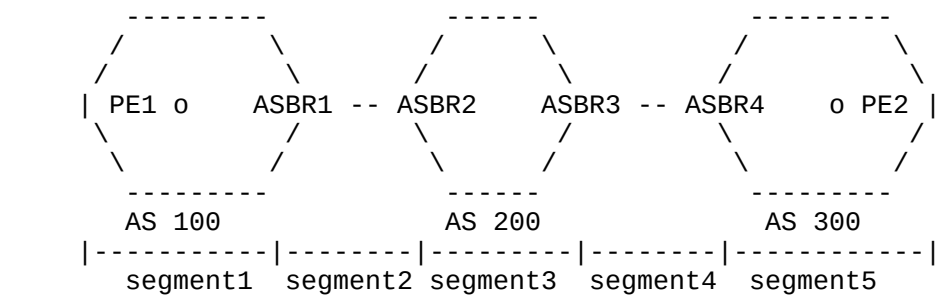
6. Inter-Region Segmentation

6.1. Area vs. Region

RFC 7524 is for MVPN/VPLS inter-area segmentation and does not explicitly cover EVPN. However, if "area" is replaced by "region" and "ABR" is replaced by "RBR" (Regional Border Router) then everything still works, and can be applied to EVPN as well.

A region can be a sub-area, or can be an entire AS including its external links. Instead of automatic region definition based on IGP areas, a region would be defined as a BGP peer group. In fact, even with IGP area based region definition, a BGP peer group listing the PEs and ABRs in an area is still needed.

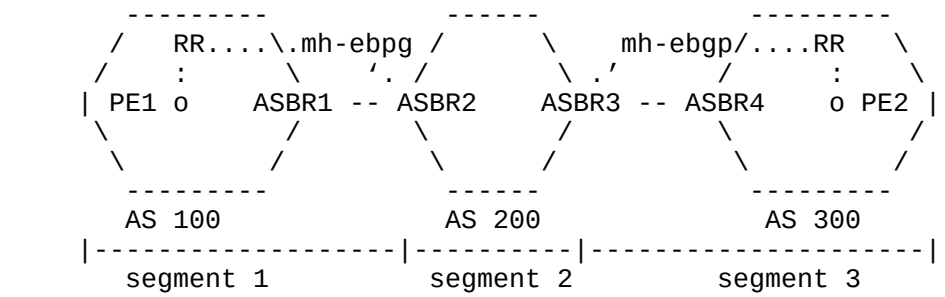
Consider the following example diagram:



The inter-as segmentation procedures specified so far (RFC 6513/6514, 7117, and Section 5 of this document) requires all ASBRs to be involved, and Ingress Replication is used between two ASBRs in different ASes.

In the above diagram, it's possible that ASBR1/4 does not support segmentation, and the provider tunnels in AS 100/300 can actually extend across the external link. In the case, the inter-region segmentation procedures can be used instead - a region is the entire (AS100 + ASBR1-ASBR2 link) or (AS300 + ASBR3-ASBR4 link). ASBR2/3 would be the RBRs, and ASBR1/4 will just be a transit core router with respect to provider tunnels.

As illustrated in the diagram below, ASBR2/3 will establish a multihop EBGP session with either a RR or directly with PEs in the neighboring AS. I/S-PMSI A-D routes from ingress PEs will not be processed by ASBR1/4. When ASBR2 re-advertises the routes into AS 200, it changes the next hop to its own address and changes PTA to specify the tunnel type/identification in its own AS. When ASBR3 re-advertises I/S-PMSI A-D routes into the neighboring AS 300, it changes the next hop to its own address and changes PTA to specify the tunnel type/identification in the neighboring region 3. Now the segment is rooted at ASBR3 and extends across the external link to PEs.



6.2. Per-region Aggregation

Notice that every I/S-PMSI route from each PE will be propagated throughout all the ASes or regions. They may also trigger corresponding Leaf A-D routes depending on the types of tunnels used in each region. This may become too many - routes and corresponding tunnels. To address this concern, the I-PMSI routes from all PEs in a AS/region can be aggregated into a single I-PMSI route originated from the RBRs, and traffic from all those individual I-PMSI tunnels will be switched into the single I-PMSI tunnel. This is like the MVPN Inter-AS I-PMSI route originated by ASBRs.

The MVPN Inter-AS I-PMSI A-D route can be better called as per-AS I-PMSI A-D route, to be compared against the (per-PE) Intra-AS I-PMSI A-D routes originated by each PE. In this document we will call it as per-region I-PMSI A-D route, in case we want to apply the aggregation at regional level. The per-PE I-PMSI routes will not be propagated to other regions. If multiple RBRs are connected to a region, then each will advertise such a route, with the same route key (Section 3.1). Similar to the per-PE I-PMSI A-D routes, RBRs/PEs in a downstream region will each select a best one from all those re-advertised by the upstream RBRs, hence will only receive traffic injected by one of them.

MVPN does not aggregate S-PMSI routes from all PEs in an AS like it does for I-PMSIs routes, because the number of PEs that will advertise S-PMSI routes for the same (s,g) or (*,g) is small. This is also the case for EVPN, i.e., there is no per-region S-PMSI routes.

Notice that per-region I-PMSI routes can also be used to address backwards compatibility issue, as discussed in Section 5.3.

The per-region I-PMSI route uses an embedded EC in NLRI to identify a region. As long as it uniquely identify the region and the RBRs for the same region uses the same EC it is permitted. In the case where an AS number or area ID is needed, the following can be used:

- For a two-octet AS number, a Transitive Two-Octet AS-Specific EC of sub-type 0x09 (Source AS), with the Global Administrator sub-field set to the AS number and the Local Administrator sub-field set to 0.

- For a four-octet AS number, a Transitive Four-Octet AS-Specific EC of sub-type 0x09 (Source AS), with the Global Administrator sub-field set to the AS number and the Local Administrator sub-field set to 0.

o For an area ID, a Transitive IPv4-Address-Specific EC of any sub-type.

Uses of other particular ECs may be specified in other documents.

