

Network Working Group
Internet-Draft
Intended status: Proposed Standard
Expires: November 2, 2016

A. Verma
Juniper Networks

J. Drake
Juniper Networks

R. Molina
Ericsson Inc.

W. Lin
Juniper Networks

May 2, 2016

Vpls Best-site id
draft-anshuverma-bess-vpls-best-site-id-02.txt

Abstract

With network-based applications becoming prevalent, solutions that provide connectivity over wide area become more attractive for customers. In small-to-medium enterprise sector, Virtual Private LAN Service (VPLS), is a very useful service provider offering. It creates an emulated LAN segments fully capable of learning and forwarding Ethernet MAC addresses.

Today, in VPLS implementations, within the context of a VPLS PE (VE), a single-site is selected from which all PWs are rooted. The site-election mechanism is usually hard-coded by different vendors (e.g. minimum or maximum site-id), and as such, is outside end-users control. This offers no flexibility to end-users as it forces them to define the site-id allocation scheme well in advance, or deal with the consequences of a suboptimal site-id election. Moreover, whenever the elected site-id is declared down, the traffic to and from all other sites hosted within the same VE is impacted as well.

This draft defines protocol extensions to keep core-facing pseudowires (PWs) established at all times, regardless of the events

taking place on the attachment-circuit (AC) segment when using the BGP-based signaling procedures.

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on November 2, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Introduction.....	3
2. Conventions used in this document.....	5
3. Modifications to Layer 2 Info Extended Community.....	5
4. Best-site functionality.....	6
5. Remote mac-flush mechanism.....	8
6. Security Considerations.....	9
7. IANA Considerations.....	9
8. References.....	9
8.1. Normative References.....	9
8.2. Informative References.....	10
9. Authors Addresses.....	11

1. Introduction

As the popularity of VPLS services continue to expand, Service Provider requirements for a scalable multi-homed solution are becoming increasingly demanding. As dictated by RFC4762 BGP-VPLS RFC, every PE participating in a VPLS domain must be fully meshed through a bidirectional pseudowire (PW). This set of PWs is built attending to the signaling information (label-block) advertised by each PE. The label-block used to build any given PW, will be the one matching the local site being elected as 'representative' of the VPLS domain within a given PE. As stated in RFC4762, if this site is ever declared 'down', a compliant implementation will need to either withdraw the corresponding label-block, or announce that the affected site is no longer reachable. In either case, the PW will end up being destroyed, which will have a considerable impact on other local sites relying on this specific PW. Furthermore, as a considerable amount of cycles are spent in destroying/re-building affected PWs, the overall convergence period will be severely impacted for those critical multi-homed sites that need a rapid transition to a backup PE.

This draft defines protocol extensions to keep core-facing pseudowires established at all times, regardless of the events taking place on the attachment-circuit segment when using the BGP-based signaling procedures defined in [RFC4761].

Today, in VPLS implementations, within the context of a VPLS_PE (VE), a single-site is selected from which all PWs are rooted. The site-election mechanism is usually hard-coded by different vendors (e.g. minimum or maximum site-id), and as such, is outside end-users control. This offers no flexibility to end-users as it forces them to define the site-id allocation scheme well in advance, or deal with the consequences of a suboptimal site-id election. Moreover, whenever the elected site-id is declared down, the traffic to and from all other sites hosted within the same VE is impacted as well.

In BGP VPLS MH scenarios the above pitfalls are specially acute, as not only we need to factor in the cost to bring the active PW down and run DF election in primary PE, but also in the n-DF PE and all remote-PEs within the VPLS domain. Taking into account that control-plane operation is signaled through BGP protocol, is fare to expect that many of these operations will be carried out in sequence and not in parallel, so the overall cost is usually pretty considerable in scaling scenarios.

To achieve minimal traffic disruption, this draft introduces a virtual or dummy site which will serve as the preferable or best site within each VE. Thereby, its corresponding site-id value will be defined by the end-user. But more than providing greater provisioning flexibility, the real advantage of this best-site solution relies on the capability to maintain VPLS PWs established at all times regardless of the fluctuations in AC segments.

To summarize, this best-site feature offers:

- * Greater provisioning flexibility.
- * Minimal traffic disruption for non-preferable sites in multi-site VEs (upon AC going down).
- * Convergence period would be considerably reduced in MH setups during transient intervals.

2. 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.

3. Modifications to Layer 2 Info Extended Community

The Layer 2 Info Extended Community is used to signal control information of the pseudowires to be setup. The extended community format is described in [RFC4761]. This draft recommends that the Control Flags field of this extended community be used to synchronize the best-site information amongst PEs for a given L2VPN.

```

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

```

Layer-2 Info Extended Community:

Control Flags Bit Vector:

This field contains bit flags relating to pseudowire's control information. It is augmented with the definition of one new flag field. If on a given PE VPLS instance is configured with 'best-site', it will include in its VPLS BGP NLRI a Layer 2 Info Extended Community using Control Flags field with B = 1.

```

    0 1 2 3 4 5 6 7
+-----+
|D|A|F|B|T|R|C|S| (Z = MUST Be Zero)
+-----+

```

With reference to the Control Flags Bit Vector, the following bits in the Control Flags are defined; the remaining bits, MUST be set to zero when sending and MUST be ignored when receiving this Extended Community. The signaling procedure described here is therefore backwards compatible with existing implementations.

- D Defined in l2vpn-vpls-multihoming draft
- A Defined in l2vpn-auto-site-id draft
- F Defined in l2vpn-vpls-multihoming draft
- B When the bit value is 1, the PE receiving the label-block will deem the corresponding site as the most preferable site from the remote neighbor.
When the bit value is 0, the PE receiving the label-block will rely on its legacy/default site-election algorithm.
- T/R Defined in l2vpn-fat-pw-bgp draft
- C Defined in [RFC4761]
- S Defined in [RFC4761]

4. Best-site functionality:

Traditionally, vpls path selection mechanism pick the minimum (or maximum) site-id to determine the 'preferable' local site. This 'preferable' local site serves two purposes: 1) pseudowires created from the local VE will be rooted from this site, and 2) pseudowires created from remote VEs will be built towards this elected site.

In order to provide some greater flexibility in the current pre-defined site-election process, this draft proposes a solution to give priority to these 'best-sites' in detriment of those local sites with minimum (maximum) site-ids.

This solution would be fully backward compatible as VPLS-PEs on which the proposed feature isn't enable, would simply obviate the BGP extensions previously described, and thereby, would rely on their legacy/default site-election mechanism.

Let's make use of the following example to describe our solution in more details:

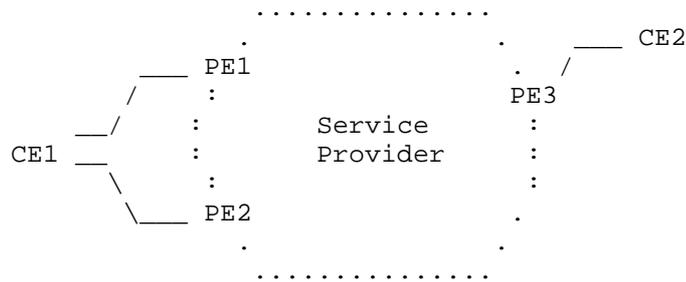


Figure 1- MH scenario with Best-site capable nodes.

A PE where 'best-site' feature is enabled in VPLS instance, behaves as a dummy site and no access interface will be associated with it. This dummy site won't be subjected to access interface down/up events; thereby, the corresponding D-bit will not be set to represent a site-down condition. The main goal here is to have a site that is permanently alive, regardless of the state of the attached circuits defined within the VPLS domain.

Each VPLS instance where a 'best-site' is defined (e.g. PE1), will signal the site's existence by setting the B-bit of the control-flags bit-vector within the L2-info extended community. Upon arrival of this BGP advertisement to the receiving PE (e.g. PE3), and only if this one is 'best-site' capable, the received B-bit will be honored and the corresponding site will be elected as the most preferable site within the remote VE (PE1).

For those neighbors where 'best-site' feature is not configured, conventional local site election will take place. For instance, if PE1 does not receive a Label-Block advertisement with B-bit set from a remote PE (PE3), it will assume that PE3 is not 'best-site' capable, and will create a pseudowire from its minimum (maximum) designated site. For the rest of the 'best-site' capable PEs, PE1 will construct pseudowires rooted at its 'best-site' site.

By proceeding to define a 'best-site' in each of the VEs across the VPLS network, we will be drastically reducing the DF transition period as no CPU cycles will need to be spent destroying and creating new pseudowires during failover events.

5. Remote mac-flush requirement:

Having a permanent pseudowire setup would not be that effective if we end up relying solely on the current implicit mac-flush mechanism. MAC addresses are automatically aged out when the pseudowire over which they are learned is deleted. This approach would collide with the proposed 'best-site' feature, in which pseudowires are kept established on a permanent basis.

An explicit-mac-flush capable implementation would ensure that MAC-to-pseudowire bindings are cleared the moment in which a DF transition is initiated. In scenarios where 'best-site' feature is enabled, no core-facing PW will be ever torn down, so previously learned MAC entries could potentially end up pointing to an invalid PW.

Thereby, to avoid potential traffic blackholes, any successful 'best-site' implementation should be capable of supporting the explicit-mac-flush mechanism depicted in [I-D.ietf-l2vpn-vpls-multihoming draft]. F-bit was introduced in the Control-Flags bit-vector, to provide a deterministic method in which any given PE can request a remote PE to flush those mac-entries learned from the former one.

Control Flags Bit Vector

```

    0 1 2 3 4 5 6 7
+-----+
|D|A|F|B|Z|Z|C|S| (Z = MUST Be Zero)
+-----+

```

When making use of this feature, a DF PE will set the 'F' bit, whereas an n-DF one will clear it when sending BGP MH advertisements. A state transition from one to zero for the 'F' bit, will be interpreted by a remote PE as an indication to flush all the MACs learned from the PE that is transitioning from DF to n-DF.

6. Security Considerations

This extension to BGP does not change the underlying security issues inherent in the existing [RFC4271].

7. IANA Considerations

8. References

8.1. Normative References

- [I-D.ietf-l2vpn-vpls-multihoming]
Kothari, B., Kompella, K., Henderickx, W., Balus, F., Uttaro, J., Palislamovic, S., and W. Lin, "BGP based Multi-homing in Virtual Private LAN Service", draft-ietf-l2vpn-vpls-multihoming-07, May 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4271] Rekhter, Y., Li, T., and S. Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, January 2006.

[RFC4761] Kompella, K. and Y. Rekhter, "Virtual Private LAN Service(VPLS) Using BGP for Auto-Discovery and Signaling", RFC 4761, January 2007.

8.2. Informative References

[RFC4447] Martini, L., Rosen, E., El-Aawar, N., Smith, T., and G. Heron, "Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)", RFC 4447, April 2006.

[RFC6624] Kompella, K., Kothari, B., and R. Cherukuri, "Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling", RFC 6624, May 2012.

9. Author's Addresses

Anshu Verma
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089, USA

Email: anshuverma@juniper.net

John Drake
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089, USA

Email: jdrake@juniper.net

Rodny Molina
Ericsson Inc.
100 Headquarters Dr,
San Jose, CA 95134

Email: rodny.molina.maldonado@ericsson.com

Wen Lin
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089, USA

Email: wlin@juniper.net

BESS Working Group
Internet-Draft
Intended status: Informational
Expires: August 23, 2016

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
February 20, 2016

Yang Data Model for BGP/MPLS L3 VPNs
draft-dhjain-bess-bgp-l3vpn-yang-01.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 August 23, 2016.

Copyright Notice

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

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

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

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	3
2.	Definitions and Acronyms	3
3.	Design of BGP L3VPN Data Model	4
3.1.	Overview	4
3.2.	VRF Specific Configuration	4
3.2.1.	VRF interface	4
3.2.2.	Route distinguisher	4
3.2.3.	Import and export route target	5
3.2.4.	Forwarding mode	5
3.2.5.	Label security	5
3.2.6.	Yang tree	5
3.3.	BGP Specific Configuration	7
3.3.1.	VPN peering	8
3.3.2.	VPN prefix limits	8
3.3.3.	Label Mode	8
3.3.4.	ASBR options	8
3.3.5.	Yang tree	8
4.	BGP Yang Module	10
5.	IANA Considerations	26
6.	Security Considerations	26
7.	Acknowledgements	26
8.	References	26
8.1.	Normative References	26

8.2. Informative References 27
Authors' Addresses 27

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 a unique identifier used in VPN routes to distinguish prefixes across different VPNs. RD is an 8 byte field as defined in the [RFC4364]. Where the first two bytes refer to type followed by 6 bytes of value. The format of the value is dependent on type. In the yang model, RDs are defined in 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 Layer 3 VPNs.
  It augments the IETF bgp yang model and IETF routing model to add L3VPN specific
  configuration and operational knobs.
```

Terms and Acronyms

AF : Address Family

AS : Autonomous System

ASBR : Autonomous Systems Border Router

BGP (bgp) : Border Gateway Protocol

CE : Customer Edge

IP (ip) : Internet Protocol

IPv4 (ipv4): Internet Protocol Version 4

IPv6 (ipv6): Internet Protocol Version 6

L3VPN: Layer 3 VPN

PE : Provider Edge

RT : Route Target

```
RD : Route Distinguisher

VPN : Virtual Private Network

VRF : Virtual Routing and Forwarding

";

revision 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 security";
        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 routing
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 candidate
for import";
}

leaf protocol {
    type enumeration {
        enum ALL {
            value "0";
            description "ALL:";
        }
        enum Direct {

```

```
        value "1";
        description "Direct:";
    }
    enum OSPF {
        value "2";
        description "OSPF:";
    }
    enum ISIS {
        value "3";
        description "ISIS:";
    }
    enum Static {
        value "4";
        description "Static:";
    }
    enum RIP {
        value "5";
        description "RIP:";
    }
    enum BGP {
        value "6";
        description "BGP:";
    }
    enum OSPFV3 {
        value "7";
        description "OSPFV3:";
    }
    enum RIPNG {
        value "8";
        description "RIPNG:";
    }
    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-istance, 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:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv4-unicast
" {
    description "Retain route targets for ASBR scenario";
    uses retain-route-targets;
    uses vpn-pfx-limit;
}
```

```
augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:l3vpn-ipv6-unicast"
{
  description "Retain route targets for ASBR scenario";
  uses retain-route-targets;
  uses vpn-pfx-limit;
}

// Label allocation mode configuration. Certain AFs only.
augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv4-unicast" {
  description "Augment BGP global AF mode for label allocation mode configuration";
  uses bgp-label-mode ;
  uses routing-tbl-limit;
}

augment "/bgp:bgp/bgp:global/bgp:afi-safis/bgp:afi-safi/bgp:ipv6-unicast" {
  description "Augment BGP global AF mode for label allocation mode configuration";
  uses bgp-label-mode ;
  uses routing-tbl-limit;
}

// 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 ASBRs";
  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

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

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

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

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

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	3
2.	Definitions and Acronyms	3
3.	Design of BGP L3VPN Data Model	4
3.1.	Overview	4
3.2.	VRF Specific Configuration	4
3.2.1.	VRF interface	4
3.2.2.	Route distinguisher	4
3.2.3.	Import and export route target	5
3.2.4.	Forwarding mode	5
3.2.5.	Label security	5
3.2.6.	Yang tree	5
3.3.	BGP Specific Configuration	7
3.3.1.	VPN peering	8
3.3.2.	VPN prefix limits	8
3.3.3.	Label Mode	8
3.3.4.	ASBR options	8
3.3.5.	Yang tree	8
4.	BGP Yang Module	10
5.	IANA Considerations	26
6.	Security Considerations	26
7.	Acknowledgements	26
8.	References	26
8.1.	Normative References	26

8.2. Informative References	27
Authors' Addresses	27

1. Introduction

YANG [RFC6020] is a data definition language that was introduced to define the contents of a conceptual data store that allows networked devices to be managed using NETCONF [RFC6241]. YANG is proving relevant beyond its initial confines, as bindings to other interfaces (e.g. ReST) and encodings other than XML (e.g. JSON) are being defined. Furthermore, YANG data models can be used as the basis of implementation for other interfaces, such as CLI and programmatic APIs.

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

1.1. Requirements Language

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

2. Definitions and Acronyms

AF: Address Family

AS: Autonomous System

ASBR: Autonomous System Border Router

BGP: Border Gateway Protocol

CE: Customer Edge

PE: Provider Edge

L3VPN: Layer 3 VPN

NETCONF: Network Configuration Protocol

RD: Route Distinguisher

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

RTFilter: Route Filter

VPN: Virtual Private Network

VRF: Virtual Routing and Forwarding

YANG: Data definition language for NETCONF

3. Design of BGP L3VPN Data Model

3.1. Overview

There are two parts of the BGP L3VPN yang data model. The first part of the model defines VRF specific parameters for L3VPN by augmenting the 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 a unique identifier used in VPN routes to distinguish prefixes across different VPNs. RD is an 8 byte field as defined in the [RFC4364]. Where the first two bytes refer to type followed by 6 bytes of value. The format of the value is dependent on type. In the yang model, RDs are defined in the 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 of 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 Layer 3 VPNs.
  It augments the IETF bgp yang model and IETF routing model to add L3VPN specific
  configuration and operational knobs.
```

Terms and Acronyms

AF : Address Family

AS : Autonomous System

ASBR : Autonomous Systems Border Router

BGP (bgp) : Border Gateway Protocol

CE : Customer Edge

IP (ip) : Internet Protocol

IPv4 (ipv4): Internet Protocol Version 4

IPv6 (ipv6): Internet Protocol Version 6

L3VPN: Layer 3 VPN

PE : Provider Edge

RT : Route Target

```
RD : Route Distinguisher

VPN : Virtual Private Network

VRF : Virtual Routing and Forwarding

";

revision 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 security";
    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 routing
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 candidate
for import";
}

leaf protocol {
    type enumeration {
        enum ALL {
            value "0";
            description "ALL:";
        }
        enum Direct {
```

```
        value "1";
        description "Direct:";
    }
    enum OSPF {
        value "2";
        description "OSPF:";
    }
    enum ISIS {
        value "3";
        description "ISIS:";
    }
    enum Static {
        value "4";
        description "Static:";
    }
    enum RIP {
        value "5";
        description "RIP:";
    }
    enum BGP {
        value "6";
        description "BGP:";
    }
    enum OSPFV3 {
        value "7";
        description "OSPFV3:";
    }
    enum RIPNG {
        value "8";
        description "RIPNG:";
    }
    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-istance, 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 Engineering Task Force
Internet-Draft
Intended status: Standards Track
Expires: January 7, 2017

V. Govindan
M. Mudigonda
A. Sajassi
Cisco Systems
G. Mirsky
Ericsson
July 6, 2016

Fault Management for EVPN networks
draft-gmsm-bess-evpn-bfd-00

Abstract

This document proposes a proactive, in-band network OAM mechanism to detect loss of continuity and miss-connection faults that affect unicast and multi-destination paths, used by Broadcast, unknown Unicast and Multicast traffic, in an EVPN network. The mechanisms proposed in the draft use the principles of the widely adopted Bidirectional Forwarding Detection protocol.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 7, 2017.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Requirements Language	3
2. Scope of the Document	3
3. Motivation for running BFD at the network layer of EVPN	3
4. Fault Detection of unicast traffic	4
5. Fault Detection of BUM traffic using ingress replication (MP2P)	5
6. Fault Detection of BUM traffic using P2MP tunnels (LSM)	5
7. BFD packet encapsulation	5
7.1. Using GAL/G-ACh encapsulation without IP headers	5
7.1.1. Ingress replication	5
7.1.1.1. Alternative encapsulation format	5
7.1.2. LSM	6
7.1.3. Unicast	6
7.1.3.1. Alternative encapsulation format	6
7.2. Using IP headers	7
8. Scalability Considerations	7
9. IANA Considerations	7
10. Security Considerations	8
11. References	8
11.1. Normative References	8
11.2. Informative References	10
Authors' Addresses	10

1. Introduction

[I-D.salam-l2vpn-evpn-oam-req-frmwk] and [I-D.oamdt-rtgwg-oam-requirement] outlines the OAM requirements of Ethernet VPN networks [RFC7432]. This document proposes mechanisms for proactive fault detection at the network(overlay) OAM layer of EVPN. EVPN fault detection mechanisms need to consider unicast and Broadcast and unknown Unicast (BUM) traffic separately since they map to different FECs in EVPN, hence this document proposes different fault detection mechanisms to suit each type using the principles of [RFC5880],[RFC5884] and Point-to-multipoint BFD [I-D.ietf-bfd-multipoint] and [I-D.ietf-bfd-multipoint-active-tail].

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. Scope of the Document

This document proposes proactive fault detection for EVPN [RFC7432] using BFD mechanisms for:

- o Unicast traffic.
- o BUM traffic using Multi-point-to-Point (MP2P) tunnels (ingress replication).
- o BUM traffic using Point-to-Multipoint (P2MP) tunnels (LSM).

This document does not discuss BFD mechanisms for:

- o EVPN variants like PBB-EVPN [RFC7623]. This will be addressed in future versions.
- o IRB solution based on EVPN [I-D.ietf-bess-evpn-inter-subnet-forwarding]. This will be addressed in future versions.
- o EVPN using other encapsulations like VxLAN, NVGRE and MPLS over GRE [I-D.ietf-bess-evpn-overlay].
- o BUM traffic using MP2MP tunnels will also be addressed in a future version of this document.

This specification describes procedures only for BFD asynchronous mode. BFD demand mode is outside the scope of this specification. Further, the use of the Echo function is outside the scope of this specification.

3. Motivation for running BFD at the network layer of EVPN

The choice of running BFD at the network layer of the OAM model for EVPN [I-D.salam-l2vpn-evpn-oam-req-frmwk] and [I-D.ooamdt-rtgwg-ooam-requirement] was made after considering the following:

- o In addition to detecting link failures in the EVPN network, BFD sessions at the network layer can be used to monitor the successful programming of labels used for setting up MP2P and P2MP

EVPN tunnels transporting Unicast and BUM traffic. The scope of reachability detection covers the ingress and the egress EVPN PE nodes and the network connecting them.

- o Monitoring a representative set of path(s) or a particular path among the multiple paths available between two EVPN PE nodes could be done by exercising the entropy labels when they are used. However paths that cannot be realized by entropy variations cannot be monitored. Fault monitoring requirements outlined by [I-D.salam-l2vpn-evpn-oam-req-frmwk] are addressed by the mechanisms proposed by this draft.

Successful establishment and maintenance of BFD sessions between EVPN PE nodes does not fully guarantee that the EVPN service is functioning. For example, an egress EVPN-PE can understand the EVPN label but could switch data to incorrect interface. However, once BFD sessions in the EVPN Network Layer reach UP state, it does provide additional confidence that data transported using those tunnels will reach the expected egress node. When the BFD session in EVPN overlay goes down that can be used as indication of the Loss-of-Connectivity defect in the EVPN underlay that would cause EVPN service failure.

4. Fault Detection of unicast traffic

The mechanisms specified in BFD for MPLS LSPs [RFC5884] [RFC7726] can be applied to bootstrap and maintain BFD sessions for unicast EVPN traffic. The discriminators required for de-multiplexing the BFD sessions MUST be exchanged using EVPN LSP ping specifying the Unicast EVPN FEC [I-D.jain-bess-evpn-lsp-ping] before establishing the BFD session. This is needed since the MPLS label stack does not contain enough information to disambiguate the sender of the packet. The usage of MPLS entropy labels take care of addressing the requirement of monitoring various paths of the multi-path server layer network [RFC6790]. Each unique realizable path between the participating PE routers MAY be monitored separately when entropy labels are used. The multi-path connectivity between two PE routers MUST be tracked by at least one representative BFD session, in which case the granularity of fault-detection would be coarser. The PE node receiving the EVPN LSP ping MUST allocate BFD discriminators using the procedures defined in [RFC7726]. Note that once the BFD session for the EVPN label is UP, either end of the BFD session MUST NOT change the local discriminator values of the BFD Control packets it generates, unless it first brings down the session as specified in [RFC5884].

5. Fault Detection of BUM traffic using ingress replication (MP2P)

Ingress replication uses separate MP2P tunnels for transporting BUM traffic from the ingress PE (head) to a set of one or more egress PEs (tails). The fault detection mechanism proposed by this document takes advantage of the fact that a unique copy is made by the head for each tail. Another key aspect to be considered in EVPN is the advertisement of the inclusive multicast route. The BUM traffic flows from a head node to a particular tail only after the head receives the inclusive multicast route containing the BUM EVPN label (downstream allocated) corresponding to the MP2P tunnel. The head-end PE performing ingress replication MUST initiate an EVPN LSP ping using the inclusive multicast FEC [I-D.jain-bess-evpn-lsp-ping] upon receiving an inclusive multicast route from a tail to bootstrap the BFD session. There MAY exist multiple BFD sessions between a head PE and an individual tail due to the usage of entropy labels [RFC6790] for an inclusive multicast FEC. The PE node receiving the EVPN LSP ping MUST allocate BFD discriminators using the procedures defined in [RFC7726]. Note that once the BFD session for the EVPN label is UP, either end of the BFD session MUST NOT change the local discriminator values of the BFD Control packets it generates, unless it first brings down the session as specified in [RFC5884].

6. Fault Detection of BUM traffic using P2MP tunnels (LSM)

TBD.

7. BFD packet encapsulation

7.1. Using GAL/G-ACh encapsulation without IP headers

7.1.1. Ingress replication

The packet contains the following labels: LSP label (transport) when not using PHP, the optional entropy label, the BUM label and the SH label [RFC7432] (where applicable). The G-ACh type is set to TBD. The G-ACh payload of the packet MUST contain the L2 header (in overlay space) followed by the IP header encapsulating the BFD packet. The MAC address of the inner packet is used to validate the <EVI, MAC> in the receiving node. The discriminator values of BFD are obtained through negotiation through the out-of-band EVPN LSP ping.

7.1.1.1. Alternative encapsulation format

A new TLV can be defined as proposed in Sec 3 of [RFC6428] to include the EVPN FEC information as a TLV following the BFD Control packet.

The format of the TLV can be reused from the EVPN Inclusive Multicast sub-TLV proposed by Fig 2 of [I-D.jain-bess-evpn-lsp-ping].

A new type (TBD3) to indicate the EVPN Inclusive Multicast SubTLV is requested from the "CC/ CV MEP-ID TLV" registry [RFC6428].

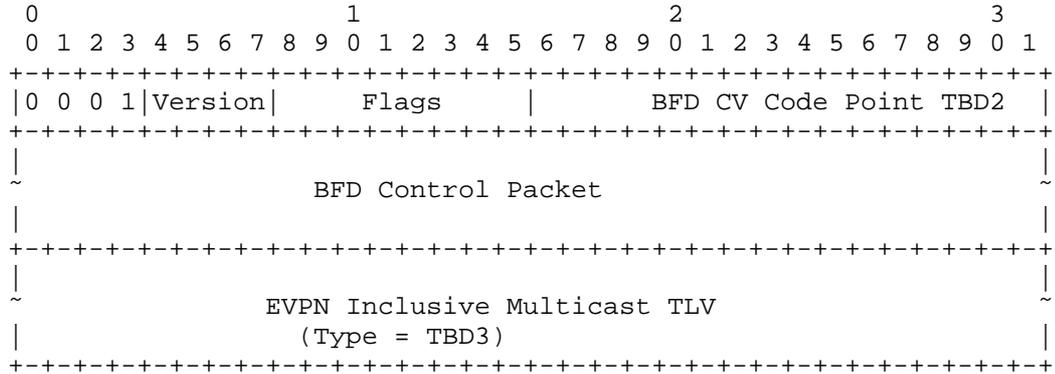


Figure 1: BFD-EVPN CV Message for EVPN Multicast (Ingress Replication)

7.1.2. LSM

TBD.

7.1.3. Unicast

The packet contains the following labels: LSP label (transport) when not using PHP, the optional entropy label and the EVPN Unicast label. The G-ACh type is set to TBD. The G-Ach payload of the packet MUST contain the L2 header (in overlay space) followed by the IP header encapsulating the BFD packet. The MAC address of the inner packet is used to validate the <EVI, MAC> in the receiving node. The discriminator values of BFD are obtained through negotiation through the out-of-band EVPN ping.

7.1.3.1. Alternative encapsulation format

A new TLV can be defined as proposed in Sec 3 of [RFC6428] to include the EVPN FEC information as a TLV following the BFD Control packet. The format of the TLV can be reused from the EVPN MAC sub-TLV proposed by Fig 1 of [I-D.jain-bess-evpn-lsp-ping]. A new type (TBD4) to indicate the EVPN MAC SubTLV is requested from the "CC/ CV MEP-ID TLV" registry [RFC6428].

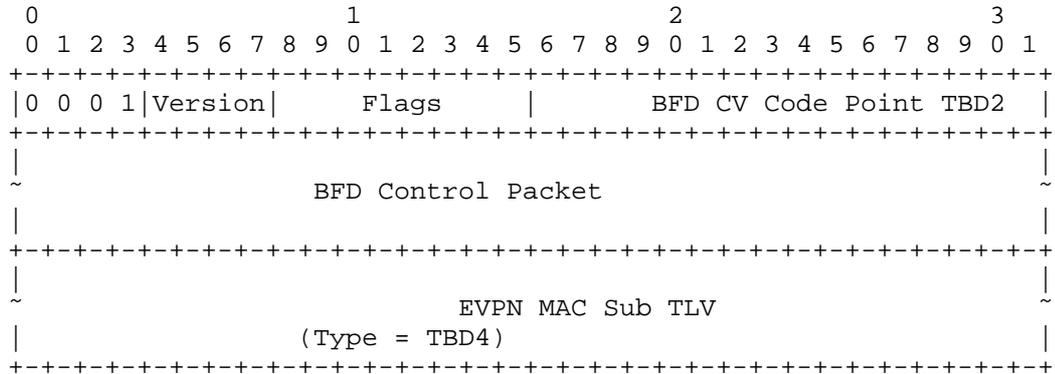


Figure 2: BFD-EVPN CV Message for EVPN Unicast

7.2. Using IP headers

The encapsulation option using IP headers will not be suited for EVPN, as using different values in the destination IP address for data and OAM (BFD) packets could cause the BFD packets to follow a different path than that of data packets. Hence this option MUST NOT be used for EVPN.

8. Scalability Considerations

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

9. IANA Considerations

IANA is requested for two channel types from the "Pseudowire Associated Channel Types" registry in [RFC4385].

TBD1 BFD-EVPN CC message

TBD2 BFD-EVPN CV message

Ed Note: Do we need a CC code point? TBD

IANA is requested to allocate the following code-points from the "CC/ CV MEP-ID TLV" registry [RFC6428]. The parent registry is the "Pseudowire Associated Channel Types" registry of [RFC4385] . All

code points within this registry shall be allocated according to the "Standards Action" procedures as specified in [RFC5226]. The items tracked in the registry will be the type, associated name, and reference. The requested values are:

TBD3 - CV code-point for BFD EVPN Inclusive multicast.

TBD4 - CV code-point for BFD EVPN Unicast.

10. Security Considerations

TBD.

11. References

11.1. Normative 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-01 (work in progress), October 2015.

[I-D.ietf-bess-evpn-overlay]

Sajassi, A., Drake, J., Bitar, N., Shekhar, R., Uttaro, J., and W. Henderickx, "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-04 (work in progress), June 2016.

[I-D.ietf-bfd-multipoint]

Katz, D., Ward, D., and J. Networks, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-08 (work in progress), April 2016.

[I-D.ietf-bfd-multipoint-active-tail]

Katz, D., Ward, D., and J. Networks, "BFD Multipoint Active Tails.", draft-ietf-bfd-multipoint-active-tail-02 (work in progress), May 2016.

[I-D.jain-bess-evpn-lsp-ping]

Jain, P., Boutros, S., and S. Salam, "LSP-Ping Mechanisms for EVPN and PBB-EVPN", draft-jain-bess-evpn-lsp-ping-03 (work in progress), May 2016.

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

- [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>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, DOI 10.17487/RFC5226, May 2008, <<http://www.rfc-editor.org/info/rfc5226>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<http://www.rfc-editor.org/info/rfc5880>>.
- [RFC5884] Aggarwal, R., Kompella, K., Nadeau, T., and G. Swallow, "Bidirectional Forwarding Detection (BFD) for MPLS Label Switched Paths (LSPs)", RFC 5884, DOI 10.17487/RFC5884, June 2010, <<http://www.rfc-editor.org/info/rfc5884>>.
- [RFC6428] Allan, D., Ed., Swallow, G., Ed., and J. Drake, Ed., "Proactive Connectivity Verification, Continuity Check, and Remote Defect Indication for the MPLS Transport Profile", RFC 6428, DOI 10.17487/RFC6428, November 2011, <<http://www.rfc-editor.org/info/rfc6428>>.
- [RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, November 2012, <<http://www.rfc-editor.org/info/rfc6790>>.
- [RFC7432] Sajassi, A., Ed., Aggarwal, R., Bitar, N., Isaac, A., Uttaro, J., Drake, J., and W. Henderickx, "BGP MPLS-Based Ethernet VPN", RFC 7432, DOI 10.17487/RFC7432, February 2015, <<http://www.rfc-editor.org/info/rfc7432>>.
- [RFC7623] Sajassi, A., Ed., Salam, S., Bitar, N., Isaac, A., and W. Henderickx, "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, DOI 10.17487/RFC7623, September 2015, <<http://www.rfc-editor.org/info/rfc7623>>.
- [RFC7726] Govindan, V., Rajaraman, K., Mirsky, G., Akiya, N., and S. Aldrin, "Clarifying Procedures for Establishing BFD Sessions for MPLS Label Switched Paths (LSPs)", RFC 7726, DOI 10.17487/RFC7726, January 2016, <<http://www.rfc-editor.org/info/rfc7726>>.

11.2. Informative References

[I-D.ooamdt-rtgwg-ooam-requirement]

Kumar, N., Pignataro, C., Kumar, D., Mirsky, G., Chen, M., Nordmark, E., Networks, J., and D. Mozes, "Overlay OAM Requirements", draft-ooamdt-rtgwg-ooam-requirement-00 (work in progress), March 2016.

[I-D.salam-l2vpn-evpn-oam-req-frmwk]

Salam, S., Sajassi, A., Aldrin, S., and J. Drake, "E-VPN Operations, Administration and Maintenance Requirements and Framework", draft-salam-l2vpn-evpn-oam-req-frmwk-02 (work in progress), January 2014.

Authors' Addresses

Vengada Prasad Govindan
Cisco Systems

Email: venggovi@cisco.com

Mudigonda Mallik
Cisco Systems

Email: mmudigon@cisco.com

Ali Sajassi
Cisco Systems

Email: sajassi@cisco.com

Gregory Mirsky
Ericsson

Email: gregory.mirsky@ericsson.com

INTERNET-DRAFT
Intended status: Proposed Standard

V. Govindan
M. Mudigonda
A. Sajassi
Cisco Systems
G. Mirsky
ZTE
D. Eastlake
Futurewei Technologies
January 2, 2020

Expires: July 1, 2020

Fault Management for EVPN networks
draft-gsm-bess-evpn-bfd-04

Abstract

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

Status of this Memo

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

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

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

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

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

Table of Contents

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

1. Introduction

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

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

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

1.1 Terminology

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

The following acronyms are used in this document.

BFD - Bidirectional Forwarding Detection [RFC5880]

BUM - Broadcast, Unknown Unicast, and Multicast

CC - Continuity Check

CV - Connectivity Verification

EVI - EVPN Instance

EVPN - Ethernet VPN [RFC7432]

FEC - Forwarding Equivalency Class

GAL - Generic Associated Channel Label [RFC5586]

LSM - Label Switched Multicast (P2MP)

LSP - Label Switched Path

MP2P - Multi-Point to Point

OAM - Operations Administration, and Maintenance

P2MP - Point to Multi-Point (LSM)

PE - Provider Edge

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

2. Scope of this Document

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

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

This document does not discuss BFD mechanisms for:

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

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

3. Motivation for Running BFD at the EVPN Network Layer

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

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

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

4. Fault Detection for Unicast Traffic

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

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

5. Fault Detection for BUM Traffic

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

5.1 Ingress Replication

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

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

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

5.2 P2MP Tunnels (Label Switched Multicast)

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

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

6. BFD Packet Encapsulation

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

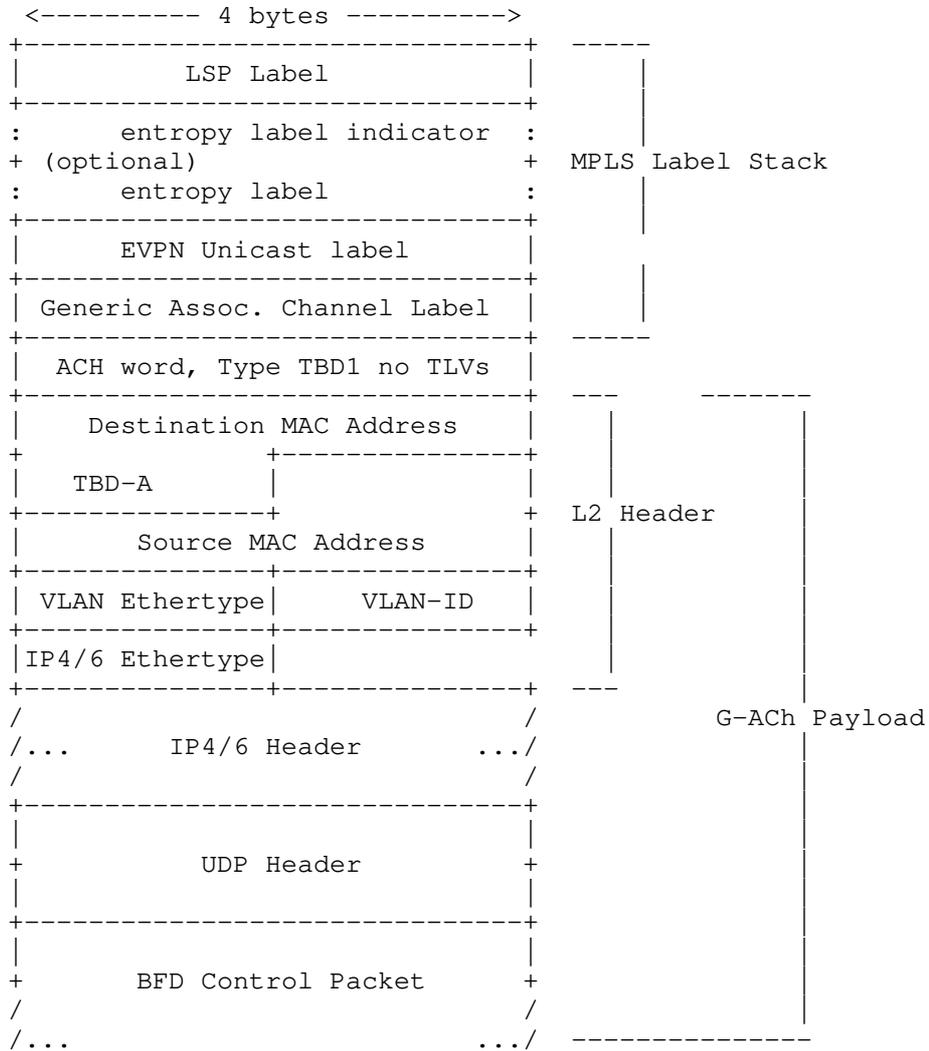
6.1 MPLS Encapsulation

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

6.1.1 Unicast

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

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



6.1.2 Ingress Replication

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

6.1.3 LSM (Label Switched Multicast, P2MP)

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

6.2 VXLAN Encapsulation

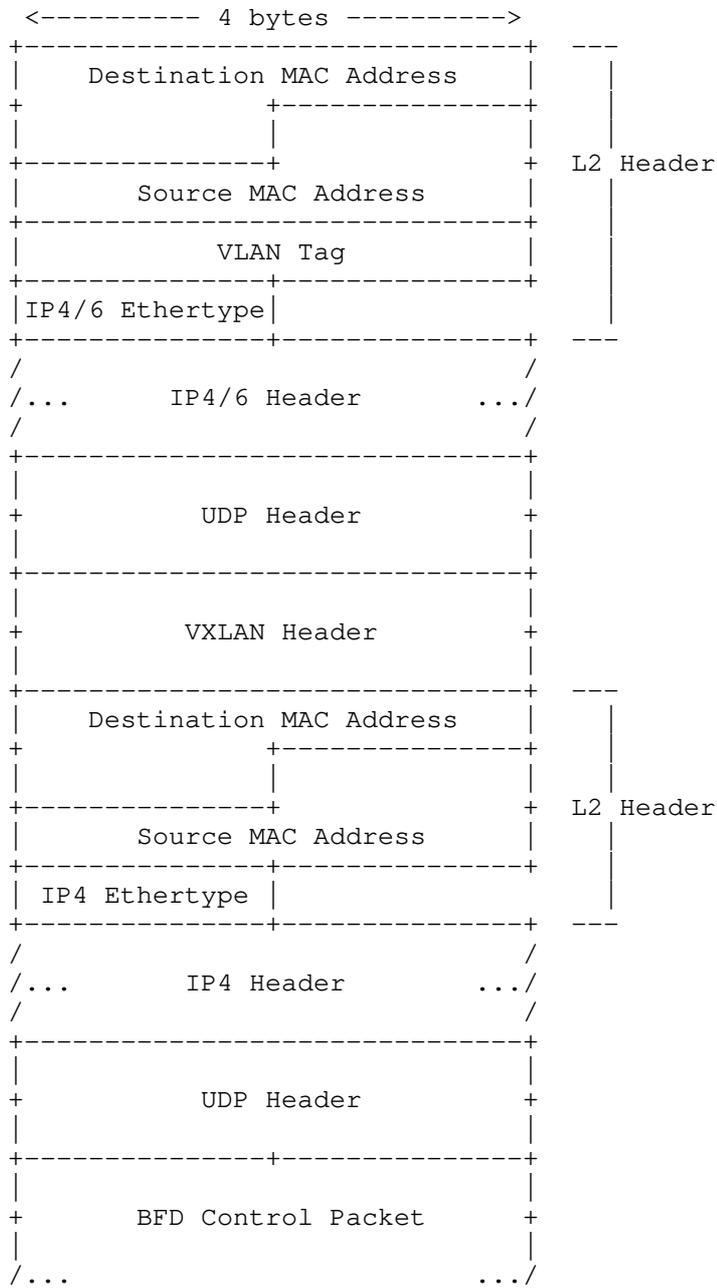
This section describes the use of the VXLAN [RFC7348] for BFD encapsulation in VXLAN based EVPN OAM. This specification conforms to [ietf-bfd-vxlan].

6.2.1 Unicast

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

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

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



6.2.2 Ingress Replication

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

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

6.2.3 LSM (Label Switched Multicast, P2MP)

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

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

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

7. BGP Distribution of BFD Discriminators

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

Unicast: MAC/IP Advertisement Route [RFC7432].

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

P2MP: TBD

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

8. Scalability Considerations

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

9. IANA Considerations

The following IANA Actions are requested.

9.1 Pseudowire Associated Channel Type

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

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

9.2 MAC Address

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

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

10. Security Considerations

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

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

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

Acknowledgement

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

Mach Chen

Normative References

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

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

Informative References

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

Authors' Addresses

Vengada Prasad Govindan
Cisco Systems

Email: venggovi@cisco.com

Mudigonda Mallik
Cisco Systems

Email: mmudigon@cisco.com

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

Email: sajassi@cisco.com

Gregory Mirsky
ZTE Corp.

Email: gregimirsky@gmail.com

Donald Eastlake, 3rd
Futurewei Technologies
2386 Panoramic Circle
Apopka, FL 32703 USA

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

Copyright, Disclaimer, and Additional IPR Provisions

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

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

BESS
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2016

J. Heitz
A. Sajassi
Cisco
J. Drake
Juniper
J. Rabadan
Nokia
June 29, 2016

Multi-homing in EVPN with Inter-AS Option B
draft-heizt-bess-evpn-option-b-00

Abstract

The BGP speaker that originates an EVPN Ethernet A-D per ES route is identified by the next-hop of the route. When the route is propagated by an ASBR as an Inter-AS Option B route, the ASBR overwrites the next-hop. This document describes a method to identify the originator of the route.

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 December 31, 2016.

Copyright Notice

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

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

Table of Contents

1. Terminology	2
2. Introduction	3
3. Solution using the Tunnel Encapsulation Attribute	4
4. Operation	4
5. Procedures at the Imposition PE	4
5.1. Primer for subsequent sections	4
5.2. PEO exists on all Type 2/5 and EAD Routes	5
5.3. Some routes do not contain PEO	5
5.4. PEO exists on EAD routes, but not on Type 2/5 routes	5
6. Security Considerations	6
7. IANA Considerations	6
8. Acknowledgements	6
9. Appendix	6
9.1. Alternative Ways to Signal PEO	6
9.1.1. Extended Community holding the IP address	6
9.1.2. Large Community holding the BGP Identifier	6
9.2. Considerations	6
10. Normative References	7
Authors' Addresses	8

1. Terminology

Inter-AS Option B: This is described in Section 10.b of [RFC4364]

EAD-per-ES: Ethernet A-D per Ethernet Segment Route.

EAD-per-EVI: Ethernet A-D per EVPN Instance Route.

EAD: EVPN Type 1 route: Ethernet Auto-discovery Route. Either an EAD-per-ES or an EAD-per-EVI route.

Type 2/5: either the EVPN Type 2 route: MAC/IP Advertisement Route or the EVPN Type 5 route: IP Prefix Route described in [I-D.ietf-bess-evpn-prefix-advertisement].

Mass Withdraw: To withdraw the route from the forwarding table. For example, a MAC route that is mass withdrawn remains in the BGP table. The MAC route is required for directing packets with the specified MAC destination address to a matching backup or alias route. When a MAC route is completely withdrawn, then the matching backup or alias routes can no longer be used for the given MAC address. The withdrawal of an EAD-per-ES route will cause the mass withdrawal of associated Type 2/5 routes as well as associated EAD-per-EVI routes.

2. Introduction

Inter-AS Option B is illustrated in Figure 1.

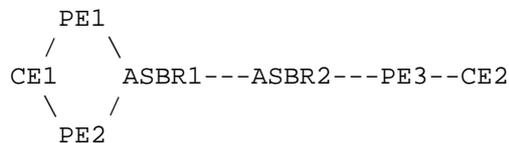


Figure 1: Inter-AS Option B

Traffic flow is from CE2 to CE1 where PE3 is an imposition PE, and PE1 and PE2 are disposition PEs.

In a multi-homing scenario, the router that performs the redundancy switchover or the load balancing (e.g. PE3) must know which router originated the Ethernet A-D routes. These redundancy functions are normally implemented on a PE, but not on an ASBR.

Quote from [RFC7432]:

"A remote PE that receives a MAC/IP Advertisement route with a non-reserved ESI SHOULD consider the advertised MAC address to be reachable via all PEs that have advertised reachability to that MAC address's EVI/ES via the combination of an Ethernet A-D per EVI route for that EVI/ES (and Ethernet tag, if applicable) AND Ethernet A-D per ES routes for that ES."

In the Intra-AS case, the remote PE identifies the "PEs that have advertised reachability" by the next-hops of the Ethernet A-D routes. In the Inter-AS option B case, ASBR1 and ASBR2 rewrite the next-hops to themselves on all EVPN route advertisements, thus losing the identity of the PE that originated an advertisement.

As a result, PE3 is unable to distinguish an EAD-per-ES route that originated at PE1 from one that originated at PE2.

3. Solution using the Tunnel Encapsulation Attribute

The Tunnel Encapsulation Attribute is specified in [I-D.ietf-idr-tunnel-encaps]. A new TLV to identify the PE of Origin is specified here. It is called PEO. The tunnel type for the PEO (suggested value 15) is to be assigned by IANA. The PEO MUST contain the Remote Endpoint Sub-TLV. The PEO must be able to uniquely identify the PE of origin within all ASes that participate in an EVPN instance.

If a BGP speaker, such as a route reflector or an ASBR, is about to re-advertise a Type 2/5 or EAD route that does not have a PEO, and will change the next-hop of that route, then it MUST add one by putting the received next-hop into the Remote Endpoint Sub-TLV of the PEO. This will ensure that all originating EVPN routes carry the necessary information for imposition PEs to function properly for aliasing and mass withdraw.

Any router that re-advertises a route that contains a PEO may modify some TLVs in the Tunnel Encapsulation Attribute attribute. However, it MUST keep the PEO unchanged. Examples are ASBR1 and ASBR2 in Figure 1.

4. Operation

For an inter-AS option B scenario, when a PE receives EVPN route(s) with PEO from an ASBR, then everything works per [RFC7432] specification including both aliasing function and mass withdraw. i.e., the imposition PE (e.g., PE3) can process mass withdraw messages (Ethernet A-D per ES route). However, if a PE receives EVPN route(s) without a PEO from an ASBR, then the mass withdraw function operates in a degenerate mode where only Ethernet A-D per EVI route can be processed (for its corresponding MAC-VRF) but not Ethernet A-D per ES route (corresponding to all the impacted MAC-VRFs). The following sections detail the procedures associated with PEO processing.

5. Procedures at the Imposition PE

5.1. Primer for subsequent sections

When routes are being compared, they must exist in the same MAC-VRF and have the same non-reserved ESI. In addition, when Type 2/5 routes and EAD-per-EVI routes are being compared, they must have the

same Ethernet Tag. Type 2/5 routes with ESI==0 do not use mass withdrawal or aliasing.

5.2. PEO exists on all Type 2/5 and EAD Routes

If all Type 2/5 and EAD routes have a PEO, then "PEs that have advertised reachability" can be identified by the PEO and the procedures of [RFC7432] can be applied without modification.

5.3. Some routes do not contain PEO

The routes that have a PEO are handled as per the previous section. The routes that do not have a PEO need the following procedures.

Type 2/5 routes without a PEO and EAD-per-EVI routes without a PEO are valid if at least one EAD-per-ES route without a PEO exists with the same next-hop. In other words: if multiple EAD-per-ES routes with the same next-hop as a Type 2/5 route exist, then the Type 2/5 route will only be mass withdrawn once all of the EAD-per-ES routes are withdrawn. This rule is necessary, because a BGP speaker may serve dual roles as ASBR and PE

[Editorial note: If it is determined that no BGP speakers exist that do not normally follow the procedures in this document (Legacy speakers) then the following sub sections may be omitted]

If an EAD-per-EVI route without a PEO is withdrawn, it will mass withdraw all Type 2/5 routes without a PEO that have the same next-hop and the same RD as the EAD-per-EVI route. This is called mass-withdraw per EVI. Note, it is not the absence of the EAD-per-EVI route that causes mass-withdrawal, but the actual withdrawal itself. If the route was never there to begin with, then no withdrawal took place.

If any entity in the network rewrites an RD, then all entities must rewrite the RD in a consistent manner, such that routes with the same RD continue to have the same RD and routes with different RDs continue to have different RDs. Note that if this condition is violated, then other network functions would also break.

5.4. PEO exists on EAD routes, but not on Type 2/5 routes

If a Type 2/5 route exists without a PEO and an EAD-per-EVI route exists with a PEO and it has the same next-hop and the same RD as the Type 2/5 route, then the Type 2/5 route shall inherit the PEO from the EAD-per-EVI route. Thereafter, section 5.2 applies.

6. Security Considerations

TBD

7. IANA Considerations

A Tunnel Encapsulation Attribute Tunnel Type for the PEO is required.

8. Acknowledgements

Thanks to Kiran Pillai, Patrice Brissette, Satya Mohanty and Keyur Patel for careful review and suggestions.

9. Appendix

9.1. Alternative Ways to Signal PEO

[Note to RFC editor: This appendix to be removed before publication]

9.1.1. Extended Community holding the IP address

The Extended Community to use must be transitive and either IPv4 Specific or IPv6 Specific as described in [RFC5701]. Thus, if it is IPv4 Specific, it will be of type 0x41 and if IPv6 Specific, it will be of type 0x40.

The extended community will hold the IP address of the PE that originates the EVPN routes.

9.1.2. Large Community holding the BGP Identifier

A PE can be uniquely identified by its BGP identifier (also called Router ID) and its AS number. A Large Community is a 4-octet AS specific extended community with a 6 octet local administrator field. The local administrator field should carry the BGP identifier.

9.2. Considerations

It may be possible to associate the EAD-per-ES route with the Type 2/5 route by matching the Administrator Subfield of the RD. However, there are too many constraints that need to be met to make this method reliable. Basically, the RD was emphatically designed to distinguish routes, not to identify them. The constraints that need to be met are:

- o The RD MUST be of Type 1. [RFC7432] recommends Type 1, but does not mandate it.

- o The Administrator subfield of the RD MUST be the same for each of these routes originated by one PE. [RFC7432] does not require this. It just says "The value field comprises an IP address of the PE", but does not say that it must be the same IP address for all. In an IPv6 only scenario, other ways will be used to assign RD.
- o The Administrator subfield of the RD MUST be unique among all PEs participating in the Inter-AS EVPN. This is likely, but not guaranteed.
- o If RDs are rewritten at AS boundaries, then the Administrator subfield MUST be rewritten in a consistent way such as to preserve the above properties.

By allowing a single EAD-per-ES route to validate all EAD-per-EVI routes and all Type 2/5 routes, some of those routes may be falsely validated. However that is the best possible outcome without a PEO. It is transient until the Type 2/5 route can be withdrawn.

The possibility of the address space of PE next-hops in one AS overlapping that of another AS was raised. In such a case, the IP address of a PE in one AS may be the same as the IP address of a different PE in another AS. Because an ASBR overwrites next-hops, this can work. The PEO contains both the ASN as well as the IP address of the originating PE, so this works too. However, EVPN route types 3 and 4 contain only the originating router's IP address, but not the originating router's ASN. Therefore, EVPN route types 3 and 4 may also need a PEO.

The possibility of making the EAD-per-EVI route mandatory was raised. This would make some of the procedures easier, because the RD of the EAD-per-EVI route can be matched with the RD of the Type 2/5 route

10. Normative References

[I-D.ietf-bess-evpn-prefix-advertisement]

Rabadan, J., Henderickx, W., Palislaamovic, S., and A. Isaac, "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-02 (work in progress), September 2015.

[I-D.ietf-idr-tunnel-encaps]

Rosen, E., Patel, K., and G. Velde, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-02 (work in progress), May 2016.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [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>>.
- [RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <<http://www.rfc-editor.org/info/rfc5701>>.
- [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>>.

Authors' Addresses

Jakob Heitz
Cisco
170 West Tasman Drive
San Jose, CA, CA 95054
USA

Email: jheitz@cisco.com

Ali Sajassi
Cisco
170 West Tasman Drive
San Jose, CA, CA 95054
USA

Email: sajassi@cisco.com

John Drake
Juniper

Email: jdrake@juniper.net

Jorge Rabadan
Nokia

Email: jorge.rabadan@nokia.com

BESS
Internet-Draft
Intended status: Standards Track
Expires: May 17, 2018

J. Heitz
A. Sajassi
Cisco
J. Drake
Juniper
J. Rabadan
Nokia
November 13, 2017

Multi-homing and E-Tree in EVPN with Inter-AS Option B
draft-heizt-bess-evpn-option-b-01

Abstract

The BGP speaker that originates an EVPN Ethernet A-D per ES route is identified by the next-hop of the route. When the route is propagated by an ASBR as an Inter-AS Option B route, the ASBR overwrites the next-hop. This document describes a method to identify the originator of the route.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on May 17, 2018.

Copyright Notice

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

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

Table of Contents

1. Terminology	2
2. Introduction	3
2.1. EVPN multi-homing and Inter-AS Option B issue	3
2.2. EVPN E-tree and Inter-AS Option B issue	4
3. Solution using the Tunnel Encapsulation Attribute	4
4. Operation	5
5. Procedures at the Imposition PE	5
5.1. Primer for subsequent sections	5
5.2. OPE exists on all Type 2/5 and EAD Routes	5
5.3. Some routes do not contain OPE	6
5.4. OPE exists on EAD routes, but not on Type 2/5 routes	6
6. Security Considerations	6
7. IANA Considerations	6
8. Acknowledgements	7
9. Appendix	7
9.1. Alternative Ways to Signal OPE	7
9.1.1. Extended Community holding the IP address	7
9.1.2. Large Community holding the BGP Identifier	7
9.2. Considerations	7
10. Normative References	8
Authors' Addresses	9

1. Terminology

Inter-AS Option B: This is described in Section 10.b of [RFC4364]

EAD-per-ES: Ethernet A-D per Ethernet Segment Route.

EAD-per-EVI: Ethernet A-D per EVPN Instance Route.

EAD: EVPN Type 1 route: Ethernet Auto-discovery Route. Either an EAD-per-ES or an EAD-per-EVI route.

Type 2/5: either the EVPN Type 2 route: MAC/IP Advertisement Route or the EVPN Type 5 route: IP Prefix Route described in [I-D.ietf-bess-evpn-prefix-advertisement].

Mass Withdraw: To withdraw the route from the forwarding table. For example, a MAC route that is mass withdrawn remains in the BGP table. The MAC route is required for directing packets with the specified MAC destination address to a matching backup or alias route. When a MAC route is completely withdrawn, then the matching backup or alias routes can no longer be used for the given MAC address. The withdrawal of an EAD-per-ES route will cause the mass withdrawal of associated Type 2/5 routes as well as associated EAD-per-EVI routes.

2. Introduction

Inter-AS Option B is illustrated in Figure 1.

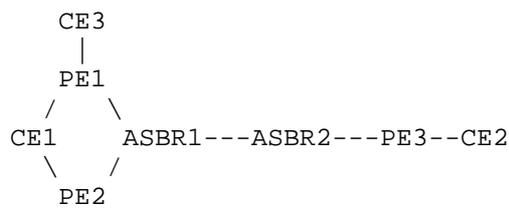


Figure 1: Inter-AS Option B

Traffic flow is from CE2 to CE1 where PE3 is an imposition PE, and PE1 and PE2 are disposition PEs. The following sections describe the issues that EVPN multi-homing and EVPN E-tree services have in these types of scenarios.

2.1. EVPN multi-homing and Inter-AS Option B issue

In a multi-homing scenario, the router that performs the redundancy switchover or the load balancing (e.g. PE3) must know which router originated the Ethernet A-D routes. These redundancy functions are normally implemented on a PE, but not on an ASBR.

Quote from [RFC7432]:

"A remote PE that receives a MAC/IP Advertisement route with a non-reserved ESI SHOULD consider the advertised MAC address to be reachable via all PEs that have advertised reachability to that MAC address's EVI/ES via the combination of an Ethernet A-D per

EVI route for that EVI/ES (and Ethernet tag, if applicable) AND Ethernet A-D per ES routes for that ES."

In the Intra-AS case, the remote PE identifies the "PEs that have advertised reachability" by the next-hops of the Ethernet A-D routes. In the Inter-AS option B case, ASBR1 and ASBR2 rewrite the next-hops to themselves on all EVPN route advertisements, thus losing the identity of the PE that originated an advertisement.

As a result, PE3 is unable to distinguish an EAD-per-ES route that originated at PE1 from one that originated at PE2.

2.2. EVPN E-tree and Inter-AS Option B issue

As described in [EVPN-Etree], leaf-to-leaf BUM traffic filtering is always performed at the disposition PE and based on the Leaf Label. The Leaf Label can be downstream allocated (ingress replication) or upstream allocated (p2mp tunnels) and is advertised in an EAD-per-ES route with ESI-0. As in the multi-homing case, the PEs must identify the PE that originated a given EAD-per-ES route, for both cases, ingress replication or p2mp tunnels, so that the leaf-to-leaf BUM filtering can be successful.

If ingress-replication is used for BUM traffic, the ingress PE must identify the originator of the ESI-0 EAD-per-ES route, program the Leaf Label and push it on the stack when sending BUM Leaf traffic to the egress PE. However, this identification of the originating PE is not possible in Inter-AS option B scenarios where ASBRs rewrite the next-hops. For instance, assuming CE2 and CE3 (Figure 1) are connected to Leaf ACs, PE1 will advertise a Leaf Label in an EAD-per-ES route for ESI-0. When CE2 sends BUM traffic, PE3 will not know what Leaf Label to use for sending traffic to PE1.

Similarly, when PE3 uses non-segmented p2mp tunnels for BUM traffic, PE3 will upstream allocate a Leaf Label and advertise it in an EAD-per-ES route, so that when sending BUM traffic with a Leaf Label, PE1 can identify that is coming from a Leaf and not forward it to CE3.

In both cases, the current Intra-AS procedures do not allow to identify the originator of the EAD-per-ES routes and therefore egress BUM filtering for leaf-to-leaf is not possible when the Leaf ACs are located on different AS'es.

3. Solution using the Tunnel Encapsulation Attribute

The Tunnel Encapsulation Attribute is specified in [I-D.ietf-idr-tunnel-encaps]. A new TLV to identify the Originating PE is specified here. It is called OPE. The tunnel type for the OPE

(suggested value 15) is to be assigned by IANA. The OPE MUST contain the Remote Endpoint Sub-TLV. The OPE must be able to uniquely identify the PE of origin within all ASes that participate in an EVPN instance.

If a BGP speaker, such as a route reflector or an ASBR, is about to re-advertise a Type 2/5 or EAD route that does not have a OPE, and will change the next-hop of that route, then it MUST add one by putting the received next-hop into the Remote Endpoint Sub-TLV of the OPE. This will ensure that all originating EVPN routes carry the necessary information for imposition PEs to function properly for aliasing and mass withdraw.

Any router that re-advertises a route that contains a OPE may modify some TLVs in the Tunnel Encapsulation Attribute attribute. However, it MUST keep the OPE unchanged. Examples are ASBR1 and ASBR2 in Figure 1.

4. Operation

For an inter-AS option B scenario, when a PE receives EVPN route(s) with OPE from an ASBR, then everything works per [RFC7432] specification including both aliasing function and mass withdraw. i.e., the imposition PE (e.g., PE3) can process mass withdraw messages (Ethernet A-D per ES route). However, if a PE receives EVPN route(s) without a OPE from an ASBR, then the mass withdraw function operates in a degenerate mode where only Ethernet A-D per EVI route can be processed (for its corresponding MAC-VRF) but not Ethernet A-D per ES route (corresponding to all the impacted MAC-VRFs). The following sections detail the procedures associated with OPE processing.

5. Procedures at the Imposition PE

5.1. Primer for subsequent sections

When routes are being compared, they must exist in the same MAC-VRF and have the same non-reserved ESI. In addition, when Type 2/5 routes and EAD-per-EVI routes are being compared, they must have the same Ethernet Tag. Type 2/5 routes with ESI==0 do not use mass withdrawal or aliasing.

5.2. OPE exists on all Type 2/5 and EAD Routes

If all Type 2/5 and EAD routes have a OPE, then "PEs that have advertised reachability" can be identified by the OPE and the procedures of [RFC7432] can be applied without modification.

5.3. Some routes do not contain OPE

The routes that have a OPE are handled as per the previous section. The routes that do not have a OPE need the following procedures.

Type 2/5 routes without a OPE and EAD-per-EVI routes without a OPE are valid if at least one EAD-per-ES route without a OPE exists with the same next-hop. In other words: if multiple EAD-per-ES routes with the same next-hop as a Type 2/5 route exist, then the Type 2/5 route will only be mass withdrawn once all of the EAD-per-ES routes are withdrawn. This rule is necessary, because a BGP speaker may serve dual roles as ASBR and PE

[Editorial note: If it is determined that no BGP speakers exist that do not normally follow the procedures in this document (Legacy speakers) then the following sub sections may be omitted]

If an EAD-per-EVI route without a OPE is withdrawn, it will mass withdraw all Type 2/5 routes without a OPE that have the same next-hop and the same RD as the EAD-per-EVI route. This is called mass-withdraw per EVI. Note, it is not the absence of the EAD-per-EVI route that causes mass-withdrawal, but the actual withdrawal itself. If the route was never there to begin with, then no withdrawal took place.

If any entity in the network rewrites an RD, then all entities must rewrite the RD in a consistent manner, such that routes with the same RD continue to have the same RD and routes with different RDs continue to have different RDs. Note that if this condition is violated, then other network functions would also break.

5.4. OPE exists on EAD routes, but not on Type 2/5 routes

If a Type 2/5 route exists without a OPE and an EAD-per-EVI route exists with a OPE and it has the same next-hop and the same RD as the Type 2/5 route, then the Type 2/5 route shall inherit the OPE from the EAD-per-EVI route. Thereafter, Section 5.2 applies.

6. Security Considerations

TBD

7. IANA Considerations

A Tunnel Encapsulation Attribute Tunnel Type for the OPE is required.

8. Acknowledgements

Thanks to Kiran Pillai, Patrice Brissette, Satya Mohanty and Keyur Patel for careful review and suggestions.

9. Appendix

9.1. Alternative Ways to Signal OPE

[Note to RFC editor: This appendix to be removed before publication]

9.1.1. Extended Community holding the IP address

The Extended Community to use must be transitive and either IPv4 Specific or IPv6 Specific as described in [RFC5701]. Thus, if it is IPv4 Specific, it will be of type 0x41 and if IPv6 Specific, it will be of type 0x40.

The Extended Community will hold the IP address of the PE that originates the EVPN routes.

9.1.2. Large Community holding the BGP Identifier

A PE can be uniquely identified by its BGP identifier (also called Router ID) and its AS number (ASN). A Large Community [RFC8092] can be used to carry the BGP identifier and the ASN. A well known Large Community needs to be allocated for this. This allocation is for the Global Administrator field. The Local Data Part 1 field should carry ASN and the Local Data Part 2 should carry the BGP identifier.

9.2. Considerations

It may be possible to associate the EAD-per-ES route with the Type 2/5 route by matching the Administrator Subfield of the RD. However, there are too many constraints that need to be met to make this method reliable. Basically, the RD was emphatically designed to distinguish routes, not to identify them. The constraints that need to be met are:

- o The RD MUST be of Type 1. [RFC7432] recommends Type 1, but does not mandate it.
- o The Administrator subfield of the RD MUST be the same for each of these routes originated by one PE. [RFC7432] does not require this. It just says "The value field comprises an IP address of the PE", but does not say that it must be the same IP address for all. In an IPv6 only scenario, other ways will be used to assign RD.

- o The Administrator subfield of the RD MUST be unique among all PEs participating in the Inter-AS EVPN. This is likely, but not guaranteed.
- o If RDs are rewritten at AS boundaries, then the Administrator subfield MUST be rewritten in a consistent way such as to preserve the above properties.

By allowing a single EAD-per-ES route to validate all EAD-per-EVI routes and all Type 2/5 routes, some of those routes may be falsely validated. However that is the best possible outcome without a OPE. It is transient until the Type 2/5 route can be withdrawn.

The possibility of the address space of PE next-hops in one AS overlapping that of another AS was raised. In such a case, the IP address of a PE in one AS may be the same as the IP address of a different PE in another AS. Because an ASBR overwrites next-hops, this can work. The OPE contains both the ASN as well as the IP address of the originating PE, so this works too. However, EVPN route types 3 and 4 contain only the originating router's IP address, but not the originating router's ASN. Therefore, EVPN route types 3 and 4 may also need a OPE.

The possibility of making the EAD-per-EVI route mandatory was raised. This would make some of the procedures easier, because the RD of the EAD-per-EVI route can be matched with the RD of the Type 2/5 route

10. Normative References

- [I-D.ietf-bess-evpn-prefix-advertisement]
Rabadan, J., Henderickx, W., Palislamovic, S., and A. Isaac, "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-02 (work in progress), September 2015.
- [I-D.ietf-idr-tunnel-encaps]
Rosen, E., Patel, K., and G. Velde, "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-02 (work in progress), May 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, DOI 10.17487/RFC4364, February 2006, <<https://www.rfc-editor.org/info/rfc4364>>.

- [RFC5701] Rekhter, Y., "IPv6 Address Specific BGP Extended Community Attribute", RFC 5701, DOI 10.17487/RFC5701, November 2009, <<https://www.rfc-editor.org/info/rfc5701>>.
- [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>>.
- [RFC8092] Heitz, J., Ed., Snijders, J., Ed., Patel, K., Bagdonas, I., and N. Hilliard, "BGP Large Communities Attribute", RFC 8092, DOI 10.17487/RFC8092, February 2017, <<https://www.rfc-editor.org/info/rfc8092>>.

Authors' Addresses

Jakob Heitz
Cisco
170 West Tasman Drive
San Jose, CA 95134
USA

Email: jheitz@cisco.com

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

Email: sajassi@cisco.com

John Drake
Juniper

Email: jdrake@juniper.net

Jorge Rabadan
Nokia

Email: jorge.rabadan@nokia.com

BESS Workgroup
Internet Draft
Intended status: Standards Track

R. Shekhar
A. Lohiya
J. Drake
Juniper

J. Rabadan
S. Sathappan
W. Henderickx
S. Palislamovic
Nokia

A. Sajassi
D. Cai
Cisco

Expires: January 8, 2017

July 7, 2016

Interconnect Solution for EVPN Overlay networks
draft-ietf-bess-dci-evpn-overlay-03

Abstract

This document describes how Network Virtualization Overlay networks (NVO) can be connected to a Wide Area Network (WAN) in order to extend the layer-2 connectivity required for some tenants. The solution analyzes the interaction between NVO networks running EVPN and other L2VPN technologies used in the WAN, such as VPLS/PBB-VPLS or EVPN/PBB-EVPN, and proposes a solution for the interworking between both.

Status of this Memo

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

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

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

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

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on January 8, 2017.

Copyright Notice

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

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

Table of Contents

1. Introduction	3
2. Decoupled Interconnect solution for EVPN overlay networks . . .	3
2.1. Interconnect requirements	4
2.2. VLAN-based hand-off	5
2.3. PW-based (Pseudowire-based) hand-off	5
2.4. Multi-homing solution on the GWs	6
2.5. Gateway Optimizations	6
2.5.1. MAC Address Advertisement Control	6
2.5.2. ARP flooding control	7
2.5.3. Handling failures between GW and WAN Edge routers . . .	7
3. Integrated Interconnect solution for EVPN overlay networks . .	8
3.1. Interconnect requirements	8
3.2. VPLS Interconnect for EVPN-Overlay networks	9
3.2.1. Control/Data Plane setup procedures on the GWs	9
3.2.2. Multi-homing procedures on the GWs	10
3.3. PBB-VPLS Interconnect for EVPN-Overlay networks	10
3.3.1. Control/Data Plane setup procedures on the GWs	10
3.3.2. Multi-homing procedures on the GWs	11
3.4. EVPN-MPLS Interconnect for EVPN-Overlay networks	11
3.4.1. Control Plane setup procedures on the GWs	11
3.4.2. Data Plane setup procedures on the GWs	13
3.4.3. Multi-homing procedures on the GWs	14
3.4.4. Impact on MAC Mobility procedures	15
3.4.5. Gateway optimizations	15
3.4.6. Benefits of the EVPN-MPLS Interconnect solution	16
3.5. PBB-EVPN Interconnect for EVPN-Overlay networks	16

3.5.1. Control/Data Plane setup procedures on the GWs	17
3.5.2. Multi-homing procedures on the GWs	17
3.5.3. Impact on MAC Mobility procedures	17
3.5.4. Gateway optimizations	17
3.6. EVPN-VXLAN Interconnect for EVPN-Overlay networks	18
3.6.1. Globally unique VNIs in the Interconnect network	18
3.6.2. Downstream assigned VNIs in the Interconnect network	19
5. Conventions and Terminology	19
6. Security Considerations	20
7. IANA Considerations	20
8. References	20
8.1. Normative References	20
8.2. Informative References	21
9. Acknowledgments	21
10. Contributors	21
11. Authors' Addresses	21

1. Introduction

[EVPN-Overlays] discusses the use of EVPN as the control plane for Network Virtualization Overlay (NVO) networks, where VXLAN, NVGRE or MPLS over GRE can be used as possible data plane encapsulation options.

While this model provides a scalable and efficient multi-tenant solution within the Data Center, it might not be easily extended to the WAN in some cases due to the requirements and existing deployed technologies. For instance, a Service Provider might have an already deployed (PBB-)VPLS or (PBB-)EVPN network that must be used to interconnect Data Centers and WAN VPN users. A Gateway (GW) function is required in these cases.

This document describes a Interconnect solution for EVPN overlay networks, assuming that the NVO Gateway (GW) and the WAN Edge functions can be decoupled in two separate systems or integrated into the same system. The former option will be referred as "Decoupled Interconnect solution" throughout the document, whereas the latter one will be referred as "Integrated Interconnect solution".

2. Decoupled Interconnect solution for EVPN overlay networks

This section describes the interconnect solution when the GW and WAN Edge functions are implemented in different systems. Figure 1 depicts the reference model described in this section.

be supported between the EVPN-Overlay network and the WAN network.

- o The following optimizations MAY be supported at the GW:
 - + Flooding reduction of unknown unicast traffic sourced from the DC Network Virtualization Edge devices (NVEs).
 - + Control of the WAN MAC addresses advertised to the DC.
 - + ARP flooding control for the requests coming from the WAN.

2.2. VLAN-based hand-off

In this option, the hand-off between the GWs and the WAN Edge routers is based on 802.1Q VLANs. This is illustrated in Figure 1 (between the GWs in NVO-1 and the WAN Edge routers). Each MAC-VRF in the GW is connected to a different VSI/MAC-VRF instance in the WAN Edge router by using a different C-TAG VLAN ID or a different combination of S/C-TAG VLAN IDs that matches at both sides.

This option provides the best possible demarcation between the DC and WAN providers and it does not require control plane interaction between both providers. The disadvantage of this model is the provisioning overhead since the service must be mapped to a S/C-TAG VLAN ID combination at both, GW and WAN Edge routers.

In this model, the GW acts as a regular Network Virtualization Edge (NVE) towards the DC. Its control plane, data plane procedures and interactions are described in [EVPN-Overlays].

The WAN Edge router acts as a (PBB-)VPLS or (PBB-)EVPN PE with attachment circuits (ACs) to the GWs. Its functions are described in [RFC4761][RFC4762][RFC6074] or [RFC7432][PBB-EVPN].

2.3. PW-based (Pseudowire-based) hand-off

If MPLS can be enabled between the GW and the WAN Edge router, a PW-based Interconnect solution can be deployed. In this option the hand-off between both routers is based on FEC128-based PWs or FEC129-based PWs (for a greater level of network automation). Note that this model still provides a clear demarcation boundary between DC and WAN, and security/QoS policies may be applied on a per PW basis. This model provides better scalability than a C-TAG based hand-off and less provisioning overhead than a combined C/S-TAG hand-off. The PW-based hand-off interconnect is illustrated in Figure 1 (between the NVO-2 GWs and the WAN Edge routers).

In this model, besides the usual MPLS procedures between GW and WAN Edge router, the GW MUST support an interworking function in each MAC-VRF that requires extension to the WAN:

- o If a FEC128-based PW is used between the MAC-VRF (GW) and the VSI (WAN Edge), the provisioning of the VCID for such PW MUST be supported on the MAC-VRF and must match the VCID used in the peer VSI at the WAN Edge router.
- o If BGP Auto-discovery [RFC6074] and FEC129-based PWs are used between the GW MAC-VRF and the WAN Edge VSI, the provisioning of the VPLS-ID MUST be supported on the MAC-VRF and must match the VPLS-ID used in the WAN Edge VSI.

2.4. Multi-homing solution on the GWs

As already discussed, single-active multi-homing, i.e. per-service load-balancing multi-homing MUST be supported in this type of interconnect. All-active multi-homing may be considered in future revisions of this document.

The GWs will be provisioned with a unique ESI per WAN interconnect and the hand-off attachment circuits or PWs between the GW and the WAN Edge router will be assigned to such ESI. The ESI will be administratively configured on the GWs according to the procedures in [RFC7432]. This Interconnect ESI will be referred as "I-ESI" hereafter.

The solution (on the GWs) MUST follow the single-active multi-homing procedures as described in [EVPN-Overlays] for the provisioned I-ESI, i.e. Ethernet A-D routes per ESI and per EVI will be advertised to the DC NVEs. The MAC addresses learned (in the data plane) on the hand-off links will be advertised with the I-ESI encoded in the ESI field.

2.5. Gateway Optimizations

The following features MAY be supported on the GW in order to optimize the control plane and data plane in the DC.

2.5.1. MAC Address Advertisement Control

The use of EVPN in the NVO networks brings a significant number of benefits as described in [EVPN-Overlays]. However, if multiple DCs are interconnected into a single EVI, each DC will have to import all of the MAC addresses from each of the other DCs.

Even if optimized BGP techniques like RT-constraint are used, the number of MAC addresses to advertise or withdraw (in case of failure) by the GWs of a given DC could overwhelm the NVEs within that DC, particularly when the NVEs reside in the hypervisors.

The solution specified in this document uses the 'Unknown MAC' route which is advertised into a given DC by each of the DC's GWs. This route is a regular EVPN MAC/IP Advertisement route in which the MAC Address Length is set to 48, the MAC address is set to 00:00:00:00:00:00, the IP length is set to 0, and the ESI field is set to the DC GW's I-ESI.

An NVE within that DC that understands the Unknown MAC route will send (unicast) a packet with an unknown unicast MAC address to one of the DCs GWs which will then forward that packet to the correct egress PE. I.e., because the ESI is set to the DC GW's I-ESI, all-active multi-homing can be applied to unknown unicast MAC addresses.

This document proposes that administrative policy determines whether and which external MAC addresses and/or the Unknown MAC route are to be advertised into a given DC. E.g., when all the DC MAC addresses are learned in the control/management plane, it may be appropriate to advertise the Unknown MAC route.

2.5.2. ARP flooding control

Another optimization mechanism, naturally provided by EVPN in the GWs, is the Proxy ARP/ND function. The GWs SHOULD build a Proxy ARP/ND cache table as per [RFC7432]. When the active GW receives an ARP/ND request/solicitation coming from the WAN, the GW does a Proxy ARP/ND table lookup and replies as long as the information is available in its table.

This mechanism is especially recommended on the GWs since it protects the DC network from external ARP/ND-flooding storms.

2.5.3. Handling failures between GW and WAN Edge routers

Link/PE failures MUST be handled on the GWs as specified in [RFC7432]. The GW detecting the failure will withdraw the EVPN routes as per [RFC7432].

Individual AC/PW failures should be detected by OAM mechanisms. For instance:

- o If the Interconnect solution is based on a VLAN hand-off, 802.lag/Y.1731 Ethernet-CFM MAY be used to detect individual AC failures on both, the GW and WAN Edge router. An individual AC failure will trigger the withdrawal of the corresponding A-D per EVI route as well as the MACs learned on that AC.
- o If the Interconnect solution is based on a PW hand-off, the LDP PW Status bits TLV MAY be used to detect individual PW failures on

both, the GW and WAN Edge router.

3. Integrated Interconnect solution for EVPN overlay networks

When the DC and the WAN are operated by the same administrative entity, the Service Provider can decide to integrate the GW and WAN Edge PE functions in the same router for obvious CAPEX and OPEX saving reasons. This is illustrated in Figure 2. Note that this model does not provide an explicit demarcation link between DC and WAN anymore.

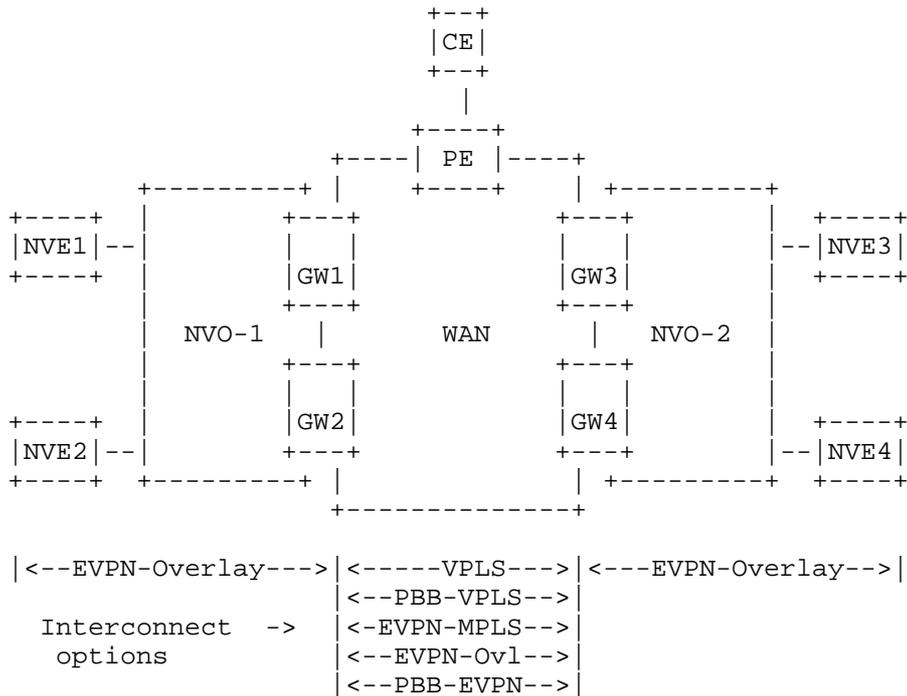


Figure 2 Integrated Interconnect model

3.1. Interconnect requirements

The solution must observe the following requirements:

- o The GW function must provide control plane and data plane interworking between the EVPN-overlay network and the L2VPN technology supported in the WAN, i.e. (PBB-)VPLS or (PBB-)EVPN, as depicted in Figure 2.
- o Multi-homing MUST be supported. Single-active multi-homing with

per-service load balancing MUST be implemented. All-active multi-homing, i.e. per-flow load-balancing, MUST be implemented as long as the technology deployed in the WAN supports it.

- o If EVPN is deployed in the WAN, the MAC Mobility, Static MAC protection and other procedures (e.g. proxy-arp) described in [RFC7432] must be supported end-to-end.
- o Any type of inclusive multicast tree MUST be independently supported in the WAN as per [RFC7432], and in the DC as per [EVPN-Overlays].

3.2. VPLS Interconnect for EVPN-Overlay networks

3.2.1. Control/Data Plane setup procedures on the GWs

Regular MPLS tunnels and TLDP/BGP sessions will be setup to the WAN PEs and RRs as per [RFC4761][RFC4762][RFC6074] and overlay tunnels and EVPN will be setup as per [EVPN-Overlays]. Note that different route-targets for the DC and for the WAN are normally required. A single type-1 RD per service may be used.

In order to support multi-homing, the GWs will be provisioned with an I-ESI (see section 2.4), that will be unique per interconnection. All the [RFC7432] procedures are still followed for the I-ESI, e.g. any MAC address learned from the WAN will be advertised to the DC with the I-ESI in the ESI field.

A MAC-VRF per EVI will be created in each GW. The MAC-VRF will have two different types of tunnel bindings instantiated in two different split-horizon-groups:

- o VPLS PWs will be instantiated in the "WAN split-horizon-group".
- o Overlay tunnel bindings (e.g. VXLAN, NVGRE) will be instantiated in the "DC split-horizon-group".

Attachment circuits are also supported on the same MAC-VRF, but they will not be part of any of the above split-horizon-groups.

Traffic received in a given split-horizon-group will never be forwarded to a member of the same split-horizon-group.

As far as BUM flooding is concerned, a flooding list will be created with the sub-list created by the inclusive multicast routes and the sub-list created for VPLS in the WAN. BUM frames received from a local attachment circuit will be flooded to both sub-lists. BUM frames received from the DC or the WAN will be forwarded to the

flooding list observing the split-horizon-group rule described above.

Note that the GWs are not allowed to have an EVPN binding and a PW to the same far-end within the same MAC-VRF in order to avoid loops and packet duplication. This is described in [EVPN-VPLS-INTEGRATION].

The optimizations procedures described in section 2.5 can also be applied to this model.

3.2.2. Multi-homing procedures on the GWs

Single-active multi-homing MUST be supported on the GWs. All-active multi-homing is not supported by VPLS.

All the single-active multi-homing procedures as described by [EVPN-Overlays] will be followed for the I-ESI.

The non-DF GW for the I-ESI will block the transmission and reception of all the bindings in the "WAN split-horizon-group" for BUM and unicast traffic.

3.3. PBB-VPLS Interconnect for EVPN-Overlay networks

3.3.1. Control/Data Plane setup procedures on the GWs

In this case, there is no impact on the procedures described in [RFC7041] for the B-component. However the I-component instances become EVI instances with EVPN-Overlay bindings and potentially local attachment circuits. M MAC-VRF instances can be multiplexed into the same B-component instance. This option provides significant savings in terms of PWs to be maintained in the WAN.

The I-ESI concept described in section 3.2.1 will also be used for the PBB-VPLS-based Interconnect.

B-component PWs and I-component EVPN-overlay bindings established to the same far-end will be compared. The following rules will be observed:

- o Attempts to setup a PW between the two GWs within the B-component context will never be blocked.
- o If a PW exists between two GWs for the B-component and an attempt is made to setup an EVPN binding on an I-component linked to that B-component, the EVPN binding will be kept operationally down. Note that the BGP EVPN routes will still be valid but not used.

- o The EVPN binding will only be up and used as long as there is no PW to the same far-end in the corresponding B-component. The EVPN bindings in the I-components will be brought down before the PW in the B-component is brought up.

The optimizations procedures described in section 2.5 can also be applied to this Interconnect option.

3.3.2. Multi-homing procedures on the GWs

Single-active multi-homing MUST be supported on the GWs.

All the single-active multi-homing procedures as described by [EVPN-Overlays] will be followed for the I-ESI for each EVI instance connected to B-component.

3.4. EVPN-MPLS Interconnect for EVPN-Overlay networks

If EVPN for MPLS tunnels, EVPN-MPLS hereafter, is supported in the WAN, an end-to-end EVPN solution can be deployed. The following sections describe the proposed solution as well as the impact required on the [RFC7432] procedures.

3.4.1. Control Plane setup procedures on the GWs

The GWs MUST establish separate BGP sessions for sending/receiving EVPN routes to/from the DC and to/from the WAN. Normally each GW will setup one (two) BGP EVPN session(s) to the DC RR(s) and one(two) session(s) to the WAN RR(s).

In order to facilitate separate BGP processes for DC and WAN, EVPN routes sent to the WAN SHOULD carry a different route-distinguisher (RD) than the EVPN routes sent to the DC. In addition, although reusing the same value is possible, different route-targets are expected to be handled for the same EVI in the WAN and the DC. Note that the EVPN service routes sent to the DC RRs will normally include a [RFC5512] BGP encapsulation extended community with a different tunnel type than the one sent to the WAN RRs.

As in the other discussed options, an I-ESI will be configured on the GWs for multi-homing. This I-ESI represents the WAN to the DC but also the DC to the WAN. Optionally, different I-ESI values MAY be configured for representing the WAN and the DC, as long as the I-ESI values are consistently configured on the redundant GWs and the same GW becomes DF for both I-ESIs.

Received EVPN routes will never be reflected on the GWs but consumed and re-advertised (if needed):

- o Ethernet A-D routes, ES routes and Inclusive Multicast routes are consumed by the GWs and processed locally for the corresponding [RFC7432] procedures.
- o MAC/IP advertisement routes will be received, imported and if they become active in the MAC-VRF MAC FIB, the information will be re-advertised as new routes with the following fields:
 - + The RD will be the GW's RD for the MAC-VRF.
 - + The ESI will be set to the I-ESI.
 - + The Ethernet-tag value will be kept from the received NLRI.
 - + The MAC length, MAC address, IP Length and IP address values will be kept from the received NLRI.
 - + The MPLS label will be a local 20-bit value (when sent to the WAN) or a DC-global 24-bit value (when sent to the DC).
 - + The appropriate Route-Targets (RTs) and [RFC5512] BGP Encapsulation extended community will be used according to [EVPN-Overlays].

The GWs will also generate the following local EVPN routes that will be sent to the DC and WAN, with their corresponding RTs and [RFC5512] BGP Encapsulation extended community values:

- o ES route for the I-ESI.
- o Ethernet A-D routes per ESI and EVI for the I-ESI. The A-D per-EVI routes sent to the WAN and the DC will have a consistent Ethernet-Tag values.
- o Inclusive Multicast routes with independent tunnel type value for the WAN and DC. E.g. a P2MP LSP may be used in the WAN whereas ingress replication may be used in the DC. The routes sent to the WAN and the DC will have a consistent Ethernet-Tag.
- o MAC/IP advertisement routes for MAC addresses learned in local attachment circuits. Note that these routes will not include the I-ESI, but ESI=0 or different from 0 for local Ethernet Segments (ES). The routes sent to the WAN and the DC will have a consistent Ethernet-Tag.

Assuming GW1 and GW2 are peer GWs of the same DC, each GW will generate two sets of local service routes: Set-DC will be sent to the DC RRs and will include A-D per EVI, Inclusive Multicast and MAC/IP

routes for the DC encapsulation and RT. Set-WAN will be sent to the WAN RRs and will include the same routes but using the WAN RT and encapsulation. GW1 and GW2 will receive each other's set-DC and set-WAN. This is the expected behavior on GW1 and GW2 for locally generated routes:

- o Inclusive multicast routes: when setting up the flooding lists for a given MAC-VRF, each GW will include its DC peer GW only in the EVPN-overlay flooding list (by default) and not the EVPN-MPLS flooding list. That is, GW2 will import two Inclusive Multicast routes from GW1 (from set-DC and set-WAN) but will only consider one of the two, having the set-DC route higher priority. An administrative option MAY change this preference so that the set-WAN route is selected first.
- o MAC/IP advertisement routes for local attachment circuits: as above, the GW will select only one, having the route from the set-DC a higher priority. As for the Inclusive multicast routes, an administrative option MAY change this priority.

Note that, irrespective of the encapsulation, EVPN routes always have higher priority than VPLS AD routes as per [EVPN-VPLS-INTEGRATION].

3.4.2. Data Plane setup procedures on the GWs

The procedure explained at the end of the previous section will make sure there are no loops or packet duplication between the GWs of the same DC (for frames generated from local ACs) since only one EVPN binding per EVI will be setup in the data plane between the two nodes. That binding will by default be added to the EVPN-overlay flooding list.

As for the rest of the EVPN tunnel bindings, they will be added to one of the two flooding lists that each GW sets up for the same MAC-VRF:

- o EVPN-overlay flooding list (composed of bindings to the remote NVEs or multicast tunnel to the NVEs).
- o EVPN-MPLS flooding list (composed of MP2P or LSM tunnel to the remote PEs)

Each flooding list will be part of a separate split-horizon-group: the WAN split-horizon-group or the DC split-horizon-group. Traffic generated from a local AC can be flooded to both split-horizon-groups. Traffic from a binding of a split-horizon-group can be flooded to the other split-horizon-group and local ACs, but never to a member of its own split-horizon-group.

When either GW1 or GW2 receive a BUM frame on an overlay tunnel, they will perform a tunnel IP SA lookup to determine if the packet's origin is the peer DC GW, i.e. GW2 or GW1 respectively. If the packet is coming from the peer DC GW, it MUST only be flooded to local attachment circuits and not to the WAN split-horizon-group (the assumption is that the peer GW would have sent the BUM packet to the WAN directly).

3.4.3. Multi-homing procedures on the GWs

Single-active as well as all-active multi-homing MUST be supported.

All the multi-homing procedures as described by [RFC7432] will be followed for the DF election for I-ESI, as well as the backup-path (single-active) and aliasing (all-active) procedures on the remote PEs/NVEs. The following changes are required at the GW with respect to the I-ESI:

- o Single-active multi-homing; assuming a WAN split-horizon-group, a DC split-horizon-group and local ACs on the GWs:
 - + Forwarding behavior on the non-DF: the non-DF MUST NOT forward BUM or unicast traffic received from a given split-horizon-group to a member of its own split-horizon-group or to the other split-horizon-group. Only forwarding to local ACs is allowed (as long as they are not part of an ES for which the node is non-DF).
 - + Forwarding behavior on the DF: the DF MUST NOT forward BUM or unicast traffic received from a given split-horizon-group to a member of his own split-horizon group or to the non-DF. Forwarding to the other split-horizon-group (except the non-DF) and local ACs is allowed (as long as the ACs are not part of an ES for which the node is non-DF).
- o All-active multi-homing; assuming a WAN split-horizon-group, a DC split-horizon-group and local ACs on the GWs:
 - + Forwarding behavior on the non-DF: the non-DF follows the same behavior as the non-DF in the single-active case but only for BUM traffic. Unicast traffic received from a split-horizon-group MUST NOT be forwarded to a member of its own split-horizon-group but can be forwarded normally to the other split-horizon-group and local ACs. If a known unicast packet is identified as a "flooded" packet, the procedures for BUM traffic MUST be followed.

- + Forwarding behavior on the DF: the DF follows the same behavior as the DF in the single-active case but only for BUM traffic. Unicast traffic received from a split-horizon-group MUST NOT be forwarded to a member of its own split-horizon-group but can be forwarded normally to the other split-horizon-group and local ACs. If a known unicast packet is identified as a "flooded" packet, the procedures for BUM traffic MUST be followed.
- o No ESI label is required to be signaled for I-ESI for its use by the non-DF in the data path. This is possible because the non-DF and the DF will never forward BUM traffic (coming from a split-horizon-group) to each other.

3.4.4. Impact on MAC Mobility procedures

MAC Mobility procedures described in [RFC7432] are not modified by this document.

Note that an intra-DC MAC move still leaves the MAC attached to the same I-ESI, so under the rules of [RFC7432] this is not considered a MAC mobility event. Only when the MAC moves from the WAN domain to the DC domain (or from one DC to another) the MAC will be learned from a different ES and the MAC Mobility procedures will kick in.

The sticky bit indication in the MAC Mobility extended community MUST be propagated between domains.

3.4.5. Gateway optimizations

All the Gateway optimizations described in section 2.5 MAY be applied to the GWs when the Interconnect is based on EVPN-MPLS.

In particular, the use of the Unknown MAC route, as described in section 2.5.1, solves some transient packet duplication issues in cases of all-active multi-homing, as explained below.

Consider the diagram in Figure 2 for EVPN-MPLS Interconnect and all-active multi-homing, and the following sequence:

- a) MAC Address M1 is advertised from NVE3 in EVI-1.
- b) GW3 and GW4 learn M1 for EVI-1 and re-advertise M1 to the WAN with I-ESI-2 in the ESI field.
- c) GW1 and GW2 learn M1 and install GW3/GW4 as next-hops following the EVPN aliasing procedures.

- d) Before NVE1 learns M1, a packet arrives at NVE1 with destination M1. If the Unknown MAC route had not been advertised into the DC, NVE1 would have flooded the packet throughout the DC, in particular to both GW1 and GW2. If the same VNI/VSID is used for both known unicast and BUM traffic, as is typically the case, there is no indication in the packet that it is a BUM packet and both GW1 and GW2 would have forwarded it. However, because the Unknown MAC route had been advertised into the DC, NVE1 will unicast the packet to either GW1 or GW2.
- e) Since both GW1 and GW2 know M1, the GW receiving the packet will forward it to either GW3 or GW4.

3.4.6. Benefits of the EVPN-MPLS Interconnect solution

Besides retaining the EVPN attributes between Data Centers and throughout the WAN, the EVPN-MPLS Interconnect solution on the GWs has some benefits compared to pure BGP EVPN RR or Inter-AS model B solutions without a gateway:

- o The solution supports the connectivity of local attachment circuits on the GWs.
- o Different data plane encapsulations can be supported in the DC and the WAN.
- o Optimized multicast solution, with independent inclusive multicast trees in DC and WAN.
- o MPLS Label aggregation: for the case where MPLS labels are signaled from the NVEs for MAC/IP Advertisement routes, this solution provides label aggregation. A remote PE MAY receive a single label per GW MAC-VRF as opposed to a label per NVE/MAC-VRF connected to the GW MAC-VRF. For instance, in Figure 2, PE would receive only one label for all the routes advertised for a given MAC-VRF from GW1, as opposed to a label per NVE/MAC-VRF.
- o The GW will not propagate MAC mobility for the MACs moving within a DC. Mobility intra-DC is solved by all the NVEs in the DC. The MAC Mobility procedures on the GWs are only required in case of mobility across DCs.
- o Proxy-ARP/ND function on the DC GWs can be leveraged to reduce ARP/ND flooding in the DC or/and in the WAN.

3.5. PBB-EVPN Interconnect for EVPN-Overlay networks

[PBB-EVPN] is yet another Interconnect option. It requires the use of GWs where I-components and associated B-components are EVI instances.

3.5.1. Control/Data Plane setup procedures on the GWs

EVPN will run independently in both components, the I-component MAC-VRF and B-component MAC-VRF. Compared to [PBB-EVPN], the DC C-MACs are no longer learned in the data plane on the GW but in the control plane through EVPN running on the I-component. Remote C-MACs coming from remote PEs are still learned in the data plane. B-MACs in the B-component will be assigned and advertised following the procedures described in [PBB-EVPN].

An I-ESI will be configured on the GWs for multi-homing, but it will only be used in the EVPN control plane for the I-component EVI. No non-reserved ESIs will be used in the control plane of the B-component EVI as per [PBB-EVPN].

The rest of the control plane procedures will follow [RFC7432] for the I-component EVI and [PBB-EVPN] for the B-component EVI.

From the data plane perspective, the I-component and B-component EVPN bindings established to the same far-end will be compared and the I-component EVPN-overlay binding will be kept down following the rules described in section 3.3.1.

3.5.2. Multi-homing procedures on the GWs

Single-active as well as all-active multi-homing MUST be supported.

The forwarding behavior of the DF and non-DF will be changed based on the description outlined in section 3.4.3, only replacing the "WAN split-horizon-group" for the B-component.

3.5.3. Impact on MAC Mobility procedures

C-MACs learned from the B-component will be advertised in EVPN within the I-component EVI scope. If the C-MAC was previously known in the I-component database, EVPN would advertise the C-MAC with a higher sequence number, as per [RFC7432]. From a Mobility perspective and the related procedures described in [RFC7432], the C-MACs learned from the B-component are considered local.

3.5.4. Gateway optimizations

All the considerations explained in section 3.4.5 are applicable to the PBB-EVPN Interconnect option.

3.6. EVPN-VXLAN Interconnect for EVPN-Overlay networks

If EVPN for Overlay tunnels is supported in the WAN and a GW function is required, an end-to-end EVPN solution can be deployed. This section focuses on the specific case of EVPN for VXLAN (EVPN-VXLAN hereafter) and the impact on the [RFC7432] procedures.

This use-case assumes that NVEs need to use the VNIs or VSIDs as a globally unique identifiers within a data center, and a Gateway needs to be employed at the edge of the data center network to translate the VNI or VSID when crossing the network boundaries. This GW function provides VNI and tunnel IP address translation. The use-case in which local downstream assigned VNIs or VSIDs can be used (like MPLS labels) is described by [EVPN-Overlays].

While VNIs are globally significant within each DC, there are two possibilities in the Interconnect network:

- a) Globally unique VNIs in the Interconnect network:
In this case, the GWs and PEs in the Interconnect network will agree on a common VNI for a given EVI. The RT to be used in the Interconnect network can be auto-derived from the agreed Interconnect VNI. The VNI used inside each DC MAY be the same as the Interconnect VNI.
- b) Downstream assigned VNIs in the Interconnect network.
In this case, the GWs and PEs MUST use the proper RTs to import/export the EVPN routes. Note that even if the VNI is downstream assigned in the Interconnect network, and unlike option B, it only identifies the <Ethernet Tag, GW> pair and not the <Ethernet Tag, egress PE> pair. The VNI used inside each DC MAY be the same as the Interconnect VNI. GWs SHOULD support multiple VNI spaces per EVI (one per Interconnect network they are connected to).

In both options, NVEs inside a DC only have to be aware of a single VNI space, and only GWs will handle the complexity of managing multiple VNI spaces. In addition to VNI translation above, the GWs will provide translation of the tunnel source IP for the packets generated from the NVEs, using their own IP address. GWs will use that IP address as the BGP next-hop in all the EVPN updates to the Interconnect network.

The following sections provide more details about these two options.

3.6.1. Globally unique VNIs in the Interconnect network

Considering Figure 2, if a host H1 in NVO-1 needs to communicate with

a host H2 in NVO-2, and assuming that different VNIs are used in each DC for the same EVI, e.g. VNI-10 in NVO-1 and VNI-20 in NVO-2, then the VNIs must be translated to a common Interconnect VNI (e.g. VNI-100) on the GWs. Each GW is provisioned with a VNI translation mapping so that it can translate the VNI in the control plane when sending BGP EVPN route updates to the Interconnect network. In other words, GW1 and GW2 must be configured to map VNI-10 to VNI-100 in the BGP update messages for H1's MAC route. This mapping is also used to translate the VNI in the data plane in both directions, that is, VNI-10 to VNI-100 when the packet is received from NVO-1 and the reverse mapping from VNI-100 to VNI-10 when the packet is received from the remote NVO-2 network and needs to be forwarded to NVO-1.

The procedures described in section 3.4 will be followed, considering that the VNIs advertised/received by the GWs will be translated accordingly.

3.6.2. Downstream assigned VNIs in the Interconnect network

In this case, if a host H1 in NVO-1 needs to communicate with a host H2 in NVO-2, and assuming that different VNIs are used in each DC for the same EVI, e.g. VNI-10 in NVO-1 and VNI-20 in NVO-2, then the VNIs must be translated as in section 3.6.1. However, in this case, there is no need to translate to a common Interconnect VNI on the GWs. Each GW can translate the VNI received in an EVPN update to a locally assigned VNI advertised to the Interconnect network. Each GW can use a different Interconnect VNI, hence this VNI does not need to be agreed on all the GWs and PEs of the Interconnect network.

The procedures described in section 3.4 will be followed, taking the considerations above for the VNI translation.

5. Conventions and 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].

AC: Attachment Circuit

BUM: it refers to the Broadcast, Unknown unicast and Multicast traffic

DF: Designated Forwarder

GW: Gateway or Data Center Gateway

DCI: Data Center Interconnect

ES: Ethernet Segment

ESI: Ethernet Segment Identifier

I-ESI: Interconnect ESI defined on the GWs for multi-homing to/from the WAN

EVI: EVPN Instance

MAC-VRF: it refers to an EVI instance in a particular node

NVE: Network Virtualization Edge

PW: Pseudowire

RD: Route-Distinguisher

RT: Route-Target

TOR: Top-Of-Rack switch

VNI/VSID: refers to VXLAN/NVGRE virtual identifiers

VSI: Virtual Switch Instance or VPLS instance in a particular PE

6. Security Considerations

This section will be completed in future versions.

7. IANA Considerations

8. References

8.1. Normative References

[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>>.

[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>>.

[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](http://www.rfc-editor.org/info/rfc7041)>.

[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](http://www.rfc-
editor.org/info/rfc7432)>.

[RFC7623] Sajassi et al., "Provider Backbone Bridging Combined with
Ethernet VPN (PBB-EVPN)", RFC 7623, September, 2015, <[http://www.rfc-
editor.org/info/rfc7623](http://www.rfc-
editor.org/info/rfc7623)>.

8.2. Informative References

[EVPN-Overlays] Sajassi-Drake et al., "A Network Virtualization
Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-02.txt,
work in progress, October, 2015

[EVPN-VPLS-INTEGRATION] Sajassi et al., "(PBB-)EVPN Seamless
Integration with (PBB-)VPLS", draft-ietf-bess-evpn-vpls-integration-
00.txt, work in progress, February, 2015

9. Acknowledgments

The authors would like to thank Neil Hart for their valuable comments
and feedback.

10. Contributors

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

Florin Balus
Wen Lin

11. Authors' Addresses

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

Senthil Sathappan
Nokia
Email: senthil.sathappan@nokia.com

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

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

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Ravi Shekhar
Juniper
Email: rshekhar@juniper.net

Anil Lohiya
Juniper
Email: alohiya@juniper.net

Dennis Cai
Cisco Systems
Email: dcai@cisco.com

John Drake
Juniper
Email: jdrake@juniper.net

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan (Ed.)
S. Sathappan
W. Henderickx
Nokia

A. Sajassi
Cisco

J. Drake
Juniper

Expires: September 3, 2018

March 2, 2018

Interconnect Solution for EVPN Overlay networks
draft-ietf-bess-dci-evpn-overlay-10

Abstract

This document describes how Network Virtualization Overlays (NVO) can be connected to a Wide Area Network (WAN) in order to extend the layer-2 connectivity required for some tenants. The solution analyzes the interaction between NVO networks running Ethernet Virtual Private Networks (EVPN) and other L2VPN technologies used in the WAN, such as Virtual Private LAN Services (VPLS), VPLS extensions for Provider Backbone Bridging (PBB-VPLS), EVPN or PBB-EVPN. It also describes how the existing technical specifications apply to the Interconnection and extends the EVPN procedures needed in some cases. In particular, this document describes how EVPN routes are processed on Gateways (GWs) that interconnect EVPN-Overlay and EVPN-MPLS networks, as well as the Interconnect Ethernet Segment (I-ES) to provide multi-homing, and the use of the Unknown MAC route to avoid MAC scale issues on Data Center Network Virtualization Edge (NVE) devices.

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on September 3, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1. Conventions and Terminology	3
2. Introduction	5
3. Decoupled Interconnect solution for EVPN overlay networks	6
3.1. Interconnect requirements	7
3.2. VLAN-based hand-off	8
3.3. PW-based (Pseudowire-based) hand-off	8
3.4. Multi-homing solution on the GWs	9
3.5. Gateway Optimizations	9
3.5.1. MAC Address Advertisement Control	9
3.5.2. ARP/ND flooding control	10
3.5.3. Handling failures between GW and WAN Edge routers	11
4. Integrated Interconnect solution for EVPN overlay networks	11
4.1. Interconnect requirements	12
4.2. VPLS Interconnect for EVPN-Overlay networks	13
4.2.1. Control/Data Plane setup procedures on the GWs	13
4.2.2. Multi-homing procedures on the GWs	14
4.3. PBB-VPLS Interconnect for EVPN-Overlay networks	14

4.3.1. Control/Data Plane setup procedures on the GWs	14
4.3.2. Multi-homing procedures on the GWs	15
4.4. EVPN-MPLS Interconnect for EVPN-Overlay networks	15
4.4.1. Control Plane setup procedures on the GWs	15
4.4.2. Data Plane setup procedures on the GWs	17
4.4.3. Multi-homing procedure extensions on the GWs	18
4.4.4. Impact on MAC Mobility procedures	20
4.4.5. Gateway optimizations	20
4.4.6. Benefits of the EVPN-MPLS Interconnect solution	21
4.5. PBB-EVPN Interconnect for EVPN-Overlay networks	22
4.5.1. Control/Data Plane setup procedures on the GWs	22
4.5.2. Multi-homing procedures on the GWs	22
4.5.3. Impact on MAC Mobility procedures	23
4.5.4. Gateway optimizations	23
4.6. EVPN-VXLAN Interconnect for EVPN-Overlay networks	23
4.6.1. Globally unique VNIs in the Interconnect network	24
4.6.2. Downstream assigned VNIs in the Interconnect network	24
5. Security Considerations	25
6. IANA Considerations	26
7. References	26
7.1. Normative References	26
7.2. Informative References	27
8. Acknowledgments	28
9. Contributors	28
10. Authors' Addresses	29

1. Conventions and Terminology

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

AC: Attachment Circuit.

ARP: Address Resolution Protocol.

BUM: refers to Broadcast, Unknown unicast and Multicast traffic.

CE: Customer Equipment.

CFM: Connectivity Fault Management.

DC and DCI: Data Center and Data Center Interconnect.

DC RR(s) and WAN RR(s): refers to the Data Center and Wide Area Network Route Reflectors, respectively.

DF and NDF: Designated Forwarder and Non-Designated Forwarder.

EVPN: Ethernet Virtual Private Network, as in [RFC7432].

EVI: EVPN Instance.

EVPN Tunnel binding: refers to a tunnel to a remote PE/NVE for a given EVI. Ethernet packets in these bindings are encapsulated with the Overlay or MPLS encapsulation and the EVPN label at the bottom of the stack.

ES and vES: Ethernet Segment and virtual Ethernet Segment.

ESI: Ethernet Segment Identifier.

GW: Gateway or Data Center Gateway.

I-ES and I-ESI: Interconnect Ethernet Segment and Interconnect Ethernet Segment Identifier. An I-ES is defined on the GWs for multi-homing to/from the WAN.

MAC-VRF: refers to an EVI instance in a particular node.

MP2P and LSM tunnels: refer to Multi-Point to Point and Label Switched Multicast tunnels.

ND: Neighbor Discovery protocol.

NVE: Network Virtualization Edge.

NVGRE: Network Virtualization using Generic Routing Encapsulation.

NVO: refers to Network Virtualization Overlays.

OAM: Operations and Maintenance.

PBB: Provider Backbone Bridging.

PE: Provider Edge.

PW: Pseudowire.

RD: Route-Distinguisher.

RT: Route-Target.

S/C-TAG: refers to a combination of Service Tag and Customer Tag in a 802.1Q frame.

TOR: Top-Of-Rack switch.

UMR: Unknown MAC Route.

VNI/VSID: refers to VXLAN/NVGRE virtual identifiers.

VPLS: Virtual Private LAN Service.

VSI: Virtual Switch Instance or VPLS instance in a particular PE.

VXLAN: Virtual eXtensible LAN.

2. Introduction

[EVPN-Overlays] discusses the use of Ethernet Virtual Private Networks (EVPN) [RFC7432] as the control plane for Network Virtualization Overlays (NVO), where VXLAN [RFC7348], NVGRE [RFC7637] or MPLS over GRE [RFC4023] can be used as possible data plane encapsulation options.

While this model provides a scalable and efficient multi-tenant solution within the Data Center, it might not be easily extended to the Wide Area Network (WAN) in some cases due to the requirements and existing deployed technologies. For instance, a Service Provider might have an already deployed Virtual Private LAN Service (VPLS) [RFC4761][RFC4762], VPLS extensions for Provider Backbone Bridging (PBB-VPLS) [RFC7041], EVPN [RFC7432] or PBB-EVPN [RFC7623] network that has to be used to interconnect Data Centers and WAN VPN users. A Gateway (GW) function is required in these cases. In fact, [EVPN-Overlays] discusses two main Data Center Interconnect solution groups: "DCI using GWs" and "DCI using ASBRs". This document specifies the solutions that correspond to the "DCI using GWs" group.

It is assumed that the NVO Gateway (GW) and the WAN Edge functions can be decoupled in two separate systems or integrated into the same system. The former option will be referred as "Decoupled Interconnect solution" throughout the document, whereas the latter one will be referred as "Integrated Interconnect solution".

The specified procedures are local to the redundant GWs connecting a DC to the WAN. The document does not preclude any combination across different DCs for the same tenant. For instance, a "Decoupled" solution can be used in GW1 and GW2 (for DC1) and an "Integrated" solution can be used in GW3 and GW4 (for DC2).

While the Gateways and WAN PEs use existing specifications in some cases, the document also defines extensions that are specific to DCI. In particular, those extensions are:

- o The Interconnect Ethernet Segment (I-ES), an Ethernet Segment that can be associated to a set of PWs or other tunnels. I-ES defined in this document is not associated with a set of Ethernet links, as per [RFC7432], but rather with a set of virtual tunnels (e.g., a set of PWs). This set of virtual tunnels is referred to as vES [VIRTUAL-ES].
- o The use of the Unknown MAC route in a DCI scenario.
- o The processing of EVPN routes on Gateways with MAC-VRFs connecting EVPN-Overlay and EVPN-MPLS networks, or EVPN-Overlay and EVPN-Overlay networks.

3. Decoupled Interconnect solution for EVPN overlay networks

This section describes the interconnect solution when the GW and WAN Edge functions are implemented in different systems. Figure 1 depicts the reference model described in this section. Note that, although not shown in Figure 1, GWs may have local ACs (Attachment Circuits).

to detect individual AC failures.

- o Support for the following optimizations at the GW:
 - + Flooding reduction of unknown unicast traffic sourced from the DC Network Virtualization Edge devices (NVEs).
 - + Control of the WAN MAC addresses advertised to the DC.
 - + Address Resolution Protocol (ARP) and Neighbor Discovery (ND) flooding control for the requests coming from the WAN.

3.2. VLAN-based hand-off

In this option, the hand-off between the GWs and the WAN Edge routers is based on VLANs [802.1Q-2014]. This is illustrated in Figure 1 (between the GWs in NVO-1 and the WAN Edge routers). Each MAC-VRF in the GW is connected to a different VSI/MAC-VRF instance in the WAN Edge router by using a different C-TAG VLAN ID or a different combination of S/C-TAG VLAN IDs that matches at both sides.

This option provides the best possible demarcation between the DC and WAN providers and it does not require control plane interaction between both providers. The disadvantage of this model is the provisioning overhead since the service has to be mapped to a C-TAG or S/C-TAG VLAN ID combination at both GW and WAN Edge routers.

In this model, the GW acts as a regular Network Virtualization Edge (NVE) towards the DC. Its control plane, data plane procedures and interactions are described in [EVPN-Overlays].

The WAN Edge router acts as a (PBB-)VPLS or (PBB-)EVPN PE with attachment circuits (ACs) to the GWs. Its functions are described in [RFC4761], [RFC4762], [RFC6074] or [RFC7432], [RFC7623].

3.3. PW-based (Pseudowire-based) hand-off

If MPLS between the GW and the WAN Edge router is an option, a PW-based Interconnect solution can be deployed. In this option the hand-off between both routers is based on FEC128-based PWs [RFC4762] or FEC129-based PWs (for a greater level of network automation) [RFC6074]. Note that this model still provides a clear demarcation boundary between DC and WAN (since there is a single PW between each MAC-VRF and peer VSI), and security/QoS policies may be applied on a per PW basis. This model provides better scalability than a C-TAG based hand-off and less provisioning overhead than a combined C/S-TAG hand-off. The PW-based hand-off interconnect is illustrated in Figure 1 (between the NVO-2 GWs and the WAN Edge routers).

In this model, besides the usual MPLS procedures between GW and WAN

Edge router [RFC3031], the GW MUST support an interworking function in each MAC-VRF that requires extension to the WAN:

- o If a FEC128-based PW is used between the MAC-VRF (GW) and the VSI (WAN Edge), the corresponding VCID MUST be provisioned on the MAC-VRF and match the VCID used in the peer VSI at the WAN Edge router.
- o If BGP Auto-discovery [RFC6074] and FEC129-based PWs are used between the GW MAC-VRF and the WAN Edge VSI, the provisioning of the VPLS-ID MUST be supported on the MAC-VRF and MUST match the VPLS-ID used in the WAN Edge VSI.

If a PW-based handoff is used, the GW's AC (or point of attachment to the EVPN Instance) uses a combination of a PW label and VLAN IDs. PWs are treated as service interfaces defined in [RFC7432].

3.4. Multi-homing solution on the GWs

EVPN single-active multi-homing, i.e. per-service load-balancing multi-homing is required in this type of interconnect.

The GWs will be provisioned with a unique ES per WAN interconnect, and the hand-off attachment circuits or PWs between the GW and the WAN Edge router will be assigned an ESI for such ES. The ESI will be administratively configured on the GWs according to the procedures in [RFC7432]. This Interconnect ES will be referred as "I-ES" hereafter, and its identifier will be referred as "I-ESI". [RFC7432] describes different ESI Types. The use of Type 0 for the I-ESI is RECOMMENDED in this document.

The solution (on the GWs) MUST follow the single-active multi-homing procedures as described in [EVPN-Overlays] for the provisioned I-ESI, i.e. Ethernet A-D routes per ES and per EVI will be advertised to the DC NVEs for the multi-homing functions, ES routes will be advertised so that ES discovery and Designated Forwarder (DF) procedures can be followed. The MAC addresses learned (in the data plane) on the hand-off links will be advertised with the I-ESI encoded in the ESI field.

3.5. Gateway Optimizations

The following GW features are optional and optimize the control plane and data plane in the DC.

3.5.1. MAC Address Advertisement Control

The use of EVPN in NVO networks brings a significant number of benefits as described in [EVPN-Overlays]. However, if multiple DCs

are interconnected into a single EVI, each DC will have to import all of the MAC addresses from each of the other DCs.

Even if optimized BGP techniques like RT-constraint [RFC4684] are used, the number of MAC addresses to advertise or withdraw (in case of failure) by the GWs of a given DC could overwhelm the NVEs within that DC, particularly when the NVEs reside in the hypervisors.

The solution specified in this document uses the 'Unknown MAC Route' (UMR) which is advertised into a given DC by each of the DC's GWs. This route is defined in [RFC7543] and is a regular EVPN MAC/IP Advertisement route in which the MAC Address Length is set to 48, the MAC address is set to 0, and the ESI field is set to the DC GW's I-ESI.

An NVE within that DC that understands and process the UMR will send unknown unicast frames to one of the DCs GWs, which will then forward that packet to the correct egress PE. Note that, because the ESI is set to the DC GW's I-ESI, all-active multi-homing can be applied to unknown unicast MAC addresses. An NVE that does not understand the Unknown MAC route will handle unknown unicast as described in [RFC7432].

This document proposes that local policy determines whether MAC addresses and/or the UMR are advertised into a given DC. As an example, when all the DC MAC addresses are learned in the control/management plane, it may be appropriate to advertise only the UMR. Advertising all the DC MAC addresses in the control/management plane is usually the case when the NVEs reside in hypervisors. Refer to [EVPN-Overlays] section 7.

It is worth noting that the UMR usage in [RFC7543] and the UMR usage in this document are different. In the former, a Virtual Spoke (V-spoke) does not necessarily learn all the MAC addresses pertaining to hosts in other V-spokes of the same network. The communication between two V-spokes is done through the DMG, until the V-spokes learn each other's MAC addresses. In this document, two leaf switches in the same DC are recommended to learn each other's MAC addresses for the same EVI. The leaf to leaf communication is always direct and does not go through the GW.

3.5.2. ARP/ND flooding control

Another optimization mechanism, naturally provided by EVPN in the GWs, is the Proxy ARP/ND function. The GWs should build a Proxy ARP/ND cache table as per [RFC7432]. When the active GW receives an ARP/ND request/solicitation coming from the WAN, the GW does a Proxy

ARP/ND table lookup and replies as long as the information is available in its table.

This mechanism is especially recommended on the GWs, since it protects the DC network from external ARP/ND-flooding storms.

3.5.3. Handling failures between GW and WAN Edge routers

Link/PE failures are handled on the GWs as specified in [RFC7432]. The GW detecting the failure will withdraw the EVPN routes as per [RFC7432].