6.3. Use of S-NH-EC

RFC 7524 specifies the use of S-NH-EC because it does not allow ABRs to change the BGP next hop when they re-advertise I/S-PMSI AD routes to downstream areas. That is only to be consistent with the MVPN Inter-AS I-PMSI A-D routes, whose next hop must not be changed when they’re re-advertised by the segmenting ABRs for reasons specific to MVPN. For EVPN, it is perfectly fine to change the next hop when RBRs re-advertise the I/S-PMSI A-D routes, instead of relying on S-NH-EC. As a result, this document specifies that RBRs change the BGP next hop when they re-advertise I/S-PMSI A-D routes and do not use S-NH-EC. if a downstream PE/RBR needs to originate Leaf A-D routes, it simply uses the BGP next hop in the corresponding I/S-PMSI A-D routes to construct Route Targets.

The advantage of this is that neither ingress nor egress PEs need to understand/use S-NH-EC, and consistent procedure (based on BGP next hop) is used for both inter-as and inter-region segmentation.

6.4. Ingress PE’s I-PMSI Leaf Tracking

RFC 7524 specifies that when an ingress PE/ASBR (re-)advertises an VPLS I-PMSI A-D route, it sets the LIR flag to 1 in the route’s PTA. Similar to the inter-as case, this is actually not really needed for EVPN. To be consistent with the inter-as case, the ingress PE does not set the LIR flag in its originated I-PMSI A-D routes, and determines the leaves based on the BGP next hops in its received I-PMSI A-D routes, as specified in Section 5.2.

The same backward compatibility issue exists, and the same solution as in the inter-as case applies, as specified in Section 5.3.

7. Intra-region Segmentation and Assisted Ingress Replication

[draft-rabadan-bess-evpn-optimized-ir] describes "Assisted Ingress Replication", which reduces the burden of NVEs by having them replicate to only one of a few designated replicators, which will then replicate to other relevant NVEs. The tunnel segmentation procedures can be extended to achieve the same, even with the support for MPLS encapsulation.

With inter-region segmentation, an RBR, which is a Route Reflector, changes the BGP Next Hop to one of its own addresses when it re-

advertises an I/S-PMSI route to other regions, and sets the LIR bit in the PTA Flag field when necessary, but it does not do so when re-advertising to NVEs in its own region. If it does that even when re-advertising to local NVEs, then it becomes a replicator as in [draft-rabadan-bess-evpn-optimized-ir]: NVEs will respond with Leaf AD routes to individual I-PMSI routes from NVEs, but targeted to the re-advertising RBR of the selected best one (out of all those same routes re-advertised by different RBRs). so that the sending NVEs will only replicate to the RBRs, which will in turn replicate to NVEs.

In case of MPLS encapsulation, for split-horizon purpose, NVEs MUST set the LIR bit in their I-PMSI A-D routes to trigger corresponding Leaf A-D routes from RBRs, with different labels advertised in the Leaf A-D routes for different NVEs, so that RBRs know the source NVEs of incoming packets, and will not relay the traffic back to the source NVE.

A RNVE (Regular, or legacy NVE that does not support the procedures discussed in this section) replicate traffic directly to all NVEs/ RNVEs. RNVEs can be identified by the lack of indication as discussed in Section 5.3 in their I-PMSI A-D routes. In case of MPLS encapsulation, NVEs and RNVEs advertise a label in their I-PMSI A-D routes, and RBRs MUST not change that when re-advertise the routes. Note that, the label is advertised even though an NVE sets the LIR bit.

A RNVE is not able to send back Leaf A-D routes, so RBRs won’t relay received traffic to them. An ingress NVE (legacy or not) always send to RNVEs directly. For comparison, in inter-as scenario (Section 5.3) an ASBR is elected to relay traffic but in this intra-region case, it is reasonable for the ingress NVE to send to RNVEs directly - it is feasible and simpler.

7.1. Reducing Leaf A-D Routes

To address the possible concern with too many Leaf A-D routes (every NVE responds with one to its selected RBR for each I-PMSI A-D route), a RBR can clear the LIR bit when it re-advertises the I-PMSI routes so that no Leaf A-D routes will be triggered for the per-PE I-PMSI routes. It also originates a per-region I-PMSI A-D route (Section 6.2), but instead of into other regions, it is back into the same region. The route has the LIR bit set so that NVEs will respond with a Leaf A-D route, allowing a RBR to determine the set of NVEs that it is responsible for relaying incoming traffic to.

The per-region I-PMSI A-D routes from the RBRs and corresponding Leaf A-D routes from NVEs are comparable to the Replicator-AR and Leaf-AR routes with the Optimized IR method (Selective Mode).

This is also comparable to the per-region aggregation discussed earlier, only that the per-region I-PMSI A-D route is advertised back to the same region instead of to other regions. Similarly, the RBRs could terminate the per-PE I-PMSI A-D routes if there are no RNVEs.

7.2. Mix of inter-region and intra-region segmentation

Some more details may need to be spelled out when intra-region segmentation is used for IR optimization while in the mean time inter-region segmentation is used, with RNVEs present in different regions.

8. Multi-homing Support

If multi-homing does not span across different ASes or regions, existing procedures work with segmenation. If an ES is multi-homed to PEs in different ASes or regions, additional procedures are needed to work with segmentation. The procedures are well understood but omitted here until the requirement becomes clear.

9. EVPN DCI

In addition to inter-as/region segmentation uses cases, EVPN Overlay DC Interconnect is another important use case for EVPN tunnel segmentation.

Section 5.1.1.1 and 5.1.1.2 of [draft-ietf-bess-evpn-overlay] discuss two options of interconnecting EVPN Overlay DCs. With the GW option, DC EVPNs and Interconnect EVPN (DCI) are independent and terminate at the GWs. With the non-GW option, DC EVPNs and Interconnect EVPN form an integral EVPN, just like EVPN inter-as option-B. The GW option is discussed in details in section 3.4 of [draft-ietf-bess-dci-evpn-overlay].