Individual AC/PW failures may be detected by OAM mechanisms. For instance:

- o If the Interconnect solution is based on a VLAN hand-off, Ethernet-CFM [802.1AG][Y.1731] may be used to detect individual AC failures on both, the GW and WAN Edge router. An individual AC failure will trigger the withdrawal of the corresponding A-D per EVI route as well as the MACs learned on that AC.
- o If the Interconnect solution is based on a PW hand-off, the Label Distribution Protocol (LDP) PW Status bits TLV [RFC6870] may be used to detect individual PW failures on both, the GW and WAN Edge router.

4. Integrated Interconnect solution for EVPN overlay networks

When the DC and the WAN are operated by the same administrative entity, the Service Provider can decide to integrate the GW and WAN Edge PE functions in the same router for obvious CAPEX and OPEX saving reasons. This is illustrated in Figure 2. Note that this model does not provide an explicit demarcation link between DC and WAN anymore. Although not shown in Figure 2, note that the GWs may have local ACs.

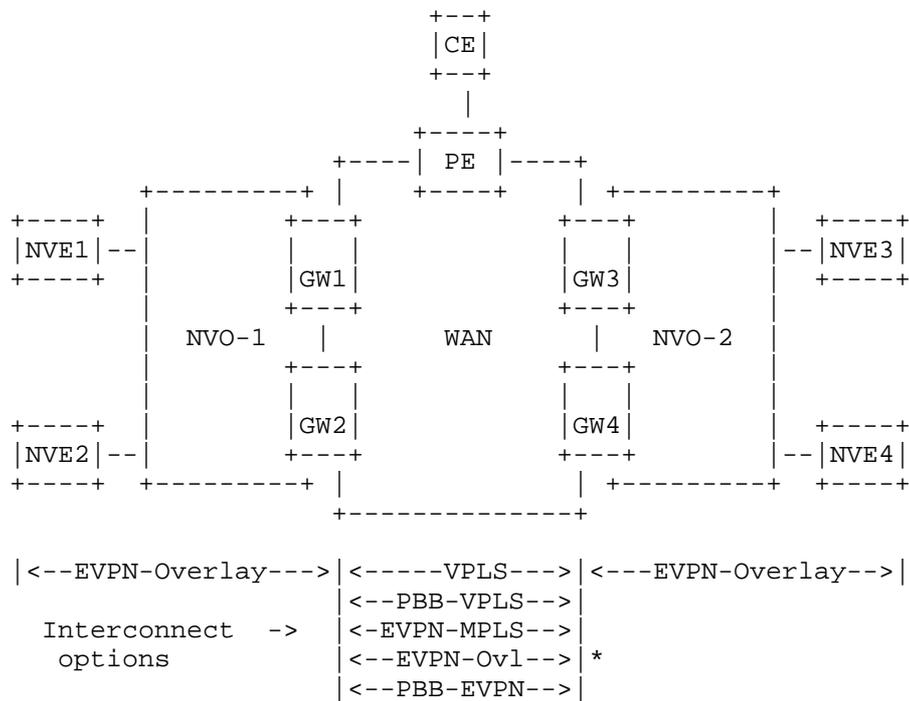


Figure 2 Integrated Interconnect model

* EVPN-Ovl stands for EVPN-Overlay (and it's an Interconnect option).

4.1. Interconnect requirements

The Integrated Interconnect solution meets the following requirements:

- o Control plane and data plane interworking between the EVPN-overlay network and the L2VPN technology supported in the WAN, irrespective of the technology choice, i.e. (PBB-)VPLS or (PBB-)EVPN, as depicted in Figure 2.
- o Multi-homing, including single-active multi-homing with per-service load balancing or all-active multi-homing, i.e. per-flow load-balancing, as long as the technology deployed in the WAN supports it.
- o Support for end-to-end MAC Mobility, Static MAC protection and other procedures (e.g. proxy-arp) described in [RFC7432] as long as EVPN-MPLS is the technology of choice in the WAN.

- o Independent inclusive multicast trees in the WAN and in the DC. That is, the inclusive multicast tree type defined in the WAN does not need to be the same as in the DC.

4.2. VPLS Interconnect for EVPN-Overlay networks

4.2.1. Control/Data Plane setup procedures on the GWs

Regular MPLS tunnels and TLDP/BGP sessions will be setup to the WAN PEs and RRs as per [RFC4761], [RFC4762], [RFC6074] and overlay tunnels and EVPN will be setup as per [EVPN-Overlays]. Note that different route-targets for the DC and for the WAN are normally required (unless [RFC4762] is used in the WAN, in which case no WAN route-target is needed). A single type-1 RD per service may be used.

In order to support multi-homing, the GWs will be provisioned with an I-ESI (see section 3.4), that will be unique per interconnection. The I-ES in this case will represent the group of PWs to the WAN PEs and GWs. All the [RFC7432] procedures are still followed for the I-ES, e.g. any MAC address learned from the WAN will be advertised to the DC with the I-ESI in the ESI field.

A MAC-VRF per EVI will be created in each GW. The MAC-VRF will have two different types of tunnel bindings instantiated in two different split-horizon-groups:

- o VPLS PWs will be instantiated in the "WAN split-horizon-group".
- o Overlay tunnel bindings (e.g. VXLAN, NVGRE) will be instantiated in the "DC split-horizon-group".

Attachment circuits are also supported on the same MAC-VRF (although not shown in Figure 2), but they will not be part of any of the above split-horizon-groups.

Traffic received in a given split-horizon-group will never be forwarded to a member of the same split-horizon-group.

As far as BUM flooding is concerned, a flooding list will be composed of the sub-list created by the inclusive multicast routes and the sub-list created for VPLS in the WAN. BUM frames received from a local Attachment Circuit (AC) will be forwarded to the flooding list. BUM frames received from the DC or the WAN will be forwarded to the flooding list observing the split-horizon-group rule described above.

Note that the GWs are not allowed to have an EVPN binding and a PW to the same far-end within the same MAC-VRF, so that loops and packet duplication are avoided. In case a GW can successfully establish

both, an EVPN binding and a PW to the same far-end PE, the EVPN binding will prevail and the PW will be brought operationally down.

The optimizations procedures described in section 3.5 can also be applied to this model.

4.2.2. Multi-homing procedures on the GWs

This model supports single-active multi-homing on the GWs. All-active multi-homing is not supported by VPLS, therefore it cannot be used on the GWs.

In this case, for a given EVI, all the PWs in the WAN split-horizon-group are assigned to I-ES. All the single-active multi-homing procedures as described by [EVPN-Overlays] will be followed for the I-ES.

The non-DF GW for the I-ES will block the transmission and reception of all the PWs in the "WAN split-horizon-group" for BUM and unicast traffic.

4.3. PBB-VPLS Interconnect for EVPN-Overlay networks

4.3.1. Control/Data Plane setup procedures on the GWs

In this case, there is no impact on the procedures described in [RFC7041] for the B-component. However the I-component instances become EVI instances with EVPN-Overlay bindings and potentially local attachment circuits. A number of MAC-VRF instances can be multiplexed into the same B-component instance. This option provides significant savings in terms of PWs to be maintained in the WAN.

The I-ESI concept described in section 4.2.1 will also be used for the PBB-VPLS-based Interconnect.

B-component PWs and I-component EVPN-overlay bindings established to the same far-end will be compared. The following rules will be observed:

- o Attempts to setup a PW between the two GWs within the B-component context will never be blocked.
- o If a PW exists between two GWs for the B-component and an attempt is made to setup an EVPN binding on an I-component linked to that B-component, the EVPN binding will be kept operationally down. Note that the BGP EVPN routes will still be valid but not used.

- o The EVPN binding will only be up and used as long as there is no PW to the same far-end in the corresponding B-component. The EVPN bindings in the I-components will be brought down before the PW in the B-component is brought up.

The optimizations procedures described in section 3.5 can also be applied to this Interconnect option.

4.3.2. Multi-homing procedures on the GWs

This model supports single-active multi-homing on the GWs. All-active multi-homing is not supported by this scenario.

The single-active multi-homing procedures as described by [EVPN-Overlays] will be followed for the I-ES for each EVI instance connected to the B-component. Note that in this case, for a given EVI, all the EVPN bindings in the I-component are assigned to the I-ES. The non-DF GW for the I-ES will block the transmission and reception of all the I-component EVPN bindings for BUM and unicast traffic. When learning MACs from the WAN, the non-DF MUST NOT advertise EVPN MAC/IP routes for those MACs.

4.4. EVPN-MPLS Interconnect for EVPN-Overlay networks

If EVPN for MPLS tunnels, EVPN-MPLS hereafter, is supported in the WAN, an end-to-end EVPN solution can be deployed. The following sections describe the proposed solution as well as the impact required on the [RFC7432] procedures.

4.4.1. Control Plane setup procedures on the GWs

The GWs MUST establish separate BGP sessions for sending/receiving EVPN routes to/from the DC and to/from the WAN. Normally each GW will setup one BGP EVPN session to the DC RR (or two BGP EVPN sessions if there are redundant DC RRs) and one session to the WAN RR (or two sessions if there are redundant WAN RRs).

In order to facilitate separate BGP processes for DC and WAN, EVPN routes sent to the WAN SHOULD carry a different route-distinguisher (RD) than the EVPN routes sent to the DC. In addition, although reusing the same value is possible, different route-targets are expected to be handled for the same EVI in the WAN and the DC. Note that the EVPN service routes sent to the DC RRs will normally include a [TUNNEL-ENCAP] BGP encapsulation extended community with a different tunnel type than the one sent to the WAN RRs.

As in the other discussed options, an I-ES and its assigned I-ESI

will be configured on the GWs for multi-homing. This I-ES represents the WAN EVPN-MPLS PEs to the DC but also the DC EVPN-Overlay NVEs to the WAN. Optionally, different I-ESI values are configured for representing the WAN and the DC. If different EVPN-Overlay networks are connected to the same group of GWs, each EVPN-Overlay network MUST get assigned a different I-ESI.

Received EVPN routes will never be reflected on the GWs but consumed and re-advertised (if needed):

- o Ethernet A-D routes, ES routes and Inclusive Multicast routes are consumed by the GWs and processed locally for the corresponding [RFC7432] procedures.
- o MAC/IP advertisement routes will be received, imported and if they become active in the MAC-VRF, the information will be re-advertised as new routes with the following fields:
 - + The RD will be the GW's RD for the MAC-VRF.
 - + The ESI will be set to the I-ESI.
 - + The Ethernet-tag value will be kept from the received NLRI.
 - + The MAC length, MAC address, IP Length and IP address values will be kept from the received NLRI.
 - + The MPLS label will be a local 20-bit value (when sent to the WAN) or a DC-global 24-bit value (when sent to the DC for encapsulations using a VNI).
 - + The appropriate Route-Targets (RTs) and [TUNNEL-ENCAP] BGP Encapsulation extended community will be used according to [EVPN-Overlays].

The GWs will also generate the following local EVPN routes that will be sent to the DC and WAN, with their corresponding RTs and [TUNNEL-ENCAP] BGP Encapsulation extended community values:

- o ES route(s) for the I-ESI(s).
- o Ethernet A-D routes per ES and EVI for the I-ESI(s). The A-D per-EVI routes sent to the WAN and the DC will have consistent Ethernet-Tag values.
- o Inclusive Multicast routes with independent tunnel type value for the WAN and DC. E.g. a P2MP LSP may be used in the WAN whereas ingress replication may be used in the DC. The routes

sent to the WAN and the DC will have a consistent Ethernet-Tag.

- o MAC/IP advertisement routes for MAC addresses learned in local attachment circuits. Note that these routes will not include the I-ESI, but ESI=0 or different from 0 for local multi-homed Ethernet Segments (ES). The routes sent to the WAN and the DC will have a consistent Ethernet-Tag.

Assuming GW1 and GW2 are peer GWs of the same DC, each GW will generate two sets of the above local service routes: Set-DC will be sent to the DC RRs and will include A-D per EVI, Inclusive Multicast and MAC/IP routes for the DC encapsulation and RT. Set-WAN will be sent to the WAN RRs and will include the same routes but using the WAN RT and encapsulation. GW1 and GW2 will receive each other's set-DC and set-WAN. This is the expected behavior on GW1 and GW2 for locally generated routes:

- o Inclusive multicast routes: when setting up the flooding lists for a given MAC-VRF, each GW will include its DC peer GW only in the EVPN-MPLS flooding list (by default) and not the EVPN-Overlay flooding list. That is, GW2 will import two Inclusive Multicast routes from GW1 (from set-DC and set-WAN) but will only consider one of the two, having the set-WAN route higher priority. An administrative option MAY change this preference so that the set-DC route is selected first.
- o MAC/IP advertisement routes for local attachment circuits: as above, the GW will select only one, having the route from the set-WAN a higher priority. As with the Inclusive multicast routes, an administrative option MAY change this priority.

4.4.2. Data Plane setup procedures on the GWs

The procedure explained at the end of the previous section will make sure there are no loops or packet duplication between the GWs of the same EVPN-Overlay network (for frames generated from local ACs) since only one EVPN binding per EVI (or per Ethernet Tag in case of VLAN-aware bundle services) will be setup in the data plane between the two nodes. That binding will by default be added to the EVPN-MPLS flooding list.

As for the rest of the EVPN tunnel bindings, they will be added to one of the two flooding lists that each GW sets up for the same MAC-VRF:

- o EVPN-overlay flooding list (composed of bindings to the remote NVEs or multicast tunnel to the NVEs).

- o EVPN-MPLS flooding list (composed of MP2P or LSM tunnel to the remote PEs)

Each flooding list will be part of a separate split-horizon-group: the WAN split-horizon-group or the DC split-horizon-group. Traffic generated from a local AC can be flooded to both split-horizon-groups. Traffic from a binding of a split-horizon-group can be flooded to the other split-horizon-group and local ACs, but never to a member of its own split-horizon-group.

When either GW1 or GW2 receive a BUM frame on an MPLS tunnel including an ESI label at the bottom of the stack, they will perform an ESI label lookup and split-horizon filtering as per [RFC7432] in case the ESI label identifies a local ESI (I-ESI or any other non-zero ESI).

4.4.3. Multi-homing procedure extensions on the GWs

This model supports single-active as well as all-active multi-homing.

All the [RFC7432] multi-homing procedures for the DF election on I-ES(s) as well as the backup-path (single-active) and aliasing (all-active) procedures will be followed on the GWs. Remote PEs in the EVPN-MPLS network will follow regular [RFC7432] aliasing or backup-path procedures for MAC/IP routes received from the GWs for the same I-ESI. So will NVEs in the EVPN-Overlay network for MAC/IP routes received with the same I-ESI.

As far as the forwarding plane is concerned, by default, the EVPN-Overlay network will have an analogous behavior to the access ACs in [RFC7432] multi-homed Ethernet Segments.

The forwarding behavior on the GWs is described below:

- o Single-active multi-homing; assuming a WAN split-horizon-group (comprised of EVPN-MPLS bindings), a DC split-horizon-group (comprised of EVPN-Overlay bindings) and local ACs on the GWs:
 - + Forwarding behavior on the non-DF: the non-DF MUST block ingress and egress forwarding on the EVPN-Overlay bindings associated to the I-ES. The EVPN-MPLS network is considered to be the core network and the EVPN-MPLS bindings to the remote PEs and GWs will be active.
 - + Forwarding behavior on the DF: the DF MUST NOT forward BUM or unicast traffic received from a given split-horizon-group to a member of his own split-horizon group. Forwarding to other

split-horizon-groups and local ACs is allowed (as long as the ACs are not part of an ES for which the node is non-DF). As per [RFC7432] and for split-horizon purposes, when receiving BUM traffic on the EVPN-Overlay bindings associated to an I-ES, the DF GW SHOULD add the I-ESI label when forwarding to the peer GW over EVPN-MPLS.

- + When receiving EVPN MAC/IP routes from the WAN, the non-DF MUST NOT re-originate the EVPN routes and advertise them to the DC peers. In the same way, EVPN MAC/IP routes received from the DC MUST NOT be advertised to the WAN peers. This is consistent with [RFC7432] and allows the remote PE/NVEs know who the primary GW is, based on the reception of the MAC/IP routes.
- o All-active multi-homing; assuming a WAN split-horizon-group (comprised of EVPN-MPLS bindings), a DC split-horizon-group (comprised of EVPN-Overlay bindings) and local ACs on the GWs:
 - + Forwarding behavior on the non-DF: the non-DF follows the same behavior as the non-DF in the single-active case but only for BUM traffic. Unicast traffic received from a split-horizon-group MUST NOT be forwarded to a member of its own split-horizon-group but can be forwarded normally to the other split-horizon-groups and local ACs. If a known unicast packet is identified as a "flooded" packet, the procedures for BUM traffic MUST be followed.
 - + Forwarding behavior on the DF: the DF follows the same behavior as the DF in the single-active case but only for BUM traffic. Unicast traffic received from a split-horizon-group MUST NOT be forwarded to a member of its own split-horizon-group but can be forwarded normally to the other split-horizon-group and local ACs. If a known unicast packet is identified as a "flooded" packet, the procedures for BUM traffic MUST be followed. As per [RFC7432] and for split-horizon purposes, when receiving BUM traffic on the EVPN-Overlay bindings associated to an I-ES, the DF GW MUST add the I-ESI label when forwarding to the peer GW over EVPN-MPLS.
 - + Contrary to the single-active multi-homing case, both DF and non-DF re-originate and advertise MAC/IP routes received from the WAN/DC peers, adding the corresponding I-ESI so that the remote PE/NVEs can perform regular aliasing as per [RFC7432].

The example in Figure 3 illustrates the forwarding of BUM traffic originated from an NVE on a pair of all-active multi-homing GWs.

section 3.5.1, solves some transient packet duplication issues in cases of all-active multi-homing, as explained below.

Consider the diagram in Figure 2 for EVPN-MPLS Interconnect and all-active multi-homing, and the following sequence:

- a) MAC Address M1 is advertised from NVE3 in EVI-1.
- b) GW3 and GW4 learn M1 for EVI-1 and re-advertise M1 to the WAN with I-ESI-2 in the ESI field.
- c) GW1 and GW2 learn M1 and install GW3/GW4 as next-hops following the EVPN aliasing procedures.
- d) Before NVE1 learns M1, a packet arrives at NVE1 with destination M1. If the Unknown MAC Route had not been advertised into the DC, NVE1 would have flooded the packet throughout the DC, in particular to both GW1 and GW2. If the same VNI/VSID is used for both known unicast and BUM traffic, as is typically the case, there is no indication in the packet that it is a BUM packet and both GW1 and GW2 would have forwarded it, creating packet duplication. However, because the Unknown MAC Route had been advertised into the DC, NVE1 will unicast the packet to either GW1 or GW2.
- e) Since both GW1 and GW2 know M1, the GW receiving the packet will forward it to either GW3 or GW4.

4.4.6. Benefits of the EVPN-MPLS Interconnect solution

The [EVPN-Overlays] "DCI using ASBRs" solution and the GW solution with EVPN-MPLS Interconnect may be seen similar since they both retain the EVPN attributes between Data Centers and throughout the WAN. However the EVPN-MPLS Interconnect solution on the GWs has significant benefits compared to the "DCI using ASBRs" solution:

- o As in any of the described GW models, this solution supports the connectivity of local attachment circuits on the GWs. This is not possible in a "DCI using ASBRs" solution.
- o Different data plane encapsulations can be supported in the DC and the WAN, while a uniform encapsulation is needed in the "DCI using ASBRs" solution.
- o Optimized multicast solution, with independent inclusive multicast trees in DC and WAN.

- o MPLS Label aggregation: for the case where MPLS labels are signaled from the NVEs for MAC/IP Advertisement routes, this solution provides label aggregation. A remote PE MAY receive a single label per GW MAC-VRF as opposed to a label per NVE/MAC-VRF connected to the GW MAC-VRF. For instance, in Figure 2, PE would receive only one label for all the routes advertised for a given MAC-VRF from GW1, as opposed to a label per NVE/MAC-VRF.
- o The GW will not propagate MAC mobility for the MACs moving within a DC. Mobility intra-DC is solved by all the NVEs in the DC. The MAC Mobility procedures on the GWs are only required in case of mobility across DCs.
- o Proxy-ARP/ND function on the DC GWs can be leveraged to reduce ARP/ND flooding in the DC or/and in the WAN.

4.5. PBB-EVPN Interconnect for EVPN-Overlay networks

PBB-EVPN [RFC7623] is yet another Interconnect option. It requires the use of GWs where I-components and associated B-components are part of EVI instances.

4.5.1. Control/Data Plane setup procedures on the GWs

EVPN will run independently in both components, the I-component MAC-VRF and B-component MAC-VRF. Compared to [RFC7623], the DC C-MACs are no longer learned in the data plane on the GW but in the control plane through EVPN running on the I-component. Remote C-MACs coming from remote PEs are still learned in the data plane. B-MACs in the B-component will be assigned and advertised following the procedures described in [RFC7623].

An I-ES will be configured on the GWs for multi-homing, but its I-ESI will only be used in the EVPN control plane for the I-component EVI. No non-reserved ESIs will be used in the control plane of the B-component EVI as per [RFC7623], that is, the I-ES will be represented to the WAN PBB-EVPN PEs using shared or dedicated B-MACs.

The rest of the control plane procedures will follow [RFC7432] for the I-component EVI and [RFC7623] for the B-component EVI.

From the data plane perspective, the I-component and B-component EVPN bindings established to the same far-end will be compared and the I-component EVPN-overlay binding will be kept down following the rules described in section 4.3.1.

4.5.2. Multi-homing procedures on the GWs

This model supports single-active as well as all-active multi-homing.

The forwarding behavior of the DF and non-DF will be changed based on the description outlined in section 4.4.3, only replacing the "WAN split-horizon-group" for the B-component, and using [RFC7623] procedures for the traffic sent or received on the B-component.

4.5.3. Impact on MAC Mobility procedures

C-MACs learned from the B-component will be advertised in EVPN within the I-component EVI scope. If the C-MAC was previously known in the I-component database, EVPN would advertise the C-MAC with a higher sequence number, as per [RFC7432]. From a Mobility perspective and the related procedures described in [RFC7432], the C-MACs learned from the B-component are considered local.

4.5.4. Gateway optimizations

All the considerations explained in section 4.4.5 are applicable to the PBB-EVPN Interconnect option.

4.6. EVPN-VXLAN Interconnect for EVPN-Overlay networks

If EVPN for Overlay tunnels is supported in the WAN and a GW function is required, an end-to-end EVPN solution can be deployed. While multiple Overlay tunnel combinations at the WAN and the DC are possible (MPLSoGRE, nvGRE, etc.), VXLAN is described here, given its popularity in the industry. This section focuses on the specific case of EVPN for VXLAN (EVPN-VXLAN hereafter) and the impact on the [RFC7432] procedures.

The procedures described in section 4.4 apply to this section too, only replacing EVPN-MPLS for EVPN-VXLAN control plane specifics and using [EVPN-Overlays] "Local Bias" procedures instead of section 4.4.3. Since there are no ESI-labels in VXLAN, GWs need to rely on "Local Bias" to apply split-horizon on packets generated from the I-ES and sent to the peer GW.

This use-case assumes that NVEs need to use the VNIs or VSIDs as a globally unique identifiers within a data center, and a Gateway needs to be employed at the edge of the data center network to translate the VNI or VSID when crossing the network boundaries. This GW function provides VNI and tunnel IP address translation. The use-case in which local downstream assigned VNIs or VSIDs can be used (like MPLS labels) is described by [EVPN-Overlays].

While VNIs are globally significant within each DC, there are two possibilities in the Interconnect network:

- a) Globally unique VNIs in the Interconnect network:
In this case, the GWs and PEs in the Interconnect network will agree on a common VNI for a given EVI. The RT to be used in the Interconnect network can be auto-derived from the agreed Interconnect VNI. The VNI used inside each DC MAY be the same as the Interconnect VNI.
- b) Downstream assigned VNIs in the Interconnect network.
In this case, the GWs and PEs MUST use the proper RTs to import/export the EVPN routes. Note that even if the VNI is downstream assigned in the Interconnect network, and unlike option (a), it only identifies the <Ethernet Tag, GW> pair and not the <Ethernet Tag, egress PE> pair. The VNI used inside each DC MAY be the same as the Interconnect VNI. GWs SHOULD support multiple VNI spaces per EVI (one per Interconnect network they are connected to).

In both options, NVEs inside a DC only have to be aware of a single VNI space, and only GWs will handle the complexity of managing multiple VNI spaces. In addition to VNI translation above, the GWs will provide translation of the tunnel source IP for the packets generated from the NVEs, using their own IP address. GWs will use that IP address as the BGP next-hop in all the EVPN updates to the Interconnect network.

The following sections provide more details about these two options.

4.6.1. Globally unique VNIs in the Interconnect network

Considering Figure 2, if a host H1 in NVO-1 needs to communicate with a host H2 in NVO-2, and assuming that different VNIs are used in each DC for the same EVI, e.g. VNI-10 in NVO-1 and VNI-20 in NVO-2, then the VNIs MUST be translated to a common Interconnect VNI (e.g. VNI-100) on the GWs. Each GW is provisioned with a VNI translation mapping so that it can translate the VNI in the control plane when sending BGP EVPN route updates to the Interconnect network. In other words, GW1 and GW2 MUST be configured to map VNI-10 to VNI-100 in the BGP update messages for H1's MAC route. This mapping is also used to translate the VNI in the data plane in both directions, that is, VNI-10 to VNI-100 when the packet is received from NVO-1 and the reverse mapping from VNI-100 to VNI-10 when the packet is received from the remote NVO-2 network and needs to be forwarded to NVO-1.

The procedures described in section 4.4 will be followed, considering that the VNIs advertised/received by the GWs will be translated accordingly.

4.6.2. Downstream assigned VNIs in the Interconnect network

In this case, if a host H1 in NVO-1 needs to communicate with a host H2 in NVO-2, and assuming that different VNIs are used in each DC for the same EVI, e.g. VNI-10 in NVO-1 and VNI-20 in NVO-2, then the VNIs MUST be translated as in section 4.6.1. However, in this case, there is no need to translate to a common Interconnect VNI on the GWs. Each GW can translate the VNI received in an EVPN update to a locally assigned VNI advertised to the Interconnect network. Each GW can use a different Interconnect VNI, hence this VNI does not need to be agreed on all the GWs and PEs of the Interconnect network.

The procedures described in section 4.4 will be followed, taking the considerations above for the VNI translation.

5. Security Considerations

This document applies existing specifications to a number of Interconnect models. The Security Considerations included in those documents, such as [RFC7432], [EVPN-Overlays], [RFC7623], [RFC4761] and [RFC4762] apply to this document whenever those technologies are used.

As discussed, [EVPN-Overlays] discusses two main DCI solution groups: "DCI using GWs" and "DCI using ASBRs". This document specifies the solutions that correspond to the "DCI using GWs" group. It is important to note that the use of GWs provide a superior level of security on a per tenant basis, compared to the use of ASBRs. This is due to the fact that GWs need to perform a MAC lookup on the frames being received from the WAN, and they apply security procedures, such as filtering of undesired frames, filtering of frames with a source MAC that matches a protected MAC in the DC or application of MAC duplication procedures defined in [RFC7432]. On ASBRs though, traffic is forwarded based on a label or VNI swap and there is usually no visibility of the encapsulated frames, which can carry malicious traffic.

In addition, the GW optimizations specified in this document, provide additional protection of the DC Tenant Systems. For instance, the MAC address advertisement control and Unknown MAC Route defined in section 3.5.1 protect the DC NVEs from being overwhelmed with an excessive number MAC/IP routes being learned on the GWs from the WAN. The ARP/ND flooding control described in 3.5.2 can reduce/suppress broadcast storms being injected from the WAN.

Finally, the reader should be aware of the potential security implications of designing a DCI with the Decoupled Interconnect solution (section 3) or the Integrated Interconnect solution (section 4). In the Decoupled Interconnect solution the DC is typically easier

to protect from the WAN, since each GW has a single logical link to one WAN PE, whereas in the Integrated solution, the GW has logical links to all the WAN PEs that are attached to the tenant. In either model, proper control plane and data plane policies should be put in place in the GWs in order to protect the DC from potential attacks coming from the WAN.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

[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>>.

[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>>.

[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>>.

[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>>.

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

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC

2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<http://www.rfc-editor.org/info/rfc8174>>.

[TUNNEL-ENCAP] Rosen et al., "The BGP Tunnel Encapsulation Attribute", draft-ietf-idr-tunnel-encaps-08, work in progress, January 11, 2018.

[RFC7623] Sajassi et al., "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, September, 2015, <<http://www.rfc-editor.org/info/rfc7623>>.

[EVPN-Overlays] Sajassi-Drake et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-11.txt, work in progress, January 2018.

[RFC7543] Jeng, H., Jalil, L., Bonica, R., Patel, K., and L. Yong, "Covering Prefixes Outbound Route Filter for BGP-4", RFC 7543, DOI 10.17487/RFC7543, May 2015, <<https://www.rfc-editor.org/info/rfc7543>>.

7.2. Informative 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>>.

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

[RFC7637] Garg, P., et al., "NVGRE: Network Virtualization using Generic Routing Encapsulation", RFC 7637, September, 2015

[RFC4023] Worster, T., Rekhter, Y., and E. Rosen, Ed., "Encapsulating MPLS in IP or Generic Routing Encapsulation (GRE)", RFC 4023, DOI 10.17487/RFC4023, March 2005, <<http://www.rfc-editor.org/info/rfc4023>>.

[Y.1731] ITU-T Recommendation Y.1731, "OAM functions and mechanisms for Ethernet based networks", July 2011.

[802.1AG] IEEE 802.1AG_2007, "IEEE Standard for Local and Metropolitan Area Networks - Virtual Bridged Local Area Networks Amendment 5: Connectivity Fault Management", January 2008.

[802.1Q-2014] IEEE 802.1Q-2014, "IEEE Standard for Local and metropolitan area networks--Bridges and Bridged Networks", December 2014.

[RFC6870] Muley, P., Ed., and M. Aissaoui, Ed., "Pseudowire Preferential Forwarding Status Bit", RFC 6870, DOI 10.17487/RFC6870, February 2013, <<http://www.rfc-editor.org/info/rfc6870>>.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<http://www.rfc-editor.org/info/rfc3031>>.

[VIRTUAL-ES] Sajassi et al., "EVPN Virtual Ethernet Segment", draft-sajassi-bess-evpn-virtual-eth-segment-03, work in progress, February 2018.

8. Acknowledgments

The authors would like to thank Neil Hart, Vinod Prabhu and Kiran Nagaraj for their valuable comments and feedback. We would also like to thank Martin Vigoureux and Alvaro Retana for his detailed review and comments.

9. Contributors

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

Ravi Shekhar
Anil Lohiya
Wen Lin
Juniper Networks

Florin Balus
Patrice Brissette
Cisco

Senad Palislamovic
Nokia

Dennis Cai
Alibaba

10. Authors' Addresses

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

Senthil Sathappan
Nokia
Email: senthil.sathappan@nokia.com

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

Ali Sajassi
Cisco
Email: sajassi@cisco.com

John Drake
Juniper
Email: jdrake@juniper.net

BESS Working Group

Internet Draft

Intended Status: Proposed Standard

Expires: January 9, 2017

P. Brissette
A. Sajassi
Cisco System
H. Shah
Ciena Corporation
Z. Li
Huawei Technologies
I. Chen
Ericsson
K. Tiruveedhula
Juniper Networks
I. Hussain
Infinera Corporation
J. Rabadan
Nokia

July 8, 2016

Yang Data Model for EVPN
draft-ietf-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. Any "add-on" features such as EVPN IRB, EVPN overlay, etc. are for future investigation. This document mainly focuses on EVPN and Ethernet-Segment instance framework.

Status of this Memo

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

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

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

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

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

Copyright and License Notice

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

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

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	4
2. Specification of Requirements	5
3. EVPN YANG Model	5
3.1. Overview	5
3.2. Ethernet-Segment Model	6
3.3. EVPN Model	6
4. YANG Module	7
4.1. Ethernet Segment Yang Module	7
4.2. EVPN Yang Module	9
5. Security Considerations	11
6. IANA Considerations	11
7. Acknowledgments	11
8. References	12
8.1. Normative References	12
8.2. Informative References	12
Authors' Addresses	12

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 EVPN-based services. The definition work is undertaken initially by a smaller working group with members representing various vendors and service providers. The EVPN basic framework consist of two modules: EVPN and Ethernet-Segment. These models are completely orthogonal. They usually work in pair but user can definitely use one or the other for its own need.

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

- o Configuration
- o Operational State
- o Executables (Actions)
- o 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 a specific interface. That 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 where no EVPN instance is required.

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:

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

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

Following documents will be covered at that time:

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

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

3.2 Ethernet-Segment Model

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

```

module: ietf-ethernet-segment
  +--rw ethernet-segments
  |   +--rw ethernet-segment* [name]
  |   |   +--rw name string
  |   |   +--rw (ac-or-pw)?
  |   |   |   +---:(ac)
  |   |   |   |   +--rw ac? string
  |   |   |   +---:(pw)
  |   |   |   |   +--rw pw? string
  |   |   +--rw ethernet-segment-identifier? uint32
  |   |   +--rw (active-mode)
  |   |   |   +---:(single-active)
  |   |   |   |   +--rw single-active-mode? empty
  |   |   |   +---:(all-active)
  |   |   |   |   +--rw all-active-mode? empty
  |   |   +--rw pbb-parameters {ethernet-segment-pbb-params}?
  |   |   |   +--rw backbone-src-mac? yang:mac-address
  |   |   +--rw bgp-parameters
  |   |   |   +--rw common
  |   |   |   |   +--rw rd-rt* [route-distinguisher]
  |   |   |   |   |   +--rw route-distinguisher string
  |   |   |   |   |   +--rw vpn-target* [rt-value]
  |   |   |   |   |   |   +--rw rt-value string
  |   |   |   |   |   |   +--rw rt-type bgp-rt-type
  |   |   +--rw df-election
  |   |   |   +--rw (df-election-method)?
  |   |   |   |   +---:(highest-random-weight)
  |   |   |   |   |   +--rw hrw? boolean
  |   |   +--rw election-wait-time? uint32
  |   +--rw ead-evi-route? boolean

```

```

+--ro ethernet-segments-state
  +--ro ethernet-segment-state* [name]
    +--ro name string
    +--ro service-type? string
    +--ro status? status-type
    +--ro (ac-or-pw)?
      | +--:(ac)
      | | +--ro ac? string
      | +--:(pw)
      | | +--ro pw? string
    +--ro interface-status? status-type
    +--ro ethernet-segment-identifier? uint32
    +--ro active-mode? string
    +--ro pbb-parameters {ethernet-segment-pbb-params}?
      | +--ro backbone-src-mac? yang:mac-address
    +--ro bgp-parameters
      | +--ro common
      | | +--ro rd-rt* [route-distinguisher]
      | | | +--ro route-distinguisher string
      | | | +--ro vpn-target* [rt-value]
      | | | | +--ro rt-value string
      | | | | +--ro rt-type bgp-rt-type
    +--ro df-election
      | +--ro hrw-enabled? boolean
      | +--ro election-wait-time? uint32
    +--ro ead-evi-route-enabled? boolean
    +--ro esi-label? string
    +--ro member*
      | +--ro ip-address? inet:ip-address
    +--ro df*
      +--ro service-identifier? uint32
      +--ro vlan? uint32
      +--ro ip-address? inet:ip-address

```

3.3 EVPN Model

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

```

module: ietf-evpn
  +--rw evpn
    +--rw common
      | +--rw (replication-type)?
      | | +--:(ingress-replication)
      | | | +--rw ingress-replication? boolean

```

```

|      +---:(p2mp-replication)
|          +---rw p2mp-replication?          boolean
+---rw evpn-instances
|   +---rw evpn-instance* [name]
|       +---rw name                          string
|       +---rw evi?                          uint32
|       +---rw pbb-parameters {evpn-pbb-params}?
|           | +---rw source-bmac?  yang:hex-string
+---rw bgp-parameters
|   +---rw common
|       +---rw rd-rt* [route-distinguisher]
|           +---rw route-distinguisher  string
|           +---rw vpn-target* [rt-value]
|               +---rw rt-value         string
|               +---rw rt-type          bgp-rt-type
+---rw arp-proxy?                          boolean
+---rw arp-suppression?                    boolean
+---rw nd-proxy?                          boolean
+---rw nd-suppression?                    boolean
+---rw underlay-multicast?                boolean
+---rw flood-unknown-unicast-supression?  boolean
+---ro evpn-instances-state
|   +---ro evpn-instance-state*
|       +---ro name?                        string
|       +---ro evi?                        uint32
|       +---ro pbb-parameters
|           | +---ro source-bmac?  yang:hex-string
+---ro bgp-parameters
|   +---ro common
|       +---ro rd-rt* [route-distinguisher]
|           +---ro route-distinguisher  string
|           +---ro vpn-target* [rt-value]
|               +---ro rt-value         string
|               +---ro rt-type          bgp-rt-type
+---ro advertise-mac-suppression-enabled?  boolean
+---ro arp-proxy-enabled?                 boolean
+---ro arp-suppression-enabled?          boolean
+---ro nd-proxy-enabled?                 boolean
+---ro nd-suppression-enabled?          boolean
+---ro underlay-multicast-enabled?       boolean
+---ro flood-unknown-unicast-suppression-enabled?  boolean
+---ro routes
|   +---ro ethernet-auto-discovery-route*
|       | +---ro rd-rt* [route-distinguisher]
|       | | +---ro route-distinguisher  string
|       | | +---ro vpn-target* [rt-value]
|       | | | +---ro rt-value         string
|       | | | +---ro ethernet-segment-identifier?  uint32

```



```

|         |--ro next-hop?   inet:ip-address
|         |--ro detail
|           |--ro attributes
|             |--ro extended-community*  string
|           |--ro bestpath?   empty
+--ro ip-prefix-route*
  |--ro rd-rt* [route-distinguisher]
  |   |--ro route-distinguisher  string
  |   |--ro vpn-target* [rt-value]
  |   |--ro rt-value  string
  |--ro ethernet-segment-identifier?  uint32
  |--ro ip-prefix?  inet:ip-prefix
  |--ro path*
  |   |--ro next-hop?   inet:ip-address
  |   |--ro label?     mpls:mpls-label
  |   |--ro detail
  |     |--ro attributes
  |       |--ro extended-community*  string
  |     |--ro bestpath?   empty
+--ro statistics
  |--ro tx-count?  uint32
  |--ro rx-count?  uint32
  |--ro detail
    |--ro broadcast-tx-count?  uint32
    |--ro broadcast-rx-count?  uint32
    |--ro multicast-tx-count?  uint32
    |--ro multicast-rx-count?  uint32
    |--ro unicast-tx-count?    uint32
    |--ro unicast-rx-count?    uint32

```

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@2016-07-08.yang"
module ietf-ethernet-segment {
  namespace "urn:ietf:params:xml:ns:yang:ietf-ethernet-segment";
  prefix "es";

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

  import ietf-evpn {
    prefix "evpn";
  }

```

```
    }

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

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

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

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

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

    /* Features */

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

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

    /* Typedefs */

    typedef status-type {
      type enumeration {
        enum up {
          description "Status is up";
        }
      }
    }
```

```
    enum down {
      description "Status is down";
    }
  }
  description "status type";
}

/* 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";
    }
    choice ac-or-pw {
      description "ac-or-pw";
      case ac {
        leaf ac {
          type string;
          description "Eventual reference to standard " +
            "attachment circuit definition";
        }
      }
      case pw {
        leaf pw {
          type string;
          description "Eventual reference to standard " +
            "pseudowire definition";
        }
      }
    }
  }
  leaf ethernet-segment-identifier {
    type uint32;
    description "Ethernet segment identifier (esi)";
  }
  choice active-mode {
    mandatory true;
    description "Choice of active mode";
    case single-active {
      leaf single-active-mode {
        type empty;
        description "single-active-mode";
      }
    }
  }
}
```

```
    }
    case all-active {
      leaf all-active-mode {
        type empty;
        description "all-active-mode";
      }
    }
  }
  container pbb-parameters {
    if-feature ethernet-segment-pbb-params;
    description "PBB configuration";
    leaf backbone-src-mac {
      type yang:mac-address;
      description "backbone-src-mac, only if this is a PBB";
    }
  }
  uses evpn:bgp-parameters-grp {
    if-feature ethernet-segment-bgp-params;
  }
  container df-election {
    description "df-election";
    choice df-election-method {
      description "Choice of df election method";
      case highest-random-weight {
        leaf hrw {
          type boolean;
          description "Enable (TRUE) or disable (FALSE) " +
            "highest random weight";
        }
      }
    }
  }
  leaf election-wait-time {
    type uint32;
    description "election-wait-time";
  }
  leaf ead-evi-route {
    type boolean;
    default false;
    description "Enable (true) or disable (false) ead-evi-route";
  }
  description "An ethernet segment";
}

container ethernet-segments-state {
  config false;
  description "Ethernet segmet operational state";
}
```

```
list ethernet-segment-state {
  key "name";
  leaf name {
    type string;
    description "Name of the ethernet segment";
  }
  leaf service-type {
    type string;
    description "service-type";
  }
  leaf status {
    type status-type;
    description "Ethernet segment status";
  }
  choice ac-or-pw {
    description "ac-or-pw";
    case ac {
      leaf ac {
        type string;
        description "Name of attachment circuit";
      }
    }
    case pw {
      leaf pw {
        type string;
        description "Name of pseudowire";
      }
    }
  }
  leaf interface-status {
    type status-type;
    description "interface status";
  }
  leaf ethernet-segment-identifier {
    type uint32;
    description "Ethernet segment identifier (esi)";
  }
  leaf active-mode {
    type string;
    description "Single-active-mode/all-active-mode";
  }
  container pbb-parameters {
    if-feature "ethernet-segment-pbb-params";
    description "PBB configuration";
    leaf backbone-src-mac {
      type yang:mac-address;
      description "backbone-src-mac, only if this is a PBB";
    }
  }
}
```

```
    }
    uses evpn:bgp-parameters-grp {
      if-feature ethernet-segment-bgp-params;
    }
    container df-election {
      description "df-election";
      leaf hrw-enabled {
        type boolean;
        description "hrw-enabled is enabled (TRUE) " +
          "or disabled (FALSE)";
      }
      leaf election-wait-time {
        type uint32;
        description "election-wait-time";
      }
    }
    leaf ead-evi-route-enabled {
      type boolean;
      description "ead-evi-route is enabled (TRUE) " +
        "or disabled (FALSE)";
    }
    leaf esi-label {
      type string;
      description "esi-label";
    }
    list member {
      leaf ip-address {
        type inet:ip-address;
        description "ip-address";
      }
      description "member of the ethernet segment";
    }
    list df {
      leaf service-identifier {
        type uint32;
        description "service-identifier";
      }
      leaf vlan {
        type uint32;
        description "vlan";
      }
      leaf ip-address {
        type inet:ip-address;
        description "ip-address";
      }
      description "df of an evpn instance's vlan";
    }
  }
  description "An ethernet segment";
```

```
    }  
  }  
}
```

<CODE ENDS>

4.2 EVPN Yang Module

```
<CODE BEGINS> file "ietf-evpn@2016-07-08.yang"  
module ietf-evpn {  
  namespace "urn:ietf:params:xml:ns:yang:ietf-evpn";  
  prefix "evpn";  
  
  import ietf-inet-types {  
    prefix "inet";  
  }  
  
  import ietf-mpls {  
    prefix "mpls";  
  }  
  
  import ietf-yang-types {  
    prefix "yang";  
  }  
  
  organization "ietf";  
  contact "ietf";  
  description "evpn";  
  
  revision "2016-07-08" {  
    description " - Added operational state " +  
      " - Added a configuration knob to enable/disable " +  
      " underlay-multicast " +  
      " - Added a configuration knob to enable/disable " +  
      " flooding of unknow unicast " +  
      " - Added several configuration knobs " +  
      " to manage ARP and ND" +  
      "";  
    reference "";  
  }  
  
  revision "2016-06-23" {  
    description "WG document adoption";  
    reference "";  
  }  
  
  revision "2015-10-15" {
```

```
    description "Initial revision";
    reference   "";
  }

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

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

  /* 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-rd-grp {
    description "BGP RD grouping";
    leaf route-distinguisher {
      type string;
      description "BGP RD";
    }
  }

  grouping bgp-rd-rt-grp {
    description "BGP RD-RT grouping";
    list rd-rt {
      key "route-distinguisher";
      leaf route-distinguisher {
        type string;
        description "BGP RD";
      }
    }
    list vpn-target {
```

```
        key "rt-value";
        leaf rt-value {
            type string;
            description "BGP route target";
        }
        description "List of route targets";
    }
    description "List of RD";
}

grouping bgp-parameters-grp {
    description "BGP parameters grouping";
    container bgp-parameters {
        description "BGP parameters";
        container common {
            description "Common BGP parameters";
            uses bgp-rd-rt-grp {
                refine "rd-rt" {
                    max-elements 1;
                }
                augment "rd-rt/vpn-target" {
                    description "Add type of RT";
                    leaf rt-type {
                        type bgp-rt-type;
                        mandatory true;
                        description "Type of RT";
                    }
                }
            }
        }
    }
}

grouping common-route-parameters-grp {
    description "common-route-parameters-grp";
    uses bgp-rd-rt-grp;
    leaf ethernet-segment-identifier {
        type uint32;
        description "Ethernet segment identifier (esi)";
    }
}

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

```
    }
    leaf label {
      type mpls:mpls-label;
      description "label";
    }
  }

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

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

/* EVPN YANG Model */

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

```
    }
    case p2mp-replication {
      leaf p2mp-replication {
        type boolean;
        description "p2mp-replication";
      }
    }
  }
}
container evpn-instances {
  description "evpn-instances";
  list evpn-instance {
    key "name";
    description "An EVPN instance";
    leaf name {
      type string;
      description "Name of EVPN instance";
    }
    leaf evi {
      type uint32;
      description "evi";
    }
    container pbb-parameters {
      if-feature "evpn-pbb-params";
      description "PBB parameters";
      leaf source-bmac {
        type yang:hex-string;
        description "source-bmac";
      }
    }
  }
  uses bgp-parameters-grp {
    if-feature "evpn-bgp-params";
  }
  leaf arp-proxy {
    type boolean;
    default false;
    description "Enable (TRUE) or disable (FALSE) ARP proxy";
  }
  leaf arp-suppression {
    type boolean;
    default false;
    description "Enable (TRUE) or disable (FALSE) " +
      "ARP suppression";
  }
  leaf nd-proxy {
    type boolean;
    default false;
    description "Enable (TRUE) or disable (FALSE) ND proxy";
  }
}
```



```
        "is enabled (TRUE) " +
        "or disabled (FALSE)";
    }
    leaf arp-proxy-enabled {
        type boolean;
        description "arp-proxy is enabled (TRUE) " +
            "or disabled (FALSE)";
    }
    leaf arp-suppression-enabled {
        type boolean;
        description "arp-suppression is enabled (TRUE) " +
            "or disabled (FALSE)";
    }
    leaf nd-proxy-enabled {
        type boolean;
        description "nd-proxy is enabled (TRUE) " +
            "or disabled (FALSE)";
    }
    leaf nd-suppression-enabled {
        type boolean;
        description "nd-suppression is enabled (TRUE) " +
            "or disabled (FALSE)";
    }
    leaf underlay-multicast-enabled {
        type boolean;
        description "underlay-multicast is enabled (TRUE) " +
            "or disabled (FALSE)";
    }
    leaf flood-unknown-unicast-suppression-enabled {
        type boolean;
        description "flood-unknown-unicast-suppression is " +
            "enabled (TRUE) or disabled (FALSE)";
    }
    container routes {
        description "routes";
        list ethernet-auto-discovery-route {
            uses common-route-parameters-grp;
            leaf ethernet-tag {
                type uint32;
                description "An ethernet tag (etag) indentifying a " +
                    "broadcast domain";
            }
            list path {
                uses next-hop-label-grp;
                uses path-detail-grp;
                description "path";
            }
            description "ethernet-auto-discovery-route";
        }
    }
}
```

```
    }
  list mac-ip-advertisement-route {
    uses common-route-parameters-grp;
    leaf ethernet-tag {
      type uint32;
      description "An ethernet tag (etag) indentifying a " +
        "broadcast domain";
    }
    leaf mac-address {
      type yang:hex-string;
      description "Route mac address";
    }
    leaf mac-address-length {
      type uint8 {
        range "0..48";
      }
      description "mac address length";
    }
    leaf ip-prefix {
      type inet:ip-prefix;
      description "ip-prefix";
    }
    list path {
      uses next-hop-label-grp;
      uses next-hop-label2-grp;
      uses path-detail-grp;
      description "path";
    }
    description "mac-ip-advertisement-route";
  }
  list inclusive-multicast-ethernet-tag-route {
    uses common-route-parameters-grp;
    leaf originator-ip-prefix {
      type inet:ip-prefix;
      description "originator-ip-prefix";
    }
    list path {
      uses next-hop-label-grp;
      uses path-detail-grp;
      description "path";
    }
    description "inclusive-multicast-ethernet-tag-route";
  }
  list ethernet-segment-route {
    uses common-route-parameters-grp;
    leaf originator-ip-prefix {
      type inet:ip-prefix;
      description "originator ip-prefix";
    }
  }
}
```

```
    }
    list path {
      leaf next-hop {
        type inet:ip-address;
        description "next-hop";
      }
      uses path-detail-grp;
      description "path";
    }
    description "ethernet-segment-route";
  }
  list ip-prefix-route {
    uses common-route-parameters-grp;
    leaf ip-prefix {
      type inet:ip-prefix;
      description "ip-prefix";
    }
    list path {
      uses next-hop-label-grp;
      uses path-detail-grp;
      description "path";
    }
    description "ip-prefix route";
  }
}
container statistics {
  description "Statistics";
  leaf tx-count {
    type uint32;
    description "transmission count";
  }
  leaf rx-count {
    type uint32;
    description "receive count";
  }
}
container detail {
  description "Detailed statistics";
  leaf broadcast-tx-count {
    type uint32;
    description "broadcast transmission count";
  }
  leaf broadcast-rx-count {
    type uint32;
    description "broadcast receive count";
  }
  leaf multicast-tx-count {
    type uint32;
    description "multicast transmission count";
  }
}
```


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

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

Himanshu Shah
Ciena Corporation
EMail: hshah@ciena.com

Zhenbin Li

Huawei Technologies
EMail: lizhenbin@huawei.com

Helen Chen
Ericsson
EMail: ichen@kuatrotech.com

Kishore Tiruveedhula
Juniper Networks
EMail: kishoret@juniper.net

Iftekar Hussain
Infinera Corporation
EMail: ihussain@infinera.com

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

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

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

March 11, 2019

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

Abstract

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

Status of this Memo

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

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

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

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

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

Copyright and License Notice

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

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

Convention

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

Table of Contents

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

1. Introduction

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

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

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

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

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

- o Configuration
- o Operational State
- o Notifications

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

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

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

2. Specification of Requirements

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

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

3. EVPN YANG Model

3.1. Overview

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

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

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

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

Following documents will be covered at that time:

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

3.2 Ethernet-Segment Model

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

module: ietf-ethernet-segment

```

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

```

3.3 EVPN Model

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

```

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

```


following high level configuration areas:

4.1 Ethernet Segment Yang Module

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

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

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

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

  import iETF-interfaces {
    prefix "if";
  }

  import iETF-pseudowires {
    prefix "pw";
  }

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

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

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

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

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

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

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

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

/* Features */
```

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

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

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

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

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

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

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

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

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

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

4.2 EVPN Yang Module

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

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

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

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

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

  import ietf-l2vpn {
    prefix "l2vpn";
  }

  import ietf-pseudowires {
    prefix "pw";
  }

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

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

```
description    "evpn";

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

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

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

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

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

```
    }

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

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

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

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

    /* Identities */

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

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

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

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

    /* Typedefs */

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

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

/* Groupings */

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

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

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

grouping path-detail-grp {
```

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

/* EVPN YANG Model */

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

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

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

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

```
    }
    leaf ethernet-tag {
      type uint32;
      description "An ethernet tag (etag) indentifying a " +
        "broadcast domain";
    }
    leaf mac-address {
      type yang:mac-address;
      description "Route mac address";
    }
    leaf mac-address-length {
      type uint8 {
        range "0..48";
      }
      description "mac address length";
    }
    leaf ip-prefix {
      type inet:ip-prefix;
      description "ip-prefix";
    }
    list path {
      uses next-hop-label-grp;
      uses next-hop-label2-grp;
      uses path-detail-grp;
      description "path";
    }
    description "mac-ip-advertisement-route";
  }
  list inclusive-multicast-ethernet-tag-route {
    uses route-rd-rt-grp;
    leaf originator-ip-prefix {
      type inet:ip-prefix;
      description "originator-ip-prefix";
    }
    list path {
      uses next-hop-label-grp;
      uses path-detail-grp;
      description "path";
    }
    description "inclusive-multicast-ethernet-tag-route";
  }
  list ethernet-segment-route {
    uses route-rd-rt-grp;
    leaf ethernet-segment-identifier {
      type es:ethernet-segment-identifier-type;
      description "Ethernet segment identifier (esi)";
    }
    leaf originator-ip-prefix {
```

```
        type inet:ip-prefix;
        description "originator ip-prefix";
    }
    list path {
        leaf next-hop {
            type inet:ip-address;
            description "next-hop";
        }
        uses path-detail-grp;
        description "path";
    }
    description "ethernet-segment-route";
}
list ip-prefix-route {
    uses route-rd-rt-grp;
    leaf ethernet-segment-identifier {
        type es:ethernet-segment-identifier-type;
        description "Ethernet segment identifier (esi)";
    }
    leaf ip-prefix {
        type inet:ip-prefix;
        description "ip-prefix";
    }
    list path {
        uses next-hop-label-grp;
        uses path-detail-grp;
        description "path";
    }
    description "ip-prefix route";
}
}
container statistics {
    config false;
    description "Statistics";
    leaf tx-count {
        type yang:zero-based-counter32;
        description "transmission count";
    }
    leaf rx-count {
        type yang:zero-based-counter32;
        description "receive count";
    }
}
container detail {
    description "Detailed statistics";
    leaf broadcast-tx-count {
        type yang:zero-based-counter32;
        description "broadcast transmission count";
    }
}
```



```

    description "Augment for an L2VPN instance and EVPN association";
    leaf evpn-instance {
        type evpn-instance-ref;
        description "Reference to an EVPN instance";
    }
}

augment "/ni:network-instances/ni:network-instance/ni:ni-type" +
    "/l2vpn:l2vpn" {
    when "l2vpn:type = 'l2vpn:vpls-instance-type'" {
        description "Constraints only for VPLS pseudowires";
    }
    description "Augment for VPLS instance";
    container vpls-constraints {
        must "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:pw/l2vpn:name]" +
            "    /evpn-pw/local-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:primary-pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:primary-pw/l2vpn:name]" +
            "    /evpn-pw/local-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:backup-pw/l2vpn:name]" +
            "    /evpn-pw/remote-id)) and " +
            "not(boolean(/pw:pseudowires/pw:pseudowire" +
            "    [pw:name = current()/../l2vpn:endpoint" +
            "    /l2vpn:backup-pw/l2vpn:name]" +
            "    /evpn-pw/local-id))" {
            description "A VPLS pseudowire must not be EVPN PW";
        }
        description "VPLS constraints";
    }
}

/* Notifications */

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

```

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

5. Security Considerations

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

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

6. IANA Considerations

None.

7. References

7.1. Normative References

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

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

7.2. Informative References

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

Authors' Addresses

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

Himanshu Shah
Ciena Corporation
EMail: hshah@ciena.com

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

Iftekar Hussain
Infinera Corporation
EMail: ihussain@infinera.com

Kishore Tiruveedhula
Juniper Networks
EMail: kishoret@juniper.net

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

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

Zhenbin Li
Huawei Technologies
EMail: lizhenbin@huawei.com

BESS Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 26, 2016

H. Shah, Ed.
Ciena Corporation
P. Brissette, Ed.
Cisco Systems, Inc.
I. Chen, Ed.
Ericsson
I. Hussain, Ed.
Infinera Corporation
B. Wen, Ed.
Comcast
June 24, 2016

YANG Data Model for MPLS-based L2VPN
draft-ietf-bess-l2vpn-yang-00.txt

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on December 26, 2016.

Copyright Notice

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

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

Table of Contents

1.	Introduction	2
2.	Specification of Requirements	3
3.	L2VPN YANG Model	3
3.1.	Overview	3
3.2.	L2VPN Common	7
3.2.1.	ac-templates	7
3.2.2.	pw-templates	7
3.3.	Point-to-Point and Multipoint service	7
3.3.1.	ac list	7
3.3.2.	pw list	7
3.3.3.	redundancy-grp choice	8
3.3.4.	endpoint container	8
3.3.5.	vpws-instances and bridge-table-instances container	8
3.4.	Operational State	9
3.5.	Yang tree	9
4.	YANG Module	19
5.	Security Considerations	45
6.	IANA Considerations	45
7.	Acknowledgments	45
8.	References	45
8.1.	Normative References	46
8.2.	Informative References	46
	Appendix A. Example Configuration	49
	Appendix B. Contributors	49
	Authors' Addresses	50

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 defines a YANG data model for MPLS based Layer 2 VPN services (L2VPN) [RFC4664] and includes switching between the local

attachment circuits. The L2VPN model covers point-to-point VPWS and Multipoint VPLS services. These services use signaling of Pseudowires across MPLS networks using LDP [RFC4447][RFC4762] or BGP[RFC4761].

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

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

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

The data model uses following constructs for configuration and management:

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

The current document focuses on definition of configuration and state objects. The future revisions is expected to cover the actions and notifications aspects of the model.

The L2VPN data object model uses the instance centric approach. The attributes of each service; VPWS, VPLS, etc are specified for a given service instance.

2. Specification of Requirements

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

3. L2VPN YANG Model

3.1. Overview

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 operations state object holds read-only information of objects that has either been configured or dynamically created.

The IETF working group has defined the VPWS and VPLS services that leverages the pseudowire technologies defined by the PWE3 working group. A large number of RFCs from these working groups cover this subject matter. Hence, it is prudent that this document state the scope of the MPLS L2VPN object model definitions.

The following documents are within the scope. This is not an exhaustive list but a representation of documents that are covered for this work:

- o Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3) [RFC3916]
- o Pseudo-wire Emulation Edge-to-Edge (PWE3) Architecture [RFC3985]
- o IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3) [RFC4446]
- o Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) [RFC4447]
- o Encapsulation Methods for Transport of Ethernet over MPLS Networks [RFC4448]
- o Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN [RFC4385]
- o Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3) [RFC5254]
- o An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge [RFC5659]
- o Segmented Pseudowire [RFC6073]
- o Framework for Layer 2 Virtual Private Networks [RFC4664]
- o Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks [RFC4665]
- o Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling [RFC4761]

- o Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling [RFC4762]
- o Attachment Individual Identifier (AII) Types for Aggregation [RFC5003]
- o Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs) [RFC6074]
- o Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network [RFC6391]
- o Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling [RFC6624]
- o Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging [RFC7041]
- o LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS) [RFC7361]
- o Using the generic associated channel label for Pseudowire in the MPLS Transport Profile [RFC6423]
- o Pseudowire status for static pseudowire [RFC6478]

The specifics of pseudowire over MPLS-TP LSPs is in scope. However, the initial effort addresses definitions of object models that are commonly deployed.

The IETF work in L2VPN and PWE3 working group relating to L2TP, OAM, multicast (e.g. p2mp, etree, etc) and access specific protocols such as G.8032, MSTP, etc is out-of-scope for this document.

The following is the high level view of the L2VPN data model.

```
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

3.2. L2VPN Common

3.2.1. ac-templates

The ac-templates container does not exist. The AC will be referenced from definitions by IEEE and/or MEF.

3.2.2. 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.

3.3. Point-to-Point and Multipoint service

3.3.1. ac list

AC resides within endpoint container as member of ac-or-pw-or-redundancy-grp.

3.3.2. 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.

3.3.3. redundancy-grp choice

The redundancy-grp is a generic redundancy construct which can hold primary and backup members of AC and PWs. This flexibility permits combinations of -

- o primary and backup AC
- o primary and backup PW
- o primary AC and backup PW
- o primary PW and backup AC

3.3.4. 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.

3.3.5. vpws-instances and bridge-table-instances container

The vpws-instance container contains a list of vpws-instances. 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 contains 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.

3.4. 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.

3.5. Yang tree

```

module: ietf-l2vpn
  +--rw l2vpn
    |   +--rw common
    |   |   +--rw pw-templates
    |   |   |   +--rw pw-template* [name]
    |   |   |   |   +--rw name          string
    |   |   |   |   +--rw mtu?          uint16
    |   |   |   |   +--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?          uint16
    |   |   |   |   +--rw mac-aging-timer? uint32
    |   |   |   |   +--rw pbb-parameters
    |   |   |   |   |   +--rw (component-type)?
    |   |   |   |   |   |   +--:(i-component)
    |   |   |   |   |   |   |   +--rw i-sid?          i-sid-type
    |   |   |   |   |   |   |   +--rw backbone-src-mac? yang:mac-address
    |   |   |   |   |   |   +--:(b-component)
    |   |   |   |   |   |   |   +--rw bind-b-component? bridge-table-instance-ref
    |   |   |   |   +--rw bgp-parameters
    |   |   |   |   |   +--rw common
    |   |   |   |   |   |   +--rw route-distinguisher? string
    |   |   |   |   |   |   +--rw vpn-target* [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?                  uint16
|   +--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?      mpls:mpls-label
|   +--rw receive-label?       mpls:mpls-label
|   +--rw tunnel-policy?       string
+--rw endpoint* [name]
|   +--rw name                    string
|   +--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
|
|   +--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? uint16
  +--rw mac-aging-timer? uint32
  +--rw service-type? l2vpn-service-type
  +--rw discovery-type? l2vpn-discovery-type
  +--rw signaling-type l2vpn-signaling-type
  +--rw bgp-parameters
    +--rw common
      +--rw route-distinguisher? string
      +--rw vpn-target* [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? uint16
  +--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? mpls:mpls-label
      +--rw receive-label? mpls:mpls-label
    +--:(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)

```



```

|--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-target* [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* [name]
  |--ro name                string
  |--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?       uint16
  |   |   |--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? mpls:mpls-label
  |   |   |--ro receive-label? mpls:mpls-label
  |   |   |--ro tunnel-policy? string
  |--:(redundancy-grp)
  |   |--ro (primary)
  |   |   |--:(primary-pw)
  |   |   |   |--ro primary-pw* [name]
  |   |   |   |   |--ro name        string
  |   |   |   |   |--ro state?     operational-state-type
  |   |   |   |   |--ro mtu?       uint16
  |   |   |   |   |--ro mac-withdraw? 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? mpls:mpls-label
        | +--ro receive-label? mpls:mpls-label
        +---:(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?             uint16
      +--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? mpls:mpls-label
          | +--ro receive-label? mpls:mpls-label
          +---:(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?      uint16
        +--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?  mpls:mpls-label
            +--ro receive-label?  mpls:mpls-label
          +--:(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?      uint16
            +--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?   mpls:mpls-label
|         |         |   +--ro receive-label?    mpls:mpls-label
|         |         |   +---:(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?                uint16
|   |   |   +--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? mpls:mpls-label
|   |   |   |   |   +--ro receive-label? mpls:mpls-label
|   |   |   |   |   +---:(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

4. YANG Module

The L2VPN configuration container is logically divided into following high level config areas:

```
(CODE BEGINS) file "ietf-l2vpn@2016-05-31.yang"
module ietf-l2vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn";
  prefix "l2vpn";

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

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

  import ietf-mpls {
    prefix "mpls";
  }

  organization "ietf";
  contact "ietf";
  description "l2vpn";

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

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

revision "2015-10-08" {
  description "Second revision " +
    " - Added container vpls-instances " +
    " - Rearranged groupings and typedefs to be " +
    "   reused across vpls-instance and vpws-instances";
  reference "";
}

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

/* identities */

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

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

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

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

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

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

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

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

  identity bgp-auto-discovery {
    base "l2vpn-discovery";
    description "Border Gateway Protocol (BGP) auto-discovery of " +
```

```
        "l2vpn service";
    }

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

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

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

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

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

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

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

    /* typedefs */

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

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

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

typedef 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";
}
```

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

/* groupings */

grouping pbb-parameters-grp {
  description "PBB parameters grouping";
  container pbb-parameters {
    description "pbb-parameters";
    choice component-type {
      description "PBB component type";
      case i-component {
        leaf i-sid {
          type i-sid-type;
          description "I-SID";
        }
        leaf backbone-src-mac {
          type yang:mac-address;
          description "backbone-src-mac";
        }
      }
      case b-component {
        leaf bind-b-component {
          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-auto-discovery'" {
                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-target {
            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-auto-discovery'" {
        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-signaling'" {
        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 mpls:mpls-label;
                description "transmit lable";
            }
        }
    }
}
```

```
    }
    leaf receive-label {
      type mpls:mpls-label;
      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 of 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 of attachment 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 of attachment 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 of attachment 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 uint16;
    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 mpls:mpls-label;
    description "transmit lable";
  }
  leaf receive-label {
    type mpls:mpls-label;
    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 uint16;
    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 uint16;
                    description "pseudowire mtu";
                }
                leaf cw-negotiation {
                    type cw-negotiation-type;
                    default "preferred";
                    description
                        "control-word negotiation preference";
                }
                leaf tunnel-policy {
                    type string;
                    description "tunnel policy name";
                }
            }
        }
        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 uint16;
            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 uint16;
            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 mpls:mpls-label;
        description "transmit lable";
    }
    leaf receive-label {
        type mpls:mpls-label;
        description "receive label";
    }
    leaf tunnel-policy {
        type string;
        description "tunnel policy name";
    }
}
list endpoint {
    key "name";
    leaf name {
        type string;
        description "endpoint name";
    }
    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 uint16;
            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-discovery;
            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 uint16;
    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 pseudowire 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 uint16;
                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 "name";
                leaf name {
                    type string;
                    description "endpoint name";
                }
                leaf split-horizon-group {
                    type string;
                    description "Identify a split horizon group";
                }
                choice ac-or-pw-or-redundancy-grp {
                    description "A choice of attachment 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";
  }
}
```



```
    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

5. Security Considerations

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

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

6. IANA Considerations

None.

7. Acknowledgments

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

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>>.

8.2. Informative References

- [RFC3916] Xiao, X., Ed., McPherson, D., Ed., and P. Pate, Ed., "Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)", RFC 3916, DOI 10.17487/RFC3916, September 2004, <<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>>.

Appendix A. Example Configuration

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

Appendix B. Contributors

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

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

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

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

Zhenbin Li
Huawei Technologies
Email: lizhenbin@huawei.com

Zhuang Shunwan
Huawei Technologies
Email: Zhuangshunwan@huawei.com

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

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

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

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

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

Kishore Tiruveedhula
Juniper Networks
Email: kishoret@juniper.net

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

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

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

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Ing-When Chen
Ericsson

Email: ichen@kuatrotech.com

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

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

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

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

Abstract

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

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

Status of This Memo

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

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

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

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

Copyright Notice

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

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

Table of Contents

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

1. Introduction

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

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

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

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

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

The data model uses following constructs for configuration and management:

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

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

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

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

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

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

The L2VPN data object model augments Pseudowire data object for its definition.

The document also includes Notifications used by the L2VPN object model

2. Specification of Requirements

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

3. L2VPN YANG Model

3.1. Overview

The document defines configuration of one single container for L2VPN. Within the l2vpn container, common parameters and a list of endpoints are defined. For the point-to-point VPWS configuration, endpoint list is used with the constraint that limits the number of endpoints to be two. For the multipoint service, endpoint list is used. Each endpoint contains the common definition that is either an attachment circuit, a pseudowire or a redundancy group. The previous versions of this document represented VPWS service with definition of endpoint-a and endpoint-z while VPLS with a list of endpoints. This duplication is removed with simplified version whereby list of endpoints is used for both. When defining VPWS, the number of endpoints is constrained to two endpoints.

The l2vpn container also includes definition of common building blocks for redundancy-grp templates and pseudowire-templates.

The State objects have been consolidated with the configuration object as per the recommendations provided by the Guidelines for Yang Module Authors document.

The IETF working group has defined the VPWS and VPLS services that leverages the pseudowire technologies defined by the PWE3 working group. A large number of RFCs from these working groups cover this subject matter. Hence, it is prudent that this document state the scope of the MPLS L2VPN object model definitions.

The following documents are within the scope. This is not an exhaustive list but a representation of documents that are covered for this work:

- o Requirements for Pseudo-wire Emulation Edge-to-Edge (PWE3) [RFC3916]
- o Pseudo-wire Emulation Edge-to-Edge (PWE3) Architecture [RFC3985]
- o IANA Allocations for Pseudowire Edge to Edge Emulation (PWE3) [RFC4446]
- o Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP) [RFC8077]
- o Encapsulation Methods for Transport of Ethernet over MPLS Networks [RFC4448]
- o Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN [RFC4385]
- o Requirements for Multi-Segment Pseudowire Emulation Edge-to-Edge (PWE3) [RFC5254]
- o An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge [RFC5659]
- o Segmented Pseudowire [RFC6073]
- o Framework for Layer 2 Virtual Private Networks [RFC4664]
- o Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks [RFC4665]
- o Virtual Private LAN Service (VPLS) Using BGP for Auto-Discovery and Signaling [RFC4761]
- o Virtual Private LAN Service (VPLS) Using Label Distribution Protocol (LDP) Signaling [RFC4762]
- o Attachment Individual Identifier (AII) Types for Aggregation [RFC5003]
- o Provisioning, Auto-Discovery, and Signaling in Layer 2 Virtual Private Networks (L2VPNs) [RFC6074]
- o Flow-Aware Transport of Pseudowires over an MPLS Packet Switched Network [RFC6391]

- o Layer 2 Virtual Private Networks Using BGP for Auto-Discovery and Signaling [RFC6624]
- o Extensions to the Virtual Private LAN Service (VPLS) Provider Edge (PE) Model for Provider Backbone Bridging [RFC7041]
- o LDP Extensions for Optimized MAC Address Withdrawal in a Hierarchical Virtual Private LAN Service (H-VPLS) [RFC7361]
- o Using the generic associated channel label for Pseudowire in the MPLS Transport Profile [RFC6423]
- o Pseudowire status for static pseudowire [RFC6478]

The specifics of pseudowire over MPLS-TP LSPs is in scope. However, the initial effort addresses definitions of object models that are commonly deployed.

The IETF work in L2VPN and PWE3 working group relating to L2TP, OAM, multicast (e.g. p2mp, etree, etc) and access specific protocols such as G.8032, MSTP, etc is out-of-scope for this document.

The following is the high level view of the L2VPN data model.

```

PW // Container
    PW specific attributes

    PW template definition

template-ref Redundancy-Group // redundancy-group
    template
    attributes

Network Instance // container
    l2vpn // container

    common attributes

    BGP-parameters // container
        common attributes
        auto-discovery attributes
        signaling attributes

    // list of PWs being used
    PW // container
        template-ref PW
        attribute-override

    PBB-parameters // container
        pbb specific attributes

    VPWS-constraints // rule to limit number of endpoints to two

    // List of endpoints, where each member endpoint container is -
    PW // reference
    redundancy-grp // container
        AC // eventual reference to standard AC
        PW // reference

```

Figure 1

3.2. Latest addition

Pseudowire module is extended to include,

Multi-segment PW - a new attribute is added to pseudowire that identifies the pseudowire as a member of the multi-segment

pseudowire. Two pseudowire members in a VPWS, configures a multi-segment pseudowire at the switching PE.

Pseudowire load-balancing - The load-balancing behaviour for a pseudowire can be configured either using the FAT label that resides below the pseudowire label or Entropy label with Entropy label indicator above the pseudowire label. By default, the load-balancing is disabled.

FEC 129 related - AGI, SAI and TAI string configurations is added to facilitate FEC 129 based pseudowire configuration.

3.3. Open issues and next steps

This section provides updates on open issues and will be removed before publication. Authors believes the document has covered the topics within the scope of the document. However, there are items, such as PW Headend, VPLS IRB, etc that can be candidate for inclusion. The authors would like to progress the document to publication for general availability with current content and tackle the other topics in a follow up document.

3.4. Pseudowire Common

3.4.1. Pseudowire

Pseudowire definitions is moved to a separate container in order to allow Ethernet Segment and EVPN models can refer without having to pull down L2VPN container.

3.4.2. pw-templates

The pw-templates container contains a list of pw-template. Each pw-template defines a list of common pseudowire attributes such as PW MTU, control word support etc.

3.5. L2VPN Common

3.5.1. redundancy-group-templates

The redundancy-group-template contains a list of templates. Each template defines common attributes related to redundancy such as protection mode, reversion parameters, etc.

3.6. L2VPN instance

The network instance container defined in the network instance model [I-D.ietf-rtgwg-ni-model] identifies the L2VPN instance. One of the value defined by the ni-type used in the instance model refers to VSI (Virtual Switch Instance) to denote the L2VPN instance. The name attribute field is used as the key to refer to specific network instance. Network Instance of type VSI anchors L2VPN container with a list of endpoints which when limited to two entries represents point to point service (i.e. VPWS) while more than two endpoints represent multipoint service (i.e. VPLS). Within a service instance, a set of common attributes are defined, followed by a list of PWs and a list of endpoints.

3.6.1. common attributes

The common attributes apply to entire L2VPN instance. These attributes typically include attributes such as mac-aging-timer, BGP related parameters (if using BGP signaling), discovery-type, etc.

3.6.2. PW list

The PW list is the number of PWs that are being used for a given L2VPN instance. Each PW entry refers to PW template to inherit common attributes for the PW. The one or more attributes from the template can be overridden. It further extends definitions of more PW specific attributes such as use of control word, mac withdraw, what type of signaling (i.e. LDP or BGP), setting of the TTL, etc.

3.6.3. List of endpoints

The list of endpoints define the characteristics of the L2VPN service. In the case of VPWS, the list is limited to two entries while for VPLS, there could be many.

Each entry in the endpoint list, may hold AC, PW or redundancy-grp references. The core aspect of endpoint container is its flexible personality based on what user decides to include in it. It is future-proofed with possible extensions that can be included in the endpoint container such as Integrated Route Bridging (IRB), PW Headend, Virtual Switch Instance, etc.

The endpoint entry also includes the split-horizon attribute which defines the frame forwarding restrictions between the endpoints belonging to same split-horizon group. This construct permits multiple instances of split horizon groups with its own endpoint members. The frame forwarding restrictions does not apply between endpoints that belong to two different split horizon groups.

3.6.3.1. ac

Attachment Circuit (AC) resides within endpoint entry either as an independent entity or as a member of the redundancy group. AC is not defined in this document but references the definitions specified by other working groups and standard bodies.

3.6.3.2. pw

The Pseudo-wire resides within endpoint entry either as an independent entity or as a member of the redundancy group. The PW refers to one of the entry in the list of PWs defined with the L2VPN instance.

3.6.3.3. redundancy-grp choice

The redundancy-grp is a generic redundancy construct which can hold primary and backup members of AC and PWs. This flexibility permits combinations of -

- o primary and backup AC
- o primary and backup PW
- o primary AC and backup PW
- o primary PW and backup AC

The redundancy group also defines attributes of the type of redundancy, such as protection mode, reroute mode, reversion related parameters, etc.

3.6.4. point-to-point or multipoint service

The point-to-point service as defined for VPWS is represented by a list of endpoints and is limited to two entries by the VPWS constrain rules

The multipoint service as defined for VPLS is represented by a list of endpoints.

The list of endpoints with one entry is invalid.

The augmentation of ietf-l2vpn module is TBD. All IP addresses defined in this module are currently scoped under global VRF/table.

3.6.5. multi-segment pseudowire

The multi-segment pseudowire is expressed as configuration of two pseudowire segments at the switching PEs that provides end-to-end PW path between two terminating PEs consisting of multiple pseudowire segments.

The multi-segment pseudowire is configured at switching PE using two endpoints that consists of pseudowires of type "ms-pw-members". The VPWS service construct is used with "vpws constraint" that restricts the number of endpoints to two.

To verify consistency, a) verify that both endpoints are using ms-pw-member pseudowires and b) it is only used as for VPWS configuration at the switching PE.

3.7. Operational State

The operational state of L2VPN attributes has been consolidated with the configuration as per recommendations from the guidelines for the YANG author document.

3.8. Yang tree

```

module: ietf-pseudowires
  +--rw pseudowires
    +--rw pseudowire* [name]
      +--rw name                string
      +--ro state?              pseudowire-status-type
      +--rw template?           pw-template-ref
      +--rw mtu?                 uint16
      +--rw mac-withdraw?        boolean
      +--rw pw-loadbalance?      enumeration
      +--rw ms-pw-member?        boolean
      +--rw cw-negotiation?      cw-negotiation-type
      +--rw tunnel-policy?       string
      +--rw (pw-type)?
        +--:(configured-pw)
          +--rw peer-ip?         inet:ip-address
          +--rw pw-id?           uint32
          +--rw group-id?        uint32
          +--rw icb?              boolean
          +--rw transmit-label?  rt-types:mpls-label
          +--rw receive-label?   rt-types:mpls-label
          +--rw generalized?     boolean
          +--rw agi?              string
          +--rw saii?             string
  
```

```

    |   |--rw taii?           string
    |--:(bgp-pw)
    |   |--rw remote-pe-id?  inet:ip-address
    |--:(bgp-ad-pw)
    |   |--rw remote-ve-id?  uint16
+--rw pw-templates
  |--rw pw-template* [name]
    |--rw name               string
    |--rw mtu?               uint16
    |--rw cw-negotiation?   cw-negotiation-type
    |--rw tunnel-policy?    string

module: ietf-l2vpn
+--rw l2vpn
  |--rw redundancy-group-templates
    |--rw redundancy-group-template* [name]
      |--rw name             string
      |--rw protection-mode? enumeration
      |--rw reroute-mode?   enumeration
      |--rw dual-receive?   boolean
      |--rw revert?         boolean
      |--rw reroute-delay?  uint16
      |--rw revert-delay?   uint16

augment /ni:network-instances/ni:network-instance/ni:ni-type:
+--:(l2vpn)
  |--rw type?               identityref
  |--rw mtu?                uint16
  |--rw mac-aging-timer?   uint32
  |--rw service-type?      l2vpn-service-type
  |--rw discovery-type?    l2vpn-discovery-type
  |--rw signaling-type      l2vpn-signaling-type
  |--rw bgp-parameters
    |--rw vpn-id?          string
    |--rw rd-rt
      |--rw route-distinguisher? rt-types:route-distinguisher
      |--rw vpn-target* [route-target]
        |--rw route-target      rt-types:route-target
        |--rw route-target-type  rt-types:route-target-type
  |--rw bgp-signaling
    |--rw site-id?         uint16
    |--rw site-range?     uint16
  |--rw endpoint* [name]
    |--rw name              string
    |--rw (ac-or-pw-or-redundancy-grp)?
      |--:(ac)
        |--rw ac* [name]
          |--rw name         if:interface-ref

```

```

|         +---ro state?    operational-state-type
|         +---:(pw)
|         |         +---rw pw* [name]
|         |         +---rw name    pw:pseudowire-ref
|         |         +---ro state?  -> /pw:pseudowires/pseudowire[pw:name=current (
) /../name] /state
|         +---:(redundancy-grp)
|         |         +---rw (primary)
|         |         |         +---:(primary-ac)
|         |         |         |         +---rw primary-ac
|         |         |         |         +---rw name?    if:interface-ref
|         |         |         |         +---ro state?    operational-state-type
|         |         |         +---:(primary-pw)
|         |         |         |         +---rw primary-pw* [name]
|         |         |         |         +---rw name    pw:pseudowire-ref
|         |         |         |         +---ro state?  -> /pw:pseudowires/pseudowire[pw:name=cu
rrent () /../name] /state
|         |         |         +---rw (backup)?
|         |         |         |         +---:(backup-ac)
|         |         |         |         |         +---rw backup-ac
|         |         |         |         |         +---rw name?    if:interface-ref
|         |         |         |         |         +---ro state?    operational-state-type
|         |         |         |         +---:(backup-pw)
|         |         |         |         |         +---rw backup-pw* [name]
|         |         |         |         |         +---rw name    pw:pseudowire-ref
|         |         |         |         |         +---ro state?  -> /pw:pseudowires/pseudowire[pw:na
me=current () /../name] /state
|         |         |         +---rw precedence?    uint32
|         |         |         +---rw template?      redundancy-group-template-ref
|         |         |         +---rw protection-mode? enumeration
|         |         |         +---rw reroute-mode?   enumeration
|         |         |         +---rw dual-receive?   boolean
|         |         |         +---rw revert?         boolean
|         |         |         +---rw reroute-delay?  uint16
|         |         |         +---rw revert-delay?   uint16
|         |         |         +---rw split-horizon-group? string
|         +---rw vpws-constraints
|         +---rw pbb-parameters
|         |         +---rw (component-type)?
|         |         |         +---:(i-component)
|         |         |         |         +---rw i-sid?          i-sid-type
|         |         |         |         +---rw backbone-src-mac? yang:mac-address
|         |         |         +---:(b-component)
|         |         |         |         +---rw bind-b-component-name? l2vpn-instance-name-ref
|         |         |         |         +---ro bind-b-component-type? identityref
|         +---rw vccv-ability?    boolean
|         +---rw request-vlanid?  uint16
|         +---rw vlan-tpid?       string
|         +---rw ttl?             uint8
|         +---rw pw-type:

```

```

+--:(bgp-pw)
|   +--rw bgp-pw
|       +--rw remote-pe-id?   inet:ip-address
+--:(bgp-ad-pw)
|   +--rw bgp-ad-pw
|       +--rw remote-ve-id?   uint16

notifications:
+---n l2vpn-state-change-notification
+--ro l2vpn-instance-name?     l2vpn-instance-name-ref
+--ro l2vpn-instance-type?     -> /ni:network-instances/network-instance
[ni:name=current()../l2vpn-instance-name]/l2vpn:type
+--ro endpoint?                -> /ni:network-instances/network-instance
[ni:name=current()../l2vpn-instance-name]/l2vpn:endpoint/name
+--ro (ac-or-pw-or-redundancy-grp)?
|   +--:(ac)
|   |   +--ro ac?              -> /ni:network-instances/network-instance
[ni:name=current()../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/.
./endpoint]/ac/name
|   |   +--:(pw)
|   |   |   +--ro pw?          -> /ni:network-instances/network-instance
[ni:name=current()../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/.
./endpoint]/pw/name
|   |   +--:(redundancy-grp)
|   |   |   +--ro (primary)
|   |   |   |   +--:(primary-ac)
|   |   |   |   |   +--ro primary-ac?  -> /ni:network-instances/network-instance
[ni:name=current()../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/.
./endpoint]/primary-ac/name
|   |   |   |   |   +--:(primary-pw)
|   |   |   |   |   |   +--ro primary-pw?  -> /ni:network-instances/network-instance
[ni:name=current()../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/.
./endpoint]/primary-pw/name
|   |   |   |   |   +--ro (backup)?
|   |   |   |   |   |   +--:(backup-ac)
|   |   |   |   |   |   |   +--ro backup-ac?  -> /ni:network-instances/network-instance
[ni:name=current()../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/.
./endpoint]/backup-ac/name
|   |   |   |   |   |   |   +--:(backup-pw)
|   |   |   |   |   |   |   |   +--ro backup-pw?  -> /ni:network-instances/network-instance
[ni:name=current()../l2vpn-instance-name]/l2vpn:endpoint[l2vpn:name=current()/.
./endpoint]/backup-pw/name
|   |   |   |   |   |   |   +--ro state?
|   |   |   |   |   |   |   |   identityref

```

Figure 2

4. YANG Module

The L2VPN configuration container is logically divided into following high level config areas:

```

<CODE BEGINS> file "ietf-pseudowires@2018-10-17.yang"
module ietf-pseudowires {
  namespace "urn:ietf:params:xml:ns:yang:ietf-pseudowires";
  prefix "pw";

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

```



```
}

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

organization "ietf";
contact "ietf";
description "Pseudowire YANG model";

revision "2018-10-17" {
  description "Second revision " +
    " - Added group-id and attachment identifiers " +
    "";
  reference "";
}

revision "2017-06-26" {
  description "Initial revision " +
    " - Created a new model for pseudowires, which used " +
    " to be defined within the L2VPN model " +
    "";
  reference "";
}

/* Typedefs */

typedef pseudowire-ref {
  type leafref {
    path "/pw:pseudowires/pw:pseudowire/pw:name";
  }
  description "A type that is a reference to a pseudowire";
}

typedef pw-template-ref {
  type leafref {
    path "/pw:pseudowires/pw:pw-templates/pw:pw-template/pw:name";
  }
  description "A type that is a reference to a pw-template";
}

typedef cw-negotiation-type {
  type enumeration {
    enum "non-preferred" {
      description "No preference for control-word";
    }
    enum "preferred" {
      description "Prefer to have control-word negotiation";
    }
  }
}
```

```
    }
  }
  description "control-word negotiation preference type";
}

typedef pseudowire-status-type {
  type bits {
    bit pseudowire-forwarding {
      position 0;
      description "Pseudowire is forwarding";
    }
    bit pseudowire-not-forwarding {
      position 1;
      description "Pseudowire is not forwarding";
    }
    bit local-attachment-circuit-receive-fault {
      position 2;
      description "Local attachment circuit (ingress) receive " +
        "fault";
    }
    bit local-attachment-circuit-transmit-fault {
      position 3;
      description "Local attachment circuit (egress) transmit " +
        "fault";
    }
    bit local-PSN-facing-PW-receive-fault {
      position 4;
      description "Local PSN-facing PW (ingress) receive fault";
    }
    bit local-PSN-facing-PW-transmit-fault {
      position 5;
      description "Local PSN-facing PW (egress) transmit fault";
    }
    bit PW-preferential-forwarding-status {
      position 6;
      description "Pseudowire preferential forwarding status";
    }
    bit PW-request-switchover-status {
      position 7;
      description "Pseudowire request switchover status";
    }
  }
  description
    "Pseudowire status type, as registered in the IANA " +
    "Pseudowire Status Code Registry";
}

/* Data */
```

```
container pseudowires {
  description "Configuration management of pseudowires";
  list pseudowire {
    key "name";
    description "A pseudowire";
    leaf name {
      type string;
      description "pseudowire name";
    }
    leaf state {
      type pseudowire-status-type;
      config false;
      description "pseudowire operation status";
      reference "RFC 4446 and IANA Pseudowire Status Codes " +
        "Registry";
    }
    leaf template {
      type pw-template-ref;
      description "pseudowire template";
    }
    leaf mtu {
      type uint16;
      description "PW MTU";
    }
    leaf mac-withdraw {
      type boolean;
      default false;
      description "Enable (true) or disable (false) MAC withdraw";
    }
    leaf pw-loadbalance {
      type enumeration {
        enum "disabled" {
          value 0;
          description "load-balancing disabled";
        }
        enum "fat-pw" {
          value 1;
          description "load-balance using FAT label below PW label";
        }
        enum "entropy" {
          value 2;
          description "load-balance using ELI/EL above PW label";
        }
      }
      description "PW load-balancing";
    }
    leaf ms-pw-member {
      type boolean;
    }
  }
}
```

```
    default false;
    description "Enable (true) or disable (false) not a member of MS-PW";
  }
  leaf cw-negotiation {
    type cw-negotiation-type;
    description "cw-negotiation";
  }
  leaf tunnel-policy {
    type string;
    description "tunnel policy name";
  }
  choice pw-type {
    description "A choice of pseudowire type";
    case configured-pw {
      leaf peer-ip {
        type inet:ip-address;
        description "peer IP address";
      }
      leaf pw-id {
        type uint32;
        description "pseudowire id";
      }
      leaf group-id {
        type uint32;
        description "group id";
      }
    }
    leaf icb {
      type boolean;
      description "inter-chassis backup";
    }
    leaf transmit-label {
      type rt-types:mpls-label;
      description "transmit lable";
    }
    leaf receive-label {
      type rt-types:mpls-label;
      description "receive label";
    }
    leaf generalized {
      type boolean;
      description "generalized pseudowire id FEC element";
    }
    leaf agi {
      type string;
      description "attachment group identifier";
    }
    leaf sai {
      type string;
    }
  }
}
```

```
        description "source attachment individual identifier";
    }
    leaf taii {
        type string;
        description "target attachment individual identifier";
    }
}
case bgp-pw {
    leaf remote-pe-id {
        type inet:ip-address;
        description "remote pe id";
    }
}
case bgp-ad-pw {
    leaf remote-ve-id {
        type uint16;
        description "remote ve id";
    }
}
}
}
container pw-templates {
    description "pw-templates";
    list pw-template {
        key "name";
        description "pw-template";
        leaf name {
            type string;
            description "name";
        }
        leaf mtu {
            type uint16;
            description "pseudowire mtu";
        }
        leaf cw-negotiation {
            type cw-negotiation-type;
            default "preferred";
            description
                "control-word negotiation preference";
        }
        leaf tunnel-policy {
            type string;
            description "tunnel policy name";
        }
    }
}
}
}
```

```
<CODE ENDS>
<CODE BEGINS> file "ietf-l2vpn@2019-05-28.yang"
module ietf-l2vpn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-l2vpn";
  prefix "l2vpn";

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

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

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

  import ietf-interfaces {
    prefix "if";
  }

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

  import ietf-pseudowires {
    prefix "pw";
  }

  organization "ietf";
  contact "ietf";
  description "l2vpn";

  revision "2019-05-28" {
    description "Ninth revision " +
      " - Used bgp parameters hierarchy common to L2VPN and EVPN " +
      "";
    reference "";
  }

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

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

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

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

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

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

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

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

/* identities */

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

/* typedefs */
```

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

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

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

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

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

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

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

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

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

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

/* groupings */

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

```

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

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

```

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

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

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

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

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

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

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

/* L2VPN YANG Model */

container l2vpn {
  description "L2VPN specific data";

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

/* augments */
```

```

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

```

```

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

```

```

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

```

```

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

```



```

        description "time-to-live";
    }
}

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

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

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

```

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

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

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

/* Notifications */

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

```

```

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

```

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

```


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

8. References

8.1. Normative References

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

8.2. Informative References

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

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

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

Appendix A. Example Configuration

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

Appendix B. Contributors

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

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

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

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

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

Zhenbin Li
Huawei Technologies
Email: lizhenbin@huawei.com

Zhuang Shunwan
Huawei Technologies
Email: Zhuangshunwan@huawei.com

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

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

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

Jorge Rabadan
Nokia

Email: jorge.rabadan@nokia.com

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

Santosh Esale
Juniper Networks
Email: sesale@juniper.net

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

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

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

Figure 4

Authors' Addresses

Himanshu Shah
Ciena Corporation

Email: hshah@ciena.com

Patrice Brissette
Cisco Systems, Inc.

Email: pbrisset@cisco.com

Ing-When Chen
The MITRE Corporation

Email: ingwherchen@mitre.org

Iftekar Hussain
Infinera Corporation

Email: ihussain@infinera.com

Bin Wen
Comcast

Email: Bin_Wen@cable.comcast.com

Kishore Tiruveedhula
Juniper Networks

Email: kishoret@juniper.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 8, 2017

T. Morin, Ed.
Orange
R. Kebler, Ed.
Juniper Networks
July 7, 2016

Multicast VPN fast upstream failover
draft-ietf-bess-mvpn-fast-failover-01

Abstract

This document defines multicast VPN extensions and procedures that allow fast failover for upstream failures, by allowing downstream PEs to take into account the status of Provider-Tunnels (P-tunnels) when selecting the upstream PE for a VPN multicast flow, and extending BGP MVPN routing so that a C-multicast route can be advertized toward a standby upstream PE.

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 8, 2017.

Copyright Notice

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

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

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	UMH Selection based on tunnel status	3
3.1.	Determining the status of a tunnel	4
3.1.1.	mVPN tunnel root tracking	5
3.1.2.	PE-P Upstream link status	5
3.1.3.	P2MP RSVP-TE tunnels	5
3.1.4.	Leaf-initiated P-tunnels	6
3.1.5.	(S,G) counter information	6
3.1.6.	BFD Discriminator	6
3.1.7.	Per PE-CE link BFD Discriminator	8
4.	Standby C-multicast route	9
4.1.	Downstream PE behavior	10
4.2.	Upstream PE behavior	11
4.3.	Reachability determination	12
4.4.	Inter-AS	12
4.4.1.	Inter-AS procedures for downstream PEs, ASBR fast failover	13
4.4.2.	Inter-AS procedures for ASBRs	13
5.	Hot leaf standby	13
6.	Duplicate packets	14
7.	IANA Considerations	14
8.	Security Considerations	15
9.	Acknowledgements	15
10.	Contributor Addresses	15
11.	References	17
11.1.	Normative References	17
11.2.	Informative References	17
	Authors' Addresses	18

1. Introduction

In the context of multicast in BGP/MPLS VPNs, it is desirable to provide mechanisms allowing fast recovery of connectivity on different types of failures. This document addresses failures of

elements in the provider network that are upstream of PEs connected to VPN sites with receivers.

Section 3 describes local procedures allowing an egress PE (a PE connected to a receiver site) to take into account the status of P-Tunnels to determine the Upstream Multicast Hop (UMH) for a given (C-S, C-G). This method does not provide a "fast failover" solution when used alone, but can be used with the following sections for a "fast failover" solution.

Section 4 describes protocol extensions that can speed up failover by not requiring any multicast VPN routing message exchange at recovery time.

Moreover, section 5 describes a "hot leaf standby" mechanism, that uses a combination of these two mechanisms. This approach has similarities with the solution described in [RFC7431] to improve failover times when PIM routing is used in a network given some topology and metric constraints.

2. Terminology

The terminology used in this document is the terminology defined in [RFC6513] and [RFC6514].

x-PMSI: I-PMSI or S-PMSI

3. UMH Selection based on tunnel status

Current multicast VPN specifications [RFC6513], section 5.1, describe the procedures used by a multicast VPN downstream PE to determine what the upstream multicast hop (UMH) is for a given (C-S,C-G).

The procedure described here is an OPTIONAL procedure that consists of having a downstream PE take into account the status of P-tunnels rooted at each possible upstream PEs, for including or not including each given PE in the list of candidate UMHs for a given (C-S,C-G) state. The result is that, if a P-tunnel is "down" (see Section 3.1), the PE that is the root of the P-Tunnel will not be considered for UMH selection, which will result in the downstream PE to failover to the upstream PE which is next in the list of candidates.

A downstream PE monitors the status of the tunnels of UMHs that are ahead of the current one. Whenever the downstream PE determines that one of these tunnels is no longer "known to down", the PE selects the UMH corresponding to that as the new UMH.

More precisely, UMH determination for a given (C-S,C-G) will consider the UMH candidates in the following order:

- o first, the UMH candidates that either (a) advertise a PMSI bound to a tunnel, where the specified tunnel is not known to be down or (b) do not advertise any x-PMSI applicable to the given (C-S,C-G) but have associated a VRF Route Import BGP attribute to the unicast VPN route for S (this is necessary to avoid incorrectly invalidating an UMH PE that would use a policy where no I-PMSI is advertized for a given VRF and where only S-PMSI are used, the S-PMSI advertisement being possibly done only after the upstream PE receives a C-multicast route for (C-S, C-G)/(C-*, C-G) to be carried over the advertized S-PMSI)
- o second, the UMH candidates that advertise a PMSI bound to a tunnel that is "down" -- these will thus be used as a last resort to ensure a graceful fallback to the basic MVPN UMH selection procedures in the hypothetical case where a false negative would occur when determining the status of all tunnels

For a given downstream PE and a given VRF, the P-tunnel corresponding to a given upstream PE for a given (C-S,C-G) state is the S-PMSI tunnel advertized by that upstream PE for this (C-S,C-G) and imported into that VRF, or if there isn't any such S-PMSI, the I-PMSI tunnel advertized by that PE and imported into that VRF.

Note that this documents assumes that if a site of a given MVPN that contains C-S is dual-homed to two PEs, then all the other sites of that MVPN would have two unicast VPN routes (VPN-IPv4 or VPN-IPv6) routes to C-S, each with its own RD.

3.1. Determining the status of a tunnel

Different factors can be considered to determine the "status" of a P-tunnel and are described in the following sub-sections. The procedure proposed here also allows that all downstream PEs don't apply the same rules to define what the status of a P-tunnel is (please see Section 6), and some of them will produce a result that may be different for different downstream PEs. Thus what is called the "status" of a P-tunnel in this section, is not a characteristic of the tunnel in itself, but is the status of the tunnel, *as seen from a particular downstream PE*. Additionally, some of the following methods determine the ability of downstream PE to receive traffic on the P-tunnel and not specifically on the status of the P-tunnel itself. This could be referred to as "P-tunnel reception status", but for simplicity, we will use the terminology of P-tunnel "status" for all of these methods.

Depending on the criteria used to determine the status of a P-tunnel, there may be an interaction with another resiliency mechanism used for the P-tunnel itself, and the UMH update may happen immediately or may need to be delayed. Each particular case is covered in each separate sub-section below.

3.1.1. mVPN tunnel root tracking

A condition to consider that the status of a P-tunnel is up is that the root of the tunnel, as determined in the PMSI tunnel attribute, is reachable through unicast routing tables. In this case the downstream PE can immediately update its UMH when the reachability condition changes.

This is similar to BGP next-hop tracking for VPN routes, except that the address considered is not the BGP next-hop address, but the root address in the PMSI tunnel attribute.

If BGP next-hop tracking is done for VPN routes, and the root address of a given tunnel happens to be the same as the next-hop address in the BGP autodiscovery route advertising the tunnel, then this mechanisms may be omitted for this tunnel, as it will not bring any specific benefit.

3.1.2. PE-P Upstream link status

A condition to consider a tunnel status as up can be that the last-hop link of the P-tunnel is up.

This method should not be used when there is a fast restoration mechanism (such as MPLS FRR [RFC4090]) in place for the link.

3.1.3. P2MP RSVP-TE tunnels

For P-Tunnels of type P2MP MPLS-TE, the status of the P-Tunnel is considered up if one or more of the P2MP RSVP-TE LSPs, identified by the P-Tunnel Attribute, are in up state. The determination of whether a P2MP RSVP-TE LSP is in up state requires Path and Resv state for the LSP and is based on procedures in [RFC4875]. In this case the downstream PE can immediately update its UMH when the reachability condition changes.

When signaling state for a P2MP TE LSP is removed (e.g. if the ingress of the P2MP TE LSP sends a PathTear message) or the P2MP TE LSP changes state from up to down as determined by procedures in [RFC4875], the status of the corresponding P-Tunnel SHOULD be re-evaluated. If the P-Tunnel transitions from up to down state, the

upstream PE, that is the ingress of the P-Tunnel, SHOULD not be considered a valid UMH.

3.1.4. Leaf-initiated P-tunnels

A PE can be removed from the UMH candidate list for a given (S,G) if the P-tunnel for this S,G (I or S , depending) is leaf triggered (PIM, mLDP), but for some reason internal to the protocol the upstream one-hop branch of the tunnel from P to PE cannot be built. In this case the downstream PE can immediately update its UMH when the reachability condition changes.

3.1.5. (S,G) counter information

In cases, where the downstream node can be configured so that the maximum inter-packet time is known for all the multicast flows mapped on a P-tunnel, the local per-(C-S,C-G) traffic counter information for traffic received on this P-tunnel can be used to determine the status of the P-tunnel.

When such a procedure is used, in context where fast restoration mechanisms are used for the P-tunnels, downstream PEs should be configured to wait before updating the UMH, to let the P-tunnel restoration mechanism happen. A configurable timer MUST be provided for this purpose, and it is recommended to provide a reasonable default value for this timer.

This method can be applicable for instance when a (S,G) flow is mapped on an S-PMSI.

In cases where this mechanism is used in conjunction with Hot leaf standby, then no prior knowledge of the rate of the multicast streams is required ; downstream PEs can compare reception on the two P-tunnels to determine when one of them is down.

3.1.6. BFD Discriminator

P-tunnel status can be derived from the status of a multipoint BFD session [I-D.ietf-bfd-multipoint] whose discriminator is advertized along with an x-PMSI A-D route.

3.1.6.1. Upstream PE Procedures

When it is desired to track the P-Tunnel status using BFD, the Upstream PE MUST include the BGP-BFD Attribute in the x-PMSI A-D Route.

If a P-Tunnel is already signaled, and then it is desired to track the P-Tunnel status using BFD, x-PMSI A-D Route must be re-sent with the same attributes as before, but the BGP-BFD Attribute MUST be included.

If P-Tunnel is already signaled, and P-Tunnel status tracked using BFD and it is desired to stop tracking P-Tunnel status using BFD, then x-PMSI A-D Route MUST be re-sent with the same attributes as before, but the BGP-BFD Attribute MUST be excluded.

3.1.6.2. Downstream PE Procedures

On receiving the BFD attribute in the x-PMSI A-D Route, the Downstream PE MUST associate the received discriminator with the P-Tunnel originating from the Root PE. Once the Downstream PE start getting the BFD probes from the Upstream PE with the given discriminator, the BFD session will be declared up and will then be used to track the health of the P-Tunnel.

If the Downstream PE does not receive BFD probes for a P-Tunnel from the Upstream PE for Detection Time, the BFD session would be brought down. And, it would declare the P-tunnel associated with the discriminator as down.

Downstream PE then can then initiate a switchover of the traffic from the Primary Tunnel, to the Standby Tunnel.

When Downstream PE's P-Tunnel is already up, it receives new x-PMSI A-D Route with BGP-BFD attribute, it must accept the x-PMSI A-D Route and associate the discriminator with the P-tunnel. When the BFD probes are received with the given discriminator, the BFD session is declared up.

When Downstream PE's P-Tunnel is already up, and is tracked with BFD, and it receives new x-PMSI A-D Route without BGP-BFD attribute, it must accept the x-PMSI A-D Route the BFD session should be declared admin down. Receiver node SHOULD not switch the traffic to the Standby P-tunnel.

When such a procedure is used, in context where fast restoration mechanisms are used for the P-tunnels, leaf PEs should be configured to wait before updating the UMH, to let the P-tunnel restoration mechanism happen. A configurable timer MUST be provided for this purpose, and it is recommended to provide a reasonable default value for this timer.

3.1.6.3. BGP-BFD Attribute

This document defines and uses a new BGP attribute called the "BGP-BFD attribute". This is an optional transitive BGP attribute. The format of this attribute is defined as follows:

```

+-----+
|           Flags (1 octet)           |
+-----+
| BFD Discriminator (4 octets)       |
+-----+

```

The Flags field has the following format:

```

0 1 2 3 4 5 6 7
+---+---+---+---+
| reserved |
+---+---+---+---+

```

3.1.7. Per PE-CE link BFD Discriminator

The following approach is proposed for fast failover on PE-CE link failures, in which UMH selection for a given C-multicast route takes into account the state of a BFD session dedicated to the state of the upstream PE-CE link.

3.1.7.1. Upstream PE Procedures

For each protected PE-CE link, the upstream PE initiates a multipoint BFD session [I-D.ietf-bfd-multipoint] toward downstream PEs, with a trigger causing such a session to be torn down if the associated PE-CE link is detected as down.

For SSM groups, the upstream PE advertises a (S,G) S-PMSI A-D route or wildcard (S,*) S-PMSI A-D route for each received SSM (S,G) C-multicast route for which protection is desired. For each ASM (S,G) C-multicast route for which protection is desired, the upstream PE advertises a (S,G) S-PMSI A-D route. For each ASM (*,G) C-Multicast route for which protection is desired, the upstream PE advertises a wildcard (*,G) S-PMSI A-D route. Note that all S-PMSI A-D routes can signal the same P-Tunnel, so there is no need for a

new P-Tunnel for each S-PMSI A-D route. Multicast flows for which protection is desired is controlled by configuration/policy on the upstream PE. The protected link is the RPF PE-CE interface towards the src/RP. The upstream PE advertises the BFD discriminator of the protected link in the S-PMSI A-D route. If the route to the src/RP changes such that the RPF interface is changed to be a new PE-CE interface, then the upstream PE will update the S-PMSI A-D route with the BFD discriminator associated with the new RPF link.

3.1.7.2. Downstream PE Procedures

If an S-PMSI A-D route bound to a given C-multicast is signaled with a multipoint BFD session, then the upstream PE is considered during UMH selection for the C-multicast if and only if the corresponding BFD session is not known to be down. Whenever the BFD session goes down the Provider Tunnel will be considered down, and the downstream PE will switch to the backup Provider Tunnel. Note that the Provider Tunnel is considered down only for the C-multicast states that match to an S-PMSI A-D route which signaled the BFD discriminator of a BFD session which is down.

4. Standby C-multicast route

The procedures described below are limited to the case where the site that contains C-S is connected to exactly two PEs. The procedures require all the PEs of that MVPN to follow the single forwarder PE selection, as specified in [RFC6513]. The procedures assume that if a site of a given MVPN that contains C-S is dual-homed to two PEs, then all the other sites of that MVPN would have two unicast VPN routes (VPN-IPv4 or VPN-IPv6) routes to C-S, each with its own RD.

As long as C-S is reachable via both PEs, a given downstream PE will select one of the PEs connected to C-S as its Upstream PE with respect to C-S. We will refer to the other PE connected to C-S as the "Standby Upstream PE". Note that if the connectivity to C-S through the Primary Upstream PE becomes unavailable, then the PE will select the Standby Upstream PE as its Upstream PE with respect to C-S. When the Primary PE later becomes available, then the PE will select the Primary Upstream PE again as its Upstream PE. This is referred to as "revertive" behavior, and MUST be supported. Non-revertive behavior would refer to the behavior of continuing to select the backup PE as the UMH even after the Primary has come up. This non-revertive behavior can also be optionally supported by an implementation and would be enabled through some configuration.

For readability, in the following sub-sections, the procedures are described for BGP C-multicast Source Tree Join routes, but they apply equally to BGP C-multicast Shared Tree Join routes failover for the

case where the customer RP is dual-homed (substitute "C-RP" to "C-S").

4.1. Downstream PE behavior

When a (downstream) PE connected to some site of an MVPN needs to send a C-multicast route (C-S, C-G), then following the procedures specified in Section "Originating C-multicast routes by a PE" of [RFC6514] the PE sends the C-multicast route with RT that identifies the Upstream PE selected by the PE originating the route. As long as C-S is reachable via the Primary Upstream PE, the Upstream PE is the Primary Upstream PE. If C-S is reachable only via the Standby Upstream PE, then the Upstream PE is the Standby Upstream PE.

If C-S is reachable via both the Primary and the Standby Upstream PE, then in addition to sending the C-multicast route with an RT that identifies the Primary Upstream PE, the PE also originates and sends a C-multicast route with an RT that identifies the Standby Upstream PE. This route, that has the semantic of being a 'standby' C-multicast route, is further called a "Standby BGP C-multicast route", and is constructed as follows:

- o the NLRI is constructed as the original C-multicast route, except that the RD is the same as if the C-multicast route was built using the standby PE as the UMH (it will carry the RD associated to the unicast VPN route advertized by the standby PE for S)
- o SHOULD carry the "Standby PE" BGP Community (this is a new BGP Community, see Section 7)

The normal and the standby C-multicast routes must have their Local Preference attribute adjusted so that, if two C-multicast routes with same NLRI are received by a BGP peer, one carrying the "Standby PE" attribute and the other one *not* carrying the "Standby PE" community, then preference is given to the one *not* carrying the "Standby PE" attribute. Such a situation can happen when, for instance due to transient unicast routing inconsistencies, two different downstream PEs consider different upstream PEs to be the primary one ; in that case, without any precaution taken, both upstream PEs would process a standby C-multicast route and possibly stop forwarding at the same time. For this purpose, routes that carry the "Standby PE" BGP Community MUST have the LOCAL_PREF attribute set to zero.

Note that, when a PE advertizes such a Standby C-multicast join for an (S,G) it must join the corresponding P-tunnel.

If at some later point the local PE determines that C-S is no longer reachable through the Primary Upstream PE, the Standby Upstream PE becomes the Upstream PE, and the local PE re-sends the C-multicast route with RT that identifies the Standby Upstream PE, except that now the route does not carry the Standby PE BGP Community (which results in replacing the old route with a new route, with the only difference between these routes being the presence/absence of the Standby PE BGP Community).

4.2. Upstream PE behavior

When a PE receives a C-multicast route for a particular (C-S, C-G), and the RT carried in the route results in importing the route into a particular VRF on the PE, if the route carries the Standby PE BGP Community, then the PE performs as follows:

when the PE determines that C-S is not reachable through some other PE, the PE SHOULD install VRF PIM state corresponding to this Standby BGP C-multicast route (the result will be that a PIM Join message will be sent to the CE towards C-S, and that the PE will receive (C-S,C-G) traffic), and the PE SHOULD forward (C-S, C-G) traffic received by the PE to other PEs through a P-tunnel rooted at the PE.

Furthermore, irrespective of whether C-S carried in that route is reachable through some other PE:

- a) based on local policy, as soon as the PE receives this Standby BGP C-multicast route, the PE MAY install VRF PIM state corresponding to this BGP Source Tree Join route (the result will be that Join messages will be sent to the CE toward C-S, and that the PE will receive (C-S,C-G) traffic)
- b) based on local policy, as soon as the PE receives this Standby BGP C-multicast route, the PE MAY forward (C-S, C-G) traffic to other PEs through a P-tunnel independently of the reachability of C-S through some other PE. [note that this implies also doing (a)]

Doing neither (a), nor (b) for a given (C-S,C-G) is called "cold root standby".

Doing (a) but not (b) for a given (C-S,C-G) is called "warm root standby".

Doing (b) (which implies also doing (a)) for a given (C-S,C-G) is called "hot root standby".

Note that, if an upstream PE uses an S-PMSI only policy, it shall advertise an S-PMSI for an (S,G) as soon as it receives a C-multicast route for (S,G), normal or Standby ; i.e. it shall not wait for receiving a non-Standby C-multicast route before advertising the corresponding S-PMSI.

Section 9.3.2 of [RFC6514], describes the procedures of sending a Source-Active A-D result as a result of receiving the C-multicast route. These procedures should be followed for both the normal and Standby C-multicast routes.

4.3. Reachability determination

The standby PE can use the following information to determine that C-S can or cannot be reached through the primary PE:

- o presence/absence of a unicast VPN route toward C-S
- o supposing that the standby PE is an egress of the tunnel rooted at the Primary PE, the standby PE can determine the reachability of C-S through the Primary PE based on the status of this tunnel, determined thanks to the same criteria as the ones described in Section 3.1 (without using the UMH selection procedures of Section 3)
- o other mechanisms MAY be used

4.4. Inter-AS

If the non-segmented inter-AS approach is used, the procedures in section 4 can be applied.

When multicast VPNs are used in a inter-AS context with the segmented inter-AS approach described in section 8.2 of [RFC6514], the procedures in this section can be applied.

A pre-requisite for the procedures described below to be applied for a source of a given MVPN is:

- o that any PE of this MVPN receives two Inter-AS I-PMSI auto-discovery routes advertised by the AS of the source (or more)
- o that these Inter-AS I-PMSI autodiscovery routes have distinct Route Distinguishers (as described in item "(2)" of section 9.2 of [RFC6514]).

As an example, these conditions will be satisfied when the source is dual homed to an AS that connects to the receiver AS through two ASBR using auto-configured RDs.

4.4.1. Inter-AS procedures for downstream PEs, ASBR fast failover

The following procedure is applied by downstream PEs of an AS, for a source S in a remote AS.

Additionally to choosing an Inter-AS I-PMSI autodiscovery route advertized from the AS of the source to construct a C-multicast route, as described in section 11.1.3 [RFC6514] a downstream PE will choose a second Inter-AS I-PMSI autodiscovery route advertized from the AS of the source and use this route to construct and advertise a Standby C-multicast route (C-multicast route carrying the Standby extended community) as described in Section 4.1.

4.4.2. Inter-AS procedures for ASBRs

When an upstream ASBR receives a C-multicast route, and at least one of the RTs of the route matches one of the ASBR Import RT, the ASBR locates an Inter-AS I-PMSI A-D route whose RD and Source AS matches the RD and Source AS carried in the C-multicast route. If the match is found, and C-multicast route carries the Standby PE BGP Community, then the ASBR performs as follows:

- o if the route was received over iBGP ; the route is expected to have a LOCAL_PREF attribute set to zero and it should be re-advertized in eBGP with a MED attribute (MULTI_EXIT_DISC) set to the highest possible value (0xffff)
- o if the route was received over eBGP ; the route is expected to have a MED attribute set of 0xffff and should be re-advertized in iBGP with a LOCAL_PREF attribute set to zero

Other ASBR procedures are applied without modification.

5. Hot leaf standby

The mechanisms defined in sections Section 4 and Section 3 can be used together as follows.

The principle is that, for a given VRF (or possibly only for a given C-S,C-G):

- o downstream PEs advertise a Standby BGP C-multicast route (based on Section 4)

- o upstream PEs use the "hot standby" optional behavior and thus will forward traffic for a given multicast state as soon as they have whether a (primary) BGP C-multicast route or a Standby BGP C-multicast route for that state (or both)
- o downstream PEs accept traffic from the primary or standby tunnel, based on the status of the tunnel (based on Section 3)

Other combinations of the mechanisms proposed in Section 4) and Section 3 are for further study.

Note that the same level of protection would be achievable with a simple C-multicast Source Tree Join route advertized to both the primary and secondary upstream PEs (carrying as Route Target extended communities, the values of the VRF Route Import attribute of each VPN route from each upstream PEs). The advantage of using the Standby semantic for is that, supposing that downstream PEs always advertise a Standby C-multicast route to the secondary upstream PE, it allows to choose the protection level through a change of configuration on the secondary upstream PE, without requiring any reconfiguration of all the downstream PEs.

6. Duplicate packets

Multicast VPN specifications [RFC6513] impose that a PE only forwards to CEs the packets coming from the expected upstream PE (Section 9.1).

We highlight the reader's attention to the fact that the respect of this part of multicast VPN specifications is especially important when two distinct upstream PEs are susceptible to forward the same traffic on P-tunnels at the same time in steady state. This will be the case when "hot root standby" mode is used (Section 4), and which can also be the case if procedures of Section 3 are used and (a) the rules determining the status of a tree are not the same on two distinct downstream PEs or (b) the rule determining the status of a tree depend on conditions local to a PE (e.g. the PE-P upstream link being up).

7. IANA Considerations

Allocation is expected from IANA for the BGP "Standby PE" community. (TBC)

[Note to RFC Editor: this section may be removed on publication as an RFC.]

8. Security Considerations

9. Acknowledgements

The authors want to thank Greg Reaume, Eric Rosen, and Jeffrey Zhang for their review and useful feedback.

10. Contributor Addresses

Below is a list of other contributing authors in alphabetical order:

Rahul Aggarwal
Arktan

Email: raggarwa_1@yahoo.com

Nehal Bhau
Alcatel-Lucent, Inc.
701 E Middlefield Rd
Mountain View, CA 94043
USA

Email: Nehal.Bhau@alcatel-lucent.com

Clayton Hassen
Bell Canada
2955 Virtual Way
Vancouver
CANADA

Email: Clayton.Hassen@bell.ca

Wim Henderickx
Alcatel-Lucent
Copernicuslaan 50
Antwerp 2018
Belgium

Email: wim.henderickx@alcatel-lucent.com

Pradeep Jain
Alcatel-Lucent, Inc.
701 E Middlefield Rd
Mountain View, CA 94043
USA

Email: pradeep.jain@alcatel-lucent.com

Jayant Kotalwar
Alcatel-Lucent, Inc.
701 E Middlefield Rd
Mountain View, CA 94043
USA

Email: Jayant.Kotalwar@alcatel-lucent.com

Praveen Muley
Alcatel-Lucent
701 East Middlefield Rd
Mountain View, CA 94043
U.S.A.

Email: praveen.muley@alcatel-lucent.com

Ray (Lei) Qiu
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
U.S.A.

Email: rqiujuniper.net

Yakov Rekhter
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
U.S.A.

Email: yakov@juniper.net

Kanwar Singh
Alcatel-Lucent, Inc.
701 E Middlefield Rd
Mountain View, CA 94043
USA

Email: kanwar.singh@alcatel-lucent.com

11. References

11.1. Normative References

- [I-D.ietf-bfd-multipoint]
Katz, D., Ward, D., and S. Pallagatti, "BFD for Multipoint Networks", draft-ietf-bfd-multipoint-08 (work in progress), April 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4875] Aggarwal, R., Papadimitriou, D., and S. Yasukawa, "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007.
- [RFC6513] Aggarwal, R., Bandi, S., Cai, Y., Morin, T., Rekhter, Y., Rosen, E., Wijnands, I., and S. Yasukawa, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.
- [RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.

11.2. Informative References

- [RFC4090] Pan, P., Swallow, G., and A. Atlas, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.
- [RFC7431] Karan, A., Filsfils, C., Wijnands, IJ., and B. Decraene, "Multicast-only Fast Re-Route", RFC 7431, August 2015.

Authors' Addresses

Thomas Morin (editor)
Orange
2, avenue Pierre Marzin
Lannion 22307
France

Email: thomas.morin@orange-ftgroup.com

Robert Kebler (editor)
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
U.S.A.

Email: rkebler@juniper.net

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 25, 2021

T. Morin, Ed.
Orange
R. Kebler, Ed.
Juniper Networks
G. Mirsky, Ed.
ZTE Corp.
January 21, 2021

Multicast VPN Fast Upstream Failover
draft-ietf-bess-mvpn-fast-failover-15

Abstract

This document defines Multicast Virtual Private Network (VPN) extensions and procedures that allow fast failover for upstream failures by allowing downstream Provider Edges (PEs) to consider the status of Provider-Tunnels (P-tunnels) when selecting the Upstream PE for a VPN multicast flow. The fast failover is enabled by using RFC 8562 Bidirectional Forwarding Detection (BFD) for Multipoint Networks and the new BGP Attribute - BFD Discriminator. Also, the document introduces a new BGP Community, Standby PE, extending BGP Multicast VPN routing so that a C-multicast route can be advertised toward a Standby Upstream PE.