The non-GW option can only be used when PEs can use VNI/VSID that has local significance (like mpls labels), and the GW option must be used otherwise. With the GW option, mac lookup must be performed when traffic comes from where non-local VNI/VSID are used. Otherwise, label/VNI/VSID switching can be used (typical inter-as option-B behavior).

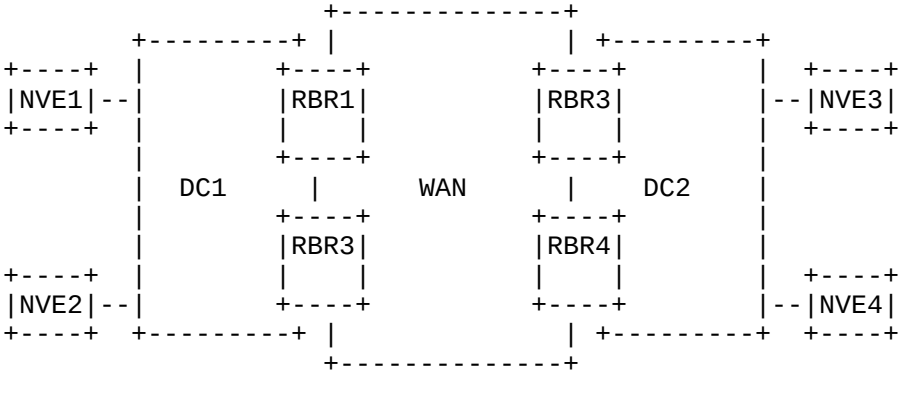
Note that with either option, BUM traffic forwarding can be based on tunnel stitching instead of mac lookup (except if IR is used together with non-local VNI/VSID), because BUM traffic goes to all PEs on

corresponding provider tunnels instead of to targeted PEs. The following sections discusses some specific details for each option.

9.1. Non-GW Option

The non-GW option can be easily compared to EVPN/mpls inter-region scenario where a region spans an entire AS - assuming that each DC is in its own AS that is different from the DCI's and other DCs'..

Consider the following diagram:



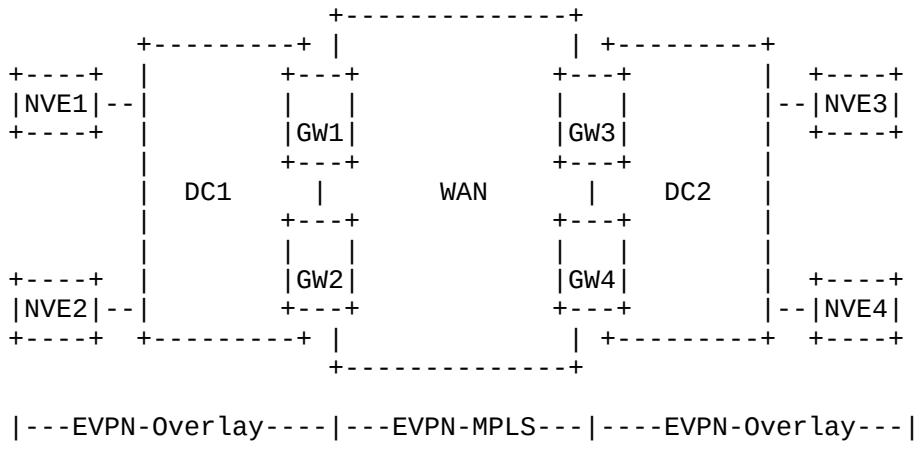
Data Center Interconnect without Gateway

The RBRs are WAN Edge routers. They re-advertise I/S-PMSI routes from one side to the other, following the previous described segmentation procedures. For example, the Inclusive Multicast Route from NVE1 is re-advertised into the WAN side by both RBR 1 and RBR2, with the LIR flag bit set in the PTA, and then re-advertised into DC2 by RBR3/4. NVE3/4 could both choose either the one re-advertised by RBR3 or by RBR4, or could each choose a different one (e.g., NVE3 chooses the one re-advertised by RBR3 while NVE4 chooses the one re-advertised by RBR4). Each either joins the advertised PIM tunnel or send a corresponding Leaf A-D route to the re-advertiser of the chosen best route. RBR3 and/or RBR4 repeat the process, followed by RBR1 and/or RBR2 doing the same. At the end, a segmented tunnel is established to reach all NVE3/4. When BUM traffic arrives on RBR1/2 from NVE1 via the tunnel segment in DC1, the multicast VXLAN encapsulation is removed and the traffic is directly switched into the segment in the WAN w/o going through mac lookup.

The per-region aggregation method (Section 6.2) can be used to limit the I-PMSI A-D routes to each DC.

9.2. GW option

Consider the following diagram adapted from section 3.4 of [draft-ietf-bess-dci-evpn-overlay]:



The GWs consumes EVPN routes from the DC side and re-originate new ones into the WAN side, and vice versa. All GWs will advertise their own I-PMSI A-D route to the DC and WAN side, but only the DF on an internal ESI (I-ESI) for the local DC will forward BUM traffic from one EVPN domain to the other. For example, BUM traffic from NVE1 will reach both GW1 and GW2, but only the DF, say GW1, will forward to the WAN side. The traffic will then reach both GW3 and GW4, but again only the DF (for the I-ESI for DC2, say GW4) will forward traffic into DC2.

In [draft-ietf-bess-dci-evpn-overlay], the traffic forwarding by GWs is based on mac lookup - because of global significance of VNIs in DCs, the VXLAN encapsulation cannot indicate to which remote NVE a known unicast packet should be forwarded to. However for BUM traffic, this is not a problem - a BUM packet only need to be put onto the appropriate tunnel. As a result, the DF GW on the I-ESI for a local DC can stitch all incoming BUM tunnels from local NVEs to its tunnel on the WAN side, and stitch all incoming BUM tunnels from remote GWs in the DCI into its tunnel on the DC side. This way, BUM traffic will be switched via label/VNI/VSID or multicast vxlan tunnel destination, bypassing mac lookup. Note that, this works only if Ingress Replication is not used for BUM traffic in an EVPN Overlay

DC, because in that case the only way to distinguish BUM traffic from known uncast traffic is by checking mac address of the packets.

Because the I-PMSI routes/tunnels are terminated in each DC/DCI, the I-PMSI routes originated by GWs are somewhat similar to the per-region I-PMSI routes discussed in the previous section. However, the per-region I-PMSI routes from RBRs in the same DC have the same route key and NVEs will only receive traffic from one of the RBRs based on best route selection, while the per-GW I-PMSI routes are distinct and all NVEs receive traffic from the same one of the GWs because only the DF on the I-ESI can forward traffic.

10. Security Considerations

This document does not seem to introduce new security risks, though this may be revised after further review and scrutiny.

11. Acknowledgements

The authors thank Eric Rosen, John Drake, and Ron Bonica for their comments and suggestions.

12. References

12.1. Normative References

[I-D.ietf-bess-ir]
Rosen, E., Subramanian, K., and J. Zhang, "Ingress Replication Tunnels in Multicast VPN", draft-ietf-bess-ir-00 (work in progress), January 2015.

[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>.

[RFC7117] Aggarwal, R., Ed., Kamite, Y., Fang, L., Rekhter, Y., and C. Kodeboniya, "Multicast in Virtual Private LAN Service (VPLS)", RFC 7117, DOI 10.17487/RFC7117, February 2014, <http://www.rfc-editor.org/info/rfc7117>.

[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>.

[RFC7524] Rekhter, Y., Rosen, E., Aggarwal, R., Morin, T., Grosclaude, I., Leymann, N., and S. Saad, "Inter-Area Point-to-Multipoint (P2MP) Segmented Label Switched Paths (LSPs)", RFC 7524, DOI 10.17487/RFC7524, May 2015, <http://www.rfc-editor.org/info/rfc7524>.

12.2. Informative References

[I-D.ietf-bess-dci-evpn-overlay]
Rabadan, J., Sathappan, S., Henderickx, W., Palislamovic, S., Balus, F., Sajassi, A., and D. Cai, "Interconnect Solution for EVPN Overlay networks", draft-ietf-bess-dci-evpn-overlay-00 (work in progress), January 2015.

[I-D.ietf-bess-evpn-overlay]
Sajassi, A., Drake, J., Bitar, N., Isaac, A., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-01 (work in progress), February 2015.

[I-D.rabadan-bess-evpn-optimized-ir]
Rabadan, J., Sathappan, S., Henderickx, W., Sajassi, A., and A. Isaac, "Optimized Ingress Replication solution for EVPN", draft-rabadan-bess-evpn-optimized-ir-00 (work in progress), October 2014.

[I-D.wijnands-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-wijnands-bier-architecture-05 (work in progress), March 2015.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <http://www.rfc-editor.org/info/rfc6513>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <http://www.rfc-editor.org/info/rfc6514>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks, Inc.

EMail: zzhang@juniper.net

Wen Lin
Juniper Networks, Inc.

EMail: wlin@juniper.net

Jorge Rabadan
Alcatel-Lucent

EMail: jorge.rabadan@alcatel-lucent.com

Keyur Patel
Cisco Systems

EMail: keyupate@cisco.com

BESS
Internet-Draft
Updates: 7432 (if approved)
Intended status: Standards Track
Expires: October 23, 2016

Z. Zhang
W. Lin
Juniper Networks
J. Rabadan
Nokia
K. Patel
Cisco Systems

April 21, 2016

Updates on EVPN BUM Procedures
draft-zzhang-bess-evpn-bum-procedure-updates-03

Abstract

This document specifies procedure updates for broadcast, unknown unicast, and multicast (BUM) traffic in Ethernet VPNs (EVPN), including selective multicast, and provider tunnel segmentation.

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 October 23, 2016.

Copyright Notice

Table of Contents

1. Terminology

To be added

2. Introduction

RFC 7432 specifies procedures to handle broadcast, unknown unicast, and multicast (BUM) traffic in Section 11, 12 and 16, using Inclusive Multicast Ethernet Tag Route. A lot of details are referred to RFC

7117 (VPLS Multicast). In particular, selective multicast is briefly mentioned for Ingress Replication but referred to RFC 7117.

RFC 7117 specifies procedures for using both inclusive tunnels and selective tunnels, similar to MVPN procedures specified in RFC 6513 and RFC 6514. A new SAFI "MCAST-VPLS" is introduced, with two types of NLRIs that match MVPN's S-PMSI A-D routes and Leaf A-D routes. The same procedures can be applied to EVPN selective multicast for both Ingress Replication and other tunnel types, but new route types need to be defined under the same EVPN SAFI.

MVPN uses terms I-PMSI and S-PMSI A-D Routes. For consistency and convenience, this document will use the same I/S-PMSI terms for VPLS and EVPN. In particular, EVPN's Inclusive Multicast Ethernet Tag Route and VPLS's VPLS A-D route carrying PTA (PMSI Tunnel Attribute) for BUM traffic purpose will all be referred to as I-PMSI A-D routes. Depending on the context, they may be used interchangeably.

MVPN provider tunnels and EVPN/VPLS BUM provider tunnels, which are referred to as MVPN/EVPN/VPLS provider tunnels in this document for simplicity, can be segmented for technical or administrative reasons, which are summarized in Section 2.1 of this document. RFC 6513/6514 cover MVPN inter-as segmentation, RFC 7117 covers VPLS multicast inter-as segmentation, and RFC 7524 (Seamless MPLS Multicast) covers inter-area segmentation for both MVPN and VPLS.

There is a difference between MVPN and VPLS multicast inter-as segmentation. For simplicity, EVPN uses the same procedures as in MVPN. All ASBRs can re-advertise their choice of the best route. Each can become the root of its intra-AS segment and inject traffic it receives from its upstream, while each downstream PE/ASBR will only pick one of the upstream ASBRs as its upstream. This is also the behavior even for VPLS in case of inter-area segmentation.

For inter-area segmentation, RFC 7524 requires the use of Inter-area P2MP Segmented Next-Hop Extended Community (S-NH-EC), and the setting of "Leaf Information Required" (LIR) flag in PTA in certain situations. Either of these could be optional in case of EVPN. Removing these requirements would make the segmentation procedures transparent to ingress and egress PEs.