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 July 25, 2021.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction	3
2.	Conventions used in this document	4
2.1.	Requirements Language	4
2.2.	Terminology	4
2.3.	Acronyms	4
3.	UMH Selection Based on Tunnel Status	5
3.1.	Determining the Status of a Tunnel	6
3.1.1.	MVPN Tunnel Root Tracking	7
3.1.2.	PE-P Upstream Link Status	7
3.1.3.	P2MP RSVP-TE Tunnels	7
3.1.4.	Leaf-initiated P-tunnels	8
3.1.5.	(C-S, C-G) Counter Information	8
3.1.6.	BFD Discriminator Attribute	9
3.1.7.	Per PE-CE Link BFD Discriminator	13
3.1.8.	Operational Considerations for Monitoring P-Tunnel's Status	13
4.	Standby C-multicast Route	14
4.1.	Downstream PE Behavior	15
4.2.	Upstream PE Behavior	16
4.3.	Reachability Determination	17
4.4.	Inter-AS	18
4.4.1.	Inter-AS Procedures for downstream PEs, ASBR Fast Failover	18
4.4.2.	Inter-AS Procedures for ASBRs	19
5.	Hot Root Standby	19
6.	Duplicate Packets	20
7.	IANA Considerations	20
7.1.	Standby PE Community	20
7.2.	BFD Discriminator	20
7.3.	BFD Discriminator Optional TLV Type	21
8.	Security Considerations	22
9.	Acknowledgments	22
10.	Contributor Addresses	22
11.	References	24
11.1.	Normative References	24
11.2.	Informative References	26

Authors' Addresses	26
------------------------------	----

1. Introduction

It is assumed that the reader is familiar with the workings of multicast MPLS/BGP IP VPNs as described in [RFC6513] and [RFC6514].

In the context of multicast in BGP/MPLS VPNs [RFC6513], it is desirable to provide mechanisms allowing fast recovery of connectivity on different types of failures. This document addresses failures of elements in the provider network that are upstream of PEs connected to VPN sites with receivers.

Section 3 describes local procedures allowing an egress PE (a PE connected to a receiver site) to take into account the status of P-tunnels to determine the Upstream Multicast Hop (UMH) for a given (C-S, C-G). One of the optional methods uses [RFC8562] and the new BGP Attribute - BFD Discriminator. None of these methods provide a "fast failover" solution when used alone, but can be used together with the mechanism described in Section 4 for a "fast failover" solution.

Section 4 describes an optional BGP extension, a new Standby PE Community. that can speed up failover by not requiring any multicast VPN (MVPN) routing message exchange at recovery time.

Section 5 describes a "hot leaf standby" mechanism that can be used to improve failover time in MVPN. The approach combines mechanisms defined in Section 3 and Section 4, and has similarities with the solution described in [RFC7431] to improve failover times when PIM routing is used in a network given some topology and metric constraints.

The procedures described in this document are optional and allow an operator to provide protection for multicast services in BGP/MPLS IP VPNs. An operator would enable these mechanisms using a method discussed in Section 3 combined with the redundancy provided by a standby PE connected to the multicast flow source. PEs that support these mechanisms would converge faster and thus provide a more stable multicast service. In the case that a BGP implementation does not recognize or is configured not to support the extensions defined in this document, the implementation will continue to provide the multicast service, as described in [RFC6513].

2. Conventions used in this document

2.1. Requirements Language

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

2.2. Terminology

The terminology used in this document is the terminology defined in [RFC6513] and [RFC6514].

The term 'upstream' (lower case) throughout this document refers to links and nodes that are upstream to a PE connected to VPN sites with receivers of a multicast flow.

The term 'Upstream' (capitalized) throughout this document refers to a PE or an Autonomous System Border Router (ASBR) at which (S,G) or (*,G) data packets enter the VPN backbone or the local AS when traveling through the VPN backbone.

2.3. Acronyms

PMSI: P-Multicast Service Interface

I-PMSI: Inclusive PMSI

S-PMSI: Selective PMSI

x-PMSI: Either an I-PMSI or an S-PMSI

P-tunnel: Provider-Tunnels

UMH: Upstream Multicast Hop

VPN: Virtual Private Network

MVPN: Multicast VPN

RD: Route Distinguisher

RP: Rendezvous Point

NLRI: Network Layer Reachability Information

VRF: VPN Routing and Forwarding Table

MED: Multi-Exit Discriminator

P2MP: Point-to-Multipoint

3. UMH Selection Based on Tunnel Status

Section 5.1 of [RFC6513] describes procedures used by a multicast VPN downstream PE to determine the Upstream Multicast Hop (UMH) for a given (C-S, C-G).

For a given downstream PE and a given VRF, the P-tunnel corresponding to a given Upstream PE for a given (C-S, C-G) state is the S-PMSI tunnel advertised by that Upstream PE for this (C-S, C-G) and imported into that VRF, or if there isn't any such S-PMSI, the I-PMSI tunnel advertised by that PE and imported into that VRF.

The procedure described here is an optional procedure that is based on a downstream PE taking into account the status of P-tunnels rooted at each possible Upstream PE, for including or not including each given PE in the list of candidate UMHs for a given (C-S, C-G) state. If it is not possible to determine whether a P-tunnel's current status is Up, the state shall be considered "not known to be Down", and it may be treated as if it is Up so that attempts to use the tunnel are acceptable. The result is that, if a P-tunnel is Down (see Section 3.1), the PE that is the root of the P-tunnel will not be considered for UMH selection. This will result in the downstream PE failing over to use the next Upstream PE in the list of candidates. Some downstream PEs could arrive at a different conclusion regarding the tunnel's state because the failure impacts only a subset of branches. Because of that, the procedures of Section 9.1.1 of [RFC6513] are applicable when using I-PMSI P-tunnels. That document is a foundation for this document, and its processes all apply here.

There are three options specified in Section 5.1 of [RFC6513] for a downstream PE to select an Upstream PE.

- o The first two options select the Upstream PE from a candidate PE set either based on an IP address or a hashing algorithm. When used together with the optional procedure of considering the P-tunnel status as in this document, a candidate Upstream PE is included in the set if it either:
 - A. advertises an x-PMSI bound to a tunnel, where the specified tunnel's state is not known to be Down, or,

- B. does not advertise any x-PMSI applicable to the given (C-S, C-G) but has associated a VRF Route Import BGP Extended Community to the unicast VPN route for S. That is necessary to avoid incorrectly invalidating a UMH PE that would use a policy where no I-PMSI is advertised for a given VRF and where only S-PMSI are used. The S-PMSI can be advertised only after the Upstream PE receives a C-multicast route for (C-S, C-G)/(C-*, C-G) to be carried over the advertised S-PMSI.

If the resulting candidate set is empty, then the procedure is repeated without considering the P-tunnel status.

- o The third option uses the installed UMH Route (i.e., the "best" route towards the C-root) as the Selected UMH Route, and its originating PE is the selected Upstream PE. With the optional procedure of considering P-tunnel status as in this document, the Selected UMH Route is the best one among those whose originating PE's P-tunnel is not "down". If that does not exist, the installed UMH Route is selected regardless of the P-tunnel status.

3.1. Determining the Status of a Tunnel

Different factors can be considered to determine the "status" of a P-tunnel and are described in the following sub-sections. The optional procedures described in this section also handle the case when the downstream PEs do not all apply the same rules to define what the status of a P-tunnel is (please see Section 6), and some of them will produce a result that may be different for different downstream PEs. Thus, the "status" of a P-tunnel in this section is not a characteristic of the tunnel in itself, but is the tunnel status, as seen from a particular downstream PE. Additionally, some of the following methods determine the ability of a downstream PE to receive traffic on the P-tunnel and not specifically on the status of the P-tunnel itself. That could be referred to as "P-tunnel reception status", but for simplicity, we will use the terminology of P-tunnel "status" for all of these methods.

Depending on the criteria used to determine the status of a P-tunnel, there may be an interaction with another resiliency mechanism used for the P-tunnel itself, and the UMH update may happen immediately or may need to be delayed. Each particular case is covered in each separate sub-section below.

An implementation may support any combination of the methods described in this section and provide a network operator with control to choose which one to use in the particular deployment.

3.1.1. MVPN Tunnel Root Tracking

When determining if the status of a P-tunnel is Up, a condition to consider is whether the root of the tunnel, as specified in the x-PMSI Tunnel attribute, is reachable through unicast routing tables. In this case, the downstream PE can immediately update its UMH when the reachability condition changes.

That is similar to BGP next-hop tracking for VPN routes, except that the address considered is not the BGP next-hop address but the root address in the x-PMSI Tunnel attribute. BGP next-hop tracking monitors BGP next-hop address changes in the routing table. In general, when a change is detected, it performs a next-hop scan to find if any of the next hops in the BGP table is affected and updates it accordingly.

If BGP next-hop tracking is done for VPN routes and the root address of a given tunnel happens to be the same as the next-hop address in the BGP A-D Route advertising the tunnel, then checking, in unicast routing tables, whether the tunnel root is reachable, will be unnecessary duplication and thus will not bring any specific benefit.

3.1.2. PE-P Upstream Link Status

When determining if the status of a P-tunnel is Up, a condition to consider is whether the last-hop link of the P-tunnel is Up. Conversely, if the last-hop link of the P-tunnel is Down, then this can be taken as an indication that the P-tunnel is Down.

Using this method when a fast restoration mechanism (such as MPLS FRR [RFC4090]) is in place for the link requires careful consideration and coordination of defect detection intervals for the link and the tunnel. When using multi-layer protection, particular consideration must be given to the interaction of defect detections at different network layers. It is recommended to use longer detection intervals at the higher layers. Some recommendations suggest using a multiplier of 3 or larger, e.g., 10 msec detection for the link failure detection and at least 100 msec for the tunnel failure detection. In many cases, it is not practical to use both protection methods simultaneously because uncorrelated timers might cause unnecessary switchovers and destabilize the network.

3.1.3. P2MP RSVP-TE Tunnels

For P-tunnels of type P2MP MPLS-TE, the status of the P-tunnel is considered Up if the sub-LSP to this downstream PE is in the Up state. The determination of whether a P2MP RSVP-TE LSP is in the Up state requires Path and Resv state for the LSP and is based on

procedures specified in [RFC4875]. As a result, the downstream PE can immediately update its UMH when the reachability condition changes.

When using this method and if the signaling state for a P2MP TE LSP is removed (e.g., if the ingress of the P2MP TE LSP sends a PathTear message) or the P2MP TE LSP changes state from Up to Down as determined by procedures in [RFC4875], the status of the corresponding P-tunnel MUST be re-evaluated. If the P-tunnel transitions from Up to Down state, the Upstream PE that is the ingress of the P-tunnel MUST NOT be considered as a valid candidate UMH.

3.1.4. Leaf-initiated P-tunnels

An Upstream PE MUST be removed from the UMH candidate list for a given (C-S, C-G) if the P-tunnel (I-PMSI or S-PMSI) for this (S, G) is leaf-triggered (PIM, mLDP), but for some reason, internal to the protocol, the upstream one-hop branch of the tunnel from P to PE cannot be built. As a result, the downstream PE can immediately update its UMH when the reachability condition changes.

3.1.5. (C-S, C-G) Counter Information

In cases where the downstream node can be configured so that the maximum inter-packet time is known for all the multicast flows mapped on a P-tunnel, the local per-(C-S, C-G) traffic counter information for traffic received on this P-tunnel can be used to determine the status of the P-tunnel.

When such a procedure is used, in the context where fast restoration mechanisms are used for the P-tunnels, a configurable timer MUST be set on the downstream PE to wait before updating the UMH to let the P-tunnel restoration mechanism execute its actions. Determining that a tunnel is probably down by waiting for enough packets to fail to arrive as expected is a heuristic and operational matter that depends on the maximum inter-packet time. A timeout of three seconds is a generally suitable default waiting period to ascertain that the tunnel is down, though other values would be needed for atypical conditions.

In cases where this mechanism is used in conjunction with the method described in Section 5, no prior knowledge of the rate or maximum inter-packet time on the multicast streams is required; downstream PEs can periodically compare actual packet reception statistics on the two P-tunnels to determine when one of them is down. The detailed specification of this mechanism is outside the scope of this document.

3.1.6. BFD Discriminator Attribute

The P-tunnel status may be derived from the status of a multipoint BFD session [RFC8562] whose discriminator is advertised along with an x-PMSI A-D Route. A P2MP BFD session can be instantiated using a mechanism other than the BFD Discriminator attribute, e.g., MPLS LSP Ping ([I-D.mirsky-mppls-p2mp-bfd]). The description of these methods is outside the scope of this document.

This document defines the format and ways of using a new BGP attribute called the "BFD Discriminator". It is an optional transitive BGP attribute. Thus it is expected that an implementation that does not recognize or is configured not to support this attribute, as if the attribute was unrecognized, follows procedures defined for optional transitive path attributes in Section 5 of [RFC4271]. In Section 7.2, IANA is requested to allocate the codepoint value (TBA2). The format of this attribute is shown in Figure 1.

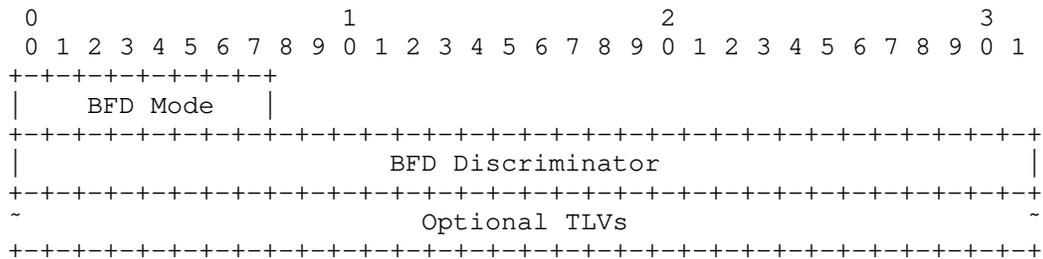


Figure 1: Format of the BFD Discriminator Attribute

Where:

BFD Mode field is one octet long. This specification defines the P2MP BFD Session as value 1 Section 7.2.

BFD Discriminator field is four octets long.

Optional TLVs is the optional variable-length field that MAY be used in the BFD Discriminator attribute for future extensions. TLVs MAY be included in a sequential or nested manner. To allow for TLV nesting, it is advised to define a new TLV as a variable-length object. Figure 2 presents the Optional TLV format TLV that consists of:

- * Type - a one-octet-long field that characterizes the interpretation of the Value field (Section 7.3)
- * Length - a one-octet-long field equal to the length of the Value field in octets
- * Value - a variable-length field.

All multibyte fields in TLVs defined in this specification are in network byte order.

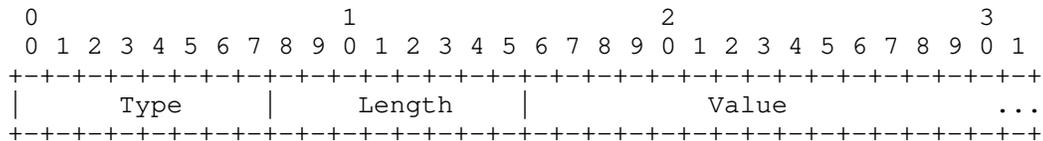


Figure 2: Format of the Optional TLV

An optional Source IP Address TLV is defined in this document. The Source IP Address TLV MUST be used when the value of the BFD Mode field's value is P2MP BFD Session. The BFD Discriminator attribute that does not include the Source IP Address TLV MUST be handled according to the "attribute discard" approach, as defined in [RFC7606]. For the Source IP Address TLV fields are set as follows:

- o The Type field is set to 1 Section 7.3.
- o The Length field is 4 for the IPv4 address family and 16 for the IPv6 address family. The TLV is considered malformed if the field is set to any other value.
- o The Value field contains the address associated with the MultipointHead of the P2MP BFD session.

The BFD Discriminator attribute MUST be considered malformed if its length is smaller than 11 octets or if Optional TLVs are present, but not well-formed. If the attribute is deemed to be malformed, the UPDATE message SHALL be handled using the approach of Attribute Discard per [RFC7606].

3.1.6.1. Upstream PE Procedures

To enable downstream PEs to track the P-tunnel status using a point-to-multipoint (P2MP) BFD session the Upstream PE:

- o MUST initiate the BFD session and set `bfd.SessionType = MultipointHead` as described in [RFC8562];
- o when transmitting BFD Control packets MUST set the IP destination address of the inner IP header to the internal loopback address 127.0.0.1/32 for IPv4 [RFC1122]. For IPv6, it MUST use the loopback address `::1/128` [RFC4291].
- o MUST use the IP address included in the Source IP Address TLV of the BFD Discriminator attribute as the source IP address when transmitting BFD Control packets;
- o MUST include the BFD Discriminator attribute in the x-PMSI A-D Route with the value set to My Discriminator value;
- o MUST periodically transmit BFD Control packets over the x-PMSI P-tunnel after the P-tunnel is considered established. Note that the methods to declare that a P-tunnel has been established are outside the scope of this specification.

If the tracking of the P-tunnel by using a P2MP BFD session is enabled after the x-PMSI A-D Route has been already advertised, the x-PMSI A-D Route MUST be re-sent with the only change between the previous advertisement and the new advertisement to be the inclusion of the BFD Discriminator attribute.

If the x-PMSI A-D Route is advertised with P-tunnel status tracked using the P2MP BFD session, and it is desired to stop tracking P-tunnel status using BFD, then:

- o x-PMSI A-D Route MUST be re-sent with the only change between the previous advertisement and the new advertisement be the exclusion of the BFD Discriminator attribute;
- o the P2MP BFD session MUST be deleted. The session MAY be deleted after some configurable delay, which should have a reasonable default.

3.1.6.2. Downstream PE Procedures

Upon receiving the BFD Discriminator attribute in the x-PMSI A-D Route, the downstream PE:

- o MUST associate the received BFD Discriminator value with the P-tunnel originating from the Upstream PE and the IP address of the Upstream PE;
- o MUST create a P2MP BFD session and set `bfd.SessionType = MultipointTail` as described in [RFC8562];
- o to properly demultiplex BFD session MUST use:

the IP address in the Source IP Address TLV included the BFD Discriminator attribute in the x-PMSI A-D Route;

the value of the BFD Discriminator field in the BFD Discriminator attribute;

the x-PMSI Tunnel Identifier [RFC6514] the BFD Control packet was received on.

After the state of the P2MP BFD session is up, i.e., `bfd.SessionState == Up`, the session state will then be used to track the health of the P-tunnel.

According to [RFC8562], if the downstream PE receives Down or AdminDown in the State field of the BFD Control packet or associated with the BFD session Detection Timer associated with the BFD session expires, the BFD session is down, i.e., `bfd.SessionState == Down`. When the BFD session state is Down, then the P-tunnel associated with the BFD session MUST be considered down. If the site that contains C-S is connected to two or more PEs, a downstream PE will select one as its Primary Upstream PE, while others are considered as Standby Upstream PEs. In such a scenario, when the P-tunnel is considered down, the downstream PE MAY initiate a switchover of the traffic from the Primary Upstream PE to the Standby Upstream PE only if the Standby Upstream PE is deemed to be in the Up state. That MAY be determined from the state of a P2MP BFD session with the Standby Upstream PE as the MultipointHead.

If the downstream PE's P-tunnel is already established when the downstream PE receives the new x-PMSI A-D Route with BFD Discriminator attribute, the downstream PE MUST associate the value of BFD Discriminator field with the P-tunnel and follow procedures listed above in this section if and only if the x-PMSI A-D Route was properly processed as per [RFC6514], and the BFD Discriminator attribute was validated.

If the downstream PE's P-tunnel is already established, its state being monitored by the P2MP BFD session set up using the BFD Discriminator attribute, and the downstream PE receives the new

x-PMSI A-D Route without the BFD Discriminator attribute, and the x-PMSI A-D Route was processed without any error as per the relevant specifications, the downstream PE:

- o MUST stop processing BFD Control packets for this P2MP BFD session;
- o the P2MP BFD session associated with the P-tunnel MUST be deleted. The session MAY be deleted after some configurable delay, which should have a reasonable default.
- o MUST NOT switch the traffic to the Standby Upstream PE.

3.1.7. Per PE-CE Link BFD Discriminator

The following approach is defined in response to the detection by the Upstream PE of a PE-CE link failure. Even though the provider tunnel is still up, it is desired for the downstream PEs to switch to a backup Upstream PE. To achieve that, if the Upstream PE detects that its PE-CE link fails, it MUST set the bfd.LocalDiag of the P2MP BFD session to Concatenated Path Down or Reverse Concatenated Path Down (per Section 6.8.17 [RFC5880]), unless it switches to a new PE-CE link within the time of bfd.DesiredMinTxInterval for the P2MP BFD session (in that case, the Upstream PE will start tracking the status of the new PE-CE link). When a downstream PE receives that bfd.LocalDiag code, it treats it as if the tunnel itself failed and tries to switch to a backup PE.

3.1.8. Operational Considerations for Monitoring P-Tunnel's Status

Several methods to monitor the status of a P-tunnel are described in Section 3.1.

Tracking the root of an MVPN (Section 3.1.1) concludes about the status of a P-tunnel based on the control plane information. Because, in general, the MPLS data plane is not fate-sharing with the control plane, this method might produce false positive or false negative alarms. For example, resulting in tunnels that considered as being up but are not able to reach the root, or ones that are declared down prematurely. On the other hand, because BGP next-hop tracking is broadly supported and deployed, this method might be the easiest to deploy.

Method described in Section 3.1.2 monitors the state of the data plane but only for an egress P-PE link of a P-tunnel. As a result, network failures that affect upstream links might not be detected using this method and the MVPN convergence would be determined by the convergence of the BGP control plane.

Using the state change of a P2MP RSVP-TE LSP as the trigger to re-evaluate the status of the P-tunnel (Section 3.1.3) relies on the mechanism used to monitor the state of the P2MP LSP.

The method described in Section 3.1.4 is simple and is safe from causing false alarms, e.g., considering a tunnel operationally up even though its data path has a defect or, conversely, declaring a tunnel failed when it is unaffected. But the method applies to a sub-set of MVPNs, those that use the leaf-triggered x-PMSI tunnels.

Though some MVPN might be used to provide a multicast service with predictable interpacket interval (Section 3.1.5), the number of such cases seem limited.

Monitoring the status of a P-tunnel using p2mp BFD session (Section 3.1.6) may produce the most accurate and expedient failure notification of all monitoring methods discussed. On the other hand, it requires careful consideration of the additional load of BFD onto network and PE nodes. Operators should consider the rate of BFD Control packets transmitted by root PEs combined with the number of such PEs in the network. In addition, the number of P2MP BFD sessions per PE determines the amount of state information that a PE maintains.

4. Standby C-multicast Route

The procedures described below are limited to the case where the site that contains C-S is connected to two or more PEs, though, to simplify the description, the case of dual-homing is described. In the case where more than two PEs are connected to the C-s site, selection of the Standby PE can be performed using one of the methods of selecting a UMH. Details of the selection are outside the scope of this document. The procedures require all the PEs of that MVPN to follow the same UMH selection procedure, as specified in [RFC6513], whether the PE selected based on its IP address, the hashing algorithm described in section 5.1.3 of [RFC6513], or Installed UMH Route. The consistency of the UMH selection method used among all PEs is expected to be provided by the management plane. The procedures assume that if a site of a given MVPN that contains C-S is dual-homed to two PEs, then all the other sites of that MVPN would have two unicast VPN routes (VPN-IPv4 or VPN-IPv6) to C-S, each with its own RD.

As long as C-S is reachable via both PEs, a given downstream PE will select one of the PEs connected to C-S as its Upstream PE for C-S. We will refer to the other PE connected to C-S as the "Standby Upstream PE". Note that if the connectivity to C-S through the Primary Upstream PE becomes unavailable, then the PE will select the

Standby Upstream PE as its Upstream PE for C-S. When the Primary PE later becomes available, then the PE will select the Primary Upstream PE again as its Upstream PE. Such behavior is referred to as "revertive" behavior and MUST be supported. Non-revertive behavior refers to the behavior of continuing to select the backup PE as the UMH even after the Primary has come up. This non-revertive behavior MAY also be supported by an implementation and would be enabled through some configuration. Selection of the behavior, revertive or non-revertive, is an operational issue, but it MUST be consistent on all PEs in the given MVPN. While revertive is considered the default behavior, there might be cases where the switchover to the standby tunnel does not affect other services and provides the required quality of service. An operator might use non-revertive behavior to avoid unnecessary in such case switchover and thus minimize disruption to the multicast service.

For readability, in the following sub-sections, the procedures are described for BGP C-multicast Source Tree Join routes, but they apply equally to BGP C-multicast Shared Tree Join routes for the case where the customer RP is dual-homed (substitute "C-RP" to "C-S").

4.1. Downstream PE Behavior

When a (downstream) PE connected to some site of an MVPN needs to send a C-multicast route (C-S, C-G), then following the procedures specified in Section 11.1 of [RFC6514], the PE sends the C-multicast route with an RT that identifies the Upstream PE selected by the PE originating the route. As long as C-S is reachable via the Primary Upstream PE, the Upstream PE is the Primary Upstream PE. If C-S is reachable only via the Standby Upstream PE, then the Upstream PE is the Standby Upstream PE.

If C-S is reachable via both the Primary and the Standby Upstream PE, then in addition to sending the C-multicast route with an RT that identifies the Primary Upstream PE, the downstream PE also originates and sends a C-multicast route with an RT that identifies the Standby Upstream PE. The route that has the semantics of being a "standby" C-multicast route is further called a "Standby BGP C-multicast route", and is constructed as follows:

- o the NLRI is constructed as the C-multicast route with an RT that identifies the Primary Upstream PE, except that the RD is the same as if the C-multicast route was built using the Standby Upstream PE as the UMH (it will carry the RD associated to the unicast VPN route advertised by the Standby Upstream PE for S and a Route Target derived from the Standby Upstream PE's UMH route's VRF RT Import EC);

- o MUST carry the "Standby PE" BGP Community (this is a new BGP Community. Section 7.1 requested IANA to allocate value TBA1).

The Local Preference attribute of the normal and the standby C-multicast route needs to be adjusted. so that, if a BGP peer receives two C-multicast routes with the same NLRI, one carrying the "Standby PE" community and the other one not carrying the "Standby PE" community, then preference is given to the one not carrying the "Standby PE" community. Such a situation can happen when, for instance, due to transient unicast routing inconsistencies or lack of support of the Standby PE community, two different downstream PEs consider different Upstream PEs to be the primary one. In that case, without any precaution taken, both Upstream PEs would process a standby C-multicast route and possibly stop forwarding at the same time. For this purpose, routes that carry the Standby PE BGP Community must have the LOCAL_PREF attribute set to the value lower than the value specified as the LOCAL_PREF attribute for the route that does not carry the Standby PE BGP Community. The value of zero is RECOMMENDED.

Note that, when a PE advertises such a Standby C-multicast join for a (C-S, C-G) it MUST join the corresponding P-tunnel.

If, at some later point, the PE determines that C-S is no longer reachable through the Primary Upstream PE, the Standby Upstream PE becomes the Upstream PE, and the PE re-sends the C-multicast route with RT that identifies the Standby Upstream PE, except that now the route does not carry the Standby PE BGP Community (which results in replacing the old route with a new route, with the only difference between these routes being the absence of the Standby PE BGP Community). The new Upstream PE must set the LOCAL_PREF attribute for that C-multicast route to the same value as when the Standby PE BGP Community was included in the advertisement.

4.2. Upstream PE Behavior

When a PE supporting this specification receives a C-multicast route for a particular (C-S, C-G) for which all of the following are true:

- o the RT carried in the route results in importing the route into a particular VRF on the PE;
- o the route carries the Standby PE BGP Community; and
- o the PE determines (via a method of failure detection that is outside the scope of this document) that C-S is not reachable through some other PE (more details are in Section 4.3),

then the PE MAY install VRF PIM state corresponding to this Standby BGP C-multicast route (the result will be that a PIM Join message will be sent to the CE towards C-S, and that the PE will receive (C-S, C-G) traffic), and the PE MAY forward (C-S, C-G) traffic received by the PE to other PEs through a P-tunnel rooted at the PE.

Furthermore, irrespective of whether C-S carried in that route is reachable through some other PE:

- a) based on local policy, as soon as the PE receives this Standby BGP C-multicast route, the PE MAY install VRF PIM state corresponding to this BGP Source Tree Join route (the result will be that Join messages will be sent to the CE toward C-S, and that the PE will receive (C-S, C-G) traffic)
- b) based on local policy, as soon as the PE receives this Standby BGP C-multicast route, the PE MAY forward (C-S, C-G) traffic to other PEs through a P-tunnel independently of the reachability of C-S through some other PE. [note that this implies also doing a)]

Doing neither a) or b) for a given (C-S, C-G) is called "cold root standby".

Doing a) but not b) for a given (C-S, C-G) is called "warm root standby".

Doing b) (which implies also doing a)) for a given (C-S, C-G) is called "hot root standby".

Note that, if an Upstream PE uses an S-PMSI only policy, it shall advertise an S-PMSI for a (C-S, C-G) as soon as it receives a C-multicast route for (C-S, C-G), normal or Standby; i.e., it shall not wait for receiving a non-Standby C-multicast route before advertising the corresponding S-PMSI.

Section 9.3.2 of [RFC6513], describes the procedures of sending a Source-Active A-D Route as a result of receiving the C-multicast route. These procedures MUST be followed for both the normal and Standby C-multicast routes.

4.3. Reachability Determination

The Standby Upstream PE can use the following information to determine that C-S can or cannot be reached through the Primary Upstream PE:

- o presence/absence of a unicast VPN route toward C-S

- o supposing that the Standby Upstream PE is the egress of the tunnel rooted at the Primary Upstream PE, the Standby Upstream PE can determine the reachability of C-S through the Primary Upstream PE based on the status of this tunnel, determined thanks to the same criteria as the ones described in Section 3.1 (without using the UMH selection procedures of Section 3);
- o other mechanisms may be used.

4.4. Inter-AS

If the non-segmented inter-AS approach is used, the procedures described in Section 4.1 through Section 4.3 can be applied.

When multicast VPNs are used in an inter-AS context with the segmented inter-AS approach described in Section 9.2 of [RFC6514], the procedures in this section can be applied.

A pre-requisite for the procedures described below to be applied for a source of a given MVPN is:

- o that any PE of this MVPN receives two or more Inter-AS I-PMSI A-D Routes advertised by the AS of the source
- o that these Inter-AS I-PMSI A-D Routes have distinct Route Distinguishers (as described in item "(2)" of section 9.2 of [RFC6514]).

As an example, these conditions will be satisfied when the source is dual-homed to an AS that connects to the receiver AS through two ASBR using auto-configured RDs.

4.4.1. Inter-AS Procedures for downstream PEs, ASBR Fast Failover

The following procedure is applied by downstream PEs of an AS, for a source S in a remote AS.

Additionally to choosing an Inter-AS I-PMSI A-D Route advertised from the AS of the source to construct a C-multicast route, as described in section 11.1.3 [RFC6514], a downstream PE will choose a second Inter-AS I-PMSI A-D Route advertised from the AS of the source and use this route to construct and advertise a Standby C-multicast route (C-multicast route carrying the Standby extended community), as described in Section 4.1.

4.4.2. Inter-AS Procedures for ASBRs

When an Upstream ASBR receives a C-multicast route, and at least one of the RTs of the route matches one of the ASBR Import RT, the ASBR, that supports this specification, must try to locate an Inter-AS I-PMSI A-D Route whose RD and Source AS respectively match the RD and Source AS carried in the C-multicast route. If the match is found, and the C-multicast route carries the Standby PE BGP Community, then the ASBR implementation that supports this specification MUST be configurable to perform as follows:

- o if the route was received over iBGP and its LOCAL_PREF attribute is set to zero, then it MUST be re-advertised in eBGP with a MED attribute (MULTI_EXIT_DISC) set to the highest possible value (0xffff)
- o if the route was received over eBGP and its MED attribute set to 0xffff, then it MUST be re-advertised in iBGP with a LOCAL_PREF attribute set to zero

Other ASBR procedures are applied without modification and, when applied, MAY modify the above-listed behavior.

5. Hot Root Standby

The mechanisms defined in Section 4 and Section 3 can be used together as follows.

The principle is that, for a given VRF (or possibly only for a given (C-S, C-G)):

- o downstream PEs advertise a Standby BGP C-multicast route (based on Section 4)
- o Upstream PEs use the "hot standby" optional behavior and thus will start forwarding traffic for a given multicast state after they have a (primary) BGP C-multicast route or a Standby BGP C-multicast route for that state (or both)
- o a policy controls downstream PEs from which tunnel to accept traffic. For example, the policy could be based on the status of the tunnel or tunnel monitoring method (Section 3.1.5).

Other combinations of the mechanisms proposed in Section 4 and Section 3 are for further study.

Note that the same level of protection would be achievable with a simple C-multicast Source Tree Join route advertised to both the

primary and secondary Upstream PEs (carrying as Route Target extended communities, the values of the VRF Route Import Extended Community of each VPN route from each Upstream PEs). The advantage of using the Standby semantic is that, supposing that downstream PEs always advertise a Standby C-multicast route to the secondary Upstream PE, it allows to choose the protection level through a change of configuration on the secondary Upstream PE, without requiring any reconfiguration of all the downstream PEs.

6. Duplicate Packets

Multicast VPN specifications [RFC6513] impose that a PE only forwards to CEs the packets coming from the expected Upstream PE (Section 9.1 of [RFC6513]).

We draw the reader's attention to the fact that the respect of this part of multicast VPN specifications is especially important when two distinct Upstream PEs are susceptible to forward the same traffic on P-tunnels at the same time in the steady state. That will be the case when "hot root standby" mode is used (Section 5), and which can also be the case if procedures of Section 3 are used and a) the rules determining the status of a tree are not the same on two distinct downstream PEs or b) the rule determining the status of a tree depends on conditions local to a PE (e.g., the PE-P upstream link being up).

7. IANA Considerations

7.1. Standby PE Community

IANA is requested to allocate the BGP "Standby PE" community value (TBA1) from the Border Gateway Protocol (BGP) Well-known Communities registry using the First Come First Served registration policy.

7.2. BFD Discriminator

This document defines a new BGP optional transitive attribute, called "BFD Discriminator". IANA is requested to allocate a codepoint (TBA2) in the "BGP Path Attributes" registry to the BFD Discriminator attribute.

IANA is requested to create a new BFD Mode sub-registry in the Border Gateway Protocol (BGP) Parameters registry. The registration policies, per [RFC8126], for this sub-registry are according to Table 1.

Value	Policy
0- 175	IETF Review
176 - 249	First Come First Served
250 - 254	Experimental Use
255	IETF Review

Table 1: BFD Mode Sub-registry Registration Policies

IANA is requested to make initial assignments according to Table 2.

Value	Description	Reference
0	Reserved	This document
1	P2MP BFD Session	This document
2- 175	Unassigned	
176 - 249	Unassigned	
250 - 254	Experimental Use	This document
255	Reserved	This document

Table 2: BFD Mode Sub-registry

7.3. BFD Discriminator Optional TLV Type

IANA is requested to create a new BFD Discriminator Optional TLV Type sub-registry in Border Gateway Protocol (BGP). The registration policies, per [RFC8126], for this sub-registry are according to Table 3.

Value	Policy
0- 175	IETF Review
176 - 249	First Come First Served
250 - 254	Experimental Use
255	IETF Review

Table 3: BFD Discriminator Optional TLV Type Sub-registry Registration Policies

IANA is requested to make initial assignments according to Table 4.

Value	Description	Reference
0	Reserved	This document
1	Source IP Address	This document
2- 175	Unassigned	
176 - 249	Unassigned	
250 - 254	Experimental Use	This document
255	Reserved	This document

Table 4: BFD Discriminator Optional TLV Type Sub-registry

8. Security Considerations

This document describes procedures based on [RFC6513] and [RFC6514] and hence shares the security considerations respectively represented in these specifications.

This document uses P2MP BFD, as defined in [RFC8562], which, in turn, is based on [RFC5880]. Security considerations relevant to each protocol are discussed in the respective protocol specifications. An implementation that supports this specification MUST provide a mechanism to limit the overall amount of capacity used by the BFD traffic (as the combination of the number of active P2MP BFD sessions and the rate of BFD Control packets to process).

The methods described in Section 3.1 may produce false-negative state changes that can be the trigger for an unnecessary convergence in the control plane, ultimately negatively impacting the multicast service provided by the VPN. An operator is expected to consider the network environment and use available controls of the mechanism used to determine the status of a P-tunnel.

9. Acknowledgments

The authors want to thank Greg Reaume, Eric Rosen, Jeffrey Zhang, Martin Vigoureux, Adrian Farrel, and Zheng (Sandy) Zhang for their reviews, useful comments, and helpful suggestions.

10. Contributor Addresses

Below is a list of other contributing authors in alphabetical order:

Rahul Aggarwal
Arktan

Email: raggarwa_1@yahoo.com

Nehal Bhau
Cisco

Email: NBhau@cisco.com

Clayton Hassen
Bell Canada
2955 Virtual Way
Vancouver
CANADA

Email: Clayton.Hassen@bell.ca

Wim Henderickx
Nokia
Copernicuslaan 50
Antwerp 2018
Belgium

Email: wim.henderickx@nokia.com

Pradeep Jain
Nokia
701 E Middlefield Rd
Mountain View, CA 94043
USA

Email: pradeep.jain@nokia.com

Jayant Kotalwar
Nokia
701 E Middlefield Rd
Mountain View, CA 94043
USA

Email: Jayant.Kotalwar@nokia.com

Praveen Muley
Nokia

701 East Middlefield Rd
Mountain View, CA 94043
U.S.A.

Email: praveen.muley@nokia.com

Ray (Lei) Qiu
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
U.S.A.

Email: rqiujuniper.net

Yakov Rekhter
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
U.S.A.

Email: yakov@juniper.net

Kanwar Singh
Nokia
701 E Middlefield Rd
Mountain View, CA 94043
USA

Email: kanwar.singh@nokia.com

11. References

11.1. Normative References

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

- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [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>>.
- [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>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [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>>.
- [RFC8562] Katz, D., Ward, D., Pallagatti, S., Ed., and G. Mirsky, Ed., "Bidirectional Forwarding Detection (BFD) for Multipoint Networks", RFC 8562, DOI 10.17487/RFC8562, April 2019, <<https://www.rfc-editor.org/info/rfc8562>>.

11.2. Informative References

- [I-D.mirsky-mpls-p2mp-bfd]
Mirsky, G., Mishra, G., and D. Eastlake, "BFD for Multipoint Networks over Point-to-Multi-Point MPLS LSP", draft-mirsky-mpls-p2mp-bfd-12 (work in progress), November 2020.
- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<https://www.rfc-editor.org/info/rfc4090>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC7431] Karan, A., Filsfils, C., Wijnands, IJ., Ed., and B. Decraene, "Multicast-Only Fast Reroute", RFC 7431, DOI 10.17487/RFC7431, August 2015, <<https://www.rfc-editor.org/info/rfc7431>>.

Authors' Addresses

Thomas Morin (editor)
Orange
2, avenue Pierre Marzin
Lannion 22307
France

Email: thomas.morin@orange-ftgroup.com

Robert Kebler (editor)
Juniper Networks
1194 North Mathilda Ave.
Sunnyvale, CA 94089
U.S.A.

Email: rkebler@juniper.net

Greg Mirsky (editor)
ZTE Corp.

Email: gregimirsky@gmail.com

BESS Working Group
Internet-Draft
Intended Status: Standards Track

Ali Sajassi
Keyur Patel
Samir Thoria
Derek Yeung
Cisco

John Drake
Wen Lin
Juniper

Expires: April 17, 2016

October 17, 2015

IGMP and MLD Proxy for EVPN
draft-sajassi-bess-evpn-igmp-ml-d-proxy-00

Abstract

Ethernet Virtual Private Network (EVPN) solution [RFC 7432] is becoming pervasive in data center (DC) applications for Network Virtualization Overlay (NVO) services, for DC interconnect (DCI) services, and for next generation virtual private LAN services in service provider (SP) applications.

This draft describes how to support efficiently endpoints running IGMP for the above services over an EVPN network by incorporating IGMP proxy procedures on EVPN PEs.

Status of this Memo

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

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

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

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

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

Copyright and License Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1	Introduction	3
1.1	Terminology	4
2	IGMP Proxy	4
2.1	Proxy Reporting	4
2.1.1	IGMP Membership Report Advertisement in BGP	4
2.1.1	IGMP Leave Group Advertisement in BGP	6
2.2	Proxy Querier	7
3	Operation	7
3.1	PE with only attached hosts/VMs for a given subnet	8
3.2	PE with mixed of attached hosts/VMs and multicast source	9
3.1	PE with mixed of attached hosts/VMs, multicast source and router	9
5	BGP Encoding	9
5.1	Selective Multicast Ethernet Tag Route	9
5.2	Constructing the Selective Multicast route	11
6	Acknowledgement	12
7	Security Considerations	12
8	IANA Considerations	12
9	References	12
9.1	Normative References	12
9.2	Informative References	12
	Authors' Addresses	12

1 Introduction

Ethernet Virtual Private Network (EVPN) solution [RFC 7432] is becoming pervasive in data center (DC) applications for Network Virtualization Overlay (NVO) services, for DC interconnect (DCI) services, and for next generation virtual private LAN services in service provider (SP) applications.

In DC applications, a POD can consist of a collection of servers supported by several TOR and Spine routers. This collection of servers and routers are self contained and may have their own control protocol for intra-POD communication and orchestration. However, EVPN is used as way of standard inter-POD communication for both intra-DC and inter-DC. A subnet can span across multiple PODs and DCs. EVPN provides robust multi-tenant solution with extensive multi-homing capabilities to stretch a subnet (e.g., VLAN) across multiple PODs and DCs. There can be many hosts/VMs (e.g., several hundreds) attached to a subnet that is stretched across several PODs and DCs.

These hosts/VMs express their interests in multicast groups on a given subnet/VLAN by sending IGMP membership reports (Joins) for their interested multicast group(s). Furthermore, an IGMP router (e.g., IGMPv1) periodically sends membership queries to find out if there are hosts on that subnet still interested in receiving multicast traffic for that group. The IGMP/MLD Proxy solution described in this draft has three objectives to accomplish:

- 1) Just like ARP/ND suppression mechanism in EVPN to reduce the flooding of ARP messages over EVPN, it is also desired to have a mechanism to reduce the flood of IGMP messages (both Queries and Reports) in EVPN.
- 2) If there is no physical/virtual multicast router attached to the EVPN network for a given (*,G) or (S,G), it is desired for the EVPN network to act as a distributed anycast multicast router for all the hosts attached to that subnet.
- 3) To forward multicast traffic efficiently over EVPN network such that it only gets forwarded to the PEs that have interest in the multicast group(s) - i.e., multicast traffic will not be forwarded to the PEs that have no receivers attached to them for that multicast group. This draft shows how both of the above objectives are achieved.

The first two objectives are achieved by using IGMP/MLD proxy on the PE and the third objective is achieved by setting up a multicast tunnel (ingress replication or P2MP) only among the PEs that have interest in that multicast group(s) based on the trigger from

IGMP/MLD proxy processing. The proposed solutions for each of these objectives are discussed in the following sections.

1.1 Terminology

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

2 IGMP Proxy

IGMP Proxy mechanism is used to reduce the flooding of IGMP messages over EVPN network similar to ARP proxy used in reducing the flooding of ARP messages over EVPN. It also provided triggering mechanism for the PEs to setup their underlay multicast tunnels. IGMP Proxy mechanism consist of two components: a) Proxy for IGMP Reports and b) Proxy for IGMP Queries.

2.1 Proxy Reporting

When IGMP protocol is used between host/VMs and its first hop EVPN router (EVPN PE), Proxy-reporting is used by the EVPN PE to summarize (when possible) reports received from downstream hosts and propagate it in BGP to other PEs that are interested in the info. This is done by terminating IGMP Reports in the first hop PE, translating and exchanging the relevant information between EVPN BGP speakers. The information is again translated back to IGMP message at the recipient EVPN speaker. Thus it helps create an IGMP overlay subnet using BGP. In order to facilitate such an overlay, this document also defines a new EVPN route type NLRI (EVPN Selective Multicast Ethernet Tag route) along with its procedures to help exchange and register IGMP multicast groups [section 5].

2.1.1 IGMP Membership Report Advertisement in BGP

When a PE wants to advertise an IGMP membership report (Join) using the BGP EVPN route, it follows the following rules:

1) When the first hop PE receives several IGMP membership reports (Joins) , belonging to the same IGMP version, from different attached hosts/VMs for the same (*,G) or (S,G), it only sends a single BGP message corresponding to the very first IGMP Join. This is because BGP is a statefull protocol and no further transmission of the same report is needed. If the IGMP Join is for (*,G), then multicast group address along with the corresponding version flag (v1, v2, or v3) are set. In case of IGMPv3, exclude flag also needs to be set to indicate

that no source IP address to be excluded (e.g., include all sources "*"). If the IGMP Join is for (S,G), then besides setting multicast group address along with the version flag v3, the source IP address and the include/exclude flag must be set. It should be noted that when advertising the EVPN route for (S,G), the only valid version flag is v3 (i.e., v1 and v2 flags must be set to zero).

2) When the first hop PE receives an IGMPv3 Join for (S,G), then the PE checks to see if the source (S) is attached to self. If so, it does not send the corresponding BGP EVPN route advertisement.

3) When the first hop PE receives an IGMP version-X Join first for (*,G) and then later it receives an IGMP version-Y Join for the same (*,G), then it will readvertise the same EVPN Selective Multicast route with flag for version-Y set in addition to any previously-set version flag(s). In other words, the first hop PE does not withdraw the EVPN route before sending the new route because the flag field is not part of BGP route key processing.

4) When the first hop PE receives an IGMP version-X Join first for (*,G) and then later it receives an IGMPv3 Join for the same multicast group address but for a specific source address S, then the PE will readvertise a new EVPN Selective Multicast route with v3 flag set (and v1 and v2 reset). Include/exclude flag also need to be set accordingly. Since source IP address is used as part of BGP route key processing, it is considered as a new BGP route advertisement.

5) When a PE receives an EVPN Selective Multicast route with more than one version flag set, it will generate the corresponding IGMP report for (*,G) for each version specified in the flag field. With multiple version flags set, there should be no source IP address in the receive EVPN route. If there is, then an error should be logged. If v3 flag is set (in addition to v1 or v2), then the include/exclude flag needs to indicate "exclude". If not, then an error should be logged. The PE MUST generate an IGMP membership report (Join) for that (*,G) and each IGMP version in the version flag.

6) When a PE receives a list of EVPN Selective Multicast NLRI's in its BGP update message, each with a different source IP address and the multicast group address, and the version flag is set to v3, then the PE generates an IGMPv3 membership report with a record corresponding to the list of source IP addresses and the group address along with the proper indication of inclusion/exclusion.

7) Upon receiving EVPN Selective Multicast route(s) and before

generating the corresponding IGMP Join(s), the PE checks to see whether it has any multicast router's AC(s) (Attachment Circuits connected to multicast routers). If it has router's ACs, then the generated IGMP Join(s) are sent to those ACs. If it doesn't have any router's AC, then no IGMP Join(s) needs to be generated because sending IGMP Joins to other hosts can result in unintentionally preventing a host from joining a specific multicast group for IGMPv1 and IGMPv2 - i.e., if the PE does not receive a join from the host it will not forward multicast data to it. Per [RFC4541], when an IGMPv1 or IGMPv2 host receives a membership report for a group address that it intends to join, the host will suppress its own membership report for the same group. This message suppression is a requirement for IGMPv1 and IGMPv2 hosts. This is not a problem for hosts running IGMPv3 because there is no suppression of IGMP Membership reports.

2.1.1 IGMP Leave Group Advertisement in BGP

When a PE wants to withdraw an EVPN Selective Multicast route corresponding to an IGMPv2 Leave Group (Leave) or IGMPv3 "Leave" equivalent message, it follows the following rules:

- 1) For IGMPv1, there is no explicit membership leave; therefore, the PE needs to periodically send out an IGMP membership query to determine whether there is any host left who is interested in receiving traffic directed to this multicast group. This proxy query function will be described in more details in section 2.2.
- 2) When a PE receives an IGMPv2 Leave Group or its "Leave" equivalent message for IGMPv3 from its attached host, it checks to see if this host is the last host who is interested in this multicast group by sending a query for the multicast group. If the host was indeed the last one, then the PE re-advertises EVPN Selective Multicast route with the corresponding version flag reset. If this is the last version flag to be reset, then instead of readvertising the EVPN route with all version flags reset, the PE withdraws the EVPN route for that (*,G).
- 3) When a PE receives an EVPN Selective Multicast route for a given (*,G), it compares the received version flags from the route with its per-PER stored version flags. If the PE finds that a version flag associated with the (*,G) for the remote PE is reset, then the PE generates IGMP Leave for that (*,G) toward its local interface (if any) attached to the multicast router for that multicast group. It also removes the remote PE from the OIF list associated with that multicast group. It should be noted that the received EVPN route should at least have one version flag set. If all version flags are reset, it is an error because the PE should have received an EVPN

route withdraw for the last version flag.

4) If the reset version flag is for version-1 or if the EVPN route withdraw is for version-1, the PE removes the remote PE from its OIF list for that multicast group. If there are no more OIF entries for that multicast group (either locally or remotely), then the PE MUST stop responding to queries from the locally attached router (if any). If there is a source for that multicast group, the PE stops sending multicast traffic for that source.

2.2 Proxy Querier

As mentioned in the previous sections, each PE need to have proxy querier functionality for the following reasons:

- 1) To enable the collection of EVPN PEs providing L2VPN service to act as distributed multicast router with Anycast IP address for all attached hosts/VMs in that subnet.
- 2) To enable suppression of IGMP membership reports and queries over MPLS/IP core.
- 3) To enable generation of query messages locally to their attached host. In case of IGMPv1, the PE needs to send out an IGMP membership query to verify that at least one host on the subnet is still interested in receiving traffic directed to that group. When there is no reply to three consecutive IGMP membership queries, the PE times out the group, stops forwarding multicast traffic to the attached hosts for that (*,G), and sends a EVPN Selective Multicast route associated with that (*,G) with the version-1 flag reset or withdraws that route.

3 Operation

Consider the EVPN network of figure-1, where there is an EVPN instance configured across the PEs shown in this figure (namely PE1, PE2, and PE3). Lets consider that this EVPN instance consist of a single bridge domain (single subnet) with all the hosts, sources and the multicast router shown in this figure connected to this subnet. PE1 only has hosts connected to it. PE2 has a mix of hosts and multicast source. PE3 has a mix of hosts, multicast source, and multicast router. Further more, lets consider that for (S1,G1), R1 is used as the multicast router but for (S2, G2), distributed multicast router with Anycast IP address is used. The following subsections describe the IGMP proxy operation in different PEs with regard to whether the locally attached devices for that subnet are:

the same (*,G1), besides adding the corresponding port to its OIF list, it re-advertises the previously sent EVPN Selective Multicast route with the version-2 flag set.

Finally when PE1 receives the IMGMPv3 Join from H4 for (S2,G2), it advertises a new EVPN Selective Multicast route corresponding to it.

3.2 PE with mixed of attached hosts/VMs and multicast source

The main difference in here is that when PE2 receives IGMPv3 Join from H7 for (S2,G2), it does not advertises it in BGP because PE2 knows that S2 is attached to its local AC. PE2 adds the port associated with H7 to its OIF list for (S2,G2). The processing for IGMPv2 received from H6 is the same as the v2 Join described in previous section.

3.1 PE with mixed of attached hosts/VMs, multicast source and router

The main difference in here relative to the previous two sections is that Join messages received locally needs to be sent to the port associated with router R1. Furthermore, the Joins received via BGP need to be passed to the R1 port but filtered for all other ports.

5 BGP Encoding

This document defines a new BGP EVPN route to carry IGMP membership reports. This route type is known as:

+ 6 - Selective Multicast Ethernet Tag Route

The detailed encoding and procedures for this route type is described in subsequent section.

5.1 Selective Multicast Ethernet Tag Route

An Selective Multicast Ethernet Tag route type specific EVPN NLRI consists of the following:

RD (8 octets)
Ethernet Tag ID (4 octets)
Multicast Source Length (1 octet)
Multicast Source Address (variable)
Multicast Group Length (1 octet)
Multicast Group Address (Variable)
Originator Router Length (1 octet)
Originator Router Address (variable)
Flags (1 octets) (optional)

For the purpose of BGP route key processing, all the fields are considered to be part of the prefix in the NLRI except for the one-octet optional flag field (if included). The Flags fields are defined as follows:

0	1	2	3	4	5	6	7
reserved		IE	v3	v2	v1		

The least significant bit, bit 7 indicates support for IGMP version 1.

The second least significant bit, bit 6 indicates support for IGMP version 2.

The third least significant bit, bit 5 indicates support for IGMP version 3.

The fourth least significant bit, bit 4 indicates whether the (S, G) information carried within the route-type is of Include Group type (bit value 0) or an Exclude Group type (bit value 1). The Exclude Group type bit MUST be ignored if bit 5 is not set.

This EVPN route type is used to carry tenant IGMP multicast group information. The flag field assists in distributing IGMP membership interest of a given host/VM for a given multicast route. The version

bits help associate IGMP version of receivers participating within the EVPN domain.

The include/exclude bit helps in creating filters for a given multicast route.

5.2 Constructing the Selective Multicast route

This section describes the procedures used to construct the Selective Multicast route. Support for this route type is optional.

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

The Ethernet Tag ID MUST to zero for VLAN-based service and to a valid normalized VID for VLAN-aware bundle service.

The Multicast Source length MUST be set to length of multicast source address in bits. In case of a (*, G) Join, the Multicast Source Length is set to 0.

The Multicast Source is the Source IP address of the IGMP membership report. In case of a (*, G) Join, this field does not exist.

The Multicast Group length MUST be set to length of multicast group address in bits.

The Multicast Group is the Group address of the IGMP membership report.

The Originator Router Length is the length of the Originator Router address in bits.

The Originator Router Address is the IP address of Router Originating the prefix.

The Flags field indicates the version of IGMP protocol from which the membership report was received. It also indicates whether the multicast group had INCLUDE or EXCLUDE bit set.

IGMP protocol is used to receive group membership information from hosts/VMs by TORs. Upon receiving the hosts/VMs expression of interest of a particular group membership, this information is then forwarded to BGP EVPN using Ethernet Multicast Source Group Route NLRI. The NLRI also keeps track of receiver's IGMP protocol version and any "source filtering" for a given group membership. All Ethernet Multicast Source Group Routes are announced with ES-Import Route

Target extended communities.

6 Acknowledgement

7 Security Considerations

Same security considerations as [RFC7432].

8 IANA Considerations

Allocation of Extended Community Type and Sub-Type for EVPN.

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.

[RFC4360] S. Sangli et al, "BGP Extended Communities Attribute", February, 2006.

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

9.2 Informative References

[ETREE-FMWK] Key et al., "A Framework for E-Tree Service over MPLS Network", draft-ietf-l2vpn-etree-frwk-03, work in progress, September 2013.

[PBB-EVPN] Sajassi et al., "PBB-EVPN", draft-ietf-l2vpn-pbb-evpn-05.txt, work in progress, October, 2013.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for IGMP and MLD snooping PEs", RFC 4541, 2006.

Authors' Addresses

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Keyur Patel
Cisco
Email: keyupate@cisco.com

Samir Thoria
Cisco
Email: sthoria@cisco.com

Derek Yeung
Cisco
Email: myeung@cisco.com

John Drake
Juniper
Email: jdrake@juniper.net

Wen Lin
Juniper
Email: wlin@juniper.net

BESS Working Group
Internet-Draft
Intended Status: Standards Track

Ali Sajassi
Samir Thoria
Cisco
Keyur Patel
Derek Yeung
Arrcus
John Drake
Wen Lin
Juniper

Expires: April 28, 2017

October 28, 2016

IGMP and MLD Proxy for EVPN
draft-sajassi-bess-evpn-igmp-mld-proxy-01

Abstract

Ethernet Virtual Private Network (EVPN) solution [RFC 7432] is becoming pervasive in data center (DC) applications for Network Virtualization Overlay (NVO) and DC interconnect (DCI) services, and in service provider (SP) applications for next generation virtual private LAN services.

This draft describes how to support efficiently endpoints running IGMP for the above services over an EVPN network by incorporating IGMP proxy procedures on EVPN PEs.

Status of this Memo

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

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

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

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

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

Copyright and License Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1	Introduction	4
1.1	Terminology	5
2	IGMP Proxy	5
2.1	Proxy Reporting	5
2.1.1	IGMP Membership Report Advertisement in BGP	5
2.1.1	IGMP Leave Group Advertisement in BGP	7
2.2	Proxy Querier	8
3	Operation	8
3.1	PE with only attached hosts/VMs for a given subnet	9
3.2	PE with mixed of attached hosts/VMs and multicast source	10
3.3	PE with mixed of attached hosts/VMs, multicast source and router	10
4	All-Active Multi-Homing	10
4.1	Local IGMP Join Synchronization	11
4.2	Local IGMP Leave Group Synchronization	11
4.2.1	Remote Leave Group Synchronization	12
4.2.2	Common Leave Group Synchronization	13
5	Single-Active Multi-Homing	13
6	Discovery of Selective P-Tunnel Types	13
7	BGP Encoding	15
7.1	Selective Multicast Ethernet Tag Route	15
7.1.1	Constructing the Selective Multicast route	16
7.2	IGMP Join Synch Route	17
7.2.1	Constructing the IGMP Join Synch Route	19
7.3	IGMP Leave Synch Route	20

7.3.1 Constructing the IGMP Leave Synch Route 21

7.4 Multicast Flags Extended Community 22

7.5 EVI-RT Extended Community 23

8 Acknowledgement 24

9 Security Considerations 24

10 IANA Considerations 24

11 References 24

11.1 Normative References 24

11.2 Informative References 24

Authors' Addresses 25

1 Introduction

Ethernet Virtual Private Network (EVPN) solution [RFC 7432] is becoming pervasive in data center (DC) applications for Network Virtualization Overlay (NVO) and DC interconnect (DCI) services, and in service provider (SP) applications for next generation virtual private LAN services.

In DC applications, a POD can consist of a collection of servers supported by several TOR and Spine switches. This collection of servers and switches are self contained and may have their own control protocol for intra-POD communication and orchestration. However, EVPN is used as way of standard inter-POD communication for both intra-DC and inter-DC. A subnet can span across multiple PODs and DCs. EVPN provides robust multi-tenant solution with extensive multi-homing capabilities to stretch a subnet (e.g., VLAN) across multiple PODs and DCs. There can be many hosts/VMs (e.g., several hundreds) attached to a subnet that is stretched across several PODs and DCs.

These hosts/VMs express their interests in multicast groups on a given subnet/VLAN by sending IGMP membership reports (Joins) for their interested multicast group(s). Furthermore, an IGMP router (e.g., IGMPv1) periodically sends membership queries to find out if there are hosts on that subnet still interested in receiving multicast traffic for that group. The IGMP/MLD Proxy solution described in this draft has three objectives to accomplish:

- 1) Just like ARP/ND suppression mechanism in EVPN to reduce the flooding of ARP messages over EVPN, it is also desired to have a mechanism to reduce the flood of IGMP messages (both Queries and Reports) in EVPN.
- 2) If there is no physical/virtual multicast router attached to the EVPN network for a given (*,G) or (S,G), it is desired for the EVPN network to act as a distributed anycast multicast router for all the hosts attached to that subnet.
- 3) To forward multicast traffic efficiently over EVPN network such that it only gets forwarded to the PEs that have interest in the multicast group(s) - i.e., multicast traffic will not be forwarded to the PEs that have no receivers attached to them for that multicast group. This draft shows how the above objectives are achieved.

The first two objectives are achieved by using IGMP/MLD proxy on the PE and the third objective is achieved by setting up a multicast tunnel (ingress replication or P2MP) only among the PEs that have interest in that multicast group(s) based on the trigger from

IGMP/MLD proxy processes. The proposed solutions for each of these objectives are discussed in the following sections.