RFC 7524 assumes that segmentation happens at area borders. However, it could be at "regional" borders, where a region could be a sub-area, or even an entire AS plus its external links (Section 6). That would allow for more flexible deployment scenarios (e.g. for single-area provider networks).

This document specifies/clarifies/redefines certain/additional EVPN BUM procedures, with a salient goal that they're better aligned among MVPN, EVPN and VPLS. For brevity, only changes/additions to relevant RFC 7117 and RFC 7524 procedures are specified, instead of repeating the entire procedures. Note that these are to be applied to EVPN only, even though sometimes they may sound to be updates to RFC 7117/7524.

2.1. Reasons for Tunnel Segmentation

Tunnel segmentation may be required and/or desired because of administrative and/or technical reasons.

For example, an MVPN/VPLS/EVPN network may span multiple providers and Inter-AS Option-B has to be used, in which the end-to-end provider tunnels have to be segmented at and stitched by the ASBRs. Different providers may use different tunnel technologies (e.g., provider A uses Ingress Replication, provider B uses RSVP-TE P2MP while provider C uses mLDP). Even if they use the same tunnel technology like RSVP-TE P2MP, it may be impractical to set up the tunnels across provider boundaries.

The same situations may apply between the ASes and/or areas of a single provider. For example, the backbone area may use RSVP-TE P2MP tunnels while non-backbone areas may use mLDP tunnels.

Segmentation can also be used to divide an AS/area to smaller regions, so that control plane state and/or forwarding plane state/ burden can be limited to that of individual regions. For example, instead of Ingress Replicating to 100 PEs in the entire AS, with inter-area segmentation [RFC 7524] a PE only needs to replicate to local PEs and ABRs. The ABRs will further replicate to their downstream PEs and ABRs. This not only reduces the forwarding plane burden, but also reduces the leaf tracking burden in the control plane.

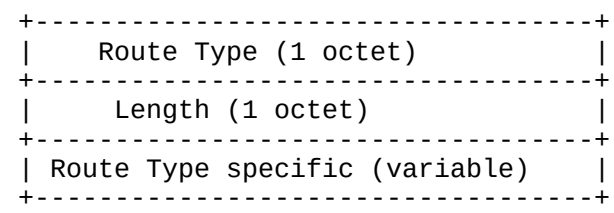
Smaller regions also have the benefit that, in case of tunnel aggregation, it is easier to find congruence among the segments of different constituent (service) tunnels and the resulting aggregation (base) tunnel in a region. This leads to better bandwidth efficiency, because the more congruent they are, the fewer leaves of the base tunnel need to discard traffic when a service tunnel's segment does not need to receive the traffic (yet it is receiving the traffic due to aggregation).

Another advantage of the smaller region is smaller BIER sub-domains. In this new multicast architecture BIER, packets carry a BitString, in which the bits correspond to edge routers that needs to receive

traffic. Smaller sub-domains means smaller BitStrings can be used without having to send multiple copies of the same packet.

3. Additional Route Types of EVPN NLRI

RFC 7432 defines the format of EVPN NLRI as the following:



So far five types have been defined:

+ 1 - Ethernet Auto-Discovery (A-D) route
+ 2 - MAC/IP Advertisement route
+ 3 - Inclusive Multicast Ethernet Tag route
+ 4 - Ethernet Segment route
+ 5 - IP Prefix Route

This document defines three additional route types:

+ 6 - Per-Region I-PMSI A-D route
+ 7 - S-PMSI A-D route
+ 8 - Leaf A-D route

The "Route Type specific" field of the type 6 and type 7 EVPN NLRIs starts with a type 1 RD, whose Administrative sub-field MUST match that of the RD in all the EVPN routes from the same advertising router for a given EVI, except the Leaf A-D route (Section 3.3).

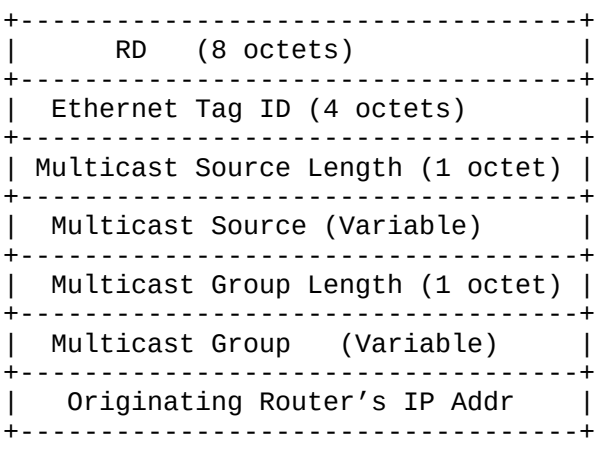
3.1. Per-Region I-PMSI A-D route

The Per-region I-PMSI A-D route has the following format. Its usage is discussed in Section 6.2.

After Ethernet Tag ID, an Extended Community (EC) is used to identify the region. Various types and sub-types of ECs provide maximum flexibility. Note that this is not an EC Attribute, but an 8-octet field embedded in the NLRI itself, following EC encoding scheme.

3.2. S-PMSI A-D route

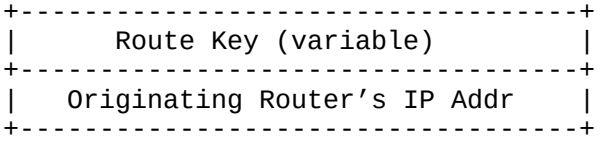
The S-PMSI A-D route has the following format:



Other than the addition of Ethernet Tag ID, it is identical to the S-PMSI A-D route as defined in RFC 7117. The procedures in RFC 7117 also apply (including wildcard functionality), except that the granularity level is per Ethernet Tag.

3.3. Leaf-AD route

The Route Type specific field of a Leaf A-D route consists of the following:



A Leaf A-D route is originated in response to a PMSI route, which could be an Inclusive Multicast Tag route, a per-region I-PMSI A-D route, an S-PMSI A-D route, or some other types of routes that may be defined in the future that triggers Leaf A-D routes. The Route Key is the "Route Type Specific" field of the route for which this Leaf A-D route is generated.