1.1 Terminology

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

2 IGMP Proxy

IGMP Proxy mechanism is used to reduce the flooding of IGMP messages over EVPN network similar to ARP proxy used in reducing the flooding of ARP messages over EVPN. It also provides triggering mechanism for the PEs to setup their underlay multicast tunnels. IGMP Proxy mechanism consist of two components: a) Proxy for IGMP Reports and b) Proxy for IGMP Queries.

2.1 Proxy Reporting

When IGMP protocol is used between host/VMs and its first hop EVPN router (EVPN PE), Proxy-reporting is used by the EVPN PE to summarize (when possible) reports received from downstream hosts and propagate it in BGP to other PEs that are interested in the info. This is done by terminating IGMP Reports in the first hop PE, translating and exchanging the relevant information among EVPN BGP speakers. The information is again translated back to IGMP message at the recipient EVPN speaker. Thus it helps create an IGMP overlay subnet using BGP. In order to facilitate such an overlay, this document also defines a new EVPN route type NLRI (EVPN Selective Multicast Ethernet Tag route) along with its procedures to help exchange and register IGMP multicast groups [section 5].

2.1.1 IGMP Membership Report Advertisement in BGP

When a PE wants to advertise an IGMP membership report (Join) using the BGP EVPN route, it follows the following rules:

1) When the first hop PE receives several IGMP membership reports (Joins) , belonging to the same IGMP version, from different attached hosts/VMs for the same (*,G) or (S,G), it only sends a single BGP message corresponding to the very first IGMP Join. This is because BGP is a statefull protocol and no further transmission of the same report is needed. If the IGMP Join is for (*,G), then multicast group address along with the corresponding version flag (v1, v2, or v3) are set. In case of IGMPv3, exclude flag also needs to be set to indicate

that no source IP address to be excluded (e.g., include all sources "*"). If the IGMP Join is for (S,G), then besides setting multicast group address along with the version flag v3, the source IP address and the include/exclude flag must be set. It should be noted that when advertising the EVPN route for (S,G), the only valid version flag is v3 (i.e., v1 and v2 flags must be set to zero).

2) When the first hop PE receives an IGMPv3 Join for (S,G), then the PE checks to see if the source (S) is attached to self. If so, it does not send the corresponding BGP EVPN route advertisement.

3) When the first hop PE receives an IGMP version-X Join first for (*,G) and then later it receives an IGMP version-Y Join for the same (*,G), then it will readvertise the same EVPN Selective Multicast route with flag for version-Y set in addition to any previously-set version flag(s). In other words, the first hop PE does not withdraw the EVPN route before sending the new route because the flag field is not part of BGP route key processing.

4) When the first hop PE receives an IGMP version-X Join first for (*,G) and then later it receives an IGMPv3 Join for the same multicast group address but for a specific source address S, then the PE will advertise a new EVPN Selective Multicast route with v3 flag set (and v1 and v2 reset). Include/exclude flag also need to be set accordingly. Since source IP address is used as part of BGP route key processing, it is considered as a new BGP route advertisement.

5) When a PE receives an EVPN Selective Multicast route with more than one version flag set, it will generate the corresponding IGMP report for (*,G) for each version specified in the flag field. With multiple version flags set, there should be no source IP address in the receive EVPN route. If there is, then an error should be logged. If v3 flag is set (in addition to v1 or v2), then the include/exclude flag needs to indicate "exclude". If not, then an error should be logged. The PE MUST generate an IGMP membership report (Join) for that (*,G) and each IGMP version in the version flag.

6) When a PE receives a list of EVPN Selective Multicast NLRIs in its BGP update message, each with a different source IP address and the multicast group address, and the version flag is set to v3, then the PE generates an IGMPv3 membership report with a record corresponding to the list of source IP addresses and the group address along with the proper indication of inclusion/exclusion.

7) Upon receiving EVPN Selective Multicast route(s) and before

generating the corresponding IGMP Join(s), the PE checks to see whether it has any multicast router's AC(s) (Attachment Circuits connected to multicast routers). If it has router's ACs, then the generated IGMP Join(s) are sent to those ACs. If it doesn't have any router's AC, then no IGMP Join(s) needs to be generated because sending IGMP Joins to other hosts can result in unintentionally preventing a host from joining a specific multicast group for IGMPv1 and IGMPv2 - i.e., if the PE does not receive a join from the host it will not forward multicast data to it. Per [RFC4541], when an IGMPv1 or IGMPv2 host receives a membership report for a group address that it intends to join, the host will suppress its own membership report for the same group. This message suppression is a requirement for IGMPv1 and IGMPv2 hosts. This is not a problem for hosts running IGMPv3 because there is no suppression of IGMP Membership reports.

2.1.1 IGMP Leave Group Advertisement in BGP

When a PE wants to withdraw an EVPN Selective Multicast route corresponding to an IGMPv2 Leave Group (Leave) or IGMPv3 "Leave" equivalent message, it follows the following rules:

- 1) For IGMPv1, there is no explicit membership leave; therefore, the PE needs to periodically send out an IGMP membership query to determine whether there is any host left who is interested in receiving traffic directed to this multicast group (this proxy query function will be described in more details in section 2.2). If there is no host left, then the PE re-advertises EVPN Selective Multicast route with the v1 version flag reset. If this is the last version flag to be reset, then instead of re-advertising the EVPN route with all version flags reset, the PE withdraws the EVPN route for that (*,G).
- 2) When a PE receives an IGMPv2 Leave Group or its "Leave" equivalent message for IGMPv3 from its attached host, it checks to see if this host is the last host who is interested in this multicast group by sending a query for the multicast group. If the host was indeed the last one, then the PE re-advertises EVPN Selective Multicast route with the corresponding version flag reset. If this is the last version flag to be reset, then instead of re-advertising the EVPN route with all version flags reset, the PE withdraws the EVPN route for that (*,G).
- 3) When a PE receives an EVPN Selective Multicast route for a given (*,G), it compares the received version flags from the route with its per-PE stored version flags. If the PE finds that a version flag associated with the (*,G) for the remote PE is reset, then the PE generates IGMP Leave for that (*,G) toward its local interface (if

any) attached to the multicast router for that multicast group. It should be noted that the received EVPN route should at least have one version flag set. If all version flags are reset, it is an error because the PE should have received an EVPN route withdraw for the last version flag. If the PE receives an EVPN Selective Multicast route withdraw, then it must remove the remote PE from the OIF list associated with that multicast group.

4) When a PE receives an EVPN Selective Multicast route withdraw, it removes the remote PE from its OIF list for that multicast group and if there are no more OIF entries for that multicast group (either locally or remotely), then the PE MUST stop responding to queries from the locally attached router (if any). If there is a source for that multicast group, the PE stops sending multicast traffic for that source.

2.2 Proxy Querier

As mentioned in the previous sections, each PE need to have proxy querier functionality for the following reasons:

- 1) To enable the collection of EVPN PEs providing L2VPN service to act as distributed multicast router with Anycast IP address for all attached hosts/VMs in that subnet.
- 2) To enable suppression of IGMP membership reports and queries over MPLS/IP core.
- 3) To enable generation of query messages locally to their attached host. In case of IGMPv1, the PE needs to send out an IGMP membership query to verify that at least one host on the subnet is still interested in receiving traffic directed to that group. When there is no reply to three consecutive IGMP membership queries, the PE times out the group, stops forwarding multicast traffic to the attached hosts for that (*,G), and sends a EVPN Selective Multicast route associated with that (*,G) with the version-1 flag reset or withdraws that route.

3 Operation

Consider the EVPN network of figure-1, where there is an EVPN instance configured across the PEs shown in this figure (namely PE1, PE2, and PE3). Lets consider that this EVPN instance consist of a single bridge domain (single subnet) with all the hosts, sources and the multicast router shown in this figure connected to this subnet. PE1 only has hosts connected to it. PE2 has a mix of hosts and

send it to any of its port for that subnet - namely ports associated with H6 and H7.

When PE1 receives the second IGMPv1 Join from H2 for the same multicast group (*,G1), it only adds that port to its OIF list but it doesn't send any EVPN BGP route because there is no change in information. However, when it receives the IGMPv2 Join from H3 for the same (*,G1), besides adding the corresponding port to its OIF list, it re-advertises the previously sent EVPN Selective Multicast route with the version-2 flag set.

Finally when PE1 receives the IMGMPv3 Join from H4 for (S2,G2), it advertises a new EVPN Selective Multicast route corresponding to it.

3.2 PE with mixed of attached hosts/VMs and multicast source

The main difference in here is that when PE2 receives IGMPv3 Join from H7 for (S2,G2), it does not advertises it in BGP because PE2 knows that S2 is attached to its local AC. PE2 adds the port associated with H7 to its OIF list for (S2,G2). The processing for IGMPv2 received from H6 is the same as the v2 Join described in previous section.

3.3 PE with mixed of attached hosts/VMs, multicast source and router

The main difference in here relative to the previous two sections is that Join messages received locally needs to be sent to the port associated with router R1. Furthermore, the Joins received via BGP need to be passed to the R1 port but filtered for all other ports.

4 All-Active Multi-Homing

Because a CE's LAG flow hashing algorithm is unknown, in an All-Active redundancy mode it must be assumed that the CE can send a given IGMP message to any one of the multi-homed PEs, either DF or non-DF - i.e., different IGMP Join messages can arrive at different PEs in the redundancy group and furthermore their corresponding Leave messages can arrive at PEs that are different from the ones received the Join messages. Therefore, all PEs attached to a given ES must coordinate IGMP Join and Leave Group (x, G) state, where x may be either '*' or a particular source S, for each [EVI, broadcast domain (BD)] on that ES. This allows the DF for that [ES, EVI, BD] to correctly advertise or withdraw a Selective Multicast Ethernet Tag (SMET) route for that (x, G) group in that [EVI, BD] when needed.

All-Active multihoming PEs for a given ES MUST support IGMP synch procedures described in this section if they want to perform IGMP proxy for hosts connects to that ES.

4.1 Local IGMP Join Synchronization

When a PE, either DF or non-DF, receives, on a given multihomed ES operating in All-Active redundancy mode, an IGMP Membership Report for (x, G), it determines the [EVI, BD] to which the IGMP Membership Report belongs. If the PE doesn't already have local IGMP Join (x, G) state for that [EVI, BD] on that ES, it instantiates local IGMP Join (x, G) state and advertises a BGP IGMP Join Synch route for that [ES, EVI, BD]. Local IGMP Join (x, G) state refers to IGMP Join (x, G) state that is created as the result of processing an IGMP Membership Report for (x, G).

The IGMP Join Synch route carries the ES-Import RT for the ES on which the IGMP Membership Report was received. Thus it may only go to the PEs attached to that ES (and not any other PEs).

When a PE, either DF or non-DF, receives an IGMP Join Synch route it installs that route and if it doesn't already have IGMP Join (x, G) state for that [ES, EVI, BD], it instantiates that IGMP Join (x,G) state - i.e., IGMP Join (x, G) state is the union of local IGMP Join (x, G) state and installed IGMP Join Synch route. If the DF is not currently advertising (originating) a SMET route for that (x, G) group in that [EVI, BD], it does so now.

When a PE, either DF or non-DF, deletes its local IGMP Join (x, G) state for that [ES, EVI, BD], it withdraws its BGP IGMP Join Synch route for that [ES, EVI, BD].

When a PE, either DF or non-DF, receives the withdrawal of an IGMP Join Synch route from another PE it removes that route. When a PE has no local IGMP Join (x, G) state and it has no installed IGMP Join Synch routes, it removes IGMP Join (x, G) state for that [ES, EVI, BD]. If the DF no longer has IGMP Join (x, G) state for that [EVI, BD] on any ES for which it is DF, it withdraws its SMET route for that (x, G) group in that [EVI, BD].

I.e., A PE advertises an SMET route for that (x, G) group in that [EVI, BD] when it has IGMP Join (x, G) state in that [EVI, BD] on at least one ES for which it is DF and it withdraws that SMET route when it does not have IGMP Join (x, G) state in that [EVI, BD] on any ES for which it is DF.

4.2 Local IGMP Leave Group Synchronization

When a PE, either DF or non-DF, receives, on a given multihomed ES operating in All-Active redundancy mode, an IGMP Leave Group message for (x, G) from the attached CE, it determines the [EVI, BD] to which the IGMPv2 Leave Group belongs. Regardless of whether it has IGMP Join (x, G) state for that [ES, EVI, BD], it initiates the (x, G) leave group synchronization procedure, which consists of the following steps:

- 1) It computes the Maximum Response Time, which is the duration of (x, G) leave group synchronization procedure. This is the product of two locally configured values, Last Member Query Count and Last Member Query Interval (described in Section 3 of [RFC2236]), plus delta, the time it takes for a BGP advertisement to propagate between the PEs attached to the multihomed ES (delta is a consistently configured value on all PEs attached to the multihomed ES).
- 2) It starts the Maximum Response Time timer. Note that the receipt of subsequent IGMP Leave Group messages or BGP Leave Synch routes for (x, G) do not change the value of a currently running Maximum Response Time timer and are ignored by the PE.
- 3) It initiates the Last Member Query procedure described in Section 3 of [RFC2236]; viz, it sends a number of Group-Specific Query (x, G) messages (Last Member Query Count) at a fixed interval (Last Member Query Interval) to the attached CE.
- 4) It advertises an IGMP Leave Synch route for that that [ES, EVI, BD]. This route notifies the other multihomed PEs attached to the given multihomed ES that it has initiated an (x, G) leave group synchronization procedure; i.e., it carries the ES-Import RT for the ES on which the IGMP Leave Group was received. It also contains the Maximum Response Time and the Leave Group Synchronization Procedure Sequence number. The latter identifies the specific (x, G) leave group synchronization procedure initiated by the advertising PE, which increments the value whenever it initiates a procedure.
- 5) When the Maximum Response Timer expires, the PE that has advertised the IGMP Leave Synch route withdraws it.

4.2.1 Remote Leave Group Synchronization

When a PE, either DF or non-DF, receives an IGMP Leave Synch route it installs that route and it starts a timer for (x, G) on the specified [ES, EVI, BD] whose value is set to the Maximum Response Time in the received IGMP Leave Synch route. Note that the receipt of subsequent IGMPv2 Leave Group messages or BGP Leave Synch routes for (x, G) do not change the value of a currently running Maximum Response Time

timer and are ignored by the PE.

4.2.2 Common Leave Group Synchronization

If a PE attached to the multihomed ES receives an IGMP Membership Report for (x, G) before the Maximum Response Time timer expires, it advertises a BGP IGMP Join Synch route for that [ES, EVI, BD]. If it doesn't already have local IGMP Join (x, G) state for that [ES, EVI, BD], it instantiates local IGMP Join (x, G) state. If the DF is not currently advertising (originating) a SMET route for that (x, G) group in that [EVI, BD], it does so now.

If a PE attached to the multihomed ES receives an IGMP Join Synch route for (x, G) before the Maximum Response Time timer expires, it installs that route and if it doesn't already have IGMP Join (x, G) state for that [EVI, BD] on that ES, it instantiates that IGMP Join (x, G) state. If the DF is not currently advertising (originating) a SMET route for that (x, G) group in that [EVI, BD], it does so now.

When the Maximum Response Timer expires a PE that has advertised an IGMP Leave Synch route, withdraws it. Any PE attached to the multihomed ES, that started the Maximum Response Time and has no local IGMP Join (x, G) state and no installed IGMP Join Synch routes, it removes IGMP Join (x, G) state for that [ES, EVI, BD]. If the DF no longer has IGMP Join (x, G) state for that [EVI, BD] on any ES for which it is DF, it withdraws its SMET route for that (x, G) group in that [EVI, BD].

5 Single-Active Multi-Homing

Note that to facilitate state synchronization after failover, the PEs attached to a multihomed ES operating in Single-Active redundancy mode should also coordinate IGMP Join (x, G) state. In this case all IGMP Join messages are received by the DF and distributed to the non-DF PEs using the procedures described above.

6 Discovery of Selective P-Tunnel Types

To allow an ingress PE that supports IGMP proxy procedures and SMET route to properly assign a selective P-tunnel supported by the receiving PEs, the ingress PE needs to discover the types of selective P-tunnels supported by the receiving PEs and select the preferred tunnel type among the ones that it has in common with the receiving PEs.

In order to support such discovery mechanism, the Multicast Flags extended community defined in section 7.2 is used. Each PE that

supports different types of P-tunnels, marks the corresponding bits and advertise this extended community along with its IMET route. Therefore, the ingress PE can discover types of P-tunnels supported by the receiving PEs. If the ingress PE does not receive this extended community along with an IMET route for a given EVI, it assumes the only P-tunnel type supported by the egress PE, is ingress replication.

If besides ingress-replication P-tunnel type, there is no other P-tunnel types in common among the participant PEs for an EVI, then the ingress PE MUST use ingress-replication P-tunnel type.

If besides ingress-replication P-tunnel type, there is one or more P-tunnel types in common among the participant PEs for an EVI, then the ingress PE can choose the P-tunnel type that it prefers.

If besides ingress-replication P-tunnel type, there is no other P-tunnel types in common among the participant PEs for an EVI, then the ingress PE MAY choose several different P-tunnel types where the union of them covers the tunnel types supported by the participant PEs for that EVI. This implies that the ingress PE replicates the multicast traffic into different P-tunnels - i.e., to replicate the multicast traffic onto P2MP mLDP P-tunnel and ingress-replication P-tunnel.

If an ingress PE uses ingress replication, then for a given (x, G) group in a given [EVI, BD]:

- 1) It sends (x, G) traffic to the set of PEs not supporting IGMP Proxy. This set consists of any PE that has advertised an Inclusive Multicast Tag route for the [EVI, BD] without the "IGMP Proxy Support" flag.
- 2) It sends (x, G) traffic to the set of PEs supporting IGMP Proxy and having listeners for that (x, G) group in that [EVI, BD]. This set consists of any PE that has advertised an Inclusive Multicast Tag route for the [EVI, BD] with the "IGMP Proxy Support" flag and that has advertised an SMET route for that (x, G) group in that [EVI, BD].

If an ingress PE's Selective P-Tunnel for a given [EVI, BD] uses P2MP and all of the PEs in the [EVI, BD] support that tunnel type and IGMP, then for a given (x, G) group in a given [EVI, BD] it sends (x, G) traffic using the Selective P-Tunnel for that (x, G) group in that [EVI, BD]. This tunnel will include those PEs that have advertised an SMET route for that (x, G) group on that [EVI, BD] (for Selective P-tunnel) but it may include other PEs as well (for Aggregate Selective P-tunnel).

7 BGP Encoding

This document defines three new BGP EVPN routes to carry IGMP membership reports. This route type is known as:

- + 6 - Selective Multicast Ethernet Tag Route
- + 7 - IGMP Join Synch Route
- + 8 - IGMP Leave Synch Route

The detailed encoding and procedures for this route type is described in subsequent section.

7.1 Selective Multicast Ethernet Tag Route

An Selective Multicast Ethernet Tag route type specific EVPN NLRI consists of the following:

```

+-----+
|  RD (8 octets)  |
+-----+
|  Ethernet Tag ID (4 octets)  |
+-----+
|  Multicast Source Length (1 octet)  |
+-----+
|  Multicast Source Address (variable)  |
+-----+
|  Multicast Group Length (1 octet)  |
+-----+
|  Multicast Group Address (Variable)  |
+-----+
|  Originator Router Length (1 octet)  |
+-----+
|  Originator Router Address (variable)  |
+-----+
|  Flags (1 octets) (optional)  |
+-----+

```

For the purpose of BGP route key processing, all the fields are considered to be part of the prefix in the NLRI except for the one-octet optional flag field (if included). The Flags fields are defined as follows:

```

      0  1  2  3  4  5  6  7
+-----+-----+-----+-----+
| reserved | IE|v3|v2|v1|
+-----+-----+-----+-----+

```

The least significant bit, bit 7 indicates support for IGMP version 1.

The second least significant bit, bit 6 indicates support for IGMP version 2.

The third least significant bit, bit 5 indicates support for IGMP version 3.

The fourth least significant bit, bit 4 indicates whether the (S, G) information carried within the route-type is of Include Group type (bit value 0) or an Exclude Group type (bit value 1). The Exclude Group type bit MUST be ignored if bit 5 is not set.

This EVPN route type is used to carry tenant IGMP multicast group information. The flag field assists in distributing IGMP membership interest of a given host/VM for a given multicast route. The version bits help associate IGMP version of receivers participating within the EVPN domain.

The include/exclude bit helps in creating filters for a given multicast route.

7.1.1 Constructing the Selective Multicast route

This section describes the procedures used to construct the Selective Multicast route. Support for this route type is optional.

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

The Ethernet Tag ID MUST be set as follows:

```

EVI is VLAN-Based or VLAN Bundle service - set to 0
EVI is VLAN-Aware Bundle service without translation - set to
the customer VID for the [EVI, BD]
EVI is VLAN-Aware Bundle service with translation - set to the
normalized Ethernet Tag ID for the [EVI, BD]

```

The Multicast Source length MUST be set to length of multicast source address in bits. In case of a (*, G) Join, the Multicast Source Length is set to 0.

The Multicast Source is the Source IP address of the IGMP membership report. In case of a (*, G) Join, this field does not exist.

The Multicast Group length MUST be set to length of multicast group address in bits.

The Multicast Group is the Group address of the IGMP membership report.

The Originator Router Length is the length of the Originator Router address in bits.

The Originator Router Address is the IP address of Router Originating the prefix. It should be noted that using the "Originating Router's IP address" field to get the PE IP address, needed for building multicast underlay tunnels, allows for inter-AS operations where BGP next hop can get over written.

The Flags field indicates the version of IGMP protocol from which the membership report was received. It also indicates whether the multicast group had INCLUDE or EXCLUDE bit set.

IGMP protocol is used to receive group membership information from hosts/VMs by TORs. Upon receiving the hosts/VMs expression of interest of a particular group membership, this information is then forwarded using Ethernet Multicast Source Group Route NLRI. The NLRI also keeps track of receiver's IGMP protocol version and any "source filtering" for a given group membership. All EVPN Selective Multicast Group routes are announced with per-EVI Route Target extended communities.

7.2 IGMP Join Synch Route

This EVPN route type is used to coordinate IGMP Join (x,G) state for a given [EVI, BD] between the PEs attached to a given ES operating in All-Active (or Single-Active) redundancy mode and it consists of following:

RD (8 octets)
Ethernet Segment Identifier (10 octets)
Ethernet Tag ID (4 octets)
Multicast Source Length (1 octet)
Multicast Source Address (variable)
Multicast Group Length (1 octet)
Multicast Group Address (Variable)
Originator Router Length (1 octet)
Originator Router Address (variable)
Flags (1 octet)

For the purpose of BGP route key processing, all the fields are considered to be part of the prefix in the NLRI except for the one-octet Flags field, whose fields are defined as follows:

0	1	2	3	4	5	6	7
reserved				IE	v3	v2	v1

The least significant bit, bit 7 indicates support for IGMP version 1. The second least significant bit, bit 6 indicates support for IGMP version 2. The third least significant bit, bit 5 indicates support for IGMP version 3. The fourth least significant bit, bit 4 indicates whether the (S, G) information carried within the route-type is of Include Group type (bit value 0) or an Exclude Group type (bit value 1). The Exclude Group type bit MUST be ignored if bit 5 is not set.

The Flags field assists in distributing IGMP membership interest of a given host/VM for a given multicast route. The version bits help associate IGMP version of receivers participating within the EVPN domain. The include/exclude bit helps in creating filters for a

given multicast route.

7.2.1 Constructing the IGMP Join Synch Route

This section describes the procedures used to construct the IGMP Join Synch route. Support for this route type is optional. If a PE does not support this route, then it MUST not indicate that it supports 'IGMP proxy' in Multicast Flag extended community for the EVIs corresponding to its multi-homed Ethernet Segments. An IGMP Join Synch route is advertised with an ES-Import Route Target extended community whose value is set to the ESI for the ES on which the IGMP Join was received.

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

The Ethernet Segment Identifier (ESI) MUST be set to the 10-octet value defined for the ES.

The Ethernet Tag ID MUST be set as follows:

EVI is VLAN-Based or VLAN Bundle service - set to 0
EVI is VLAN-Aware Bundle service without translation - set to the customer VID for the [EVI, BD]
EVI is VLAN-Aware Bundle service with translation - set to the normalized Ethernet Tag ID for the [EVI, BD]

The Multicast Source length MUST be set to length of multicast source address in bits. In case of a (*, G) Join, the Multicast Source Length is set to 0.

The Multicast Source is the Source IP address of the IGMP membership report. In case of a (*, G) Join, this field does not exist.

The Multicast Group length MUST be set to length of multicast group address in bits.

The Multicast Group is the Group address of the IGMP membership report.

The Originator Router Length is the length of the Originator Router address in bits.

The Originator Router Address is the IP address of Router Originating the prefix.

The Flags field indicates the version of IGMP protocol from which the membership report was received. It also indicates whether the multicast group had INCLUDE or EXCLUDE bit set.

7.3 IGMP Leave Synch Route This EVPN route type is used to coordinate IGMP Leave Group (x,G) state for a given [EVI, BD] between the PEs attached to a given ES operating in All-Active (or Single-Active) redundancy mode and it consists of following:

```

+-----+
|  RD (8 octets)                               |
+-----+
|  Ethernet Segment Identifier (10 octets)      |
+-----+
|  Ethernet Tag ID (4 octets)                  |
+-----+
|  Multicast Source Length (1 octet)           |
+-----+
|  Multicast Source Address (variable)         |
+-----+
|  Multicast Group Length (1 octet)           |
+-----+
|  Multicast Group Address (Variable)         |
+-----+
|  Originator Router Length (1 octet)         |
+-----+
|  Originator Router Address (variable)       |
+-----+
|  Leave Group Synchronization # (4 octets)   |
+-----+
|  Maximum Response Time (1 octet)           |
+-----+
|  Flags (1 octet)                            |
+-----+

```

For the purpose of BGP route key processing, all the fields are considered to be part of the prefix in the NLRI except for the Maximum Response Time and the one-octet Flags field, whose fields are defined as follows:

```

      0  1  2  3  4  5  6  7
+-----+-----+-----+-----+
| reserved | IE|v3|v2|v1|
+-----+-----+-----+-----+

```

The least significant bit, bit 7 indicates support for IGMP version 1. The second least significant bit, bit 6 indicates support for IGMP version 2. The third least significant bit, bit 5 indicates support for IGMP version 3. The fourth least significant bit, bit 4 indicates whether the (S, G) information carried within the route-type is of Include Group type (bit value 0) or an Exclude Group type (bit value 1). The Exclude Group type bit MUST be ignored if bit 5 is not set.

The Flags field assists in distributing IGMP membership interest of a given host/VM for a given multicast route. The version bits help associate IGMP version of receivers participating within the EVPN domain. The include/exclude bit helps in creating filters for a given multicast route.

7.3.1 Constructing the IGMP Leave Synch Route

This section describes the procedures used to construct the IGMP Join Synch route. Support for this route type is optional. If a PE does not support this route, then it MUST not indicate that it supports 'IGMP proxy' in Multicast Flag extended community for the EVIs corresponding to its multi-homed Ethernet Segments. An IGMP Join Synch route is advertised with an ES-Import Route Target extended community whose value is set to the ESI for the ES on which the IGMP Join was received.

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

The Ethernet Segment Identifier (ESI) MUST be set to the 10-octet value defined for the ES.

The Ethernet Tag ID MUST be set as follows:

```

EVI is VLAN-Based or VLAN Bundle service - set to 0
EVI is VLAN-Aware Bundle service without translation - set to
the customer VID for the [EVI, BD]
EVI is VLAN-Aware Bundle service with translation - set to the
normalized Ethernet Tag ID for the [EVI, BD]

```

The Multicast Source length MUST be set to length of multicast source address in bits. In case of a (*, G) Join, the Multicast Source Length is set to 0.

The Multicast Source is the Source IP address of the IGMP membership report. In case of a (*, G) Join, this field does not exist.

The Multicast Group length MUST be set to length of multicast group address in bits.

The Multicast Group is the Group address of the IGMP membership report.

The Originator Router Length is the length of the Originator Router address in bits.

The Originator Router Address is the IP address of Router Originating the prefix.

The Flags field indicates the version of IGMP protocol from which the membership report was received. It also indicates whether the multicast group had INCLUDE or EXCLUDE bit set.

7.4 Multicast Flags Extended Community

The 'Multicast Flags' extended community is a new EVPN extended community. EVPN extended communities are transitive extended communities with a Type field value of 6. IANA will assign a Sub-Type from the 'EVPN Extended Community Sub-Types' registry.

A PE that supports IGMP proxy on a given [EVI, BD] MUST attach this extended community to the Inclusive Multicast Ethernet Tag (IMET) route it advertises for that [EVI, BD] and it Must set the IGMP Proxy Support flag to 1. Note that an [RFC7432] compliant PE will not advertise this extended community so its absence indicates that the advertising PE does not support IGMP Proxy.

The advertisement of this extended community enables more efficient multicast tunnel setup from the source PE specially for ingress replication - i.e., if an egress PE supports IGMP proxy but doesn't have any interest in a given (x, G), it advertises its IGMP proxy capability using this extended community but it does not advertise any SMET route for that (x, G). When the source PE (ingress PE) receives such advertisements from the egress PE, it doesn't not replicate the multicast traffic to that egress PE; however, it does replicate the multicast traffic to the egress PEs that don't

advertise such capability even if they don't have any interests in that (x, G).

A Multicast Flags extended community is encoded as an 8-octet value, as follows:

```

          1                2                3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type=0x06   | Sub-Type=TBD   |      Flags (2 Octets)      |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                Reserved=0                |      Tunnel Type      |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

The low-order bit of the Flags is defined as the "IGMP Proxy Support" bit. A value of 1 means that the PE supports IGMP Proxy as defined in this document, and a value of 0 means that the PE does not support IGMP proxy. The absence of this extended community also means that the PE doesn't support IGMP proxy.

Tunnel type field is a 2-octet field with the bits set according to the following:

```

LSB = 1, indicates the support for RSVP-TE P2MP LSP
2nd LSB = 1, indicates the support for P2MP LSP
3rd LSB = 1, indicates the support for PIM-SSM
4th LSB = 1, indicates the support for PIM-SM
5th LSB = 1, indicates the support for BIDIR-PIM
6th LSB = 1, indicates the support for mLDP MP2MP LSP

```

7.5 EVI-RT Extended Community

The 'EVI-RT' extended community is a new EVPN extended community. EVPN extended communities are transitive extended communities with a Type field value of 6. IANA will assign a Sub-Type from the 'EVPN Extended Community Sub-Types' registry.

A PE that supports IGMP synch procedures for All-Active (or Single-Active) multi-homed ES, MUST attach this extended community to either IGMP Join Synch route (sec 7.2) or IGMP Leave Synch route (sec 7.3). This extended community carries the RT associated with the EVI so that the receiving PE can identify the EVI properly. The reason standard format RT is not used, is to avoid distribution of these routes beyond the group of multihoming PEs for that ES.

```

                                1                2                3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Type=0x06   | Sub-Type=TBD |           RT associated with EVI |
+-----+-----+-----+-----+-----+-----+-----+-----+
|           RT associated with the EVI (cont.)           |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

8 Acknowledgement

9 Security Considerations

Same security considerations as [RFC7432].

10 IANA Considerations

Allocation of Extended Community Type and Sub-Type for EVPN.

11 References

11.1 Normative References

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

[RFC4360] S. Sangli et al, "BGP Extended Communities Attribute", February, 2006.

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

11.2 Informative References

[ETREE-FMWK] Key et al., "A Framework for E-Tree Service over MPLS Network", draft-ietf-l2vpn-etree-frwk-03, work in progress, September 2013.

[PBB-EVPN] Sajassi et al., "PBB-EVPN", draft-ietf-l2vpn-pbb-evpn-05.txt, work in progress, October, 2013.

[RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for IGMP and MLD snooping PEs", RFC 4541, 2006.

Authors' Addresses

Ali Sajassi
Cisco
Email: sajassi@cisco.com

Samir Thoria
Cisco
Email: sthoria@cisco.com

Keyur Patel
Cisco
Email: keyur@arccus.com

Derek Yeung
Cisco
Email: Yeung@arccus.com

John Drake
Juniper
Email: jdrake@juniper.net

Wen Lin
Juniper
Email: wlin@juniper.net

BESS Workgroup
INTERNET-DRAFT
Intended Status: Standards Track

A. Sajassi
P. Brissette
Cisco
J. Uttaro
ATT
J. Drake
W. Lin
Juniper
S. Boutros
VMWare
J. Rabadan
Nokia

Expires: January 6, 2016

July 6, 2016

EVPN VPWS Flexible Cross-Connect Service
draft-sajassi-bess-evpn-vpws-fxc-00.txt

Abstract

This document describes a new EVPN VPWS VLAN-aware bundle service type referred to as flexible cross-connect service. It also describes the rationale for this new service as well as a solution to deliver such service.

Status of this Memo

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

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

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

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

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

Copyright and License Notice

Copyright (c) 2012 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1	Introduction	3
1.1	Terminology	3
2	Conflicting Requirements	4
4	Solution	6
4.1	VLAN-Unaware Flexible Xconnect - Single-Homing	7
4.2	VLAN-Aware Flexible Xconnect	8
4.3	VLAN-Unaware Flexible Xconnect - Multi-Homing	8
5	BGP Extensions	9
6	Failure Scenarios	10
6.2	EVPN VPWS service Failure	10
6.2	Attachment Circuit Failure	10
6.3	PE Port Failure	10
6.4	PE Node Failure	10
7	Security Considerations	10
8	IANA Considerations	10
9	References	11
9.1	Normative References	11
9.2	Informative References	11
	Authors' Addresses	11

1 Introduction

[EVPN-VPWS] describes a solution to deliver P2P services using BGP constructs defined in [RFC7432]. It delivers this P2P service between a pair of Attachment Circuits (ACs), where an AC can designate on a PE a port, a VLAN on a port, or a group of VLANs on a port. It also leverages multi-homing and fast convergence capabilities of [RFC7432] in delivering these VPWS services. Multi-homing capabilities include the support of single-active and all-active redundancy mode and fast convergence is provided using "mass withdraw" message in control-plane and fast protection switching using prefix independent convergence in data-plane upon node or link failure. Furthermore, the use of EVPN BGP constructs eliminates the need for multi-segment PW auto-discovery and signaling if the VPWS service need to span across multiple ASes.

Some service providers have very large number of ACs (in millions) that require tag manipulation (e.g., VLAN translation) to be back hauled across their MPLS/IP network. These service providers want to multiplex a large number of ACs across several physical interfaces (e.g., several Ethernet Segments) onto a single VPWS service tunnel in order to a) reduce number of EVPN service labels associated with VPWS service tunnels and thus the associated OAM monitoring, and b) reduce EVPN BGP signaling (e.g., not to signal each AC as it is the case in [EVPN-VPWS]).

These service provider want the above functionality without scarifying any of the capabilities of [EVPN-VPWS] including single-active and all-active multi-homing, and fast convergence.

This document presents a solution based on extensions to [EVPN-VPWS] to meet the above requirements.

1.1 Terminology

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

MAC: Media Access Control

MPLS: Multi Protocol Label Switching

OAM: Operations, Administration and Maintenance

PE: Provide Edge Node

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

EVPL: Ethernet Virtual Private Line

EPL: Ethernet Private Line

ES: Ethernet Segment

VPWS: Virtual private wire service

EVI: EVPN Instance

VPWS Service Tunnel: It is represented by a pair of EVPN service labels associated with a pair of endpoints. Each label is downstream assigned and advertised by the disposition PE through an Ethernet A-D per-EVI route. The downstream label identifies the endpoint on the disposition PE. A VPWS service tunnel can be associated with many VPWS service identifiers for VLAN-aware VPWS service where each identifier is a normalized VID.

Single-Active Mode: When a device or a network is multi-homed to two or more PEs and when only a single PE in such redundancy group can forward traffic to/from the multi-homed device or network for a given VLAN, then such multi-homing or redundancy is referred to as "Single-Active".

All-Active: When a device is multi-homed to two or more PEs and when all PEs in such redundancy group can forward traffic to/from the multi-homed device for a given VLAN, then such multi-homing or redundancy is referred to as "All-Active".

2 Conflicting Requirements

Two of the main motivations for service providers seeking a new solution are: 1) to reduce number of VPWS service tunnels by muxing large number of ACs across different physical interfaces instead of having one VPWS service tunnel per AC, and 2) to reduce the signaling of ACs as much as possible. Besides these two requirements, they also want multi-homing and fast convergence capabilities of [EVPN-VPWS].

In [EVPN-VPWS], a PE signals an AC indirectly by first associating that AC to a VPWS service tunnel (e.g., a VPWS service instance) and then signaling the VPWS service tunnel via a per-EVI Ethernet AD route with Ethernet Tag field set to a 24-bit VPWS service instance identifier (which is unique within the EVI) and ESI field set to a 10-octet identifier of the Ethernet Segment corresponding to that AC. Therefore, a PE device that receives such EVPN routes, can associate

the VPWS service tunnel to the remote Ethernet Segment, and when the remote ES fails and the PE receives the "mass withdraw" message associated with the failed ES per [RFC7432], it can update its BGP path list for that VPWS service tunnel quickly and achieve fast convergence for multi-homing scenarios. Even if fast convergence were not needed, there would still be a need for signaling each AC failure (via its corresponding VPWS service tunnel) associated with the failed ES, so that the BGP path list for each of them gets updated accordingly and the packets are sent to backup PE (in case of single-active multi-homing) or to other PEs in the redundancy group (in case of all-active multi-homing). In absence of updating the BGP path list, the traffic for that VPWS service tunnel will be black-holed.

When a single VPWS service tunnel multiplexes many ACs across number of Ethernet Segments (number of physical interfaces) and the ACs are not signaled via EVPN BGP to remote PE devices, then the remote PE devices neither know the association of the received Ethernet Segment to these ACs (and in turn to their local ACs) nor they know the association of the VPWS service tunnel (e.g., EVPN service label) to the far-end ACs - i.e, the remote PEs only know the association of their local ACs to the VPWS service tunnel but not the far-end ACs. Thus upon a connectivity failure to the ES, they don't know how to redirect traffic via another multi-homing PE to that ES. In other words, even if an ES failure is signaled via EVPN to the remote PE devices, they don't know what to do with such message because they don't know the association among the ES, their ACs, and the VPWS service tunnel.

In order to address this issue when multiplexing large number of ACs onto a single VPWS service tunnel, two mechanisms are devised: one to support VPWS services between two single-homed endpoints and another one to support VPWS services where one of the endpoints is multi-homed. An endpoint can be an AC, MAC-VRF, IP-VRF, global table, or etc.

For single-homed endpoints, it is OK not to signal each AC in BGP because upon connection failure to the ES, there is no alternative path to that endpoint. However, the ramification for not signaling an AC failure is that the traffic destined to the failed AC, is sent over MPLS/IP core and then gets discarded at the destination PE - i.e., it can waste network resources. However, when there is a connection failure, the application layer will eventually stop sending traffic and thus this wastage of network resources should be transient. Section 4.1 describes a solution for such single-homing VPWS service which is called VLAN-Unaware flexible cross-connect service.

For VPWS services where one of the endpoints is multi-homed, there

are two options:

1) to signal each AC via BGP so that the path list can be updated upon a failure that impacts those ACs. This solution is described in section 4.2 and it is called VLAN-Aware flexible cross-connect service.

2) to bundle several ACs on an ES together per destination ES (or PE) and associated such bundle to a single VPWS service tunnel. This is similar to VLAN-bundle service interface described in [EVPN-VPWS]. This solution is described in section 4.3.

4 Solution

This section describes a solution for providing a new VPWS service between two PE devices where a large number of ACs (e.g., VLANs) that span across many physical interfaces on each PE are multiplex onto a single P2P EVPN LSP tunnel. Since multiplexing is done across several physical interfaces, there can be overlapping VLAN IDs across these interfaces; therefore, in such scenarios, the VLAN IDs (VIDs) MUST be translated into unique VIDs to avoid collision. Furthermore, if the number of VLANs that are getting multiplex onto a single VPWS service tunnel, exceed 4K, then a single tag to double tag translation MUST be performed. This translation of VIDs into unique VIDs (either single or double) is referred to as "VID normalization". When single normalized VID is used, the lower 12-bit of Ethernet tag field in EVPN routes is set to that VID and when double normalized VID is used, the lower 12-bit of Ethernet tag field is set to inner VID and the higher 12-bit is set to the outer VID.

Since there is only a single P2P EVPN LSP tunnel associated with many normalized VIDs (either single or double), MPLS lookup at the disposition PE is no longer sufficient to forward the packet to the right egress endpoint/interface. Therefore, in addition to an EVPN label lookup corresponding to the VPWS service tunnel, a VID lookup (either single or double) is also required. On the disposition PE, one can think of the lookup of EVPN label results in identification of a VID table, and the lookup of normalized VID(s) in that table, results in identification of egress endpoint/interface. The tag manipulation (translation from normalized VID(s) to local VID) can be performed either as part of the VID table lookup or at the egress interface itself.

Since VID lookup (single or double) needs to be performed at the disposition PE, then VID normalization MUST be performed prior to the MPLS encapsulation on the ingress PE. This requires that both imposition and disposition PE devices be capable of VLAN tag

manipulation, such as re-write (single or double), addition, deletion (single or double), at their endpoints (e.g., their physical interfaces).

4.1 VLAN-Unaware Flexible Xconnect - Single-Homing

In this mode of operation, many ACs across several Ethernet Segments are multiplex into a single P2P EVPN LSP tunnel represented by a single VPWS service ID. VLAN-Unaware mode for this solution means that VLANs (normalized VIDs) are not signaled via EVPN BGP among the PEs. In this solution, there is only a single P2P EVPN LSP tunnel between a pair of PEs for all their ACs that are single-homed.

As discussed previously, since the VPWS service tunnel is used to multiplex ACs across different ES's (e.g., physical interfaces), the EVPN label alone is not sufficient for proper forwarding of the received packets (over MPLS/IP network) to egress interfaces. Therefore, normalized VID lookup is required in the disposition direction to forward packets to their proper egress end-points/interfaces - i.e., the EVPN label lookup identifies a VID table and subsequently, the normalized VID lookup in that table, identifies the egress interface.

In this solution, on each PE, the single-homing ACs represented by their normalized VIDs are associated with a single VPWS service tunnel (in a given EVI). The EVPN route that gets generated is an EVPN Ethernet AD per EVI route with ESI=0, Ethernet Tag field set to VPWS service instance ID, MPLS label field set to dynamically generated EVPN service label representing the EVPN VPWS service. This route is sent with an RT representing the EVI. This RT can be auto-generated from the EVI per section 5.1.2.1 of [EVPN-Overlay]. Furthermore, this route is sent with the EVPN Layer-2 Extended Community defined in section 3.1 of [EVPN-VPWS] with two new flags (defined in section 5) that indicate: 1) this VPWS service tunnel is for VLAN-unaware Flexible Cross-Connect, and 2) normalized VID type (single versus double). The receiving PE uses these new flags for consistency check and MAY generate an alarm if it detects inconsistency but doesn't bring down the VPWS service because such inconsistency may be intentional - i.e., one side is configured for VLAN-aware VPWS service and another side is configured for VLAN-unaware VPWS service.

It should be noted that in this mode of operation, a single Ethernet AD route is sent upon configuration of the first AC (ie, normalized VID). Later, when additional ACs are configured and associated with this EVPN VPWS service tunnel, the PE does not advertise any additional EVPN BGP routes. The PE only associates locally these ACs

with the already created VPWS service tunnel.

4.2 VLAN-Aware Flexible Xconnect

In this mode of operation, just as the VLAN-unaware mode, many normalized VIDs (ACs) across several different ES's/interfaces are multiplexed into a single P2P EVPN LSP tunnel; however, this single tunnel is represented by many VPWS service IDs (one per normalized VID) and these normalized VIDs are signaled using EVPN BGP.

In this solution, on each PE, the multi-homing ACs represented by their normalized VIDs are configured with a single EVI. There is no need to configure VPWS service instance ID in here. A VPWS service instance ID is derived automatically from each normalized VID. For each normalized VID on each ES, the PE generates an EVPN Ethernet AD per EVI route where ESI field represents the ES ID, the Ethernet Tag field is set to the normalized VID, MPLS label field is set to dynamically generated EVPN label representing the P2P EVPN LSP tunnel. This route is sent with an RT representing the EVI. As before, this RT can be auto-generated from the EVI per section 5.1.2.1 of [EVPN-Overlay]. Furthermore, this route is sent with the EVPN Layer-2 Extended Community defined in section 3.1 of [EVPN-VPWS] with two new flags (defined in section 5) that indicate: 1) this VPWS service tunnel is for VLAN-aware Flexible Cross-Connect, and 2) normalized VID type (single versus double). The receiving PE uses these new flags for consistency check and MAY generate an alarm if it detects inconsistency but doesn't bring down the VPWS service because such inconsistency may be intentional - i.e., one side is configured for VLAN-aware VPWS service and another side is configured for VLAN-unaware VPWS service.

It should be noted that in this mode of operation, the PE sends a single Ethernet AD route for each AC that is configured - i.e., each normalized VID that is configured per ES results in generation of an EVPN Ethernet AD per EVI.

This mode of operation provides automatic cross checking of normalized VIDs used for EVPL services because these VIDs are signaled in EVPN BGP. For example, if the same normalized VID is configured on three PE devices (instead of two) for the same EVI, then when a PE receives the second EVPN Eth-AD per EVI route, it generates an error message unless the two EVPN Eth-AD per EVI routes include the same ESI. Such cross-checking is not feasible in VLAN-unaware FXC because the normalized VIDs are not signaled.

4.3 VLAN-Unaware Flexible Xconnect - Multi-Homing

In this mode of operation, a group of normalized VIDs (ACs) on a single ES that are destined to a single endpoint/interface are multiplexed into a single P2P EVPN LSP tunnel represented by a single VPWS service ID. This mode of operation is the same as VLAN-bundle service interface of [EVPN-VPWS] except for the fact that VIDs on Ethernet frames are normalized before getting sent over the LSP tunnel.

In the previous two modes of operation, only a single EVPN VPWS service tunnel is needed per pair of PEs. However, in this mode of operation, there can be lot more service tunnels per pair of PEs - i.e, there is one tunnel per group of VIDs per pair of PEs and there can be many groups between a pair of PEs, thus resulting in many EVPN service tunnels.

5. BGP Extensions

This draft uses the EVPN Layer-2 attribute extended community defined in [EVPN-VPWS] with two additional flags added to this EC as described below. This EC is to be advertised with Ethernet A-D per EVI route per section 4.

```

+-----+
|  Type(0x06)/Sub-type(TBD)(2 octet)  |
+-----+
|  Control Flags (2 octets)            |
+-----+
|  L2 MTU (2 octets)                  |
+-----+
|  Reserved (2 octets)                |
+-----+

```

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|  MBZ           | V | M | C | P | B |   (MBZ = MUST Be Zero)
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The following bits in the Control Flags are defined; the remaining bits MUST be set to zero when sending and MUST be ignored when receiving this community.

Name	Meaning
B,P,C	per definition in [EVPN-VPWS]
M	00 mode of operation as defined in [EVPN-VPWS] 01 VLAN-aware FXC 10 VLAN-unaware FXC
V	00 operating per [EVPN-VPWS] 01 single-VID normalization 10 double-VID normalization

The M and V fields are OPTIONAL on transmission and ignored at reception for forwarding purposes. They are used for error notifications.

6 Failure Scenarios

6.2 EVPN VPWS service Failure

The failure detection of an EVPN VPWS service can be performed via OAM mechanisms such as VCCV-BFD and upon such failure detection, the switch over procedure to the backup S-PE is the same as the one described above.

6.2 Attachment Circuit Failure

6.3 PE Port Failure

6.4 PE Node Failure

In the case of PE node failure, the operation is similar to the steps described above, albeit that EVPN route withdrawals are performed by the Route Reflector instead of the PE.

7 Security Considerations

TBD.

8 IANA Considerations

TBD

9 References

9.1 Normative References

[RFC7432] Sajassi et al., "Ethernet VPN", RFC 7432, February 2015.

[EVPN-IRB] Sajassi et al., "Integrated Routing and Bridging in EVPN", draft-ietf-bess-evpn-inter-subnet-forwarding-00, work in progress, November 2014.

[EVPN-PREFIX] Rabadan et al., "IP Prefix Advertisement in EVPN", draft-ietf-bess-evpn-prefix-advertisement-02, work in progress, September 2015.

[RFC6718] Muley P., et al., "Pseudowire Redundancy", RFC 6718, August 2012.

[RFC6870] Muley P., et al., "Pseudowire Preferential Forwarding Status Bit", RFC 6870, February 2013.

9.2 Informative References

[BGP-PIC] Bashandy A. et al., "BGP Prefix Independent Convergence", draft-rtgwg-bgp-pic-02.txt, work in progress, October 2013.

Authors' Addresses

A. Sajassi
Cisco
EMail: sajassi@cisco.com

P. Brissette
Cisco
EMail: pbrisset@cisco.com

J. Uttaro
ATT

EMail: ju1738@att.com

J. Drake
Juniper
EMail: jdrake@juniper.net

S. Boutros
ATT
EMail: boutros.sami@gmail.com

W. Lin
Juniper
EMail: wlin@juniper.net

J. Rabadan
jorge.rabadan@nokia.com

BESS Workgroup
INTERNET-DRAFT
Intended Status: Standards Track

A. Sajassi
P. Brissette
Cisco
J. Uttaro
ATT
J. Drake
W. Lin
Juniper
S. Boutros
VMWare
J. Rabadan
Nokia

Expires: August 26, 2018

February 26, 2018

EVPN VPWS Flexible Cross-Connect Service
draft-sajassi-bess-evpn-vpws-fxc-03.txt

Abstract

This document describes a new EVPN VPWS service type specifically for multiplexing multiple attachment circuits across different Ethernet Segments and physical interfaces into a single EVPN VPWS service tunnel and still providing Single-Active and All-Active multi-homing. This new service is referred to as flexible cross-connect service. It also describes the rationale for this new service type as well as a solution to deliver such service.

Status of this Memo

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

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

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

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

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

Copyright and License Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1	Introduction	3
1.1	Terminology	3
2	Requirements	4
4	Solution	6
4.1	Flexible Xconnect	7
4.2	VLAN-Signaled Flexible Xconnect	8
4.2.1	Local Switching	9
5.	BGP Extensions	9
6	Failure Scenarios	11
6.1	EVPN VPWS service Failure	13
6.2	Attachment Circuit Failure	13
6.3	PE Port Failure	14
6.4	PE Node Failure	14
7	Security Considerations	14
8	IANA Considerations	14
9	References	14
9.1	Normative References	14
9.2	Informative References	15
	Authors' Addresses	15

1 Introduction

[RFC8214] describes a solution to deliver P2P services using BGP constructs defined in [RFC7432]. It delivers this P2P service between a pair of Attachment Circuits (ACs), where an AC can designate on a PE, a port, a VLAN on a port, or a group of VLANs on a port. It also leverages multi-homing and fast convergence capabilities of [RFC7432] in delivering these VPWS services. Multi-homing capabilities include the support of single-active and all-active redundancy mode and fast convergence is provided using "mass withdraw" message in control-plane and fast protection switching using prefix independent convergence in data-plane upon node or link failure [BGP-PIC]. Furthermore, the use of EVPN BGP constructs eliminates the need for multi-segment PW auto-discovery and signaling if the VPWS service need to span across multiple ASes.

Some service providers have very large number of ACs (in millions) that need to be back hauled across their MPLS/IP network. These ACs may or may not require tag manipulation (e.g., VLAN translation). These service providers want to multiplex a large number of ACs across several physical interfaces spread across one or more PEs (e.g., several Ethernet Segments) onto a single VPWS service tunnel in order to a) reduce number of EVPN service labels associated with EVPN-VPWS service tunnels and thus the associated OAM monitoring, and b) reduce EVPN BGP signaling (e.g., not to signal each AC as it is the case in [RFC8214]).

These service provider want the above functionality without scarifying any of the capabilities of [RFC8214] including single-active and all-active multi-homing, and fast convergence.

This document presents a solution based on extensions to [RFC8214] to meet the above requirements.

1.1 Terminology

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

MAC: Media Access Control

MPLS: Multi Protocol Label Switching

OAM: Operations, Administration and Maintenance

PE: Provide Edge Node

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

EVPL: Ethernet Virtual Private Line

EPL: Ethernet Private Line

ES: Ethernet Segment

VPWS: Virtual private wire service

EVI: EVPN Instance

VPWS Service Tunnel: It is represented by a pair of EVPN service labels associated with a pair of endpoints. Each label is downstream assigned and advertised by the disposition PE through an Ethernet A-D per-EVI route. The downstream label identifies the endpoint on the disposition PE. A VPWS service tunnel can be associated with many VPWS service identifiers for VLAN-signaled VPWS service where each identifier is a normalized VID.

Single-Active Mode: When a device or a network is multi-homed to two or more PEs and when only a single PE in such redundancy group can forward traffic to/from the multi-homed device or network for a given VLAN, then such multi-homing or redundancy is referred to as "Single-Active".

All-Active: When a device is multi-homed to two or more PEs and when all PEs in such redundancy group can forward traffic to/from the multi-homed device for a given VLAN, then such multi-homing or redundancy is referred to as "All-Active".

2 Requirements

Two of the main motivations for service providers seeking a new solution are: 1) to reduce number of VPWS service tunnels by multiplexing large number of ACs across different physical interfaces instead of having one VPWS service tunnel per AC, and 2) to reduce the signaling of ACs as much as possible. Besides these two requirements, they also want multi-homing and fast convergence capabilities of [RFC8214].

In [RFC8214], a PE signals an AC indirectly by first associating that AC to a VPWS service tunnel (e.g., a VPWS service instance) and then signaling the VPWS service tunnel via a per-EVI Ethernet AD route with Ethernet Tag field set to a 24-bit VPWS service instance identifier (which is unique within the EVI) and ESI field set to a 10-octet identifier of the Ethernet Segment corresponding to that AC.

Therefore, a PE device that receives such EVPN routes, can associate the VPWS service tunnel to the remote Ethernet Segment, and when the remote ES fails and the PE receives the "mass withdraw" message associated with the failed ES per [RFC7432], it can update its BGP path list for that VPWS service tunnel quickly and achieve fast convergence for multi-homing scenarios. Even if fast convergence were not needed, there would still be a need for signaling each AC failure (via its corresponding VPWS service tunnel) associated with the failed ES, so that the BGP path list for each of them gets updated accordingly and the packets are sent to backup PE (in case of single-active multi-homing) or to other PEs in the redundancy group (in case of all-active multi-homing). In absence of updating the BGP path list, the traffic for that VPWS service tunnel will be black-holed.

When a single VPWS service tunnel multiplexes many ACs across number of Ethernet Segments (number of physical interfaces) and the ACs are not signaled via EVPN BGP to remote PE devices, then the remote PE devices neither know the association of the received Ethernet Segment to these ACs (and in turn to their local ACs) nor they know the association of the VPWS service tunnel (e.g., EVPN service label) to the far-end ACs - i.e, the remote PEs only know the association of their local ACs to the VPWS service tunnel but not the far-end ACs. Thus upon a connectivity failure to the ES, they don't know how to redirect traffic via another multi-homing PE to that ES. In other words, even if an ES failure is signaled via EVPN to the remote PE devices, they don't know what to do with such message because they don't know the association among the remote ES, the remote ACs, and the VPWS service tunnel.

In order to address this issue when multiplexing large number of ACs onto a single VPWS service tunnel, two mechanisms are devised: one to support VPWS services between two single-homed endpoints and another one to support VPWS services where one of the endpoints is multi-homed. An endpoint can be an AC, MAC-VRF, IP-VRF, global table, or etc.

For single-homed endpoints, it is OK not to signal each AC in BGP because upon connection failure to the ES, there is no alternative path to that endpoint. However, the ramification for not signaling an AC failure is that the traffic destined to the failed AC, is sent over MPLS/IP core and then gets discarded at the destination PE - i.e., it can waste network resources. However, when there is a connection failure, the application layer will eventually stop sending traffic and thus this wastage of network resources should be transient. Section 4.1 describes a solution for such single-homing VPWS service.

For VPWS services where one of the endpoints is multi-homed, there

are two options:

1) to signal each AC via BGP so that the path list can be updated upon a failure that impacts those ACs. This solution is described in section 4.2 and it is called VLAN-signaled flexible cross-connect service.

2) to bundle several ACs on an ES together per destination end-point (e.g., ES, MAC-VRF, etc.) and associated such bundle to a single VPWS service tunnel. This is similar to VLAN-bundle service interface described in [RFC8214]. This solution is described in section 4.3.

4 Solution

This section describes a solution for providing a new VPWS service between two PE devices where a large number of ACs (e.g., VLANs) that span across many Ethernet Segments (i.e., physical interfaces) on each PE are multiplex onto a single P2P EVPN service tunnel. Since multiplexing is done across several physical interfaces, there can be overlapping VLAN IDs across these interfaces; therefore, in such scenarios, the VLAN IDs (VIDs) MUST be translated into unique VIDs to avoid collision. Furthermore, if the number of VLANs that are getting multiplex onto a single VPWS service tunnel, exceed 4K, then a single tag to double tag translation MUST be performed. This translation of VIDs into unique VIDs (either single or double) is referred to as "VID normalization". When single normalized VID is used, the lower 12-bit of Ethernet tag field in EVPN routes is set to that VID and when double normalized VID is used, the lower 12-bit of Ethernet tag field is set to inner VID and the higher 12-bit is set to the outer VID.

Since there is only a single EVPN VPWS service tunnel associated with many normalized VIDs (either single or double) across multiple physical interfaces, MPLS lookup at the disposition PE is no longer sufficient to forward the packet to the right egress endpoint/interface. Therefore, in addition to an EVPN label lookup corresponding to the VPWS service tunnel, a VID lookup (either single or double) is also required. On the disposition PE, one can think of the lookup of EVPN label results in identification of a VID-VRF, and the lookup of normalized VID(s) in that table, results in identification of egress endpoint/interface. The tag manipulation (translation from normalized VID(s) to local VID) can be performed either as part of the VID table lookup or at the egress interface itself.

Since VID lookup (single or double) needs to be performed at the

disposition PE, then VID normalization MUST be performed prior to the MPLS encapsulation on the ingress PE. This requires that both imposition and disposition PE devices be capable of VLAN tag manipulation, such as re-write (single or double), addition, deletion (single or double) at their endpoints (e.g., their ES's, MAC-VRFs, IP-VRFs, etc.).

4.1 Flexible Xconnect

In this mode of operation, many ACs across several Ethernet Segments are multiplex into a single EVPN VPWS service tunnel represented by a single VPWS service ID. This is the default mode of operation for FXC and the participating PEs do not need to signal the VLANs (normalized VIDs) in EVPN BGP.

With respect to the data-plane aspects of the solution, both imposition and disposition PEs are aware of the VLANs as the imposition PE performs VID normalization and the disposition PE does VID lookup and translation. In this solution, there is only a single P2P EVPN VPWS service tunnel between a pair of PEs for a set of ACs.

As discussed previously, since the EVPN VPWS service tunnel is used to multiplex ACs across different ES's (e.g., physical interfaces), the EVPN label alone is not sufficient for proper forwarding of the received packets (over MPLS/IP network) to egress interfaces. Therefore, normalized VID lookup is required in the disposition direction to forward packets to their proper egress end-points - i.e., the EVPN label lookup identifies a VID-VRF and subsequently, the normalized VID lookup in that table, identifies the egress interface.

This mode of operation is only suitable for single-homing because in multi-homing the association between EVPN VPWS service tunnel and remote AC changes during the failure and therefore the VLANs (normalized VIDs) need to be signaled.

In this solution, on each PE, the single-homing ACs represented by their normalized VIDs are associated with a single EVPN VPWS service tunnel (in a given EVI). The EVPN route that gets generated is an EVPN Ethernet AD per EVI route with ESI=0, Ethernet Tag field set to VPWS service instance ID, MPLS label field set to dynamically generated EVPN service label representing the EVPN VPWS service tunnel. This route is sent with an RT representing the EVI. This RT can be auto-generated from the EVI per section 5.1.2.1 of [EVPN-Overlay]. Furthermore, this route is sent with the EVPN Layer-2 Extended Community defined in section 3.1 of [RFC8214] with two new flags (defined in section 5) that indicate: 1) this VPWS service

tunnel is for default Flexible Cross-Connect, and 2) normalized VID type (single versus double). The receiving PE uses these new flags for consistency check and MAY generate an alarm if it detects inconsistency but doesn't bring down the VPWS service.

It should be noted that in this mode of operation, a single Ethernet AD per EVI route is sent upon configuration of the first AC (ie, normalized VID). Later, when additional ACs are configured and associated with this EVPN VPWS service tunnel, the PE does not advertise any additional EVPN BGP routes. The PE only associates locally these ACs with the already created VPWS service tunnel.

The default FXC mode can be used for multi-homing. In this mode, a group of normalized VIDs (ACs) on a single Ethernet segment that are destined to a single endpoint are multiplexed into a single EVPN VPWS service tunnel represented by a single VPWS service ID. When the default FXC mode is used for multi-homing, instead of a single EVPN VPWS service tunnel, there can be many service tunnels per pair of PEs - i.e, there is one tunnel per group of VIDs per pair of PEs and there can be many groups between a pair of PEs, thus resulting in many EVPN service tunnels.

4.2 VLAN-Signaled Flexible Xconnect

In this mode of operation, just as the default FXC mode in section 4.1, many normalized VIDs (ACs) across several different ES's/interfaces are multiplexed into a single EVPN VPWS service tunnel; however, this single tunnel is represented by many VPWS service IDs (one per normalized VID) and these normalized VIDs are signaled using EVPN BGP.

In this solution, on each PE, the multi-homing ACs represented by their normalized VIDs are configured with a single EVI. There is no need to configure VPWS service instance ID in here as it is the same as the normalized VID. For each normalized VID on each ES, the PE generates an EVPN Ethernet AD per EVI route where ESI field represents the ES ID, the Ethernet Tag field is set to the normalized VID, MPLS label field is set to dynamically generated EVPN label representing the P2P EVPN service tunnel and it is the same label for all the ACs that are multiplexed into a single EVPN VPWS service tunnel. This route is sent with an RT representing the EVI. As before, this RT can be auto-generated from the EVI per section 5.1.2.1 of [EVPN-Overlay]. Furthermore, this route is sent with the EVPN Layer-2 Extended Community defined in section 3.1 of [RFC8214] with two new flags (defined in section 5) that indicate: 1) this VPWS service tunnel is for VLAN-signaled Flexible Cross-Connect, and 2) normalized VID type (single versus double). The receiving PE uses

these new flags for consistency check and MAY generate an alarm if it detects inconsistency but doesn't bring down the VPWS service.

It should be noted that in this mode of operation, the PE sends a single Ethernet AD route for each AC that is configured - i.e., each normalized VID that is configured per ES results in generation of an EVPN Ethernet AD per EVI.

This mode of operation provides automatic cross checking of normalized VID's used for EVPL services because these VID's are signaled in EVPN BGP. For example, if the same normalized VID is configured on three PE devices (instead of two) for the same EVI, then when a PE receives the second EVPN EAD per-EVI route, it generates an error message unless the two EVPN EAD per-EVI routes include the same ESI. Such cross-checking is not feasible in default FXC mode because the normalized VID's are not signaled.

4.2.1 Local Switching

When cross-connection is between two ACs belonging to two multi-homed Ethernet Segments on the same set of multi-homing PEs, then forwarding between the two ACs MUST be performed locally during normal operation (e.g., in absence of a local link failure) - i.e., the traffic between the two ACs MUST be locally switched within the PE.

In terms of control plane processing, this means that when the receiving PE receives an Ethernet A-D per-EVI route whose ESI is a local ESI, the PE does not alter its forwarding state based on the received route. This ensures that the local switching takes precedence over forwarding via MPLS/IP network. This scheme of locally switched preference is consistent with baseline EVPN [RFC 7432] where it describes the locally switched preference for MAC/IP routes.

In such scenarios, the Ethernet A-D per EVI route should be advertised with the MPLS label either associated with the destination Attachment Circuit or with the destination Ethernet Segment in order to avoid any ambiguity in forwarding. In other words, the MPLS label cannot represent the same VID-VRF used in section 4.2 because the same normalized VID can be reachable via two Ethernet Segments. In case of using MPLS label per destination AC, then this same solution can be used for VLAN-based VPWS or VLAN-bundle VPWS services per [RFC8214].

5. BGP Extensions

This draft uses the EVPN Layer-2 attribute extended community defined in [RFC8214] with two additional flags added to this EC as described below. This EC is to be advertised with Ethernet A-D per EVI route per section 4.

```

+-----+
| Type(0x06)/Sub-type(TBD)(2 octet) |
+-----+
| Control Flags (2 octets)           |
+-----+
| L2 MTU (2 octets)                 |
+-----+
| Reserved (2 octets)                |
+-----+

```

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+-----+
| MBZ           | V | M | C | P | B | (MBZ = MUST Be Zero)
+-----+

```

The following bits in the Control Flags are defined; the remaining bits MUST be set to zero when sending and MUST be ignored when receiving this community.

Name	Meaning
B,P,C	per definition in [RFC8214]
M	00 mode of operation as defined in [RFC8214] 01 VLAN-Signaled FXC 10 Default FXC
V	00 operating per [RFC8214] 01 single-VID normalization 10 double-VID normalization

The M and V fields are OPTIONAL on transmission and ignored at reception for forwarding purposes. They are used for error notifications.

6 Failure Scenarios

Two examples will be used as an example to analyze the failure scenarios.

The first scenario is depicted in Figure 1 and shows the VLAN-signaled FXC mode with Multi-Homing. In this example:

- CE1 is connected to PE1 and PE2 via (port,vid)=(p1,1) and (p3,3) respectively. CE1's VIDs are normalized to value 1 on both PEs, and CE1 is Xconnected to CE3's VID 1 at the remote end.
- CE2 is connected to PE1 and PE2 via ports p2 and p4 respectively:
 - o (p2,1) and (p4,3) identify the ACs that are used to Xconnect CE2 to CE4's VID 2, and are normalized to value 2.
 - o (p2,2) and (p4,4) identify the ACs that are used to Xconnect CE2 to CE5's VID 3, and are normalized to value 3.

In this scenario, PE1 and PE2 advertise an AD per-EVI route per normalized VID (values 1, 2 and 3), however only two VPWS Service Tunnels are needed: VPWS Service Tunnel 1 (sv.T1) between PE1's FXC service and PE3's FXC, and VPWS Service Tunnel 2 (sv.T2) between PE2's FXC and PE3's FXC.

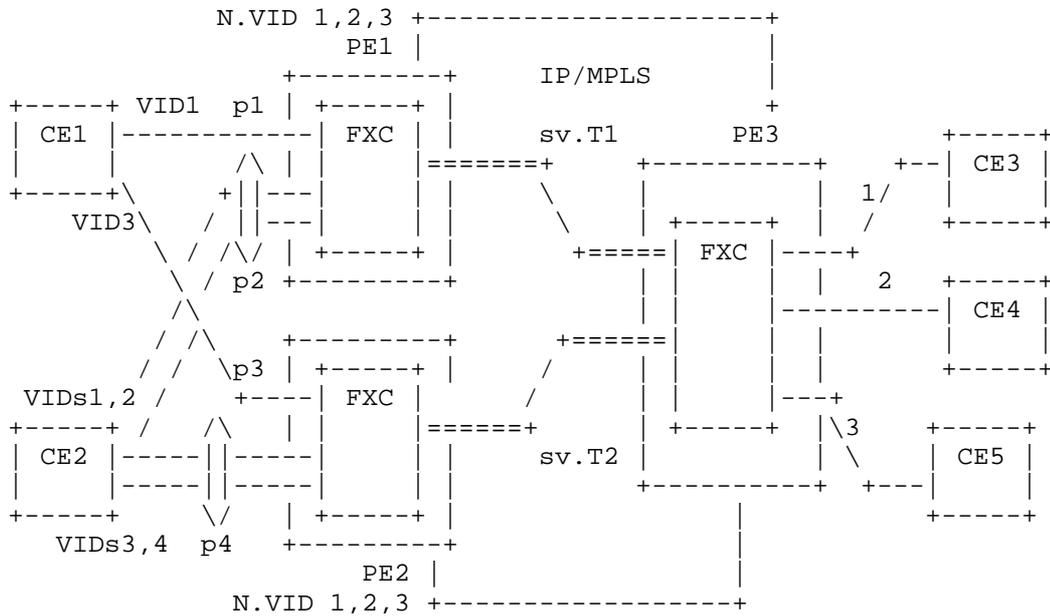


Figure 1 VLAN-Signaled Flexible Xconnect

The second scenario is a default Flexible Xconnect with Multi- Homing solution and it is depicted in Figure 2. In this case, the same VID Normalization as in the previous example is performed, however there is not an individual AD per-EVI route per normalized VID, but per bundle of ACs on an ES. That is, PE1 will advertise two AD per-EVI routes: the first one will identify the ACs on p1's ES and the second one will identify the AC2 in p2's ES. Similarly, PE2 will advertise two AD per-EVI routes.

black-hole. Application layer OAM may be used if per-VLAN fault propagation is required in this case.

6.3 PE Port Failure

In case of PE port Failure, the failure will be signaled and the other PE will take over in both cases:

- o VLAN-signaled FXC (Figure 1): a port failure, e.g. p2, triggers the withdrawal of the AD per-EVI routes for Normalized VIDs 2 and 3, as well as the withdrawal of the AD per-ES route for p2's ES. Upon receiving the fault notification, PE3 will withdraw PE1 from its path-list for the traffic coming from CE4 and CE5.

- o Default FXC (Figure 2): a port failure, e.g. p2, is signaled by route for sv.T2 will also be withdrawn. Upon receiving the fault notification, PE3 will remove PE1 from its path-list for traffic coming from CE4 and CE5.

6.4 PE Node Failure

In the case of PE node failure, the operation is similar to the steps described above, albeit that EVPN route withdrawals are performed by the Route Reflector instead of the PE.

7 Security Considerations

There are no additional security considerations beyond what is already specified in [RFC8214].

8 IANA Considerations

TBD.

9 References

9.1 Normative References

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

[RFC7432] Sajassi et al., "Ethernet VPN", RFC 7432, February 2015.

[RFC8214] Boutros et al., "Virtual Private Wire Service Support in Ethernet VPN", RFC 8214, August 2015.

9.2 Informative References

[BGP-PIC] Bashandy A. et al., "BGP Prefix Independent Convergence", draft-rtgwg-bgp-pic-02.txt, work in progress, October 2013.

[EVPN-Overlay] Sajassi et al., "A Network Virtualization Overlay Solution using EVPN", draft-ietf-bess-evpn-overlay-12, work in progress, February 2018.

Authors' Addresses

A. Sajassi
Cisco
EMail: sajassi@cisco.com

P. Brissette
Cisco
EMail: pbrisset@cisco.com

J. Uttaro
ATT
EMail: jul738@att.com

J. Drake
Juniper
EMail: jdrake@juniper.net

S. Boutros
ATT
EMail: boutros.sami@gmail.com

W. Lin
Juniper
EMail: wlin@juniper.net

J. Rabadan
jorge.rabadan@nokia.com

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
S. Sathappan
K. Nagaraj
Nokia

M. Miyake
T. Matsuda
Softbank

Expires: January 8, 2017

July 7, 2016

PBB-EVPN ISID-based CMAC-Flush
draft-snr-bess-pbb-evpn-isid-cmacflush-00

Abstract

RFC7623 defines how Provider Backbone Bridging (PBB) can be combined with Ethernet VPN (EVPN) to deploy ELAN services in very large MPLS networks. RFC7623 also describes how Single-Active Multi-homing and per-ISID Load-Balancing can be provided to access devices and aggregation networks. In order to speed up the network convergence in case of failures on Single-Active Multi-Homed Ethernet Segments, RFC7623 defines a CMAC-Flush mechanism that works for different Ethernet Segment BMAC address allocation models. This document complements those CMAC-Flush procedures for cases in which no PBB-EVPN Ethernet Segments are defined (ESI 0) and an ISID-based CMAC-Flush granularity is desired.

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on January 8, 2017.

Copyright Notice

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

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

Table of Contents

1. Problem Statement	2
2. Solution requirements	4
3. EVPN BGP Encoding for ISID-based CMAC-flush	5
4. Solution description	6
4.1 ISID-based CMAC-Flush activation procedures	6
4.2 CMAC-Flush generation	7
4.3 CMAC-Flush process upon receiving a CMAC-Flush notification	7
5. Conclusions	8
6. Conventions used in this document	8
7. Security Considerations	9
8. IANA Considerations	9
9. References	9
9.1 Normative References	9
9.2 Informative References	9
10. Acknowledgments	9
11. Contributors	9
17. Authors' Addresses	10

1. Problem Statement

RFC7623 defines how Provider Backbone Bridging (PBB) can be combined

with Ethernet VPN (EVPN) to deploy ELAN services in very large MPLS networks. RFC7623 also describes how Single-Active Multi-homing and per-ISID Load-Balancing can be provided to access devices and aggregation networks. When Access Ethernet/MPLS Networks exists, [vES] describes how virtual ES can be associated to a group of Ethernet Virtual Circuits (EVCs) or even Pseudowires (PWs). In order to speed up the network convergence in case of failures on Single-Active Multi-Homed Ethernet Segments, RFC7623 defines a CMAC-Flush mechanism that works for different Ethernet Segment BMAC address allocation models.

In some cases, the administrative entities that manage the access devices or aggregation networks, don't demand Multi-Homing Ethernet Segments (ES) from the PBB-EVPN provider, but simply multiple single-homed ES. If that is the case, the PBB-EVPN network is no longer aware of the redundancy offered by the access administrative entity. Figure 1 shows an example where the PBB-EVPN network provides four different Attachment Circuits (ACs) for ISID1, with those ACs not being part of any ES or vES (therefore they are referred to as null vES).

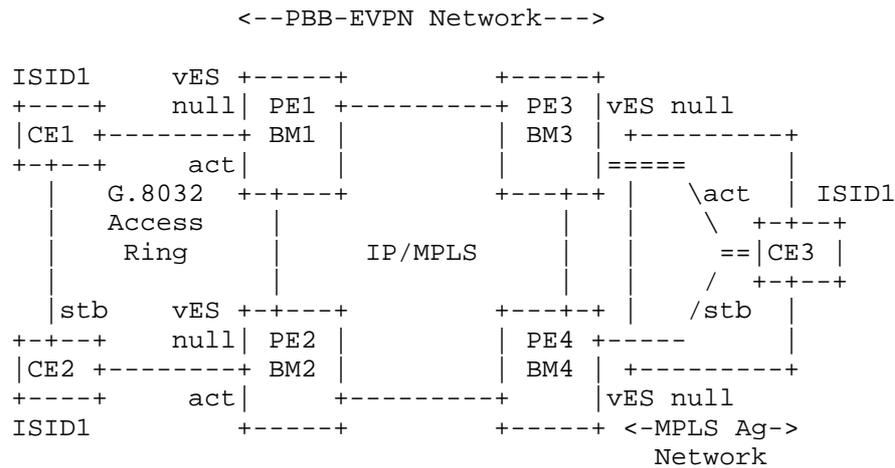


Figure 1 PBB-EVPN and non-ES based redundancy

In the example in Figure 1, CE1 and CE2 provide redundant connectivity for ISID1 through the use of G.8032 Ethernet Ring Protection Switching. CE3 provides redundant active-standby PW connectivity for ISID1. In the two cases the ACs are connected to null ES, hence the PEs will keep their ACs active and the CEs will be responsible for the per-ISID load balancing while avoiding loops.

For instance, CE2 will block its link to CE1 and CE3 will block its

forwarding path to PE4. In this situation, a failure in one of the redundant ACs will make the CEs to start using their redundant paths, however those failures will not trigger any CMAC-Flush procedures in the PEs. For example, if the active PW from CE3 fails, PE3 will not issue any CMAC-Flush message and therefore the remote PEs will continue pointing at PE3's BMAC to reach CE3's CMACs, until the CMACs age out in the ISID1 FDBs.

RFC7623 provides a CMAC-Flush solution based on a shared BMAC update along with the MAC Mobility extended community where the sequence number is incremented. However, while that procedure could be used in the example of Figure 1, it would result in unnecessary flushing of unaffected ISIDs on the remote PEs, and subsequent flooding.

This document describes an extension of the RFC7623 CMAC-Flush procedures, so that in the above failure example, PE3 can trigger a CMAC-Flush notification that makes PE1, PE2 and PE4 flush all the CMACs associated to PE3's BMAC and (only) ISID1. This new CMAC-Flush procedure explained in this document will be referred to as "PBB-EVPN ISID-based CMAC-Flush" and can be used in PBB-EVPN networks with null or non-null (virtual) Ethernet Segments.

2. Solution requirements

The following requirements must be met by the CMAC-Flush solution described in this document:

- a) The solution MUST solve black-hole scenarios in case of failures on null ES ACs (Attachment Circuits not associated to ES, that is, ESI=0) when the access device/network is responsible for the redundancy.
- b) This extension SHOULD work with Single-Active non-null ES and virtual ES, irrespective of the PE BMAC address assignment (dedicated per-ES BMAC or shared BMAC).
- c) In case of failure on the egress PE, the solution MUST provide a CMAC-Flush notification at BMAC AND ISID granularity level.
- d) The solution MUST provide a reliable CMAC-Flush notification in PBB-EVPN networks that use Route-Reflectors (RRs).
- e) The solution MUST coexist in RFC7623-compliant networks where there are systems not supporting this extension.
- f) The solution SHOULD be enabled/disabled by an administrative option on a per-PE and per-ISID basis.

3. EVPN BGP Encoding for ISID-based CMAC-flush

The solution does not use any new BGP attributes but reuses the MAC Mobility extended community as an indication of CMAC-Flush (as in RFC7623) and encodes the ISID in the Ethernet Tag field of the MAC/IP route. As a reference, Figure 2 shows the MAC Mobility extended community and the MAC/IP route that are used in this document as a CMAC-Flush notification.

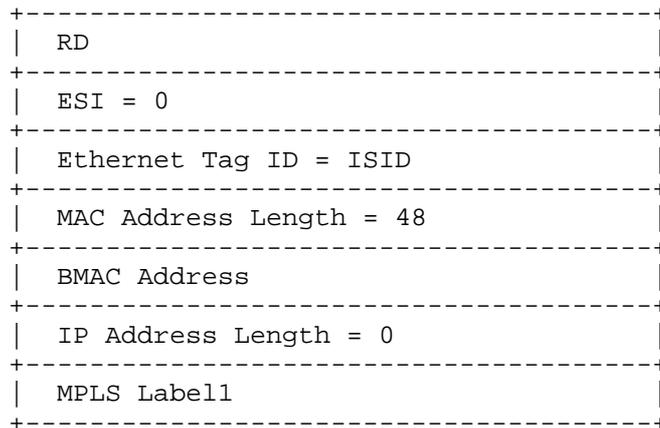
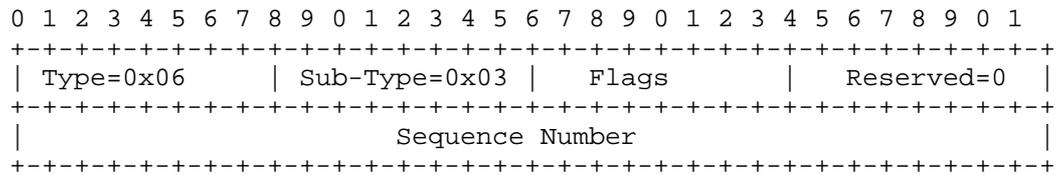


Figure 2 CMAC-Flush notification encoding: BMAC/ISID route

Where:

- o The route's RD and RT are the ones corresponding to its EVI. Alternatively to the EVI's RT, the route MAY be tagged with an RT auto-derived from the Ethernet Tag (ISID) instead. RFC7623 describes how the RT can be derived from the ISID.
- o The Ethernet Tag encodes the ISID for which the PE that receives the route must flush the CMACs upon reception of the route.
- o The MAC address field encodes the BMAC Address for which the PE that receives the route must flush the CMACs upon reception of the

route.

- o The MAC Mobility extended community is used as in RFC7623, where a delta in the sequence number between two updates for the same BMAC/ISID will be interpreted as a CMAC-flush notification for the corresponding BMAC and ISID.

All the other fields are set and used as defined in RFC7623. This document will refer to this route as the BMAC/ISID route, as opposed to the RFC7623 BMAC/0 route (BMAC route sent with Ethernet Tag = 0).

Note that this BMAC/ISID route will be accepted and reflected by any RFC7432-compliant RR, since no new attributes or values are used. A PE receiving the route will process the received BMAC/ISID update only in case of supporting the procedures described in this document.

4. Solution description

Figure 1 will be used in the description of the solution. CE1, CE2 and CE3 are connected to ACs associated to ISID1, where no (Multi-Homed) Ethernet Segments have been enabled. All the ACs are operationally active and ready to forward frames.

Enabling or disabling ISID-based CMAC-Flush SHOULD be an administrative choice on the system that MAY be configured per ISID (I-Component). When enabled on a PE:

- a) The PE will be able to generate BMAC/ISID routes as CMAC-Flush notifications for the remote PEs.
- b) The PE will be able to process BMAC/ISID routes received from remote PEs.

When ISID-based CMAC-Flush is disabled, the PE will follow the RFC7623 procedures for CMAC-flush.

These new CMAC-flush procedures are described in sections 4.1, 4.2 and 4.3 respectively:

- o ISID-based CMAC-flush activation
- o CMAC-flush notification generation upon AC failures
- o CMAC-flush process upon receiving a CMAC-Flush notification

4.1 ISID-based CMAC-Flush activation procedures

The following behavior MUST be followed by the PBB-EVPN PEs (see Figure 1):

- o As in RFC7623, each PE has previously advertised a shared BMAC in a BMAC/0 route (BM1, BM2, BM3 and BM4 respectively). This is the BMAC that each PE will use as BMAC SA (Source Address) when encapsulating the frames received on any local single-homed AC. Each PE will import the received BMAC/0 routes from the remote PEs and will install the BMACs in its B-component MAC-VRF. For instance, PE1 will advertise BM1/0 and will install BM2, BM3 and BM4 in its MAC-VRF.
- o Assuming ISID-based CMAC-Flush is activated for ISID 1, the PEs will advertise the shared BMAC with ISID 1 encoded in the Ethernet Tag. That is, PE1 will advertise BM1/1 and will receive BM2/1, BM3/1 and BM4/1. The receiving PEs MUST use these BMAC/ISID routes only for CMAC-Flush procedures and they MUST NOT be used to add/withdraw any BMAC entry in the MAC-VRFs. As per RFC7623, only BMAC/0 routes can be used to add/withdraw BMACs in the MAC-VRFs.
- o The above procedure MAY also be used for dedicated BMACs.

4.2 CMAC-Flush generation

If, for instance, there is a failure on PE1's AC, PE1 will generate an update including BM1/1 along with the MAC Mobility extended community where the Sequence Number has been incremented. The reception of the BM1/1 with a delta in the sequence number will trigger the CMAC-Flush procedures on the receiving PEs.

- o An AC going operationally down MUST generate a BMAC/ISID with a higher Sequence Number. If the AC going down makes the entire local ISID go operationally down, the PE will withdraw the BMAC/ISID route for the ISID.
- o An AC going operationally up SHOULD NOT generate any BMAC/ISID update, unless it activates its corresponding ISID, in which case the PE will advertise the BMAC/ISID route.
- o An AC receiving a CMAC-Flush notification from the access network, e.g. by G.8032, MAY propagate it to the remote PEs by generating a BMAC/ISID update with higher Sequence Number.

4.3 CMAC-Flush process upon receiving a CMAC-Flush notification

A PE receiving a CMAC-Flush notification will follow these procedures:

- o A received BMAC/ISID route (with non-zero ISID) MUST NOT add/remove any BMAC to/from the MAC-VRF.
- o An update of a previously received BMAC/ISID route with a delta Sequence Number, MUST flush all the CMACs associated to that ISID and BMAC. CMACs associated to the same ISID but different BMAC MUST NOT be flushed.
- o A received BMAC/ISID withdraw (with non-zero ISID) MUST flush all the CMACs associated to that BMAC and ISID.

Note that the CMAC-Flush procedures described in RFC7623 for BMAC/0 routes are still valid and a PE receiving RFC7623 CMAC-flush notification messages MUST observe the behavior specified in RFC7623.

5. Conclusions

The ISID-based CMAC-Flush solution described in this document has the following benefits:

- a) The solution solves black-hole scenarios in case of failures on null ES ACs, since the CMAC-flush procedures are independent of the Ethernet Segment definition.
- b) This extension can also be used with Single-Active non-null ES and virtual ES, irrespective of the PE BMAC address assignment (dedicated per-ES BMAC or shared BMAC).
- c) It provides a CMAC-Flush notification at BMAC AND ISID granularity level, therefore flushing a minimum number of CMACs and reducing the amount of flooding in the network.
- d) It provides a reliable CMAC-Flush notification in PBB-EVPN networks that use RRs. RRs will propagate the CMAC-flush notifications for all the affected ISIDs and irrespective of the order in which the notifications make it to the RR.
- e) The solution can coexist in a network with systems supporting or not supporting the CMAC-flush extensions.

6. Conventions used in this document

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

In this document, the characters ">>" preceding an indented line(s) indicates a compliance requirement statement using the key words listed above. This convention aids reviewers in quickly identifying or finding the explicit compliance requirements of this RFC.

7. Security Considerations

This section will be added in future versions.

8. IANA Considerations

9. References

9.1 Normative References

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

[RFC7623]Sajassi, A., Ed., Salam, S., Bitar, N., Isaac, A., and W. Henderickx, "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, September 2015, <<http://www.rfc-editor.org/info/rfc7623>>.

9.2 Informative References

[vES] Sajassi et al. "EVPN Virtual Ethernet Segment", draft-sajassi-bess-evpn-virtual-eth-segment-01, work-in-progress, July 6, 2015.

10. Acknowledgments

The authors want to thank Vinod Prabhu, Sriram Venkateswaran, Laxmi Padakanti, Ranganathan Boovaraghavan for their review and contributions.

11. Contributors

17. Authors' Addresses

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

Senthil Sathappan
Nokia
Email: senthil.sathappan@nokia.com

Kiran Nagaraj
Nokia
Email: kiran.nagaraj@nokia.com

Masahiro Miyake
Softbank
Email: masahiro.miyake@g.softbank.co.jp

Taku Matsuda
Softbank
Email: taku.matsuda@g.softbank.co.jp

BESS Workgroup
Internet Draft
Intended status: Standards Track

J. Rabadan, Ed.
S. Sathappan
K. Nagaraj
Nokia

M. Miyake
T. Matsuda
Softbank

Expires: January 27, 2020

July 26, 2019

PBB-EVPN ISID-based CMAC-Flush
draft-snr-bess-pbb-evpn-isid-cmacflush-06

Abstract

Provider Backbone Bridging (PBB) can be combined with Ethernet VPN (EVPN) to deploy ELAN services in very large MPLS networks (PBB-EVPN). Single-Active Multi-homing and per-ISID Load-Balancing can be provided to access devices and aggregation networks. In order to speed up the network convergence in case of failures on Single-Active Multi-Homed Ethernet Segments, PBB-EVPN defines a CMAC-Flush mechanism that works for different Ethernet Segment BMAC address allocation models. This document complements those CMAC-Flush procedures for cases in which no PBB-EVPN Ethernet Segments are defined (ESI 0) and an ISID-based CMAC-Flush granularity is desired.

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on January 27, 2020.

Copyright Notice

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

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

Table of Contents

- 1. Problem Statement 2
- 2. Solution requirements 4
- 3. EVPN BGP Encoding for ISID-based CMAC-flush 5
- 4. Solution description 6
 - 4.1 ISID-based CMAC-Flush activation procedures 7
 - 4.2 CMAC-Flush generation 7
 - 4.3 CMAC-Flush process upon receiving a CMAC-Flush notification 8
- 5. Conclusions 8
- 6. Conventions used in this document 9
- 7. Security Considerations 9
- 8. IANA Considerations 9
- 9. References 9
 - 9.1 Normative References 9
 - 9.2 Informative References 10
- 10. Acknowledgments 10
- 11. Contributors 10
- 17. Authors' Addresses 10

1. Problem Statement

[RFC7623] defines how Provider Backbone Bridging (PBB) can be

forwarding path to PE4. In this situation, a failure in one of the redundant ACs will make the CEs to start using their redundant paths, however those failures will not trigger any CMAC-Flush procedures in the PEs. For example, if the active PW from CE3 fails, PE3 will not issue any CMAC-Flush message and therefore the remote PEs will continue pointing at PE3's BMAC to reach CE3's CMACs, until the CMACs age out in the ISID1 FDBs.

[RFC7623] provides a CMAC-Flush solution based on a shared BMAC update along with the MAC Mobility extended community where the sequence number is incremented. However, while that procedure could be used in the example of Figure 1, it would result in unnecessary flushing of unaffected ISIDs on the remote PEs, and subsequent flooding.

This document describes an extension of the [RFC7623] CMAC-Flush procedures, so that in the above failure example, PE3 can trigger a CMAC-Flush notification that makes PE1, PE2 and PE4 flush all the CMACs associated to PE3's BMAC and (only) ISID1. This new CMAC-Flush procedure explained in this document will be referred to as "PBB-EVPN ISID-based CMAC-Flush" and can be used in PBB-EVPN networks with null or non-null (virtual) Ethernet Segments.

2. Solution requirements

The following requirements must be met by the CMAC-Flush solution described in this document:

- a) The solution MUST solve black-hole scenarios in case of failures on null ES ACs (Attachment Circuits not associated to ES, that is, ESI=0) when the access device/network is responsible for the redundancy.
- b) This extension SHOULD work with Single-Active non-null ES and virtual ES, irrespective of the PE BMAC address assignment (dedicated per-ES BMAC or shared BMAC).
- c) In case of failure on the egress PE, the solution MUST provide a CMAC-Flush notification at BMAC AND ISID granularity level.
- d) The solution MUST provide a reliable CMAC-Flush notification in PBB-EVPN networks that use Route-Reflectors (RRs).
- e) The solution MUST coexist in [RFC7623]-compliant networks where there are systems not supporting this extension.
- f) The solution SHOULD be enabled/disabled by an administrative

option on a per-PE and per-ISID basis.

3. EVPN BGP Encoding for ISID-based CMAC-flush

The solution does not use any new BGP attributes but reuses the MAC Mobility extended community as an indication of CMAC-Flush (as in [RFC7623]) and encodes the ISID in the Ethernet Tag field of the MAC/IP route. As a reference, Figure 2 shows the MAC Mobility extended community and the MAC/IP route that are used in this document as a CMAC-Flush notification.

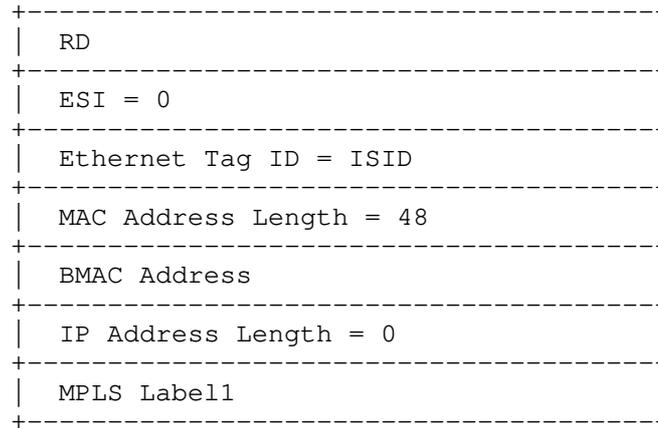
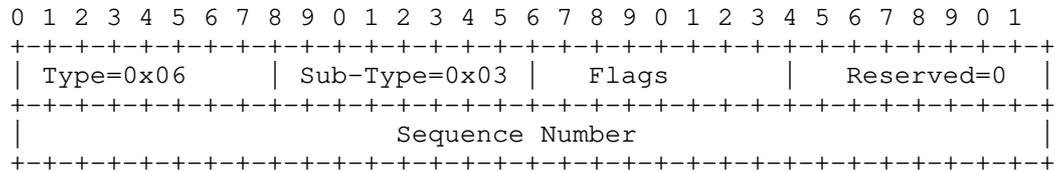


Figure 2 CMAC-Flush notification encoding: BMAC/ISID route

Where:

- o The route's RD and RT are the ones corresponding to its EVI. Alternatively to the EVI's RT, the route MAY be tagged with an RT auto-derived from the Ethernet Tag (ISID) instead. [RFC7623] describes how the RT can be derived from the ISID.
- o The Ethernet Tag encodes the ISID for which the PE that receives

the route must flush the CMACs upon reception of the route.

- o The MAC address field encodes the BMAC Address for which the PE that receives the route must flush the CMACs upon reception of the route.
- o The MAC Mobility extended community is used as in [RFC7623], where a delta in the sequence number between two updates for the same BMAC/ISID will be interpreted as a CMAC-flush notification for the corresponding BMAC and ISID.

All the other fields are set and used as defined in [RFC7623]. This document will refer to this route as the BMAC/ISID route, as opposed to the [RFC7623] BMAC/0 route (BMAC route sent with Ethernet Tag = 0).

Note that this BMAC/ISID route will be accepted and reflected by any RFC7432-compliant RR, since no new attributes or values are used. A PE receiving the route will process the received BMAC/ISID update only in case of supporting the procedures described in this document.

4. Solution description

Figure 1 will be used in the description of the solution. CE1, CE2 and CE3 are connected to ACs associated to ISID1, where no (Multi-Homed) Ethernet Segments have been enabled. All the ACs are operationally active and ready to forward frames.

Enabling or disabling ISID-based CMAC-Flush SHOULD be an administrative choice on the system that MAY be configured per ISID (I-Component). When enabled on a PE:

- a) The PE will be able to generate BMAC/ISID routes as CMAC-Flush notifications for the remote PEs.
- b) The PE will be able to process BMAC/ISID routes received from remote PEs.

When ISID-based CMAC-Flush is disabled, the PE will follow the [RFC7623] procedures for CMAC-flush.

These new CMAC-flush procedures are described in sections 4.1, 4.2 and 4.3 respectively:

- o ISID-based CMAC-flush activation
- o CMAC-flush notification generation upon AC failures

- o CMAC-flush process upon receiving a CMAC-Flush notification

4.1 ISID-based CMAC-Flush activation procedures

The following behavior MUST be followed by the PBB-EVPN PEs (see Figure 1):

- o As in [RFC7623], each PE has previously advertised a shared BMAC in a BMAC/0 route (BM1, BM2, BM3 and BM4 respectively). This is the BMAC that each PE will use as BMAC SA (Source Address) when encapsulating the frames received on any local single-homed AC. Each PE will import the received BMAC/0 routes from the remote PEs and will install the BMACs in its B-component MAC-VRF. For instance, PE1 will advertise BM1/0 and will install BM2, BM3 and BM4 in its MAC-VRF.
- o Assuming ISID-based CMAC-Flush is activated for ISID 1, the PEs will advertise the shared BMAC with ISID 1 encoded in the Ethernet Tag. That is, PE1 will advertise BM1/1 and will receive BM2/1, BM3/1 and BM4/1. The receiving PEs MUST use these BMAC/ISID routes only for CMAC-Flush procedures and they MUST NOT be used to add/withdraw any BMAC entry in the MAC-VRFs. As per [RFC7623], only BMAC/0 routes can be used to add/withdraw BMACs in the MAC-VRFs.
- o The above procedure MAY also be used for dedicated BMACs.

4.2 CMAC-Flush generation

If, for instance, there is a failure on PE1's AC, PE1 will generate an update including BM1/1 along with the MAC Mobility extended community where the Sequence Number has been incremented. The reception of the BM1/1 with a delta in the sequence number will trigger the CMAC-Flush procedures on the receiving PEs.

- o An AC going operationally down MUST generate a BMAC/ISID with a higher Sequence Number. If the AC going down makes the entire local ISID go operationally down, the PE will withdraw the BMAC/ISID route for the ISID.
- o An AC going operationally up SHOULD NOT generate any BMAC/ISID update, unless it activates its corresponding ISID, in which case the PE will advertise the BMAC/ISID route.
- o An AC receiving a CMAC-Flush notification from the access network, e.g. by G.8032, MAY propagate it to the remote PEs by generating a BMAC/ISID update with higher Sequence Number.

4.3 CMAC-Flush process upon receiving a CMAC-Flush notification

A PE receiving a CMAC-Flush notification will follow these procedures:

- o A received BMAC/ISID route (with non-zero ISID) MUST NOT add/remove any BMAC to/from the MAC-VRF.
- o An update of a previously received BMAC/ISID route with a delta Sequence Number, MUST flush all the CMACs associated to that ISID and BMAC. CMACs associated to the same ISID but different BMAC MUST NOT be flushed.
- o A received BMAC/ISID withdraw (with non-zero ISID) MUST flush all the CMACs associated to that BMAC and ISID.

Note that the CMAC-Flush procedures described in [RFC7623] for BMAC/0 routes are still valid and a PE receiving [RFC7623] CMAC-flush notification messages MUST observe the behavior specified in [RFC7623].

5. Conclusions

The ISID-based CMAC-Flush solution described in this document has the following benefits:

- a) The solution solves black-hole scenarios in case of failures on null ES ACs, since the CMAC-flush procedures are independent of the Ethernet Segment definition.
- b) This extension can also be used with Single-Active non-null ES and virtual ES, irrespective of the PE BMAC address assignment (dedicated per-ES BMAC or shared BMAC).
- c) It provides a CMAC-Flush notification at BMAC AND ISID granularity level, therefore flushing a minimum number of CMACs and reducing the amount of flooding in the network.
- d) It provides a reliable CMAC-Flush notification in PBB-EVPN networks that use RRs. RRs will propagate the CMAC-flush notifications for all the affected ISIDs and irrespective of the order in which the notifications make it to the RR.
- e) The solution can coexist in a network with systems supporting or not supporting the CMAC-flush extensions.

6. Conventions used in this document

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

7. Security Considerations

Security considerations described in [RFC7623] apply to this document. In addition, this document suggests additional procedures, that can be activated on a per ISID basis, and generate additional BGP EVPN MAC/IP advertisements in the network. The format of these additional MAC/IP routes is backwards compatible with [RFC7623] procedures and should not create any issues on receiving PEs not following this specification, however, the additional routes may consume extra memory resources on the receiving systems. Because of that, this feature should be activated only when necessary, and not by default in any PBB-EVPN PE.

8. IANA Considerations

9. References

9.1 Normative References

[RFC7623] Sajassi, A., Ed., Salam, S., Bitar, N., Isaac, A., and W. Henderickx, "Provider Backbone Bridging Combined with Ethernet VPN (PBB-EVPN)", RFC 7623, DOI 10.17487/RFC7623, September 2015, <<https://www.rfc-editor.org/info/rfc7623>>.

[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>>.

[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>>.

9.2 Informative References

[vES] Sajassi et al. "EVPN Virtual Ethernet Segment", draft-ietf-bess-evpn-virtual-eth-segment-04, work-in-progress, January, 2019.

10. Acknowledgments

The authors want to thank Vinod Prabhu, Sriram Venkateswaran, Laxmi Padakanti, Ranganathan Boovaraghavan for their review and contributions.

11. Contributors

17. Authors' Addresses

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

Senthil Sathappan
Nokia
701 E. Middlefield Road
Mountain View, CA 94043 USA
Email: senthil.sathappan@nokia.com

Kiran Nagaraj
Nokia
701 E. Middlefield Road
Mountain View, CA 94043 USA
Email: kiran.nagaraj@nokia.com

Masahiro Miyake
Softbank
Email: masahiro.miyake@g.softbank.co.jp

Taku Matsuda
Softbank
Email: taku.matsuda@g.softbank.co.jp

BESS
Internet-Draft
Updates: 6513, 6514 (if approved)
Intended status: Standards Track
Expires: January 9, 2017

Z. Zhang
R. Kebler
W. Lin
E. Rosen
Juniper Networks
July 8, 2016

MVPN/EVPN C-Multicast Routes Enhancements
draft-zzhang-bess-mvpn-evpn-cmcast-enhancements-00

Abstract

[RFC6513] and [RFC6514] specify procedures for originating, propagating, and processing "C-multicast routes". However, there are a number of MVPN use cases that are not properly or optimally handled by those procedures. This document describes those use cases, and specifies the additional procedures needed to handle them. Some of the additional procedures are also applicable to EVPN SMET routes [I-D.sajassi-bess-evpn-igmp-mld-proxy].

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on January 9, 2017.

Copyright Notice

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

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

Table of Contents

1.	Introduction	3
1.1.	Terminology	3
1.2.	MVPN C-Bidir Support with VPN Backbone being RPL	3
1.2.1.	C-multicast Routes for the MVPN-RPL Method of C-BIDIR support	4
1.2.2.	Optional use of MVPN-RPL RD with mLDP/PIM Provider Tunnels	5
1.2.3.	MVPN C-ASM Support without CE Routers	6
1.3.	Inter-AS Propagation of MVPN C-Multicast Routes	6
1.4.	EVPN Selective Multicast Ethernet Tag (SMET) Routes	8
1.5.	Provider Tunnel Segmentation with Explicit-Tracking C-Multicast Routes	9
1.5.1.	Conventional Tunnel Segmentation	9
1.5.2.	Selective Tunnel Segmentation with Untargeted Explicit-Tracking C-multicast Routes	9
2.	Specifications	10
2.1.	MVPN C-Bidir Support with VPN Backbone being RPL	10
2.1.1.	Constructing C-Multicast Share Tree Join route	10
2.1.2.	Setting Up the MVPN-RPL	12
2.2.	Inter-AS Propagation of MVPN C-Multicast Routes	12
2.2.1.	Procedures in Section 11.2 of [RFC6514]	12
2.2.2.	Ordinary BGP Propagation Procedures	13
2.3.	Provider Tunnel Segmentation with Explicit-Tracking C-Multicast Routes	13
2.3.1.	Egress PEs and RBRs	14
2.3.2.	Transit RBRs	15
2.3.3.	Ingress RBRs	15
2.3.4.	Setting Up Forwarding State on RBRs	16
2.3.5.	Other Types of Tunnels	16
3.	Security Considerations	16
4.	Acknowledgements	17

5. References	17
5.1. Normative References	17
5.2. Informative References	18
Authors' Addresses	18

1. Introduction

[RFC6513] and [RFC6514] specify procedures for originating, propagating, and processing "C-multicast routes". However, there are a number of MVPN use cases that are not properly or optimally handled by those procedures. This document describes those use cases, and specifies the additional procedures needed to handle them.

Some of the additional procedures are also applicable to EVPN SMET routes [I-D.sajassi-bess-evpn-igmp-mld-proxy]; this is discussed in Section 1.4.

1.1. Terminology

This document uses terminology from MVPN and EVPN. It is expected that the audience is familiar with the concepts and procedures defined in [RFC6513], [RFC6514], [RFC7524], [RFC7432], [I-D.zzhang-bess-evpn-bum-procedure-updates], and [I-D.sajassi-bess-evpn-igmp-mld-proxy]. Some terms are listed below for references.

- o PMSI: P-Multicast Service Interface - a conceptual interface for a PE to send customer multicast traffic to all or some PEs in the same VPN.
- o I-PMSI: Inclusive PMSI - to all PEs in the same VPN.
- o S-PMSI: Selective PMSI - to some of the PEs in the same VPN.
- o C-G-BIDIR: A bidirectional multicast group address (i.e., a group address whose IP multicast distribution tree is built by BIDIR-PIM) in customer address space.
- o RBR: Regional Border Router. A provider tunnel could be segmented, with one segment in each region. A region could be an AS, an IGP area, or even a subarea.

1.2. MVPN C-Bidir Support with VPN Backbone being RPL

In BIDIR-PIM [RFC5015], every group is associated with a "Rendezvous Point Link" (RPL). The RPL for a given group G is at the root of the BIDIR-PIM distribution tree. Links of the distribution tree that lead towards the RPL are considered to be "upstream" links, and links that lead away from the RPL are considered to be "downstream" links.

Every node on the distribution tree has one upstream link and zero or more downstream links.

Data addressed to a BIDIR-PIM group may enter the distribution tree at any node. The entry node sends the data on the upstream links and the downstream links. A node that receives the data from a downstream link sends it on its upstream link and on its other downstream links. A node that receives the data from its upstream link sends it on its downstream links. When a node that is attached to the RPL receives data from a downstream link, it forwards the data onto the RPL (as well as onto any other downstream links.) When node attached to the RPL receives data from the RPL, it forwards the data downstream.

The above is a simplified description, and ignores the fact that every link except the RPL has a Designated Forwarder (DF). Only the DF forwards traffic onto the link. However, the RPL has no DF; any node can forward traffic onto the RPL.

1.2.1. C-multicast Routes for the MVPN-RPL Method of C-BIDIR support

Section 11.1 of [RFC6513] describes a method of providing MVPN support for customers that use BIDIR-PIM. This is known as "MVPN C-BIDIR support". In this method of C-BIDIR support, the VPN backbone itself functions as the RPL. Thus this method is known as the "MVPN-RPL" method. The RPL is actually an I-PMSI or S-PMSI. The PE routers treat the I-PMSI or S-PMSI as their upstream link, and treat their VRF interfaces as downstream links.

If the MVPN-RPL method of C-BIDIR support is being used in a particular MVPN, all the PEs attached to that MVPN must be provisioned to use this method.

In the context of a given VPN, a PE with interest in receiving a particular C-BIDIR group (call it C-G-BIDIR) advertises this interest to the other PEs by originating a C-multicast Shared Tree Join route. When any PE receives traffic for the C-G-BIDIR on its PE-CE interface, it sends the data to the MVPN-RPL if and only if it has received corresponding (C-*,C-G-BIDIR) C-multicast Shared Tree Join route. Other PEs receive the traffic on the MVPN-RPL and forward to their downstream receivers. However, the procedure for constructing the C-multicast Shared Tree Join route in this case is not fully specified in [RFC6513] or [RFC6514]. The proper set of procedures are specified in Section 2.1.1 of this document.

Compared to other C-Multicast routes specified in [RFC6514], these are "untargeted" in that the RT allows all PEs in the same MVPN to

import them, while those other C-Multicast routes use a RT that identifies a VRF on a particular Upstream Multicast Hop (UMH) PE.

If a PE wants to use selective tunnel to send traffic to only a subset of the PEs on MVPN-RPL, i.e., those with downstream (C-*,C-G-BIDIR) state, per [RFC6513] [RFC6514] the PE needs to advertise a corresponding (C-*,C-G-BIDIR) S-PMSI A-D route, whose PTA specifies the tunnel to be used. In case of RSVP-TE P2MP, Ingress Replication (IR), or BIER tunnel, the Leaf Information Required (LIR) bit in the S-PMSI route's PTA is set to solicit corresponding Leaf A-D routes from those PEs with downstream (C-*,C-G-BIDIR) state. Every PE that wants to use selective tunnel for the (C-*,C-G-BIDIR) will advertise its own S-PMSI A-D route, each triggering a set of corresponding Leaf A-D routes.

Notice that the (C-*,C-G-BIDIR) C-Multicast routes from different PEs all have their own RDs so Route Reflectors (RRs) will reflect every one of them, and they already serve explicit tracking purpose (the BGP Next Hop identifies the originator of the route in non-segmentation case) - there is no need to use Leaf A-D routes triggered by the LIR bit in S-PMSI A-D routes. In case of RSVP-TE P2MP tunnel, the S-PMSI A-D routes are still needed to announce the tunnel but the LIR bit does not need to be set. In case of IR/BIER, there is no need for S-PMSI A-D routes at all.

1.2.2. Optional use of MVPN-RPL RD with mLDP/PIM Provider Tunnels

When mLDP/PIM tunnels are used, there is no need for explicit tracking as the leaves will simply send mLDP label Mapping or PIM Join messages. As a result, it's unnecessary for a PE to retain each C-Multicast route from each PE for the same C-G-BIDIR. If there is a Route Reflector (RR) in use, and it is known a priori that all the PEs/RRs/ASBRs involved in the propagation of the C-Multicast routes support BGP ADD-PATH [I-D.ietf-idr-add-paths], then the PEs could use a common RD to construct the C-Multicast routes. That way, the routes from different PEs for the same C-G-BIDIR will be considered paths for the same route and the RRs will reflect N paths to each PE. If N is significantly smaller than the number of PEs that advertises the routes, then the burden is significantly reduced for the PEs.

The reason for the need for ADD-PATH is shown with this example: both PE1 and PE2 advertise the same (C-*,C-G-BIDIR) C-Multicast route and the RR chooses the one from PE1 as the active path. Without ADD-PATH, the RR won't reflect any (C-*,C-G-BIDIR) path back to PE1, causing PE1 to think there is no other PE interested in receiving the C-G-BIDIR traffic. With ADD-PATH, it is guaranteed that even the originator of the active path will receive at least one other path. For this reason, ADD-PATH is needed and N=2 is well enough.

1.2.3. MVPN C-ASM Support without CE Routers

Current MVPN specifications is based on the fact that CEs are routers and in case of ASM one or more of the routers in customer address space, which could be a CE, a PE's VRF, or another non-PE/CE router, serves as RP. Traffic may be delivered on shared trees, switch to source specific trees, or switch back to shared trees depending the situation. There are two modes of MVPN to support ASM, all involving (C-S,C-G) MVPN Source Active (SA) A-D routes, individual (C-S,C-G) control/forwarding plane state and procedures that are not needed for a special scenario where CEs are not routers but just hosts.

From a logical point of view, this special scenario is when a VPN only involves customer networks directly connected to the PEs and no customer routers are used.. A practical example is EVPN inter-subnet multicast [I-D.lin-bess-evpn-irb-mcast], when EVPN is used to connect only servers and no customer routers are involved. In this case, it does not make sense to introduce the RP concept into the deployment and involve the MVPN SA procedures. Rather, this could be modeled as C-Bidir with MVPN-RPL and all the above discussed optimizations apply.

1.3. Inter-AS Propagation of MVPN C-Multicast Routes

Section 11.2 of [RFC6514] specifies the procedure used to propagate C-multicast routes from one AS to another. However, there are a number of problems with the procedures as specified in that RFC.

RFC6514 presumes that C-multicast routes are propagated through the ASBRs. This is analogous to RFC 4364's "Inter-AS option b". However, in some deployment scenarios, the C-multicast routes are propagated through Route Reflectors, in a manner analogous to RFC 4364's "Inter-AS option c". Strictly speaking, RFC 6514 does not allow this deployment scenario. This document updates RFC 6514 by allowing this deployment scenario to be used in place of the procedures of Section 11.2 of RFC 6514.

In some deployment scenarios, the propagation of C-multicast routes is controlled by the "Route Target Constraint" procedures of [RFC4684]. Strictly speaking, RFC 6514 does not allow this deployment scenario. This document updates RFC 6514 by allowing this deployment scenario to be used in place of the procedures of Section 11.2 of RFC 6514.

Per [RFC6514], an MVPN C-Multicast route is targeted at a particular PE, and its inter-as propagation towards the PE follows a series of ASBRs (in the reverse order) on the propagation path of one of the following:

- o The Intra-AS I-PMSI A-D route from the targeted PE, if the deployment is using non-segmented tunnels. In this scenario, the IP address of the targeted PE is encoded into the four-octet "Source AS" field (!) of the C-multicast route's NLRI.
- o The Inter-AS I-PMSI A-D route for the AS that the targeted PE is in, if the deployment is using segmented tunnel. In this scenario, the AS number of the source PE is encoded into the "Source AS" field of the C-multicast route's NLRI.

In both cases, the corresponding I-PMSI A-D route is found by looking for an I-PMSI A-D route whose NLRI consists of the C-multicast route's RD prepended to the contents of the C-multicast route's "Source AS" field. If neither Inter-AS nor Intra-AS I-PMSI A-D route is used, e.g. (C-*,C-*) S-PMSI A-D route is used, then the specified procedure will not work.

It must be noted that the RFC 6514 Section 11.2 propagation procedures cannot be applied to untargeted C-multicast routes, and cannot be applied even to targeted C-multicast routes if the infrastructure is based on IPv6 rather than IPv4.

This document updates RFC 6514 by declaring that the procedure of Section 11.2 of that document is only applicable in the case that (1) the C-multicast routes are being propagated through the ASBRs, AND (2) the propagation of those routes is not under the control of the Route Target Constraint procedures. It also updates the procedures of Section 11.2 of [RFC6514] to allow it to work without relying on I-PMSI A-D routes, whether IPv4 or IPv6 infrastructure is used.

This document also updates RFC 6514 by declaring that C-multicast routes MAY be propagated using ordinary BGP propagation procedures, which do not rely on the presence of I-PMSI A-D routes. For targeted C-multicast routes, this will result in a less optimal propagation path, but it does work in all cases. The Route Target Constraint procedures can always be used to obtain a more optimal path.

The selection of the propagation procedure for C-multicast routes is determined by provisioning.

In Section 1.2.1, the explicit tracking using C-multicast route relies on that the route's next hop is not changed so that the next hop can identify the originator. If the c-multicast routes are propagated through ASBRs, the next hop will be changed. With tunnel segmentation, this is not a problem (see Section 1.5) but if non-segmented tunnels are used, either the C-multicast route propagation must follow the Optoin C procedures and the next hop is not changed by the RRs, or the routes must carry an EC to identify the

originator. Or, the RD of a C-multicast route can be used to locate an I/S-PMSI route from the same PE, in which the Originator IP Address can be found.

1.4. EVPN Selective Multicast Ethernet Tag (SMET) Routes

[I-D.sajassi-bess-evpn-igmp-mld-proxy] defines a new EVPN route type known as an "SMET route".

The EVPN SMET routes are analogous to the MVPN C-multicast routes, in that both type of routes are used to disseminate the information that a particular egress PE has interest in a particular multicast C-flow or set of C-flows.

An EVPN SMET route contains, in its NLRI, the RD associated with the VRF from which the SMET route was originated. In addition, it is disseminated to all PEs of a given EVI. In this way, SMET routes are analogous to the MVPN C-multicast routes that are used for C-BIDIR support.

An EVPN SMET route contains, in its NLRI, the IP address of the originating PE. In this way, they are analogous to the MVPN Leaf A-D routes (They really combine the function of the MVPN C-multicast routes and the MVPN Leaf A-D routes). Similarly, they are also analogous to the C-multicast route for MVPN-RPL that carries an EC that identifies the originating PE.

In EVPN, as in MVPN, explicit tracking is required when selective tunnels are realized using IR, BIER, or RSVP-TE P2MP. The EVPN SMET routes provide this explicit tracking, so in these cases EVPN does not need explicit Leaf A-D routes. With IR/BIER, there is no need for S-PMSI route either. However, when SMET routes are used with segmented IR/BIER tunnels, more procedures are needed, just like the C-multicast route in MVPN-RPL case (Section 1.5). For that reason, given the similarity between SMET and C-Multicast routes, in this document we will use the same term C-Multicast route for EVPN SMET route as well. The two may be used interchangeably in case of EVPN.

If selective tunnels are set up using procedures that do not require explicit tracking, e.g. mLDP or PIM, the following optimization could be done, similar to MVPN-RPL with mLDP/PIM tunnels (Section 1.2.2):

- o When constructing an SMET route, put 0 as the Originator Router Address.
- o When constructing an SMET route in the context of a given EVI, have all PEs of that EVI set the RD field of the NLRI to the same

value (This is analogous to "MVPN-RPL RD" discussed in Section 1.2.2).

- o When a Route Reflector distributes the SMET routes, it uses BGP ADD-PATH to distribute at least two "paths" for a given NLRI.

1.5. Provider Tunnel Segmentation with Explicit-Tracking C-Multicast Routes

For the above MVPN-RPL and EVPN cases where C-multicast routes are used for explicit tracking without requiring corresponding S-PMSI A-D routes in case of IR/BIER selective tunnel, it works well when there is no tunnel segmentation. With tunnel segmentation [RFC6514] [RFC7524], [I-D.zzhang-bess-evpn-bum-procedure-updates] additional procedures are needed.

1.5.1. Conventional Tunnel Segmentation

Multicast forwarding needs to follow a rooted tree. With segmentation, the tree is divided into segments, with each segment rooted at either the ingress PE or a Regional Border Router (RBR). A segment is contained in a region, which could be an AS, an area, or a sub-area. The root of a segment only needs to track the leaves in its region, which are PEs or RBRs in that region. With the traditional PMSI/Leaf A-D procedures, an ingress PE/RBR sends out an I/S-PMSI route, propagated by RBRs (segmentation points), who change the tunnel identifier along the way to identify the tunnels for their segments. The Leaf A-D routes from PEs are not propagated by the RBRs. Rather, a RBR will proxy the Leaf AD routes it receives from its downstream towards its upstream RBR or PE, following the I/S-PMSI A-D routes received in the upstream region, as specified in [RFC6514] [RFC7524] [I-D.zzhang-bess-evpn-bum-procedure-updates].

1.5.2. Selective Tunnel Segmentation with Untargeted Explicit-Tracking C-multicast Routes

Without segmentation, the untargeted explicit-tracking C-Multicast routes are sent to every PE, and each PE adds the originator of the routes as leaves of the tunnel rooted at the PE.

With segmentation, untargeted explicit-tracking C-Multicast routes are propagated through segmentation points towards all ingress PEs or ASes and are merged along the way. This is like the traditional PMSI/Leaf A-D procedures but with one difference.

With the traditional PMSI/Leaf A-D procedures, the propagation is towards the originator of the PMSI A-D route and a single tree is formed. With untargeted C-Multicast routes, multiple trees are

formed, each being rooted at the ingress PE (if per-region aggregation [I-D.zhang-bess-evpn-bum-procedure-updates] is not used) or ingress RBR (if per-region aggregation is used). The roots of those trees are either the ingress PEs or the ingress RBRs, identified by all the per-PE or per-region I-PMSI A-D routes.

To form those multiple trees without requiring S-PMSI A-D routes from the ingress PEs/RBRs, this document proposes that the RBRs convert a C-multicast route originated in its own region to Leaf A-D routes, as if corresponding S-PMSI A-D routes had been received from ingress PEs/RBRs. The details are provided in Section 2.2.

2. Specifications

This section provides detailed specifications for the optional enhancements introduced above.

2.1. MVPN C-Bidir Support with VPN Backbone being RPL

2.1.1. Constructing C-Multicast Share Tree Join route

In the context of a particular VRF, a PE with downstream state for the group C-G-BIDIR originates a C-multicast Shared Tree Join route, referred to as "MVPN-RPL C-multicast Join", when the MVPN-RPL method of C-BIDIR support is being used.

The fields of the route are set as follows:

- o RD: See Section 2.1.1.2.
- o Source AS: set to zero.
- o Multicast Source Length: 4 or 16.
- o Multicast Source: set to RPA.
- o Multicast Group Length: 4 or 16.
- o Multicast Group: BIDIR-PIM group address.

Note that the RD field, and the Route Targets that are attached to the C-multicast route are different than what is specified in [RFC6514]. See following two sections.

2.1.1.1. Setting the Route Targets

Per [RFC6514], when a PE originates a C-multicast route, it "targets" the route to a specific one of the other PEs attached to the same VPN. The IP address of the targeted PE is encoded into a Route Target and attached to the C-multicast route. This ensures that the C-multicast route is processed only by the PE to which it is targeted.

However, C-multicast routes used by the MVPN-RPL method are not targeted. Rather, they must be processed by all the other PEs attached to the same MVPN. Thus we refer to these routes as "untargeted". The Route Targets attached to these routes must be such as to cause the routes to be propagated to all the other PEs of the given MVPN. By default, these will be the same Route Targets that are attached to the I-PMSI A-D routes of the MVPN.

2.1.1.2. Setting the Route Distinguisher

Per [RFC6514], the RD in a C-multicast Join Route is the RD of a VRF on the PE to which the route is targeted. However, in an MVPN-RPL C-multicast Join, the RD is set differently.

If PIM/mLDP provider tunnels are used, and it is known that all the PEs/RRs/ASBRs involved in the propagation of C-multicast routes support BGP ADD-PATH, the RD MAY be set to a value that is specially configured to be used as the RD for MVPN-RPL in a given VPN. Call this the "MVPN-RPL" RD for that VPN. In that case, all the C-multicast Joins that are providing C-BIDIR support (for a given VPN) using the MVPN-RPL method will have the same RD. This MVPN-RPL RD of a given VPN MUST NOT be used for any other purpose, or by any other VPN. See Section 1.2.2 for a discussion of when it may be advantageous to use an MVPN-RPL RD.

For other provider tunnel types, or if the above mentioned MVPN-RPL RD in case of PIM/mLDP tunnel is not feasible (e.g. BGP ADD-PATH is not supported), the RD in the C-multicast route is that of the VRF from which the route is originated.

For Global Table Multicast (GTM) using MVPN procedures [RFC7116], RFC 7116 specifies that MVPN routes use a special 0:0 RD. This document specifies that GTM use non-0:0 RDs for C-Multicast routes for C-Bidir, when the backbone is used as RPL and provider tunnels are not set up by PIM/mLDP.

2.1.2. Setting Up the MVPN-RPL

By default, the I-PMSI or (C-*,C-BIDIR) S-PMSI plays the role of MVPN-RPL. When (C-*,C-G-BIDIR) S-PMSI is used for a particular C-G-BIDIR, the following procedures are followed, depending on the type of provider tunnel used.

2.1.2.1. Ingress Replication or BIER

If Ingress Replication or BIER is used, there is no need for the ingress PE to advertise (C-*,C-G-BIDIR) S-PMSI A-D route. The ingress PE identifies the tunnel leaves to send traffic to by the C-multicast routes it receives, because each such route has a different RD and serves explicit tracking purpose. In case of IR, the label in the Intra-AS I-PMSI A-D route or (C-*,C-*) S-PMSI A-D route from a leaf is used to send traffic to the leaf. In case of BIER, the label in the same route from the ingress PE is used to send traffic.

2.1.2.2. RSVP-TE P2MP

With RSVP-TE P2MP tunnel, the ingress PE advertises (C-*,C-G-BIDIR) S-PMSI A-D route without setting the LIR bit in the route's PTA. It identifies the tunnel leaves from the C-multicast routes it receives.

2.1.2.3. PIM/mLDP

With PIM or mLDP P2MP provider tunnel, procedures in [RFC6514] are followed.

2.2. Inter-AS Propagation of MVPN C-Multicast Routes

This specification allows two methods of Inter-AS propagation for MVPN C-multicast routes. The choice of which method is used is by provisioning.

2.2.1. Procedures in Section 11.2 of [RFC6514]

The procedures in Section 11.2 of [RFC6514] are extended with the following.

The Source AS field in the NLRI of C-multicast route is set to the AS number of the UMH PE if and only if segmented inter-AS tunnels and per-AS aggregation (via Inter-AS I-PMSI A-D routes) are used. The existing procedures are used as is in this case.

Otherwise, when an egress PE constructs a C-Multicast route and the upstream PE is in a different AS from the local PE, it finds in its

VRF an Intra-AS I-PMSI A-D route or any S-PMSI A-D route from the upstream PE (the Originating Router's IP Address field of that route has the same value as the one carried in the VRF Route Import of the (unicast) route to the address carried in the Multicast Source field). The RD of the found I/S-PMSI A-D route is used as the RD of the advertised C-multicast route. The Source AS field in the C-multicast route is set to 0. If the Next Hop of the found I/S-PMSI A-D route is an EBGP neighbor of the local PE, then the PE advertises the C-multicast route to that neighbor. Otherwise the PE advertises the C-multicast route into IBGP.

When an ASBR receives a C-multicast route with the Source AS field set to 0, it uses the RD of the C-multicast route to locate an Intra-AS I-PMSI A-D route or any S-PMSI A-D route, and propagate the C-multicast route to the bgp neighbor from which the found I/S-PMSI A-D route is learned.

2.2.2. Ordinary BGP Propagation Procedures

This document specifies that C-multicast routes MAY be propagated using ordinary BGP propagation procedures, which do not rely on the presence of any I/S-PMSI A-D routes. With this method, the Source AS field in the C-Multicast route SHOULD be set to 0. For targeted C-multicast routes, this will result in a less optimal propagation path, but it does work in all cases. The Route Target Constraint procedures can always be used to obtain a more optimal path.

2.3. Provider Tunnel Segmentation with Explicit-Tracking C-Multicast Routes

This section applies when IR/BIER are used for MVPN/EVPN selective tunnels.

If per-region aggregation [I-D.zhang-bess-evpn-bum-procedure-updates] is used, this document specifies that the per-region I-PMSI A-D route MUST carry a VRF Route Import EC to identify the originator of the per-region I-PMSI A-D route. Note that, while it borrows "VRF Route Import EC" from the UMH routes, it is only used to identify the originator.

If per-region aggregation is not used, this document specifies that either per-PE I-PMSI or (C-*,C-*) S-PMSI A-D routes MUST be originated by every PE.

2.3.1. Egress PEs and RBRs

An egress PE originates MVPN C-multicast routes for MVPN-RPL as specified in previous sections of this document, or EVPN SMET routes as specified in [I-D.sajassi-bess-evpn-igmp-mld-proxy]. Recall that EVPN SMET routes may also be referred to C-Multicast routes in this document.

Explicit-tracking C-multicast routes must be processed by segmentation points, which are referred to as RBRs. When a RBR receives a C-multicast route from within its own region, and the route does not carry a flag bit that indicates the route is converted from a downstream Leaf A-D route (see descriptions below), it converts the C-multicast route into one or more Leaf A-D routes, as if it had received corresponding S-PMSI A-D routes. When a converted Leaf A-D routes reaches the ingress region, the RBR converts it back to C-multicast routes.

With per-region aggregation, the RBR in an egress region finds all active per-region I-PMSI A-D route that the RBR has in the corresponding VRF. For each of them, it makes up a (C-S,C-G) or (C-*,C-G) S-PMSI A-D route as following.

- o RD: set to the RD from the per-region I-PMSI A-D route.
- o Source/Group length/address fields: set according to the received C-multicast route.
- o Originator's IP Address: set according to the VRF Route Import EC in the per-region I-PMSI A-D route
- o Ethernet Tag ID in case of EVPN: set according to the received SMET route (which is also referred to as C-multicast route).
- o Next Hop: set according to the per-region I-PMSI A-D route.

Without per-region aggregation, a RBR finds all active per-PE I-PMSI or (C-*,C-*) S-PMSI A-D route in the VRF. For each of them it makes up a (C-S,C-G) or (C-*,C-G) S-PMSI A-D route similar to the per-region aggregation case. The only difference is that the Originator's IP Address field is set to the same as in the per-PE I-PMSI or (C-*,C-*) S-PMSI A-D route.

The made up S-PMSI A-D route is for local use only, and not propagated anywhere. A corresponding Leaf A-D route is then generated and propagated to the upstream identified by the BGP next hop in the made up S-PMSI A-D route, following existing PMSI/Leaf A-D route procedures.

2.3.2. Transit RBRs

When an upstream RBR receives a (C-S,C-G) or (C-*,C-G) Leaf A-D route, It locates the active per-PE/region I-PMSI or (C-*,C-*) S-PMSI A-D route whose RD matches the received Leaf A-D route. If no such route exists, the received Leaf A-D route is ignored until such a route appears later. It also tries to locate a corresponding active (C-S,C-G) or (C-*,C-G) S-PMSI A-D route, which could be a real one received from an upstream PE/RBR, or could be a made up one triggered by a Leaf A-D route from a different downstream. If such route exists, existing PMSI/Leaf A-D route procedures are followed.

If no such corresponding active (C-S,C-G) or (C-*,C-G) S-PMSI A-D route exists, and the located active I-PMSI or (C-*,C-*) S-PMSI A-D route has a next hop different from the Originator IP Address in the per-PE I-PMSI A-D route or (C-*,C-*) I-PMSI A-D route, or different from the address in the VRF Route Import EC in the per-region I-PMSI A-D route, the ingress region corresponding to the I-PMSI or (C-*,C-*) S-PMSI A-D route has not been reached. The RBR then makes up a (C-S,C-G) or (C-*,C-G) S-PMSI A-D route. as specified earlier, and proxies Leaf A-D routes further up.

2.3.3. Ingress RBRs

If the BGP next hop in the located active I-PMSI or (C-*,C-*) S-PMSI A-D route matches the Originator IP Address in the per-PE I/S-PMSI A-D route or the IP address in the per-region I-PMSI A-D route's VRF Route Import EC, it means the ingress region has been reached. If the corresponding (C-S,C-G) or (C-*,C-G) S-PMSI A-D route is a made up one and not actually advertised by an ingress PE/RBR, the RBR reconverts the Leaf A-D route back to C-multicast route, with a CV ("Converted") flag bit indicating that the route is not from local state learned on PE-CE interface but from state learned further downstream. The flag bit prevents other RBRs in this region to trigger Leaf A-D routes from this converted C-multicast route.

The converted C-multicast route is constructed as following:

- o RD: set to the RD of the RBR for the related IP/MAC VRF.
- o Source/Group length/address fields: set according to the received Leaf A-D route.
- o Ethernet Tag ID in case of EVPN: set according to the received Leaf A-D route.
- o Next Hop: set to the RBR's local IP Address.

The RT of the converted C-multicast route is set to the RT used for VRF but the route is only propagated to PEs/RBRs in the local region.

For EVPN SMET routes, the flag bit is part of the existing Flags field in the NLRI:

```

      0  1  2  3  4  5  6  7
+-----+-----+-----+-----+
|reserved|CV|IE|v3|v2|v1|
+-----+-----+-----+-----+

```

The IE/v3/v2/v1 are existing bits and the CV bit is the new bit to indicate that this is converted from state learned from downstream.

For MVPN C-Multicast route, the CV bit is part of a new MVPN Flag EC, to be specified in a future revision.

2.3.4. Setting Up Forwarding State on RBRs

As a RBR follows the PMSI/Leaf A-D route procedures (even though the S-PMSI A-D route may be made up and not real), it sets up forwarding state accordingly [I-D.ietf-bess-ir] [I-D.ietf-bier-mvpn]. If IR is used in the upstream region, a downstream allocated label is advertised in the PTA of the Leaf A-D route sent upstream. If BIER is used in a region, the root RBR for the segment in that region MUST advertise an S-PMSI A-D route, whether the route is actually received from upstream or made up based on received C-multicast route or Leaf A-D route, with the PTA's label field set to a label upstream-allocated by the root RBR of the segment. This allows label switching by the RBR instead of relying on (C-S,C-G) lookup based forwarding in the VRF.

2.3.5. Other Types of Tunnels

The inter-region segmented tunnel can consists of different types of tunnels, like PIM/mLDP/RSVP-TE P2MP tunnels that require advertised S-PMSI A-D routes. This is just like BIER case mentioned in the above section. The only difference is that in BIER case it is the upstream allocated label that needs to be advertised by the S-PMSI A-D routes and in PIM/mLDP/RSVP-TE P2MP case it is the tunnel identity and optionally the upstream allocated label that need to be advertised by the S-PMSI A-D routes.

3. Security Considerations

This document does not seem to introduce new security risks, though this may be revised after further review and scrutiny.

4. Acknowledgements

The authors thank Vinay Nallamothe and Kevin Wang for their comments and suggestions.

5. References

5.1. Normative References

[I-D.ietf-bess-ir]

Rosen, E., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", draft-ietf-bess-ir-03 (work in progress), April 2016.

[I-D.ietf-bier-mvpn]

Rosen, E., Sivakumar, M., Aldrin, S., Dolganow, A., and T. Przygienda, "Multicast VPN Using BIER", draft-ietf-bier-mvpn-03 (work in progress), June 2016.

[I-D.ietf-idr-add-paths]

Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", draft-ietf-idr-add-paths-15 (work in progress), May 2016.

[I-D.sajassi-bess-evpn-igmp-ml-d-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-ml-d-proxy-00 (work in progress), October 2015.

[I-D.zzhang-bess-evpn-bum-procedure-updates]

Zhang, Z., Lin, W., Rabadan, J., and K. Patel, "Updates on EVPN BUM Procedures", draft-zzhang-bess-evpn-bum-procedure-updates-03 (work in progress), April 2016.

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

[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>>.

- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <<http://www.rfc-editor.org/info/rfc5015>>.
- [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>>.
- [RFC7116] Scott, K. and M. Blanchet, "Licklider Transmission Protocol (LTP), Compressed Bundle Header Encoding (CBHE), and Bundle Protocol IANA Registries", RFC 7116, DOI 10.17487/RFC7116, February 2014, <<http://www.rfc-editor.org/info/rfc7116>>.
- [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>>.

5.2. Informative References

- [I-D.lin-bess-evpn-irb-mcast]
Lin, W., Zhang, Z., Drake, J., and J. Rabadan, "EVPN Inter-subnet Multicast Forwarding", draft-lin-bess-evpn-irb-mcast-02 (work in progress), March 2016.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

E-Mail: zzhang@juniper.net

Robert Kebler
Juniper Networks

E-Mail: rkebler@juniper.net

Wen Lin
Juniper Networks

EMail: wlin@juniper.net

Eric Rosen
Juniper Networks

EMail: erosen@juniper.net

BESS
Internet-Draft
Updates: 6513, 6514 (if approved)
Intended status: Standards Track
Expires: 18 September 2024

Z. Zhang
R. Kebler
W. Lin
Juniper Networks
E. Rosen
17 March 2024

MVPN/EVPN C-Multicast Routes Enhancements
draft-zzhang-bess-mvpn-evpn-cmcast-enhancements-04

Abstract

[RFC6513] and [RFC6514] specify procedures for originating, propagating, and processing "C-multicast routes". However, there are a number of MVPN use cases that are not properly or optimally handled by those procedures. This document describes those use cases, and specifies the additional procedures needed to handle them. Some of the additional procedures are also applicable to EVPN SMET routes [RFC9251].

Requirements Language

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

Status of This Memo

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

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

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

This Internet-Draft will expire on 18 September 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

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

Table of Contents

1.	Introduction	3
1.1.	Terminology	3
1.2.	MVPN C-Bidir Support with VPN Backbone being RPL	4
1.2.1.	C-multicast Routes for the MVPN-RPL Method of C-BIDIR support	4
1.2.2.	Optional use of MVPN-RPL RD with mLDP/PIM Provider Tunnels	5
1.2.3.	MVPN C-ASM Support without CE Routers	6
1.3.	Inter-AS Propagation of MVPN C-Multicast Routes	6
1.4.	MVPN Inter-AS Upstream PE Selection	8
1.5.	EVPN Selective Multicast Ethernet Tag (SMET) Routes	10
1.6.	Provider Tunnel Segmentation with Explicit-Tracking C-Multicast Routes	11
1.6.1.	Conventional Tunnel Segmentation	11
1.6.2.	Selective Tunnel Segmentation with Untargeted Explicit-Tracking C-multicast Routes	11
2.	Specifications	12
2.1.	MVPN C-Bidir Support with VPN Backbone being RPL	12
2.1.1.	Constructing C-Multicast Share Tree Join route	12
2.1.2.	Setting Up the MVPN-RPL	13
2.2.	Inter-AS Propagation of MVPN C-Multicast Routes	14
2.2.1.	Procedures in Section 11.2 of [RFC6514]	14
2.2.2.	Ordinary BGP Propagation Procedures	15
2.3.	Inter-AS Upstream PE Selection	16
2.4.	Duplication Prevention on the Same Inclusive Inter-AS Tunnel	16
2.4.1.	Using PE Distinguisher Labels	16
2.4.2.	Ingress ASBR Filtering Out Duplications	17
2.5.	Provider Tunnel Segmentation with Explicit-Tracking C-Multicast Routes	17
2.5.1.	Egress PEs and RBRs	18
2.5.2.	Transit RBRs	19
2.5.3.	Ingress RBRs	19
2.5.4.	Setting Up Forwarding State on RBRs	20
2.5.5.	Other Types of Tunnels	21
3.	Security Considerations	21
4.	Contributors	21

5. Acknowledgements	21
6. References	21
6.1. Normative References	21
6.2. Informative References	23
Authors' Addresses	23

1. Introduction

[RFC6513] and [RFC6514] specify procedures for originating, propagating, and processing "C-multicast routes". However, there are a number of MVPN use cases that are not properly or optimally handled by those procedures. This document describes those use cases, and specifies the additional procedures needed to handle them.

Some of the additional procedures are also applicable to EVPN SMET routes [RFC9251]; this is discussed in Section 1.5.

1.1. Terminology

This document uses terminology from MVPN and EVPN. It is expected that the audience is familiar with the concepts and procedures defined in [RFC6513], [RFC6514], [RFC7524], [RFC7432], [I-D.ietf-bess-evpn-bum-procedure-updates], and [RFC9251]. Some terms are listed below for references.

- * PMSI: P-Multicast Service Interface - a conceptual interface for a PE to send customer multicast traffic to all or some PEs in the same VPN.
- * I-PMSI: Inclusive PMSI - to all PEs in the same VPN.
- * S-PMSI: Selective PMSI - to some of the PEs in the same VPN.
- * C-G-BIDIR: A bidirectional multicast group address (i.e., a group address whose IP multicast distribution tree is built by BIDIR-PIM) in customer address space.
- * RBR: Regional Border Router. A provider tunnel could be segmented, with one segment in each region. A region could be an AS, an IGP area, or even a subarea.

1.2. MVPN C-Bidir Support with VPN Backbone being RPL

In BIDIR-PIM [RFC5015], every group is associated with a "Rendezvous Point Link" (RPL). The RPL for a given group G is at the root of the BIDIR-PIM distribution tree. Links of the distribution tree that lead towards the RPL are considered to be "upstream" links, and links that lead away from the RPL are considered to be "downstream" links. Every node on the distribution tree has one upstream link and zero or more downstream links.

Data addressed to a BIDIR-PIM group may enter the distribution tree at any node. The entry node sends the data on the upstream links and the downstream links. A node that receives the data from a downstream link sends it on its upstream link and on its other downstream links. A node that receives the data from its upstream link sends it on its downstream links. When a node that is attached to the RPL receives data from a downstream link, it forwards the data onto the RPL (as well as onto any other downstream links.) When node attached to the RPL receives data from the RPL, it forwards the data downstream.

The above is a simplified description, and ignores the fact that every link except the RPL has a Designated Forwarder (DF). Only the DF forwards traffic onto the link. However, the RPL has no DF; any node can forward traffic onto the RPL.

1.2.1. C-multicast Routes for the MVPN-RPL Method of C-BIDIR support

Section 11.1 of [RFC6513] describes a method of providing MVPN support for customers that use BIDIR-PIM. This is known as "MVPN C-BIDIR support". In this method of C-BIDIR support, the VPN backbone itself functions as the RPL. Thus this method is known as the "MVPN-RPL" method. The RPL is actually an I-PMSI or S-PMSI. The PE routers treat the I-PMSI or S-PMSI as their upstream link, and treat their VRF interfaces as downstream links.

If the MVPN-RPL method of C-BIDIR support is being used in a particular MVPN, all the PEs attached to that MVPN must be provisioned to use this method.

In the context of a given VPN, a PE with interest in receiving a particular C-BIDIR group (call it C-G-BIDIR) advertises this interest to the other PEs by originating a C-multicast Shared Tree Join route. When any PE receives traffic for the C-G-BIDIR on its PE-CE interface, it sends the data to the MVPN-RPL if and only if it has received corresponding (C-*,C-G-BIDIR) C-multicast Shared Tree Join route. Other PEs receive the traffic on the MVPN-RPL and forward to their downstream receivers. However, the procedure for constructing

the C-multicast Shared Tree Join route in this case is not fully specified in [RFC6513] or [RFC6514]. The proper set of procedures are specified in Section 2.1.1 of this document.

Compared to other C-Multicast routes specified in [RFC6514], these are "untargeted" in that the RT allows all PEs in the same MVPN to import them, while those other C-Multicast routes use a RT that identifies a VRF on a particular Upstream Multicast Hop (UMH) PE.

If a PE wants to use selective tunnel to send traffic to only a subset of the PEs on MVPN-RPL, i.e., those with downstream (C-*,C-G-BIDIR) state, per [RFC6513] [RFC6514] the PE needs to advertise a corresponding (C-*,C-G-BIDIR) S-PMSI A-D route, whose PTA specifies the tunnel to be used. In case of RSVP-TE P2MP, Ingress Replication (IR), or BIER tunnel, the Leaf Information Required (LIR) bit in the S-PMSI route's PTA is set to solicit corresponding Leaf A-D routes from those PEs with downstream (C-*,C-G-BIDIR) state. Every PE that wants to use selective tunnel for the (C-*,C-G-BIDIR) will advertise its own S-PMSI A-D route, each triggering a set of corresponding Leaf A-D routes.

Notice that the (C-*,C-G-BIDIR) C-Multicast routes from different PEs all have their own RDs so Route Reflectors (RRs) will reflect every one of them, and they already serve explicit tracking purpose (the BGP Next Hop identifies the originator of the route in non-segmentation case) - there is no need to use Leaf A-D routes triggered by the LIR bit in S-PMSI A-D routes. In case of RSVP-TE P2MP tunnel, the S-PMSI A-D routes are still needed to announce the tunnel but the LIR bit does not need to be set. In case of IR/BIER, there is no need for S-PMSI A-D routes at all.

1.2.2. Optional use of MVPN-RPL RD with mLDP/PIM Provider Tunnels

When mLDP/PIM tunnels are used, there is no need for explicit tracking as the leaves will simply send mLDP label Mapping or PIM Join messages. As a result, it's unnecessary for a PE to retain each C-Multicast route from each PE for the same C-G-BIDIR. If there is a Route Reflector (RR) in use, and it is known apriori that all the PEs/RRs/ASBRs involved in the propagation of the C-Multicast routes support BGP ADD-PATH [RFC7911], then the PEs could use a common RD to construct the C-Multicast routes. That way, the routes from different PEs for the same C-G-BIDIR will be considered paths for the same route and the RRs will reflect N paths to each PE. If N is significantly smaller than the number of PEs that advertises the routes, then the burden is significantly reduced for the PEs.

The reason for the need for ADD-PATH is shown with this example: both PE1 and PE2 advertise the same (C-*,C-G-BIDIR) C-Multicast route and the RR chooses the one from PE1 as the active path. Without ADD-PATH, the RR won't reflect any (C-*,C-G-BIDIR) path back to PE1, causing PE1 to think there is no other PE interested in receiving the C-G-BIDIR traffic. With ADD-PATH, it is guaranteed that even the originator of the active path will receive at least one other path. For this reason, ADD-PATH is needed and N=2 is well enough.

1.2.3. MVPN C-ASM Support without CE Routers

Current MVPN specifications is based on the fact that CEs are routers and in case of ASM one or more of the routers in customer address space, which could be a CE, a PE's VRF, or another non-PE/CE router, serves as RP. Traffic may be delivered on shared trees, switch to source specific trees, or switch back to shared trees depending the situation. There are two modes of MVPN to support ASM, all involving (C-S,C-G) MVPN Source Active (SA) A-D routes, individual (C-S,C-G) control/forwarding plane state and procedures that are not needed for a special scenario where CEs are not routers but just hosts.

From a logical point of view, this special scenario is when a VPN only involves customer networks directly connected to the PEs and no customer routers are used. A practical example is EVPN inter-subnet multicast [I-D.ietf-bess-evpn-irb-mcast], when EVPN is used to connect only servers and no customer routers are involved. In this case, it does not make sense to introduce the RP concept into the deployment and involve the MVPN SA procedures. Rather, this could be modeled as C-Bidir with MVPN-RPL and all the above discussed optimizations apply.

1.3. Inter-AS Propagation of MVPN C-Multicast Routes

Section 11.2 of [RFC6514] specifies the procedure used to propagate C-multicast routes from one AS to another. However, there are a number of problems with the procedures as specified in that RFC.

RFC6514 presumes that C-multicast routes are propagated through the ASBRs. This is analogous to RFC 4364's "Inter-AS option b". However, in some deployment scenarios, the C-multicast routes are propagated through Route Reflectors, in a manner analogous to RFC 4364's "Inter-AS option c". Strictly speaking, RFC 6514 does not allow this deployment scenario. This document updates RFC 6514 by allowing this deployment scenario to be used in place of the procedures of Section 11.2 of RFC 6514.

In some deployment scenarios, the propagation of C-multicast routes is controlled by the "Route Target Constraint" procedures of [RFC4684]. Strictly speaking, RFC 6514 does not allow this deployment scenario. This document updates RFC 6514 by allowing this deployment scenario to be used in place of the procedures of Section 11.2 of RFC 6514.

Per [RFC6514], an MVPN C-Multicast route is targeted at a particular PE, and its inter-as propagation towards the PE follows a series of ASBRs (in the reverse order) on the propagation path of one of the following:

- * The Intra-AS I-PMSI A-D route from the targeted PE, if the deployment is using non-segmented tunnels. In this scenario, the IP address of the targeted PE is encoded into the four-octet "Source AS" field (!) of the C-multicast route's NLRI.
- * The Inter-AS I-PMSI A-D route for the AS that the targeted PE is in, if the deployment is using segmented tunnel. In this scenario, the AS number of the source PE is encoded into the "Source AS" field of the C-multicast route's NLRI.

In both cases, the corresponding I-PMSI A-D route is found by looking for an I-PMSI A-D route whose NLRI consists of the C-multicast route's RD prepended to the contents of the C-multicast route's "Source AS" field. If neither Inter-AS nor Intra-AS I-PMSI A-D route is used, e.g. (C-*,C-*) S-PMSI A-D route is used, then the specified procedure will not work.

It must be noted that the RFC 6514 Section 11.2 propagation procedures cannot be applied to untargeted C-multicast routes, and cannot be applied even to targeted C-multicast routes if the infrastructure is based on IPv6 rather than IPv4.

This document updates RFC 6514 by declaring that the procedure of Section 11.2 of that document is only applicable in the case that (1) the C-multicast routes are being propagated through the ASBRs, AND (2) the propagation of those routes is not under the control of the Route Target Constraint procedures. It also updates the procedures of Section 11.2 of [RFC6514] to allow it to work without relying on I-PMSI A-D routes, whether IPv4 or IPv6 infrastructure is used. Additional enhancement is also specified in Section 2.2.1 to allow it to work with Global Table Multicast (GTM) using MVPN procedures [RFC7716] as well.

This document also updates RFC 6514 by declaring that C-multicast routes MAY be propagated using ordinary BGP propagation procedures, which do not rely on the presence of I-PMSI A-D routes. For targeted

C-multicast routes, this will result in a less optimal propagation path, but it does work in all cases. The Route Target Constraint procedures can always be used to obtain a more optimal path.

The selection of the propagation procedure for C-multicast routes is determined by provisioning.

In Section 1.2.1, the explicit tracking using C-multicast route relies on that the route's next hop is not changed so that the next hop can identify the originator. If the c-multicast routes are propagated through ASBRs, the next hop will be changed. With tunnel segmentation, this is not a problem (see Section 1.6) but if non-segmented tunnels are used, either the C-multicast route propagation must follow the Option C procedures and the next hop is not changed by the RRs, or the routes must carry an EC to identify the originator. Or, the RD of a C-multicast route can be used to locate an I/S-PMSI route from the same PE, in which the Originator IP Address can be found.

1.4. MVPN Inter-AS Upstream PE Selection

Consider the following scenario:

A multicast source is multi-homed to PE1 and PE2 in the same source AS1. ASBR1 in AS1 connects to ASBR2 in another AS2. In AS2, egress PE3 selects PE1 while egress PE4 selects PE2 as their upstream PE respectively, because they use the "Installed UMH Route" as the "Selected UMH Route" (as defined in Section 5.1.3 of [RFC6513]).

Suppose inter-as tunnel segmentation is used. Following Section 11.1.3 of [RFC6514], PE3 and PE4 will construct their C-multicast routes with the same NLRI key (in particular with the same RD from the Inter-AS I-PMSI A-D route originated by ASBR1) but with one different Route Target - PE3's C-multicast route carries the RT corresponding to PE1's VRF while PE4's C-multicast route carries the RT corresponding to PE2's VRF. ASBR2 will re-advertise only one of the two C-multicast routes to ASBR1. Assuming it is the one with a RT corresponding to PE1, then only PE1 will transmit corresponding traffic.

If selective tunnels are used, PE4 that chooses PE2 as the upstream PE will not join the selective tunnel advertised by PE1 so it will not receive traffic.

With the new method for inter-as propagation of C-multicast routes described in the previous section, this traffic blackholing problem can be resolved if PE3 and PE4 construct their C-multicast routes with different RDs, e.g. with the RD from the chosen UMH route

instead of the RD from the Inter-AS I-PMSI A-D route. That way, PE1 will receive the C-multicast route from PE3 and PE2 will receive the C-multicast route from PE4. Both will transmit traffic but PE3 and PE4 will only receive the traffic via the selective tunnel that they join hence no duplication or blackholing.

Notice that this also removes the pre-requisite in Section 4.4 of [RFC9026].

However, there are still two problems. First, while there is no duplication or blackholing issue when selective tunnels are used, two copies of traffic are sent inter-AS, possibly through many common paths before reaching the egress PEs (imagine that there are a string of ASes between AS1 and AS2). This is not an efficient use of inter-AS resources.

Choosing upstream PE based on installed UMH route allows different egress PEs to choose different upstream PEs (typically the closest upstream PE), so it is desired for certain intra-as deployment scenarios but apparently it is not desired for PEs in other ASes to choose different upstream PEs. This problem can actually be solved if PEs always do "Single Forwarder Selection" (the default method described in Section 5.1.3 of [RFC6513]) for sources in other ASes while (if provisioned so) selecting upstream PE based on installed UMH routes for sources in the local AS.

The second problem is that, when inclusive inter-as tunnels are used, if both PE1 and PE2 send the same traffic, ASBR1 will inject duplicate traffic into the same inter-as tunnel, while PE3 and PE4 has no way to distinguish the source PE of each copy.

There are two solutions to the second problem. The first solution is that ASBRs advertise PE Distinguisher (PED) labels (Section 8 of [RFC6514]) via a PED attribute attached to their Inter-AS I-PMSI A-D routes, and push a label that identifies the ingress PE when it sends a packet into the inclusive inter-AS tunnel, and an egress PE discards traffic not from its chosen upstream PE.

The other solution is for the ingress ASBR to only accept traffic from one ingress PE and forward into the inclusive inter-as tunnel. This does not require egress PEs to discard traffic based on an additional PED label, but does require the ingress ASBR to participate upstream PE selection and do IP forwarding in a VRF for the source VPN, so that it can choose the copy to accept and forward. Because it may not have local receivers, it needs to receive C-multicast routes from egress PEs who will receive corresponding traffic from it, and import the routes into its local VRF.

1.5. EVPN Selective Multicast Ethernet Tag (SMET) Routes

[RFC9251] defines a EVPN route type known as an "SMET route".

The EVPN SMET routes are analogous to the MVPN C-multicast routes, in that both type of routes are used to disseminate the information that a particular egress PE has interest in a particular multicast C-flow or set of C-flows.

An EVPN SMET route contains, in its NLRI, the RD associated with the VRF from which the SMET route was originated. In addition, it is disseminated to all PEs of a given EVI. In this way, SMET routes are analogous to the MVPN C-multicast routes that are used for C-BIDIR support.

An EVPN SMET route contains, in its NLRI, the IP address of the originating PE. In this way, they are analogous to the MVPN Leaf A-D routes (They really combine the function of the MVPN C-multicast routes and the MVPN Leaf A-D routes). Similarly, they are also analogous to the C-multicast route for MVPN-RPL that carries an EC that identifies the originating PE.

In EVPN, as in MVPN, explicit tracking is required when selective tunnels are realized using IR, BIER, or RSVP-TE P2MP. The EVPN SMET routes provide this explicit tracking, so in these cases EVPN does not need explicit Leaf A-D routes. With IR/BIER, there is no need for S-PMSI route either. However, when SMET routes are used with segmented IR/BIER tunnels, more procedures are needed, just like the C-multicast route in MVPN-RPL case (Section 1.6). For that reason, given the similarity between SMET and C-Multicast routes, in this document we will use the same term C-Multicast route for EVPN SMET route as well. The two may be used interchangeably in case of EVPN.

If selective tunnels are set up using procedures that do not require explicit tracking, e.g. mLDP or PIM, the following optimization could be done, similar to MVPN-RPL with mLDP/PIM tunnels (Section 1.2.2):

- * When constructing an SMET route, put 0 as the Originator Router Address.
- * When constructing an SMET route in the context of a given EVI, have all PEs of that EVI set the RD field of the NLRI to the same value (This is analogous to "MVPN-RPL RD" discussed in Section 1.2.2).
- * When a Route Reflector distributes the SMET routes, it uses BGP ADD-PATH to distribute at least two "paths" for a given NLRI.

1.6. Provider Tunnel Segmentation with Explicit-Tracking C-Multicast Routes

For the above MVPN-RPL and EVPN cases where C-multicast routes are used for explicit tracking without requiring corresponding S-PMSI A-D routes in case of IR/BIER selective tunnel, it works well when there is no tunnel segmentation. With tunnel segmentation [RFC6514] [RFC7524], [I-D.ietf-bess-evpn-bum-procedure-updates] additional procedures are needed.

1.6.1. Conventional Tunnel Segmentation

Multicast forwarding needs to follow a rooted tree. With segmentation, the tree is divided into segments, with each segment rooted at either the ingress PE or a Regional Border Router (RBR). A segment is contained in a region, which could be an AS, an area, or a sub-area. The root of a segment only needs to track the leaves in its region, which are PEs or RBRs in that region. With the traditional PMSI/Leaf A-D procedures, an ingress PE/RBR sends out an I/S-PMSI route, propagated by RBRs (segmentation points), who change the tunnel identifier along the way to identify the tunnels for their segments. The Leaf A-D routes from PEs are not propagated by the RBRs. Rather, a RBR will proxy the Leaf AD routes it receives from its downstream towards its upstream RBR or PE, following the I/S-PMSI A-D routes received in the upstream region, as specified in [RFC6514] [RFC7524] [I-D.ietf-bess-evpn-bum-procedure-updates].

1.6.2. Selective Tunnel Segmentation with Untargeted Explicit-Tracking C-multicast Routes

Without segmentation, the untargeted explicit-tracking C-Multicast routes are sent to every PE, and each PE adds the originator of the routes as leaves of the tunnel rooted at the PE.

With segmentation, untargeted explicit-tracking C-Multicast routes are propagated through segmentation points towards all ingress PEs or ASes and are merged along the way. This is like the traditional PMSI/Leaf A-D procedures but with one difference.

With the traditional PMSI/Leaf A-D procedures, the propagation is towards the originator of the PMSI A-D route and a single tree is formed. With untargeted C-Multicast routes, multiple trees are formed, each being rooted at the ingress PE (if per-region aggregation [I-D.ietf-bess-evpn-bum-procedure-updates] is not used) or ingress RBR (if per-region aggregation is used). The roots of those trees are either the ingress PEs or the ingress RBRs, identified by all the per-PE or per-region I-PMSI A-D routes.

To form those multiple trees without requiring S-PMSI A-D routes from the ingress PEs/RBRs, this document proposes that the RBRs convert a C-multicast route originated in its own region to Leaf A-D routes, as if corresponding S-PMSI A-D routes had been received from ingress PEs/RBRs. The details are provided in Section 2.2.

2. Specifications

This section provides detailed specifications for the optional enhancements introduced above.

2.1. MVPN C-Bidir Support with VPN Backbone being RPL

2.1.1. Constructing C-Multicast Share Tree Join route

In the context of a particular VRF, a PE with downstream state for the group C-G-BIDIR originates a C-multicast Shared Tree Join route, referred to as "MVPN-RPL C-multicast Join", when the MVPN-RPL method of C-BIDIR support is being used.

The fields of the route are set as follows:

- * RD: See Section 2.1.1.2.
- * Source AS: set to zero.
- * Multicast Source Length: 4 or 16.
- * Multicast Source: set to RPA.
- * Multicast Group Length: 4 or 16.
- * Multicast Group: BIDIR-PIM group address.

Note that the RD field, and the Route Targets that are attached to the C-multicast route are different than what is specified in [RFC6514]. See following two sections.

2.1.1.1. Setting the Route Targets

Per [RFC6514], when a PE originates a C-multicast route, it "targets" the route to a specific one of the other PEs attached to the same VPN. The IP address of the targeted PE is encoded into a Route Target and attached to the C-multicast route. This ensures that the C-multicast route is processed only by the PE to which it is targeted.

However, C-multicast routes used by the MVPN-RPL method are not targeted. Rather, they must be processed by all the other PEs attached to the same MVPN. Thus we refer to these routes as "untargeted". The Route Targets attached to these routes must be such as to cause the routes to be propagated to all the other PEs of the given MVPN. By default, these will be the same Route Targets that are attached to the I-PMSI A-D routes of the MVPN.

2.1.1.2. Setting the Route Distinguisher

Per [RFC6514], the RD in a C-multicast Join Route is the RD of a VRF on the PE to which the route is targeted. However, in an MVPN-RPL C-multicast Join, the RD is set differently.

If PIM/mLDP provider tunnels are used, and it is known that all the PEs/RRs/ASBRs involved in the propagation of C-multicast routes support BGP ADD-PATH, the RD MAY be set to a value that is specially configured to be used as the RD for MVPN-RPL in a given VPN. Call this the "MVPN-RPL" RD for that VPN. In that case, all the C-multicast Joins that are providing C-BIDIR support (for a given VPN) using the MVPN-RPL method will have the same RD. This MVPN-RPL RD of a given VPN MUST NOT be used for any other purpose, or by any other VPN. See Section 1.2.2 for a discussion of when it may be advantageous to use an MVPN-RPL RD.

For other provider tunnel types, or if the above mentioned MVPN-RPL RD in case of PIM/mLDP tunnel is not feasible (e.g. BGP ADD-PATH is not supported), the RD in the C-multicast route is that of the VRF from which the route is originated.

For Global Table Multicast (GTM) using MVPN procedures [RFC7716], RFC 7716 specifies that MVPN routes use a special 0:0 RD. This document specifies that GTM use non-0:0 RDs for C-Multicast routes for C-Bidir, when the backbone is used as RPL and provider tunnels are not set up by PIM/mLDP.

2.1.2. Setting Up the MVPN-RPL

By default, the I-PMSI or (C-*,C-BIDIR) S-PMSI plays the role of MVPN-RPL. When (C-*,C-G-BIDIR) S-PMSI is used for a particular C-G-BIDIR, the following procedures are followed, depending on the type of provider tunnel used.

2.1.2.1. Ingress Replication or BIER

If Ingress Replication or BIER is used, there is no need for the ingress PE to advertise (C-*,C-G-BIDIR) S-PMSI A-D route. The ingress PE identifies the tunnel leaves to send traffic to by the C-multicast routes it receives, because each such route has a different RD and serves explicit tracking purpose. In case of IR, the label in the Intra-AS I-PMSI A-D route or (C-*,C-*) S-PMSI A-D route from a leaf is used to send traffic to the leaf. In case of BIER, the label in the same route from the ingress PE is used to send traffic.

2.1.2.2. RSVP-TE P2MP

With RSVP-TE P2MP tunnel, the ingress PE advertises (C-*,C-G-BIDIR) S-PMSI A-D route without setting the LIR bit in the route's PTA. It identifies the tunnel leaves from the C-multicast routes it receives.

2.1.2.3. PIM/mLDP

With PIM or mLDP P2MP provider tunnel, procedures in [RFC6514] are followed.

2.2. Inter-AS Propagation of MVPN C-Multicast Routes

This specification allows two methods of Inter-AS propagation for MVPN C-multicast routes. The choice of which method is used is by provisioning.

2.2.1. Procedures in Section 11.2 of [RFC6514]

The procedures in Section 11.2 of [RFC6514] are extended with the following.

The Source AS field in the NLRI of C-multicast route is set to the AS number of the UMH PE if and only if segmented inter-AS tunnels and per-AS aggregation (via Inter-AS I-PMSI A-D routes) are used. The existing procedures are used as is in this case.

Otherwise, when an egress PE constructs a C-Multicast route and the upstream PE is in a different AS from the local PE, it finds in its VRF an Intra-AS I-PMSI A-D route or any S-PMSI A-D route from the upstream PE (the Originating Router's IP Address field of that route has the same value as the one carried in the VRF Route Import of the (unicast) route to the address carried in the Multicast Source field). The RD of the found I/S-PMSI A-D route is used as the RD of the advertised C-multicast route. The Source AS field in the C-multicast route is set to 0. If the Next Hop of the found I/S-PMSI

A-D route is an EBGp neighbor of the local PE, then the PE advertises the C-multicast route to that neighbor. Otherwise the PE advertises the C-multicast route into IBGP.

When an ASBR receives a C-multicast route with the Source AS field set to 0, it uses the RD of the C-multicast route to locate an Intra-AS I-PMSI A-D route or any S-PMSI A-D route, and propagate the C-multicast route to the bgp neighbor from which the found I/S-PMSI A-D route is learned.

In the case of GTM, when a PE originates an Intra-AS I-PMSI A-D route or an S-PMSI A-D route in the case of non-segmented inter-as tunnels, the route's RD field MUST be set to a unique non-0:0 RD. In the case of IPv4 infrastructure, the VRF Route Import EC that the PE attaches to its VPN-IP routes MAY be used. This allows an ASBR to find the PMSI A-D route from the UMH PE when it propagates a C-Multicast route as described in the previous paragraph.

2.2.2. Ordinary BGP Propagation Procedures

This document specifies that C-multicast routes MAY be propagated using ordinary BGP propagation procedures, which do not rely on the presence of any I/S-PMSI A-D routes. With this method, the construction of C-Multicast A-D routes always follows the same procedures, whether the source is in the same or different AS. Specifically, the 3rd and 5th paragraphs of Section 11.1.3 of [RFC6514] are quoted here:

From the selected UMH route, the local PE extracts (a) the ASN of the upstream PE (as carried in the Source AS Extended Community of the route), and (b) the C-multicast Import RT of the VRF on the upstream PE (the value of this C-multicast Import RT is the value of the VRF Route Import Extended Community carried by the route). The Source AS field in the C-multicast route is set to that AS. The Route Target Extended Community of the C-multicast route is set to that C-multicast Import RT.

...

... the RD of the advertised MCAST-VPN NLRI is set to the RD of the VPN-IP route that contains the address carried in the Multicast Source field.

For targeted C-multicast routes, this will result in a less optimal propagation path, but it does work in all cases. The Route Target Constraint procedures can always be used to obtain a more optimal path.

2.3. Inter-AS Upstream PE Selection

This document allows that, when selecting upstream PE for a source not in the local AS, the Single Forwarder Selection method, i.e., the default procedure in Section 5.1.3 of [RFC6513] is used, even if the method of using the installed UMH route as the selected UMH route is provisioned (to be used for sources in the local AS only).

2.4. Duplication Prevention on the Same Inclusive Inter-AS Tunnel

The procedures in this section are only applicable when inclusive inter-AS tunnels advertised in Inter-AS I-PMSI A-D routes are used and it is known that an ingress ASBR may receive duplicate traffic from different ingress PEs in the same local AS. One of the following two methods is provisioned consistently on all PEs and ingress ASBRs of a VPN.

2.4.1. Using PE Distinguisher Labels

With this method, an ingress ASBR that may receive duplicate traffic from different PEs and inject into the same inclusive inter-AS tunnels use a PED label to identify the upstream PE of the traffic, so that egress PEs can discard traffic not from their selected upstream PE.

When an ASBR advertises an Inter-AS I-PMSI A-D route, it includes a PE Distinguisher (PED) Labels attribute [RFC6514]. The attribute lists one label for each PE in the corresponding AS, and the labels are allocated from a Domain-wide Common Block (DCB, [I-D.ietf-bess-mvpn-evpn-aggregation-label]). When an ingress ASBR forwards traffic it receives from a local ingress PE, it needs to push the label assigned to the ingress PE and advertised in the PED attribute of corresponding Inter-AS I-PMSI A-D route. Because the labels are assigned from the DCB, they do not need to be swapped along the way. Downstream and upstream assigned labels could be used as well, but that requires the ASBRs swap PED labels along the way (in addition to tunnel label swapping) so they are not discussed here.

Note that if intra-AS tunnel aggregation is used in the ingress AS, the ingress PE SHOULD use the same PED label and the ingress ASBR MUST NOT push the PED label again when forwarding traffic into the inclusive inter-as tunnel.

2.4.2. Ingress ASBR Filtering Out Duplications

With this method, an ingress ASBR performs IP forwarding for traffic that goes onto inclusive tunnels [I-D.zzhang-bess-mvpn-evpn-segmented-forwarding] after discarding traffic not from the upstream PE that it chooses.

The ingress ASBR MUST be provisioned with a VRF for each VPN with local PEs, and with a C-multicast Import RT for the VRF. The Inter-AS I-PMSI A-D route that it advertises for the VPN MUST carry a VRF Route Import Extended Community (EC) that has the value of the C-multicast Import RT for the VRF. This is similar to that a PE includes a VRF Route Import EC in VPN-IP routes that it originates.

When an egress PE constructs a C-multicast routes, if the source is in a different AS, the ingress ASBR that advertises the Inter-AS I-PMSI A-D route installed by this egress PE is chosen as the upstream PE. The RD and AS number in the Inter-AS I-PMSI A-D route are used to construct the C-multicast route, and a C-multicast Import RT (for importing the constructed C-multicast route into the ingress ASBR's VRF) is included, with the value of this RT being the value of the VRF Route Import EC carried by the Inter-AS I-PMSI A-D route.

When an ingress ASBR receives a C-multicast route and imports the route into one of its local VRFs (because of the RT constructed as above), it treats as if a PIM/IGMP join was received on the inter-AS inclusive tunnel. It selects its own upstream PE and originates a corresponding C-multicast route. Corresponding traffic received from the selected upstream PE is then routed into the inter-AS inclusive tunnel.

2.5. Provider Tunnel Segmentation with Explicit-Tracking C-Multicast Routes

This section applies when IR/BIER are used for MVPN/EVPN selective tunnels.

If per-region aggregation [I-D.ietf-bess-evpn-bum-procedure-updates] is used, this document specifies that the per-region I-PMSI A-D route MUST carry a VRF Route Import EC to identify the originator of the per-region I-PMSI A-D route. Note that, while it borrows "VRF Route Import EC" from the UMH routes, it is only used to identify the originator.

If per-region aggregation is not used, this document specifies that either per-PE I-PMSI or (C-*,C-*) S-PMSI A-D routes MUST be originated by every PE.

2.5.1. Egress PEs and RBRs

An egress PE originates MVPN C-multicast routes for MVPN-RPL as specified in previous sections of this document, or EVPN SMET routes as specified in [RFC9251]. Recall that EVPN SMET routes may also be referred to C-Multicast routes in this document.

Explicit-tracking C-multicast routes must be processed by segmentation points, which are referred to as RBRs. When a RBR receives a C-multicast route from within its own region, and the route does not carry a flag bit that indicates the route is converted from a downstream Leaf A-D route (see descriptions below), it converts the C-multicast route into one or more Leaf A-D routes, as if it had received corresponding S-PMSI A-D routes. When a converted Leaf A-D routes reaches the ingress region, the RBR converts it back to C-multicast routes.

With per-region aggregation, the RBR in an egress region finds all active per-region I-PMSI A-D route that the RBR has in the corresponding VRF. For each of them, it makes up a (C-S,C-G) or (C-*,C-G) S-PMSI A-D route as following.

- * RD: set to the RD from the per-region I-PMSI A-D route.
- * Source/Group length/address fields: set according to the received C-multicast route.
- * Originator's IP Address: set according to the VRF Route Import EC in the per-region I-PMSI A-D route
- * Ethernet Tag ID in case of EVPN: set according to the received SMET route (which is also referred to as C-multicast route).
- * Next Hop: set according to the per-region I-PMSI A-D route.

Without per-region aggregation, a RBR finds all active per-PE I-PMSI or (C-*,C-*) S-PMSI A-D route in the VRF. For each of them it makes up a (C-S,C-G) or (C-*,C-G) S-PMSI A-D route similar to the per-region aggregation case. The only difference is that the Originator's IP Address field is set to the same as in the per-PE I-PMSI or (C-*,C-*) S-PMSI A-D route.

A corresponding Leaf A-D route is then generated and propagated to the upstream identified by the BGP next hop in the made up S-PMSI A-D route, following existing PMSI/Leaf A-D route procedures.

If the egress region uses Ingress Replication, the made up S-PMSI A-D route is not propagated anywhere. If the egress region uses PIM or RSVP-TE/mLDP P2MP tunnel, the S-PMSI A-D route is advertised into the egress region to announce the tunnel to be used. If the egress region uses BIER or aggregated RSVP-TE/mLDP P2MP tunnel, the S-PMSI A-D route is also advertised into the egress region and carry an upstream allocated label. The label may be at the per S-PMSI A-D route level or at per VPN/BD level. In the former case, label switching at the RBR can be used. In the latter case, IP lookup in the corresponding VRF or BD is needed.

2.5.2. Transit RBRs

When an upstream RBR receives a (C-S,C-G) or (C-*,C-G) Leaf A-D route, It locates the active per-PE/region I-PMSI or (C-*,C-*) S-PMSI A-D route whose RD matches the received Leaf A-D route. If no such route exists, the received Leaf A-D route is ignored until such a route appears later. It also tries to locate a corresponding active (C-S,C-G) or (C-*,C-G) S-PMSI A-D route, which could be a real one received from an upstream PE/RBR, or could be a made up one triggered by a Leaf A-D route from a different downstream. If such route exists, existing PMSI/Leaf A-D route procedures are followed.

If no such corresponding active (C-S,C-G) or (C-*,C-G) S-PMSI A-D route exists, and the located active I-PMSI or (C-*,C-*) S-PMSI A-D route has a next hop different from the Originator IP Address in the per-PE I-PMSI A-D route or (C-*,C-*) I-PMSI A-D route, or different from the address in the VRF Route Import EC in the per-region I-PMSI A-D route, the ingress region corresponding to the I-PMSI or (C-*,C-*) S-PMSI A-D route has not been reached. The RBR then makes up a (C-S,C-G) or (C-*,C-G) S-PMSI A-D route. as specified earlier, and proxies Leaf A-D routes further up. Similarly, the S-PMSI A-D route may be advertised into the transit region.

2.5.3. Ingress RBRs

If the BGP next hop in the located active I-PMSI or (C-*,C-*) S-PMSI A-D route matches the Originator IP Address in the per-PE I/S-PMSI A-D route or the IP address in the per-region I-PMSI A-D route's VRF Route Import EC, it means the ingress region has been reached. If the corresponding (C-S,C-G) or (C-*,C-G) S-PMSI A-D route is a made up one and not actually advertised by an ingress PE/RBR, and the RBR does not have corresponding local (C-S,C-G) or (C-*,C-G) state, it reconverts the Leaf A-D route back to C-multicast route, with a CV ("Converted") flag bit indicating that the route is not from local state learned on PE-CE interface but from state learned further downstream. The flag bit prevents other RBRs in this region to trigger Leaf A-D routes from this converted C-multicast route.

The converted C-multicast route is constructed as following:

- * RD: set to the RD of the RBR for the related IP/MAC VRF.
- * Source/Group length/address fields: set according to the received Leaf A-D route.
- * Ethernet Tag ID in case of EVPN: set according to the received Leaf A-D route.
- * Next Hop: set to the RBR's local IP Address.

The RT of the converted C-multicast route is set to the RT used for VRF but the route is only propagated to PEs/RBRs in the local region.

For EVPN SMET routes, the flag bit is part of the existing Flags field in the NLRI:

```

      0  1  2  3  4  5  6  7
      +--+--+--+--+--+--+--+--+
      |reserved|CV|IE|v3|v2|v1|
      +--+--+--+--+--+--+--+--+

```

The IE/v3/v2/v1 are existing bits and the CV bit is the new bit to indicate that this is converted from state learned from downstream.

For MVPN C-Multicast route, the CV bit is part of a new MVPN Flag EC, to be specified in a future revision.

2.5.4. Setting Up Forwarding State on RBRs

As a RBR follows the PMSI/Leaf A-D route procedures (even though the S-PMSI A-D route may be made up and not real), it sets up forwarding state accordingly [RFC7988] [RFC8556]. If IR is used in the upstream region, a downstream allocated label is advertised in the PTA of the Leaf A-D route sent upstream. If BIER is used in a region, the root RBR for the segment in that region MUST advertise an S-PMSI A-D route, whether the route is actually received from upstream or made up based on received C-multicast route or Leaf A-D route, with the PTA's label field set to a label upstream-assigned by the root RBR of the segment. This allows label switching by the RBR instead of relying on (C-S,C-G) lookup based forwarding in the VRF.

2.5.5. Other Types of Tunnels

The inter-region segmented tunnel can consist of different types of tunnels, like PIM/mLDP/RSVP-TE P2MP tunnels that require advertised S-PMSI A-D routes. This is just like BIER case mentioned in the above section. The only difference is that in BIER case it is the upstream allocated label that needs to be advertised by the S-PMSI A-D routes and in PIM/mLDP/RSVP-TE P2MP case it is the tunnel identity and optionally the upstream allocated label that need to be advertised by the S-PMSI A-D routes.

3. Security Considerations

This document does not seem to introduce new security risks, though this may be revised after further review and scrutiny.

4. Contributors

The following also contributed to this document.

Vinod N Kumar
Juniper Networks
Email: vinkumar@juniper.net

5. Acknowledgements

The authors thank Vinay Nallamothe, Kevin Wang, and Sambasiva Rao for their comments and suggestions.

6. References

6.1. Normative References

- [I-D.ietf-bess-evpn-bum-procedure-updates]
Zhang, Z. J., Lin, W., Rabadan, J., Patel, K., and A. Sajassi, "Updates on EVPN BUM Procedures", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-bum-procedure-updates-14, 18 November 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-bess-evpn-bum-procedure-updates-14>>.
- [I-D.ietf-bess-mvpn-evpn-aggregation-label]
Zhang, Z. J., Rosen, E. C., Lin, W., Li, Z., and I. Wijnands, "MVPN/EVPN Tunnel Aggregation with Common Labels", Work in Progress, Internet-Draft, draft-ietf-bess-mvpn-evpn-aggregation-label-14, 4 October 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-bess-mvpn-evpn-aggregation-label-14>>.

- [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>>.
- [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, <<https://www.rfc-editor.org/info/rfc4684>>.
- [RFC5015] Handley, M., Kouvelas, I., Speakman, T., and L. Vicisano, "Bidirectional Protocol Independent Multicast (BIDIR-PIM)", RFC 5015, DOI 10.17487/RFC5015, October 2007, <<https://www.rfc-editor.org/info/rfc5015>>.
- [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>>.
- [RFC7524] Rekhter, Y., Rosen, E., Aggarwal, R., Morin, T., Grosclaude, I., Leymann, N., and S. Saad, "Inter-Area Point-to-Multipoint (P2MP) Segmented Label Switched Paths (LSPs)", RFC 7524, DOI 10.17487/RFC7524, May 2015, <<https://www.rfc-editor.org/info/rfc7524>>.
- [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>>.
- [RFC7911] Walton, D., Retana, A., Chen, E., and J. Scudder, "Advertisement of Multiple Paths in BGP", RFC 7911, DOI 10.17487/RFC7911, July 2016, <<https://www.rfc-editor.org/info/rfc7911>>.
- [RFC7988] Rosen, E., Ed., Subramanian, K., and Z. Zhang, "Ingress Replication Tunnels in Multicast VPN", RFC 7988, DOI 10.17487/RFC7988, October 2016, <<https://www.rfc-editor.org/info/rfc7988>>.

- [RFC8556] Rosen, E., Ed., Sivakumar, M., Przygienda, T., Aldrin, S., and A. Dolganow, "Multicast VPN Using Bit Index Explicit Replication (BIER)", RFC 8556, DOI 10.17487/RFC8556, April 2019, <<https://www.rfc-editor.org/info/rfc8556>>.
- [RFC9251] Sajassi, A., Thoria, S., Mishra, M., Patel, K., Drake, J., and W. Lin, "Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Proxies for Ethernet VPN (EVPN)", RFC 9251, DOI 10.17487/RFC9251, June 2022, <<https://www.rfc-editor.org/info/rfc9251>>.

6.2. Informative References

- [I-D.ietf-bess-evpn-irb-mcast]
Lin, W., Zhang, Z. J., Drake, J., Rosen, E. C., Rabadan, J., and A. Sajassi, "EVPN Optimized Inter-Subnet Multicast (OISM) Forwarding", Work in Progress, Internet-Draft, draft-ietf-bess-evpn-irb-mcast-11, 4 March 2024, <<https://datatracker.ietf.org/doc/html/draft-ietf-bess-evpn-irb-mcast-11>>.
- [I-D.zzhang-bess-mvpn-evpn-segmented-forwarding]
Zhang, Z. J. and J. Xie, "MVPN/EVPN Segmentated Forwarding Options", Work in Progress, Internet-Draft, draft-zzhang-bess-mvpn-evpn-segmented-forwarding-00, 20 December 2018, <<https://datatracker.ietf.org/doc/html/draft-zzhang-bess-mvpn-evpn-segmented-forwarding-00>>.
- [RFC9026] Morin, T., Ed., Kebler, R., Ed., and G. Mirsky, Ed., "Multicast VPN Fast Upstream Failover", RFC 9026, DOI 10.17487/RFC9026, April 2021, <<https://www.rfc-editor.org/info/rfc9026>>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks
Email: zzhang@juniper.net

Robert Kebler
Juniper Networks
Email: rkebler@juniper.net

Wen Lin
Juniper Networks
Email: wlin@juniper.net

Eric Rosen
Email: erosen52@gmail.com