The general procedures of Leaf A-D route are first specified in RFC 6514 for MVPN. The principles apply to VPLS and EVPN as well. RFC 7117 has details for VPLS Multicast, and this document points out some specifics for EVPN, e.g. in Section 5.

4. Selective Multicast

RFC 7117 specifies Selective Multicast for VPLS. Other than that different route types and formats are specified with EVPN SAFI for S-PMSI A-D and Leaf A-D routes (Section 3), all procedures in RFC 7117 with respect to Selective Multicast apply to EVPN as well, including wildcard procedures.

5. Inter-AS Segmentation

5.1. Changes to Section 7.2.2 of RFC 7117

The first paragraph of Section 7.2.2.2 of RFC 7117 says:

> "... The best route procedures ensure that if multiple ASBRs, in an AS, receive the same Inter-AS A-D route from their EBGP neighbors, only one of these ASBRs propagates this route in Internal BGP (IBGP). This ASBR becomes the root of the intra-AS segment of the inter-AS tree and ensures that this is the only ASBR that accepts traffic into this AS from the inter-AS tree."

The above VPLS behavior requires complicated VPLS specific procedures for the ASBRs to reach agreement. For EVPN, a different approach is used and the above quoted text is not applicable to EVPN.

The Leaf A-D based procedure is used for each ASBR who re-advertises into the AS to discover the leaves on the segment rooted at itself. This is the same as the procedures for S-PMSI in RFC 7117 itself.

The following text at the end of the second bullet:

> "..... If, in order to instantiate the segment, the ASBR needs to know the leaves of the tree, then the ASBR obtains this information from the A-D routes received from other PEs/ASBRs in the ASBR’s own AS."

is changed to the following:

> "..... If, in order to instantiate the segment, the ASBR needs to know the leaves of the tree, then the ASBR MUST set the LIR flag to 1 in the PTA to trigger Leaf A-D routes from egress PEs and downstream ASBRs. It MUST be (auto-)configured with an import RT, which controls acceptance of leaf A-D routes by the ASBR."

Accordingly, the following paragraph in Section 7.2.2.4:

> "If the received Inter-AS A-D route carries the PMSI Tunnel attribute with the Tunnel Identifier set to RSVP-TE P2MP LSP, then the ASBR that originated the route MUST establish an RSVP-TE P2MP LSP with the local PE/ASBR as a leaf. This LSP MAY have been established before the local PE/ASBR receives the route, or it MAY be established after the local PE receives the route."

is changed to the following:

> "If the received Inter-AS A-D route has the LIR flag set in its PTA, then a receiving PE must originate a corresponding Leaf A-D route, and a receiving ASBR must originate a corresponding Leaf A-D route if and only if it received and imported one or more corresponding Leaf A-D routes from its downstream IBGP or EBGP peers, or it has non-null downstream forwarding state for the PIM/mLDP tunnel that instantiates its downstream intra-AS segment. The ASBR that (re-)advertised the Inter-AS A-D route then establishes a tunnel to the leaves discovered by the Leaf A-D routes."

5.2. I-PMSI Leaf Tracking

An ingress PE does not set the LIR flag in its I-PMSI's PTA, even with Ingress Replication or RSVP-TE P2MP tunnels. It does not rely on the Leaf A-D routes to discover leaves in its AS, and Section 11.2 of RFC 7432 explicitly states that the LIR flag must be set to zero.

An implementation of RFC 7432 might have used the Originating Router's IP Address field of the Inclusive Multicast Ethernet Tag routes to determine the leaves, or might have used the Next Hop field instead. Within the same AS, both will lead to the same result.

With segmentation, an ingress PE MUST determine the leaves in its AS from the BGP next hops in all its received I-PMSI A-D routes, so it does not have to set the LIR bit set to request Leaf A-D routes. PEs within the same AS will all have different next hops in their I-PMSI A-D routes (hence will all be considered as leaves), and PEs from other ASes will have the next hop in their I-PMSI A-D routes set to addresses of ASBRs in this local AS, hence only those ASBRs will be considered as leaves (as proxies for those PEs in other ASes). Note

that in case of Ingress Replication, when an ASBR re-advertises IBGP I-PMSI A-D routes, it MUST advertise the same label for all those for the same Ethernet Tag ID and the same EVI. When an ingress PE builds its flooding list, multiple routes may have the same (nexthop, label) tuple and they will only be added as a single branch in the flooding list.

5.3. Backward Compatibility

The above procedures assume that all PEs are upgraded to support the segmentation procedures:

- An ingress PE uses the Next Hop instead of Originating Router's IP Address to determine leaves for the I-PMSI tunnel.
- An egress PE sends Leaf A-D routes in response to I-PMSI routes, if the PTA has the LIR flag set (by the re-advertising ASBRs).
- In case of Ingress Replication, when an ingress PE builds its flooding list, multiple I-PMSI routes may have the same (nexthop, label) tuple and only a single branch for those will be added in the flooding list.

If a deployment has legacy PEs that does not support the above, then a legacy ingress PE would include all PEs (including those in remote ASes) as leaves of the inclusive tunnel and try to send traffic to them directly (no segmentation), which is either undesired or not possible; a legacy egress PE would not send Leaf A-D routes so the ASBRs would not know to send external traffic to them.

To address this backward compatibility problem, the following procedure can be used (see Section 6.2 for per-PE/AS/region I-PMSI A-D routes):

- An upgraded PE indicates in its per-PE I-PMSI A-D route that it supports the new procedures. Details will be provided in a future revision.
- All per-PE I-PMSI A-D routes are restricted to the local AS and not propagated to external peers.
- The ASBRs in an AS originate per-region I-PMSI A-D routes and advertise to their external peers to advertise tunnels used to carry traffic from the local AS to other ASes. Depending on the types of tunnels being used, the LIR flag in the PTA may be set, in which case the downstream ASBRs and upgraded PEs will send Leaf A-D routes to pull traffic from their upstream ASBRs. In a particular downstream AS, one of the ASBRs is elected, based on

the per-region I-PMSI A-D routes for a particular source AS, to send traffic from that source AS to legacy PEs in the downstream AS. The traffic arrives at the elected ASBR on the tunnel announced in the best per-region I-PMSI A-D route for the source AS, that the ASBR has selected of all those that it received over EBGP or IBGP sessions. Details of the election procedure will be provided in a future revision.

- In an ingress AS, if and only if an ASBR has active downstream receivers (PEs and ASBRs), which are learned either explicitly via Leaf AD routes or implicitly via PIM join or mLDP label mapping, the ASBR originates a per-PE I-PMSI A-D route (i.e., regular Inclusive Multicast Ethernet Tag route) into the local AS, and stitches incoming per-PE I-PMSI tunnels into its per-region I-PMSI tunnel. With this, it gets traffic from local PEs and send to other ASes via the tunnel announced in its per-region I-PMSI A-D route.

Note that, even if there is no backward compatibility issue, the above procedures have the benefit of keeping all per-PE I-PMSI A-D routes in their local ASes, greatly reducing the flooding of the routes and their corresponding Leaf A-D routes (when needed), and the number of inter-as tunnels.

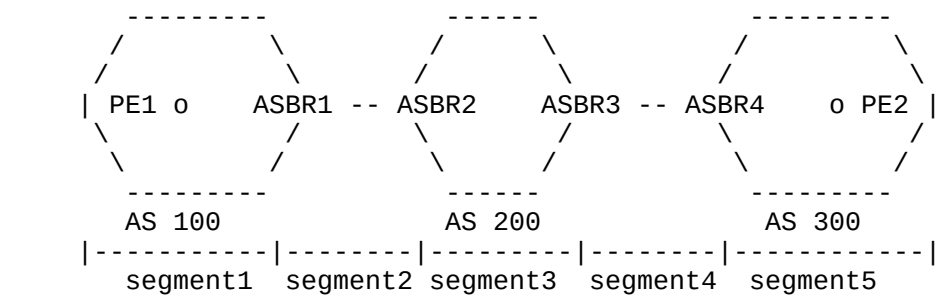
6. Inter-Region Segmentation

6.1. Area vs. Region

RFC 7524 is for MVPN/VPLS inter-area segmentation and does not explicitly cover EVPN. However, if "area" is replaced by "region" and "ABR" is replaced by "RBR" (Regional Border Router) then everything still works, and can be applied to EVPN as well.

A region can be a sub-area, or can be an entire AS including its external links. Instead of automatic region definition based on IGP areas, a region would be defined as a BGP peer group. In fact, even with IGP area based region definition, a BGP peer group listing the PEs and ABRs in an area is still needed.

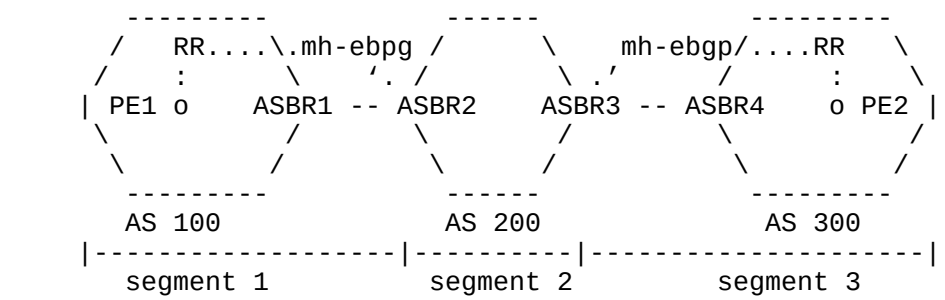
Consider the following example diagram:



The inter-as segmentation procedures specified so far (RFC 6513/6514, 7117, and Section 5 of this document) requires all ASBRs to be involved, and Ingress Replication is used between two ASBRs in different ASes.

In the above diagram, it's possible that ASBR1/4 does not support segmentation, and the provider tunnels in AS 100/300 can actually extend across the external link. In the case, the inter-region segmentation procedures can be used instead - a region is the entire (AS100 + ASBR1-ASBR2 link) or (AS300 + ASBR3-ASBR4 link). ASBR2/3 would be the RBRs, and ASBR1/4 will just be a transit core router with respect to provider tunnels.

As illustrated in the diagram below, ASBR2/3 will establish a multihop EBGP session with either a RR or directly with PEs in the neighboring AS. I/S-PMSI A-D routes from ingress PEs will not be processed by ASBR1/4. When ASBR2 re-advertises the routes into AS 200, it changes the next hop to its own address and changes PTA to specify the tunnel type/identification in its own AS. When ASBR3 re-advertises I/S-PMSI A-D routes into the neighboring AS 300, it changes the next hop to its own address and changes PTA to specify the tunnel type/identification in the neighboring region 3. Now the segment is rooted at ASBR3 and extends across the external link to PEs.



6.2. Per-region Aggregation

Notice that every I/S-PMSI route from each PE will be propagated throughout all the ASes or regions. They may also trigger corresponding Leaf A-D routes depending on the types of tunnels used in each region. This may become too many - routes and corresponding tunnels. To address this concern, the I-PMSI routes from all PEs in a AS/region can be aggregated into a single I-PMSI route originated from the RBRs, and traffic from all those individual I-PMSI tunnels will be switched into the single I-PMSI tunnel. This is like the MVPN Inter-AS I-PMSI route originated by ASBRs.

The MVPN Inter-AS I-PMSI A-D route can be better called as per-AS I-PMSI A-D route, to be compared against the (per-PE) Intra-AS I-PMSI A-D routes originated by each PE. In this document we will call it as per-region I-PMSI A-D route, in case we want to apply the aggregation at regional level. The per-PE I-PMSI routes will not be propagated to other regions. If multiple RBRs are connected to a region, then each will advertise such a route, with the same route key (Section 3.1). Similar to the per-PE I-PMSI A-D routes, RBRs/PEs in a downstream region will each select a best one from all those re-advertised by the upstream RBRs, hence will only receive traffic injected by one of them.

MVPN does not aggregate S-PMSI routes from all PEs in an AS like it does for I-PMSIs routes, because the number of PEs that will advertise S-PMSI routes for the same (s,g) or (*,g) is small. This is also the case for EVPN, i.e., there is no per-region S-PMSI routes.

Notice that per-region I-PMSI routes can also be used to address backwards compatibility issue, as discussed in Section 5.3.

The per-region I-PMSI route uses an embedded EC in NLRI to identify a region. As long as it uniquely identifies the region and the RBRs for the same region uses the same EC it is permitted. In the case where an AS number or area ID is needed, the following can be used:

- For a two-octet AS number, a Transitive Two-Octet AS-Specific EC of sub-type 0x09 (Source AS), with the Global Administrator sub-field set to the AS number and the Local Administrator sub-field set to 0.

- For a four-octet AS number, a Transitive Four-Octet AS-Specific EC of sub-type 0x09 (Source AS), with the Global Administrator sub-field set to the AS number and the Local Administrator sub-field set to 0.

- For an area ID, a Transitive IPv4-Address-Specific EC of any sub-type.

Uses of other particular ECs may be specified in other documents.

6.3. Use of S-NH-EC

RFC 7524 specifies the use of S-NH-EC because it does not allow ABRs to change the BGP next hop when they re-advertise I/S-PMSI AD routes to downstream areas. That is only to be consistent with the MVPN Inter-AS I-PMSI A-D routes, whose next hop must not be changed when they’re re-advertised by the segmenting ABRs for reasons specific to MVPN. For EVPN, it is perfectly fine to change the next hop when RBRs re-advertise the I/S-PMSI A-D routes, instead of relying on S-NH-EC. As a result, this document specifies that RBRs change the BGP next hop when they re-advertise I/S-PMSI A-D routes and do not use S-NH-EC. if a downstream PE/RBR needs to originate Leaf A-D routes, it simply uses the BGP next hop in the corresponding I/S-PMSI A-D routes to construct Route Targets.

The advantage of this is that neither ingress nor egress PEs need to understand/use S-NH-EC, and consistent procedure (based on BGP next hop) is used for both inter-as and inter-region segmentation.

6.4. Ingress PE’s I-PMSI Leaf Tracking

RFC 7524 specifies that when an ingress PE/ASBR (re-)advertises an VPLS I-PMSI A-D route, it sets the LIR flag to 1 in the route’s PTA. Similar to the inter-as case, this is actually not really needed for EVPN. To be consistent with the inter-as case, the ingress PE does not set the LIR flag in its originated I-PMSI A-D routes, and determines the leaves based on the BGP next hops in its received I-PMSI A-D routes, as specified in Section 5.2.

The same backward compatibility issue exists, and the same solution as in the inter-as case applies, as specified in Section 5.3.

7. Multi-homing Support

If multi-homing does not span across different ASes or regions, existing procedures work with segmentation. If an ES is multi-homed to PEs in different ASes or regions, additional procedures are needed to work with segmentation. The procedures are well understood but omitted here until the requirement becomes clear.

8. Security Considerations

This document does not seem to introduce new security risks, though this may be revised after further review and scrutiny.

9. Acknowledgements

The authors thank Eric Rosen, John Drake, and Ron Bonica for their comments and suggestions.

10. Contributors

The following also contributed to this document through their earlier work in EVPN selective multicast.

Junlin Zhang
Huawei Technologies
Huawei Bld., No.156 Beiqing Rd.
Beijing 100095
China

Email: jackey.zhang@huawei.com

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

Email: lizhenbin@huawei.com

11. References

11.1. Normative References

[I-D.ietf-bess-ir] Rosen, E., Subramanian, K., and J. Zhang, "Ingress Replication Tunnels in Multicast VPN", draft-ietf-bess-ir-00 (work in progress), January 2015.

[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>.

[RFC7117] Aggarwal, R., Ed., Kamite, Y., Fang, L., Rekhter, Y., and C. Kodeboniya, "Multicast in Virtual Private LAN Service (VPLS)", RFC 7117, DOI 10.17487/RFC7117, February 2014, <http://www.rfc-editor.org/info/rfc7117>.

[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>.

[RFC7524] Rekhter, Y., Rosen, E., Aggarwal, R., Morin, T., Grosclaude, I., Leymann, N., and S. Saad, "Inter-Area Point-to-Multipoint (P2MP) Segmented Label Switched Paths (LSPs)", RFC 7524, DOI 10.17487/RFC7524, May 2015, <http://www.rfc-editor.org/info/rfc7524>.

11.2. Informative References

[I-D.ietf-bess-dci-evpn-overlay]
Rabadan, J., Sathappan, S., Henderickx, W., Palislamovic, S., Balus, F., Sajassi, A., and D. Cai, "Interconnect Solution for EVPN Overlay networks", draft-ietf-bess-dci-evpn-overlay-00 (work in progress), January 2015.

[I-D.ietf-bess-evpn-overlay]
Sajassi, A., Drake, J., Bitar, N., Isaac, A., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-01 (work in progress), February 2015.

[I-D.rabadan-bess-evpn-optimized-ir]
Rabadan, J., Sathappan, S., Henderickx, W., Sajassi, A., and A. Isaac, "Optimized Ingress Replication solution for EVPN", draft-rabadan-bess-evpn-optimized-ir-00 (work in progress), October 2014.

[I-D.wijnands-bier-architecture]
Wijnands, I., Rosen, E., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast using Bit Index Explicit Replication", draft-wijnands-bier-architecture-05 (work in progress), March 2015.

[RFC6513] Rosen, E., Ed. and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", RFC 6513, DOI 10.17487/RFC6513, February 2012, <http://www.rfc-editor.org/info/rfc6513>.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, DOI 10.17487/RFC6514, February 2012, <http://www.rfc-editor.org/info/rfc6514>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Wen Lin
Juniper Networks

EMail: wlin@juniper.net

Jorge Rabadan
Nokia

EMail: jorge.rabadan@nokia.com

Keyur Patel
Cisco Systems

EMail: keyupate@cisco.com